Proceedings of the ASME 2022 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2022
August 14-August 17, 2022, St. Louis, USA

DETC2022/90895

NATURAL LANGUAGE PROCESSING FOR CONTENT ANALYSIS OF COMMUNICATION IN COLLABORATIVE DESIGN

Sachin H. Lokesh
Computer Science and Artificial Intelligence
Plaksha University
Mohali, Punjab 140306 India
Email: sachin.lokesh@plaksha.edu.in

Ashish M. Chaudhari
Institute of Data, Systems, and Society
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: amchaudhari@mit.edu

Joseph Thekinen[*]
Mechanical and Manufacturing Engineering
University of Calgary
Calgary, AB T2N 1N4, Canada
Email: joseph.thekinen@ucalgary.ca

Jitesh H. Panchal
Mechanical Engineering
Purdue University
West Lafayette, IN 47909
Email: panchal@purdue.edu

ABSTRACT

We address the problem of content analysis in text-based engineering design communication. Existing methods to characterize communication content in engineering design are manual or qualitative, which is tedious for large datasets. We formulate the characterization of communication messages as an intent classification task. We identify two intents—Intent 1 captures the presence and flow of information, Intent 2 captures specific topics about design parameters and objectives. We compare the predictive accuracy of convolutional LSTM, character-based convolutional LSTM, XLNet, and BERT models for the intent classification task. The results of our comparison show that the XLNet model predicts Intents 1 and 2 with 88% and 81% accuracy, respectively, on text data collected from 40 teams in a design experiment with university students. We analyze the differences in communication patterns between high- and low-performing teams. Time-series studies show that high-performing teams have more responsive communication and a higher consistency of information exchange

1 Introduction

Understanding communication in engineering systems design is necessary for effectively structuring interaction pathways [1, 2] and requirements analysis during the early stages of systems design [3,4]. In a typical project setting, most design activities occur through interpersonal meetings, project reports, and the exchange of data [5]. How much and what engineers communicate among each other influences the resource costs, project timings, and performance outcomes [6, 7]. There is a need to understand both the amount and content of interactions. A characterization of design communication would help identify dialog characteristics that result in superior outcomes and the characteristics that do not.

Existing approaches for the content analysis of design communication are time consuming and may not be feasible for large-scale systems design projects. Many researchers have observed and studied the textual records of design team dialog to better understand factors that drive overall team performance [8, 9]. Empirical analysis of such data requires manual labeling of digitized and transcribed textual records for predefined codes [10]. While this approach can garner important re-

[*] Address all correspondence to this author.

search insights, its application in large-scale text records or recurring design projects is limited due to the involvement of human raters. Any other form of interaction data such as frequency of communication and numerical data may be insufficiently rich for analysis purposes [7].

Recent advances in natural language processing (NLP) provide a promising avenue for automated content analysis. Typically, the NLP methods formulate the communication modeling problem into common tasks such as supervised intent classification [11] and unsupervised topic modeling [12, 13]. A basic intent classification problem involves tagging given text with the highest likelihood label from a pre-defined set of labels. On the other hand, a topic modeling problem provides an unsupervised learning of topics, but requires post-hoc human assessment of extracted topics. Both tasks require text data to train, with the additional requirement of a labeled dataset in some cases. A large training dataset can be beneficial but may not be necessary. Recent transfer learning methods allow a pre-trained NLP model to be transferred to an unseen dataset by retraining on a relatively small number of new observations [14]. Models such as Glove [15], BERT [16], and XLnet [17] are able to learn representations from a large unlabelled text corpus [18]. Learned representations of these models are adapted to a target task such as text classification through fine tuning or training a linear classifier using a small number of labeled data-sets.

Current design applications emphasize the rule-based topic modeling methods for identifying design concepts, understanding product usage, and managing expertise. Some examples include the synthesis of knowledge database from design records [19], mining keywords for directed ideation [20], identifying product affordances [21, 22], fault detection in requirements [23], and characterizing individuals' ability [24].

