Improving the Generalizability of Collaborative Dialogue Analysis With Multi-Feature Embeddings

Ayesha Enayet
University of Central Florida
Orlando, FL USA
ayeshaenayet@knights.ucf.edu

Gita Sukthankar
University of Central Florida
Orlando, FL USA
gitars@eecs.ucf.edu

Abstract

Conflict prediction in communication is integral to the design of virtual agents that support successful teamwork by providing timely assistance. The aim of our research 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 paper 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 paper 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. (2021). Our results show that MFeEmb serves as an excellent domain-agnostic representation for meta-pretraining a few-shot model on collaborative multiparty dialogues.

1 Introduction

For many natural language processing applications, the ability to learn features that generalize well across multiple datasets is a key desideratum (Saikia et al., 2020). This paper 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 paper 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 (Rahimi and Litman, 2020) and emergent leadership (Maese et al., 2021). Frequency of communication is not in itself a good predictor of team performance, but a meta-analysis conducted by Marlow et al. (2018) 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” (Jehn, 1995), 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. This paper focuses on 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 (Devlin et al., 2018) show good performance across multiple NLP tasks on in-domain test sets but are less robust to domain shift. Previous work identified that dialogue acts and sentiment sequences are informative features that predict conflict reliably even at the earliest stage of team problem-solving (Enayet and Sukthankar, 2021a); however, classifiers constructed using these features still experience lackluster transfer performance when applied to new datasets, particularly when detecting high conflict examples (Enayet and Sukthankar, 2021b).

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 paper 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. (2021). 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.

2 Related Work

Previous studies on group interaction tasks such as conflict prediction (Rahimi and Litman, 2020), disruptive talk detection (Park et al., 2022), group satisfaction (Lai and Murray, 2018), and task performance prediction (Kubasova et al., 2019; Murray and Oertel, 2018) 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 (Pugh et al., 2022), 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 (Jiang and Zhai, 2007) or explicitly identifying feature correspondences between the domains (Blitzer et al., 2006). An alternate approach is to learn a single representation that generalizes well across multiple domains. This can be done using few-shot learning (Wang et al., 2020), 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.

Triantafillou et al. (2021) introduced a method that improves few-shot generalizability by making use of multiple datasets in order to learn a universal template. Dvornik et al. (2020) 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. 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 (Ben-David et al., 2006). Few-shot frameworks such as Meta-pretraining then Meta-Learning (MTM) (Deng et al., 2019) 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. (2021) introduced a method to quantify the generalizability of a few-shot classifier under supervised, unsupervised and semi-supervised settings. This paper uses 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.

3 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.

3.1 Datasets

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

1. **Teams corpus** (Litman et al., 2016): 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 (Rahimi and Litman, 2020). 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** (Huang et al., 2022): This dataset consists of 67 teams of three people participating in a simulated search and rescue task within the Minecraft game environment. Participants completed 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** (Enayet and Sukthankar, 2020): 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 an issue resolution rubric (described in Appendix G).

3.2 Multi-Feature Embedding (MFeEmb)

This paper 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 semantic 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 (Le and Mikolov, 2014) to learn embeddings (see Appendix B). 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 in-domain 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 USE-DAC (Universal Sentence Encoder Dialogue Act Classifier, described in Appendix A). 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 the 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. We have made our code available at `https://github.com/ayeshaEnayet/MFeEmb.git`.

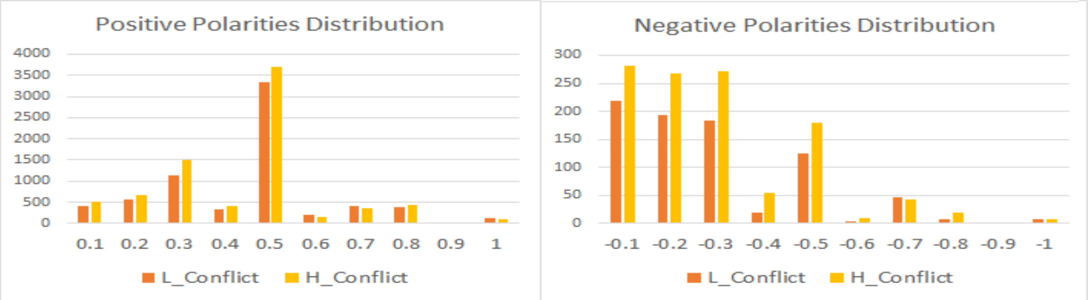

Figure 1: Sentiment polarity distribution of the high conflict vs. low conflict classes in the Teams dataset

3.3 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 1).