Existing work on modeling text communication and studying them in the context of team performance in engineering design focuses on quantitative factors of design dialog such as interaction frequency, number, and length of text messages [7, 9, 25]. Arguably, however, the content of the design dialog offers richer information for analysis [5, 26]. Despite the importance of dialog content, such studies have been limited to only structured communications such as design documents and project reports [27, 28]. Unstructured communication such as face-to-face meetings or short design dialog through emails is still unexplored in the context of team performance. This is largely due to the challenge of manually labeling large datasets.

We address this gap in this paper using state-of-the-art supervised NLP models. We compare Convolutional Long Short-Term Memory model, Character-based Convolutional Long Short-Term Memory, XLNet, and BERT models for classifying two pre-defined intents— one captures information flow, and the other captures the topic discussed. Further, the paper discusses how communications of design teams vary across high-performing and low performing teams using the predicted intents of design conversations. This work addresses the following questions:

1. What is the relative performance of different supervised natural language processing models for intent classification of design communication?
2. How do design intents of messages communicated between team members compare between high and low performing teams?

Our dataset consists of text-based conversations of 40 engineering student teams designing mechanical components of an automotive engine [29]. Each data sample consists of individual text messages exchanged between design actors. The training data consists of 528 data samples for Intent 1 and 495 data samples for Intent 2 (7.5%-8% of all data samples). We apply the NLP models and select the most accurate model using weighted F1-score and prediction accuracy. We found that the XLNet model achieved the highest accuracy — 88% on Intent 1 and 81% on Intent 2. We used the XLNet model to predict the intents for the rest of the dataset. The 40 teams are classified into high and low performing teams by measuring the quality of their final design. We perform a time-series analysis of communication intents to observe the differences between high and low performing teams.

Results show that high-performing teams have more responsive communication, maintain a consistent information flow throughout the project duration, and monitor objective values more frequently than low-performing teams. We discuss the implications of our findings in the context of design assistants that can perform intelligent interventions in collaborative design.

The rest of the paper is structured as follows. Section 2 describes the training data and selected communication intents. Section 3 details the selected NLP models and their network architectures. Section 4 presents the predictive accuracy results and the exploratory analysis of model predictions in the context of design performance. Section 5 discusses our observations on communication intents of high and low performing teams. Section 6 summarizes the conclusion of the study.

2 Overview of the Communication Data and the Intent Classification Task

2.1 Collaborative Design Experiment

The communication dataset corresponds to messages between student teams designing an automotive engine. The purpose of this design task is to simulate and observe a collaborative design process between distributed team members. The design task involves teams of five individually optimizing the dimensions of separate but mutually dependent engine components such as piston, connecting rod, crankshaft, flywheel, and piston pin. Each subsystem is characterized by two design variables, making the total number of design variables as 10. The subsystems are interdependent because they share some design variables between each other [29]. Each team member works on their assigned subsystem individually but can communicate with others through one-to-one chatboxes. The system objectives are to maximize the factor of safety and minimize the engine's total mass. The total mass is the sum of subsystem masses and the the overall factors of safety is the minimum of all factors of safety.

During the activity, the team members could exchange text messages with other members pairwise through one-to-one channels. A group discussion was not allowed. In total, 40 teams of 5 members each completed this design activity. The original experiment included different treatments based on whether designers used catalogs versus sequential design iteration and whether the global information like values of shared design variables was readily available to the team members. [29] Here, we merge communication data from different experiment conditions so that the training data for NLP models is robust and represents diverse design dialogues.

The collected communication data involves timestamped records including the sender role, receiver role and the content of text messages. The language variety of the communication data is mainly US English and all subjects are undergraduate engineering students. Table 1 describes a conversation between two subjects in the roles of piston and crankshaft. Out of 7691 available text messages, 35% messages had less than 20 words, and 32% of the messages contained between 20 to 40 words. The maximum length of messages was 204 words.

TABLE 1: A sample prose between piston and crankshaft designers

Sender	**Message**
crankshaft	Whats the effect of diameter on your results?
piston	Lower diameter, lowers my mass and FOS
piston	I am trying to get the lowest mass with a factor of safety around 2
crankshaft	Right now I have a total mass of 1.49 and FOS of 4.55
piston	that's a pretty high mass, any chamce you can bring it down and keep FOS just over 2?

2.2 Intent Description

The characteristics of the available data determine the selection of intents. Among formal and informal interactions, designers may discuss emerging inconsistencies and collective system goals, in addition to the issues in focused working, managerial decisions, and background chatter [8]. The communication in this study primarily involves interdisciplinary interactions between individuals working on different but interconnected design components. Therefore, we identify two intents to capture the information flow (e.g., information exchange about variables and objectives and who requests or provides that information) and design topics (e.g., design interdependence due to shared variables and sensitivity of the design objectives). We identify distinct classes for each intent. Each text message is characterized based on the class of intent they fall in.

The first intent (Intent 1) aims to understand the direction of information flow between the designers. A human rater classified each text message into one of three types based on this intent.

1. *Asking for information*: requesting values of shared design variables, e.g., "does a smaller value of (c) will result in an overall higher FOS?";
2. *Providing information*: sending values of own or shared design variables and subsystem performance, e.g., "the value of 45 seems decent, lower if we need to reduce mass"; and
3. *Other* (Neither sending nor receiving): background chatter, e.g., "all right, I can definitely work with that."

The second intent (Intent 2) captures the topic discussed in the text message. A human rater tagged the text messages depending on the type of topic discussed:

1. *Dependencies between design parameters*: how different design sub-components depend on each other, e.g., "My crankshaft depends on your bore diameter. I don't yet know how";
2. *Exploration of design parameter values*: finding an appropriate value for a design parameter, e.g., "How low can you go on the piston bore diameter?";
3. *Tradeoffs between objectives*: how to trade one improvement in one objective for other, e.g., "How appreciable is the change in the mass if we prioritize factor of safety" ;
4. *Monitoring objective values*: discussing the state of the system in terms of the values of system objectives, e.g., "Mass of 7.96 has been the best achieved thus far";
5. *Effects of design parameters on objectives*: understanding the strength and direction of design parameters on given objectives, e.g., "Just reduced thickness to reduce mass, increased FoS a bit";
6. *Selected design parameter values for objectives*: finding an appropriate value of a design variable for a specific objective, e.g., "The mass was 0.24 with the diameter to 70'.'; and
7. *Other*: background chatter, e.g., "yeah, that is not on me.".

The classes in Intent 2 are independent in the sense of axiomatic design [30]. A design team has designers working on interdependent components with input design parameters and output functional requirements (objectives). Due the interconnected nature of these components, the designers may discuss each others' design parameters and their effects on the objectives. On the output side, multiple conflicting objectives (e.g., mass and factor of safety) require discussions of trade-off. Finally, the classes are differentiated based on whether the communication is about qualitative relationships or numbers are being exchanges. This distinction is important if the goal is to extract the numerical data being exchanged.

Section 3 presents text pre-processing and supervised learning methods that operate on the interdisciplinary text messages and the aforementioned intents.

3 Methodology

3.1 Text Pre-processing

We adopted a series of steps to prepare the training data. First, we expanded the abbreviation and design symbols to train the model with the relevant vocabulary without losing information. Second, word tokenization is a way of separating words in the text messages into smaller units called tokens [31]. Third, removal of stopwords which do not add much value to the sentence, e.g., 'i', 'me', 'my', 'myself', 'we', 'our'. Fourth, we converted all text to lowercase. The text data collected in our design experiments had design symbols and abbreviations which were expanded to map with relevant vocabulary, e.g., tf=flywheelthickness, r2=length-diameter ratio. Finally, using Natural Language Toolkit [32] we perform lemmatisation to transform all the inflected forms of a word in the sentence to a root form, e.g., determined to determine, increasing to increase.