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 act distributions and n-grams in Appendix H).

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. 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.

3.4 Synthetic Datasets

To further improve generalization, we augment our training data with synthetic datasets generated using synonym replacement, as proposed by Wei and Zou (2019). 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 facilitate generalization.

In our experiments, the basic synonym replacement did not significantly change the intent and sentiment of the utterances. To show the robustness of dialogue acts and sentiment sequences towards data augmentation, we utilize TextAttack (Morris et al., 2020), 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 the original Game2 dataset (Figure 2). The 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 a 14% change in sentiment sequences (Appendix J).

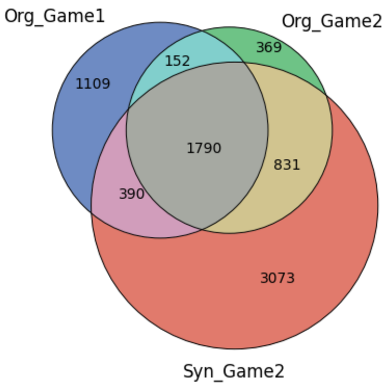

Figure 2: Vocabulary overlap between original Game1, original Game2, and the Game2 adversarially generated dataset

4 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 evaluate our proposed MFeEmb 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 (2018). 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 deviation of F1-Scores after 15 runs. For the SVM and logistic regression classification experiments, to improve the readability, we only report the best-performing classifier results measured by mean F1 score in Section 5; for the results of both classifiers, see Appendix K. '*' denotes that logistic regression was the top performer, and '+' denotes cases where the SVM was the best.

4.1 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 results of both classifiers. For DAs and sentiment sequences, we always use the Dynamic Memory model (DM) of Doc2Vec.

4.2 Few-Shot Learning (FsText)

For few-shot learning, we use the method proposed by Bailey and Chopra (2018) 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.

4.3 Concatenation Ensemble

Due to the small size of the training set, we apply the synonym replacement technique proposed by Wei and Zou (2019) to augment the training data as described in Section 3.4. 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.

4.4 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 (Enayet and Sukthankar, 2021a) using the proposed dialogue act only and sentiment only embeddings. Note that our results are not directly comparable to what was reported in the previous work because we use a reduced training set; thus, we reimplemented the embeddings. We also compare MFeEmb to the commonly used BERT based embedding (Appendix D).

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 (Sec. 4.3) trained on the synonym replaced augmented dataset, 3) a variation of few-shot learning method (FsText) (Bailey and Chopra, 2018) 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.

5 Results

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

5.1 Similarity Based Evaluation

First, we quantify the potential generalization of the representation using the similarity measure proposed by Bontonou et al. (2021). The similarity measure is given by:

$$intra(c) = \frac{1}{k(k-1)} \sum_{\substack{i \\ y_i=c}} \sum_{\substack{j\neq i \\ y_j=c}} \cos(f_i, f_j) \quad (1)$$

$$inter(c, \tilde{c}) = \frac{1}{k^2} \sum_{\substack{i \\ y_i=c}} \sum_{\substack{j\neq i \\ y_j=\tilde{c}}} \cos(f_i, f_j) \quad (2)$$

$$similarity = \frac{1}{N} \sum_{c=1}^{N} (intra(c) - \max_{c\neq\tilde{c}}(inter(c, \tilde{c}))) \quad (3)$$

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 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.

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	0.446+	-0.016	**0.458+**

Table 1: Similarity-based generalizability analysis. **Word_Emb**: Distributed Bag of Words document embedding. **MFeEmb**: Multi-Feature Embedding generated using the Dynamic Memory model of Doc2Vec. The similarity score of MFeEmb accurately predicts higher classification accuracy. '*' denotes the logistic regression results, and '+' denotes the SVM results.

5.2 MFeEmb Performance Summary

Figure 3 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 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 (Enayet and Sukthankar, 2021a). Our most consistent model, MFeEmb with FsText, had a significantly higher F1 score on the high conflict class compared to baseline models (see Table 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.