Among the labeled data samples, certain intent types of Intent 2 data represented fewer data. Table 2 provides the corresponding number of samples for each intent class after data augmentation. This imbalance influences a model to perform poorly on the classification of intents. The data augmentation [33] technique helps overcome the class imbalance problem by generating synthetic samples of minority classes. We performed additional data augmentation using back translation [34] where sentence were translated to French language and back to the English language, e.g."thats a pretty high mass, any chance you can bring it down and keep FOS just over 2?" was translated to French"c'est une masse assez élevée, n'importe quel chance de la réduire et de garder le FOS juste au-dessus de 2" ? and back to English "that's a pretty high mass, any luck getting it down and keeping the FOS just above 2?". We also used nlpaug library [35] to perform synonym replacement, e.g., "Can you send me current values" to "Can you send me present values", and keyboard augmentation, e.g., "reduce for weight savings" to "reduc el for weight savings".

TABLE 2: Summary of data labels of each intent type in the training and test data.

Intent 1: Information flow		
Class	**Training**	**Test**
Providing information	318	46
Asking for information	142	21
Other	68	9
Intent 2: Topic type		
Class	**Training**	**Test**
Dependencies between design parameters	65	9
Exploration of design parameter values	71	10
Tradeoffs between objectives	62	8
Monitoring objective values	78	12
Effects of design parameters on objectives	82	11
Selected design parameter values for objectives	59	9
None of the above	78	12

3.2 Supervised Natural Language Processing Models

The intent classification task aims to tag utterance or sentences with a predefined label. Intent classification models can be either rule-based and deep learning-based. The rule based methods aim to classify the intents using pre-defined rules and requires deep domain knowledge where as deep learning-based models aim to classifies the intents by learning the inherent patterns between text and their labels [36]. Classical NLP models require handcrafted features to be fed into a classifier for intent prediction. Using pre-trained word embedding and language models, deep learning based intent classification models are able to map the text vector into a low-dimensional feature vector and extract the key features to predict the intent [37]. We approach the intent classification task using supervised deep-learning models.

3.2.1 Convolutional Long Short-Term Memory

Zhu et al. [38] propose a Convolutional Long Short-Term Memory (CNNBiLSTM) model. Various hybrid models of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures learn to extract a local feature and context dependencies in a sentence. In this model, we train a hybrid Convolutional Long Short-Term Memory (CNN-LSTM) model on top of 100 dimensional (100 D) pre-trained word vector obtained from a pre-trained GloVe word embedding [15]. Table 3 presents the network architecture of the CNN-LSTM model.

One-dimensional CNN extracts local features of higher-

TABLE 3: Summary of the Convolutional Long Short-Term Memory (CNNLSTM) model with hyper-parameters

Layer Name	Output shape	Hyperparameters value
Embedding layer	((None, 128, 100))	[trainable=False]
Conv1D	((None, 128, 100))	[padding=same]
MaxPooling1D	(None, 64, 100)	[pool size=2]
Bidirectional(LSTM)	(None, 64, 256)	[dimension=128, dropout=0.3]
GlobalMaxPooling1D	(None, 256)	-
Dense	(None, 512)	[units=512, activation=relu]
Dropout	(None, 512)	[rate=0.4]
Dense	(None, number of intents)	[activation=softmax]

level phrases from design conversations. We used 100 filters and kernel size five as hyper-parameters for the CNN layer. The output from this CNN layer is passed through a one-dimensional max-pooling layer to reduce the dimension by retaining only the important features in the sequence. The output feature vectors are then fed through bi-directional LSTM which learns word dependencies and text structures of the design conversations from both the directions. The semantically rich vector output from a LSTM layer is passed through the global max pooling to get a single vector. The features from this output vector is then passed through the dense layer with 512 neurons, followed by dropout layer which prevents the over fitting during training by randomly turning of the inputs to 0 with a frequency of 0.4. In the last layer, we used the dense layer with softmax activation function to predict the intents of a given design conversation. During training the network we used categorical cross-entropy function as loss function and Adam optimizer (learning rate=0.0003) to minimize the loss.

3.2.2 Character Convolutional Long Short-Term Memory

The character-level convolutional neural network (Char-CNN) was first proposed by Zhang et al. [39]. The Char-CNN architecture does not require knowledge of the semantic or syntactic structure of the short sentence [40]. However, a design conversation is a short text where the context or semantic meaning would be lost if conversation were not observed over frame of a time. Therefore, we explored a hybrid Char-CNN and Long short term memory (LSTM) architecture to predict the intents of a design conversations.

Table 4 shows the network architecture for Char-CNN-LSTM model. The model takes a input character tokens of length 200 and transforms them to a vectors of vocab size dimension before passing them through a one dimensional (1D) CNN layers. We used 3 layers of convolutional network with 100 filters in each layer. The output from the each CNN layer is passed through an 1D max-pooling layer to reduce the dimension of a sequence while retaining important features. The resultant output feature vector is fed through bi-directional LSTM, which learns long range dependencies and bi-directional structures of characters. The output vectors from LSTM are then passed through the global max pooling to get a single vector representing the sequence of characters. The features from this output vector are then passed through a dense layer and followed by a dropout layer. We used a dense layer with softmax activation function as the last layer to predict the intents of a given design conversation. During training the model we used categorical cross-entropy function as loss function and Adam optimizer(learning rate=0.0004)to minimize the loss.

TABLE 4: Summary of the Character Convolutional Long Short-Term Memory (Char-CNNLSTM) model with hyper-parameters

Layer Name	Output shape	Hyperparameter values
Embedding layer	((None, 200, 59))	[trainable=True]
Conv1D	((None, 200, 100))	[Kernelsize =5, padding=same]
MaxPooling1D	(None, 100, 100)	[pool size=2]
Conv1D	((None, 100, 100))	[Kernelsize =4, padding=same]
MaxPooling1D	(None, 50, 100)	[pool size=2]
Conv1D	((None, 50, 100))	[Kernelsize=3, padding=same]
MaxPooling1D	(None, 25, 100)	[pool size=2]
Bidirectional(LSTM)	(None, 25, 256)	[dimension=128, dropout=0.3]
GlobalMaxPooling1D	(None, 256)	-
Dense	(None, 512)	[units=512, activation=relu]
Dropout	(None, 512)	[rate=0.4]
Dense	(None, number of intents)	[activation=softmax]

3.2.3 XLNet

The XLNet model was proposed by Yang et al [17]. XLNet is a generalized auto regressive language model which is an extension of the Transformer-XL model [41]. XLNet Model is pre-trained using an autoregressive method to learn the bidirectional contexts by maximizing the expected likelihood over all permutations of the input sequence factorization. BERT neglects dependency between the masked positions and suffers from a pretrain-finetune discrepancy, whereas XLNet model overcomes this limitation.

We used a pre-trained XLNetForSequenceClassification model from the Hugging face transformers library [42]. This sequence classification implementation of XLNet model consists of pre-trained XLNet base-cased model, with a sequence classification layer on top of the pooled output of the XLNet model. The XLNet-base-cased model has 12 layers, 12 attention heads in each attention layer and 768 hidden units which is a dimensionality of the encoder layers and the pooler layer.

The data-set was prepared for model training using the encode-plus method of XLNetTokenizer [43] where data was tokenized by splitting a text sequence into words or subwords using a Subword Regularization algorithm of SentencePiece [44] package and Special token [SEP] [CLS] were added at end of the tokenized sequence. For each tokenized input sentence, we created input ids, attention masks. The input Id and attention mask sequence are then padded to a max sequence length of 512. In the end, using data loaders, we passed the prepared engine design dataset to train the XLNetForSequenceClassification model for predicting the intent of the design conversation.