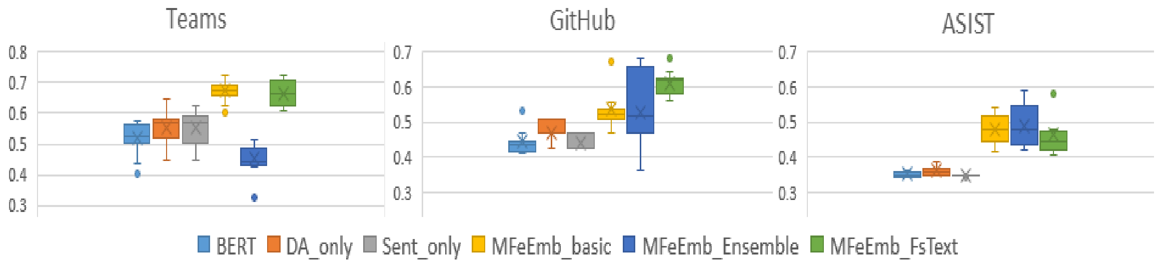

Figure 3: Performance of MFeEmb vs. other embedding choices from prior work.

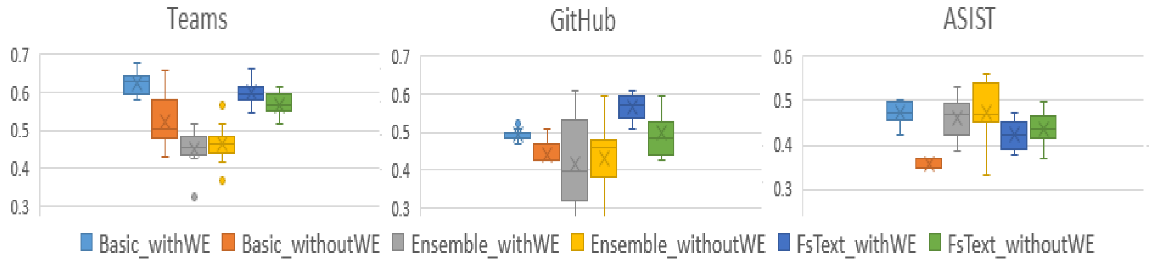

Figure 4: Performance of MFeEmb with and without word embedding (WE).

To analyze the importance of incorporating word embedding in MFeEmb, we compare the performance 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 the 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 4). The following sections present a more in-depth evaluation of each experimental setup.

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**

Table 2: Summary of high conflict class F1 scores. '*' denotes the logistic regression results, and '+' denotes the SVM results.

5.3 SVM and Logistic Regression

Table 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 regression 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 4).

5.4 Concatenation Ensemble Model

Table 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.

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 3: Detailed performance evaluation of MFeEmb. '*' denotes the logistic regression results, and '+' denotes the SVM results.

5.5 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 vocabulary'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 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.

6 Conclusion and Future Work

This paper introduces a multi-feature embedding (MFeEmb) to improve the generalizability of multiparty dialogue models under resource scarcity scenarios. We propose the use of 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.

7 Limitations

This paper 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.

8 Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-20-1-0008 and the National Institute of Mental Health of the National Institutes of Health under Award Number P50MH126231. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the National Institutes of Health, and the University of Central Florida.

References

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

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

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

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

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

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

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

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

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

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

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

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

Jing Jiang and ChengXiang Zhai. 2007. 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. Association for Computational Linguistics.

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

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

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Diane Litman, Susannah Paletz, Zahra Rahimi, Stefani Allegretti, and Caitlin Rice. 2016. 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.

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

Shannon Marlow, Christina Lacerenza, Jensine Paoletti, C. Shawn Burke, and Eduardo Salas. 2018. 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.

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

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

Kyungjin Park, Hyunwoo Sohn, Wookhee Min, Bradford Mott, Krista Glazewski, C Hmelo-Silver, and James Lester. 2022. 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*.

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

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

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

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

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3).

Jason Wei and Kai Zou. 2019. 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. Association for Computational Linguistics.

A Dialogue Act Classification

We use a dialogue act classifier (USE-DAC) to map dialogues to the sequence of DAs, where each DA in a sequence corresponds to the utterance of the dialogue. Utterances are tagged according to the SwDA-DAMSL tagset[1] which contains 42 tags, and one sequence is generated per dialogue. Our dialogue act classifier uses the Universal Sentence Encoder (USE) module available at TensorFlow Hub[2]. After extensive experiments, we identified that USE with three dense layers performs best on transfer tasks. We selected the USE Transformer-based Architecture model with three dense layers and a softmax activation function. We fine-tune USE on the SwDA dataset and use the classifier to tag the utterances of the test and training datasets. We selected the USE transformer-based model because it is itself trained on dialogue and discussion forum datasets. The test accuracy of the classification model is 72%.

B Doc2Vec

Doc2Vec (Le and Mikolov, 2014) is an unsupervised method to learn paragraph vectors from text of arbitrary size. We represent each dialogue as 1) sequence of utterances, 2) sequence of DAs, and 3) sentiment polarities. We pass these sequences through Doc2Vec to generate representations. We use the Doc2Vec implementation from the python Gensim library with an epoch size of 5, negative sampling 5, window size 5, and alpha 0.065.

C SVM & Logistic Regression

We use the classifier implementations from the scikit-learn library. The SVM was trained using the RBF kernel function and the default parameters. The parameters for logistic regression were: Cs=10, class_weight=None, cv=10, dual=False, fit_intercept=True, intercept_scaling=1.0, max_iter=1000, multi_class='ovr', n_jobs=None, penalty='l2', random_state=5434, refit=True, scoring='accuracy', solver='lbfgs', tol=0.001, verbose=False. Table 6 shows the full results of both the SVM and Logistic Regression classifiers.

D BERT Baseline

We use the bert_en_uncased_L12_H768_A12 model available at TensorFlow Hub[3] 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. The total number of parameters in the model is: 10,948,301.

E FsText

The original FsText works by using the pre-trained word2Vec embedding model word2vec-google-news-300 available through the gensim.downloader module. For MFeEmb we generate the embedding using Doc2Vec with the same parameters mentioned in Appendix B. The second phase uses a cosine similarity-based classification model that does not involve machine learning.

F Concatenation Ensemble

Our Bidirectional LSTM models for each feature has an embedding layer, an LSTM layer, one dropout layer, and one deep layer. The LSTM uses a sigmoid activation function and is trained using the Adam optimizer with a learning rate=0.01. The output shape of each individual model is (None, 100). The total number of parameters of the concatenation ensemble is: 2,365,081.

G GitHub Dataset Conflict Scoring

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.

H Dialogue Act Frequency Distribution Analysis

Figure 6 shows the frequency distribution for dialogue acts in the low conflict and high conflict classes of the Teams dataset. We divided the

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

[2] https://tfhub.dev/google/universal-sentence-encoder-large/2

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

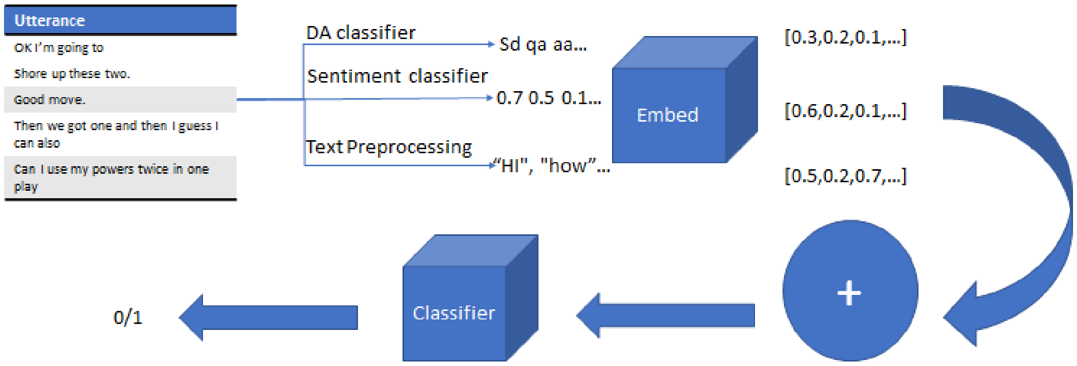

Figure 5: Utterances are classified using the dialogue act classifier to produce a sequence of DAs and the sentiment classifier to produce a time series of sentiment polarities. Along with the text data, these sequences are used to create MFeEmb using the Dynamic Memory model of Doc2Vec. The few shot learning and data augmentation options are not shown in the figure.

dialogue acts into good communication indicators, bad communication indicators and questions. Figure 7 shows the complete dialogue act frequency distribution. Table 4 shows the n-gram frequency distribution of dialogue acts across all datasets. Figure 8 visualizes the separation between low conflict and high conflict classes using both the MFeEmb and word embedding. To see if profanities were a reliable indication of conflict, we also examined profanity vocabulary differences. 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'].

I Synthetic Datasets

Synthetic datasets were generated using: https://github.com/jasonwei20/eda_nlp.