Upon some experimentation, we used the batch size of 2, max sequence length 512, Adam-optimizer with the learning rate of 3×10^{-5}, and eight epochs to train the model.

3.2.4 BERT

Bert is a general purpose language representation model based on the transformer architecture [16]. In this implementation, we used BERT-Base architecture which has 12-layer, 768-hidden-nodes, 12-attention-heads and 110 million parameters.

The engine design data was prepared for the model by tokenizing text sequences by splitting a text sequence into words or subwords using BertTokenizer. Speacial token [CLS] and [SEP] were added in the begging and end of the tokenized sequence. The tokenized sequence is then converted to ids through a lookup table and Padded to the maximum hyperparameter so each sentence is of same length. The BERT encoder is then fed with prepared input sequence data. Table 5 presents the details of the model architecture.

The data is processed within the BERT architecture through a self-attention layer and a feed-forward neural network of the encoders. In the end, each token will output an embedding of hidden size 768. In the end We fed the resulting embedding vector of [cls] token from the Bert model to a dropout layer and the output from a dropout layer to a dense layer. In the end we used the dense layer with softmax activation function for predicting the intents of design conversation. We trained the model for fifteen epochs with batch size of 16 using Adam optimizer at a learning rate of 1×10^{-5}.

TABLE 5: Summary of the BERT model with hyperparameters value

Layer Name	Output shape	Hyperparameters value
BertModelLayer	(None, 768)	-
Lambda	(None, 768)	-
Dropout	(None, 768)	[rate=0.5]
Dense	(None, 768)	[units=768, activation=”tanh”]
Dropout	(None, 768)	[rate=0.5]
Dense	(None, Number of intents)	[activation=”softmax”]

4 Results

4.1 Performance of Machine Learning Models

The dataset contains 7690 text message instances exchanged by the design offices across 40 design sessions. For Intent 1, we manually labeled 604 data samples (7.85%) which were split into 478 samples for training, 50 samples for validation, and 76 samples for testing. For Intent 2, we manually labeled 390 (5.07%) data labels and added 176 synthetic samples; the overall labeled data was split into 424 samples for training, 71 samples for validation and 71 samples for testing. Table 2 describes the distribution of data samples in the training and test set. We use two metrics to evaluate the performance of machine learning models:

1. *Accuracy*, which is the number of classifications a model correctly predicts divided by the total number of predictions, and
2. *Weighted F1-score*, which is the weighted average value of each class F1 scores considering their class support.

Recall, also known as the true positive rate (TPR), measures how many of the positive cases our model is able to correctly predict. Precision indicates how many positive predictions are true. Sections 4.1.1 and 4.1.2 describes the performance of machine learning models on Intents 1 and 2 respectively.

4.1.1 Intent 1: Classification of Information Flow

Table 6 shows the performance of machine learning models on classifying the information flow in text messages. The XLNet model provides the highest accuracy and F1 score for all three information labels. This shows that a large pre-trained XLnet model can better learn the patterns even when fine-tuned with a small label dataset. The Char-CNNBiLSTM model's better performance compared to CNNBiLSTM and BERT models highlights the importance of feature extraction at the character level using convolution operation when the number of classes are less.

4.1.2 Intent 2: Labeling Topic Types

Table 7 shows the performance of machine learning models on labeling Intent 2. Similar to Intent 1, the XLNet model gives the highest accuracy and F1 score compared to other models across all classes. This shows that the pre-trained XLnet model is able to better classify the topic type class even when there are few labelled samples. Compared to other models, the Char-CNNBiLSTM model has the lower accuracy and F1-Score for most classes. Further analysis is needed to understand the poor accuracy of character-level encoding when modeling many classes.