J Results on Adversarially Generated Dataset

This section presents results on the adversarially generated dataset (Synthetic Game 2) created using TextAttack[4]. 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 2 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.

K SVM & Logistic Regression Results

Table 6, 7, 8 provides the detailed results of both the SVM and logistic regression classifiers under different experimental settings.

[4] https://github.com/QData/TextAttack

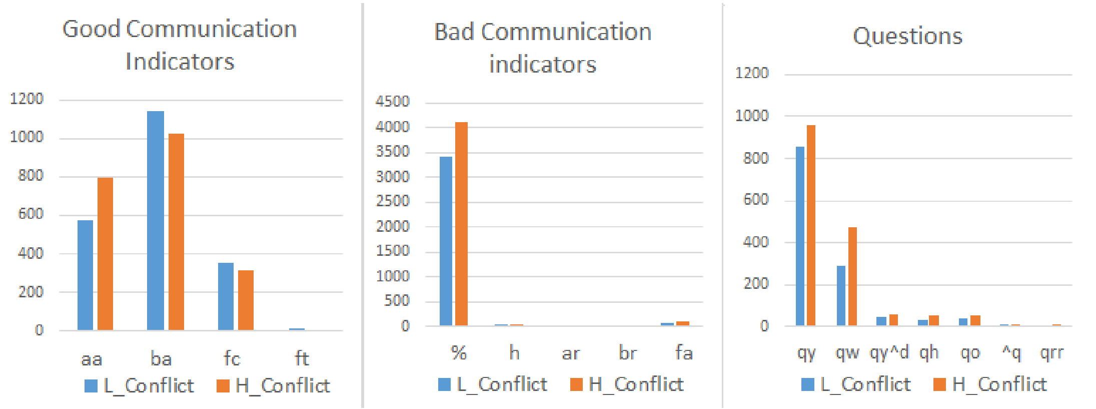

Figure 6: Dialogue act frequency distribution in high and low conflict classes for the Teams dataset. Dialogue acts were divided into good and bad communication indicators.

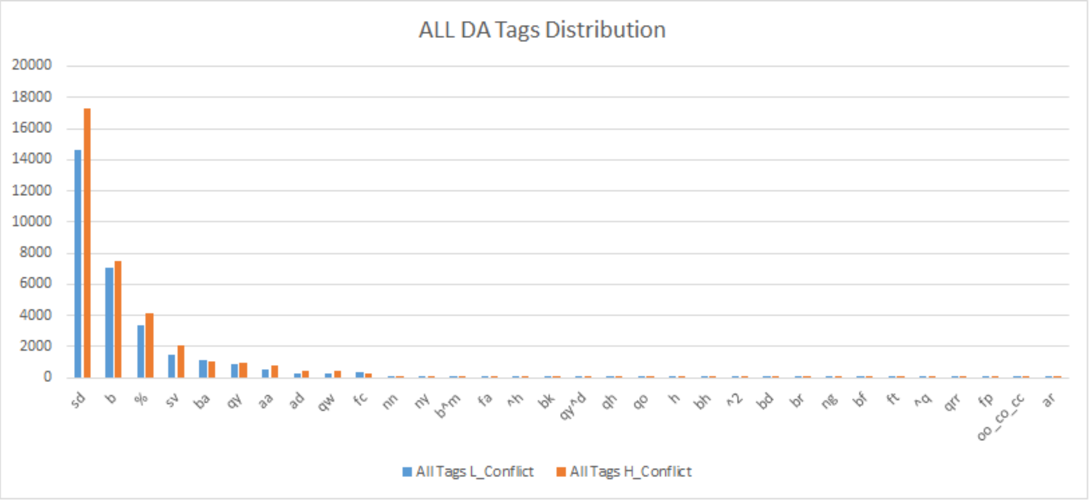

Figure 7: Complete dialogue act frequency distribution for high and low conflict classes in the Teams dataset.

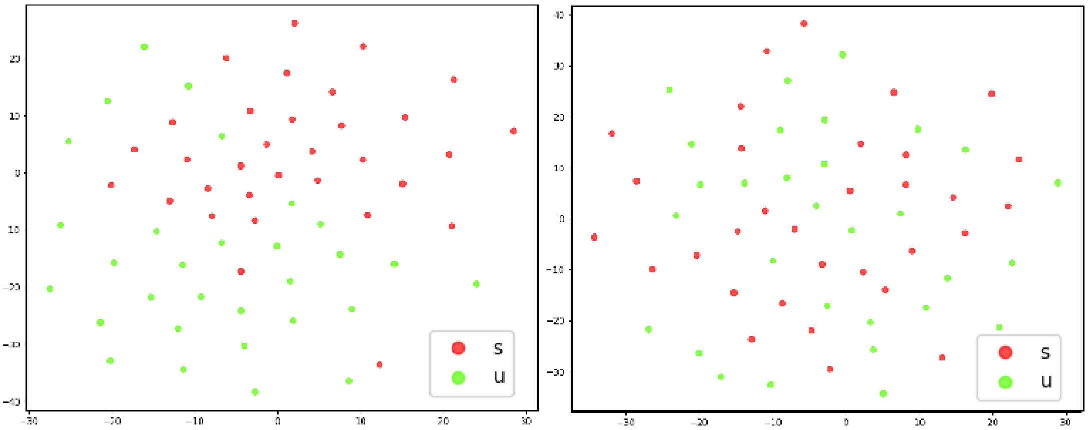

Figure 8: Comparison of the MFeEmb and word embedding distribution on the 2D plane. Multi-feature embedding showed better clustering, with most instances of one of the classes occupying the lower left and the other occupying the upper right. On the other hand, word embeddings are very intermixed. s: low conflict (successful dialogue), u: high conflict (unsuccessful dialogue).

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)
ASIST	(sd),(qy),(sv)	(sd,sd),(sd,qy),(qy,sd)	(sd,sd,sd), (sd,qy,sd), (qy,sd, sd)	(sd,sd,sd,sd), (sd,qy,sd,sd), (sd,sd,qy,sd)	(sd,sd,sd,sd,sd), (sd,sd,sd,sd, qy), (sd,qy,sd,sd,sd),

Table 4: 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.

Game2 Synthetic Dataset Results				
Train model	**Teams Game1** F1_score (std)	**SynReplace** F1_score (std)	**GitReplace** F1_score (std)	**ASISTReplace** F1_score (std)
MFeEmb (Basic)	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)

Table 5: MFeEmb results on the Game2 synthetic dataset generated using TextAttack

Both 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)* 0.369 (0.0)+	0.489 (0.080)* 0.425 (0.0)+	0.425 (0.091)* 0.348 (0.0)+
MFeEmb_Team1_dbow	0.533 (0.068)* 0.369 (0.0)+	0.437 (0.025)* 0.425 (0.0)+	0.347 (0.002)* 0.348 (0.0)+
MFeEmb_Team1_dm	0.625 (0.0295)+ 0.569 (0.045)*	0.495 (0.012)+ 0.428 (0.0)*	0.473 (0.023)+ 0.393 (0.032)*
MFeEmb_SynReplace	0.558(0.035)+ 0.542 (0.045)*	0.296(0.025)* 0.248 (0.0)+	0.318 (0.00)* 0.318 (0.00)+
MFeEmb_GitReplace	**0.676 (0.033)+** 0.593 (0.056)*	0.409 (0.039)* 0.248 (0.0)+	0.411 (0.041)* 0.318 (0.00)+
MFeEmb_ASISTReplace	0.675 (0.041)+ 0.643 (0.044)*	**0.537 (0.060)*** 0.248 (0.0)+	**0.480 (0.042)*** 0.318 (0.00)+

Table 6: Results for both the SVM and logistic regression classifiers side by side.

Word_Emb		**MFeEmb**	
Teams Game2			
similarity	F1_score	similarity	F1_score
-0.067	0.470* 0.369+	-0.016	**0.628+** 0.561*
GitHub			
similarity	F1_score	similarity	F1_score
-0.067	0.463* 0.425+	-0.017	**0.501+** 0.439*
ASIST			
similarity	F1_score	similarity	F1_score
-0.067	0.446+ 0.348*	-0.016	**0.458+** 0.394*

Table 7: Similarity-based generalizability analysis. ’*’ denotes the logistic regression results, and ’+’ denotes the SVM results.

High Conflict Class Prediction Summary		
Method	**GitHub**	**ASIST**
BERT_SynReplace	0.431	0.347
DA_only_Team1	0.320* 0.250+	0.311* 0.216+
Senti_only_Team1	0.207* 0.090+	0.300* 0.036+
MFeEmb_FsText_Team1	**0.564**	**0.478**

Table 8: Summary of high conflict class F1_scores.’*’ denotes the logistic regression results, and ’+’ denotes the SVM results.