4.2 Time Series Study of Communication Intent

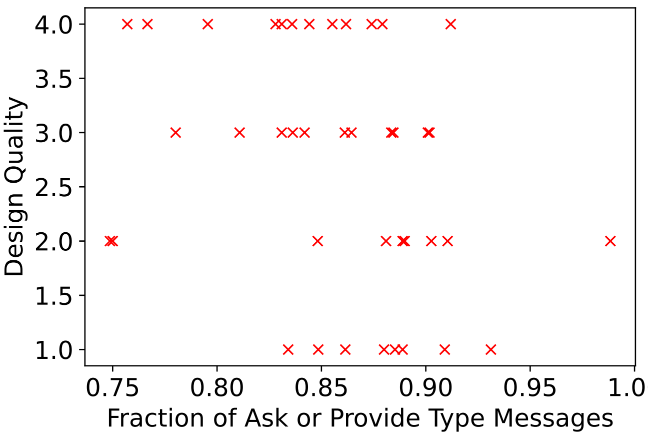
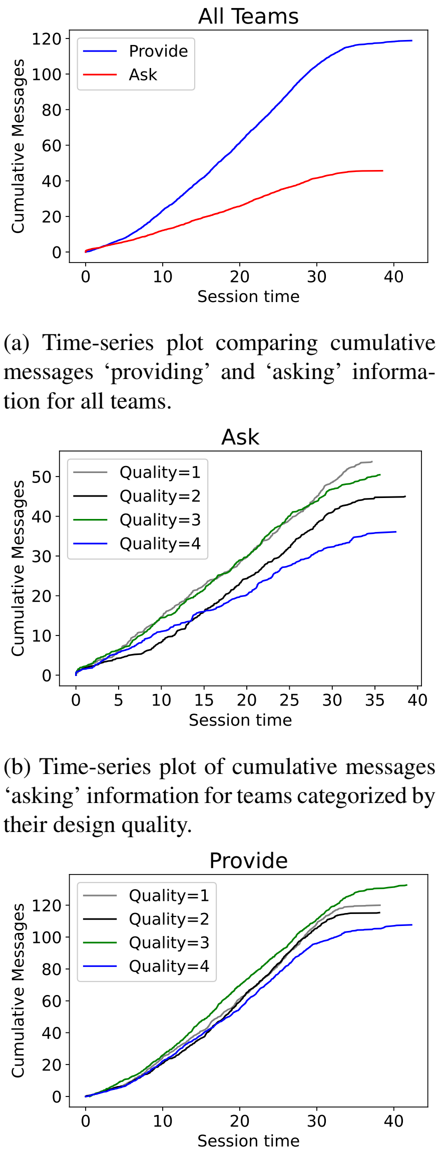


FIGURE 1: Fraction of all text messages exchanging information (either 'Asking for information' or 'Providing information') versus design quality for all 40 teams.

We observed that the XLNet model provides the highest accuracy. Therefore, we use this model to label the information flow (Intent 1) and topic type (Intent 2) for the remaining text messages. We use the resulting labels to study the effect of communication characteristics on the design quality. The teams are assigned a quality score ranging from one to four based on the system performance of the final design. The system performance is evaluated based on the factor of safety and the total mass of the engine. A design with higher system performance has a higher quality score. Table 8 summarizes key communication statistics for teams in each quality category. The column 'Average Messages' for a quality category is the average number of messages exchanged per team by all the teams in that quality category. The column 'Intent 1 Fraction' is the fraction of total messages either 'Asking' or 'Providing' information. The column 'Intent 2 Fraction' denote the fraction of total messages classified as one of the six topic types identified in Section 2.2. The column 'Intent 2 Fraction' ranged from 0.66 to 0.76, which means the topics we identified captured the topics discussed in more than two-thirds



(a) Time-series plot comparing cumulative messages 'providing' and 'asking' information for all teams.

(b) Time-series plot of cumulative messages 'asking' information for teams categorized by their design quality.

(c) Time-series plot of cumulative messages 'providing' information for teams categorized by their design quality.

FIGURE 2: Time-series plot of Intent 1 predictions.

TABLE 6: Performance of various machine learning models on information flow Intent type.

Metrics	CNNBiLSTM	CharCNNBiLSTM	XLNet	BERT
Accuracy	0.80	0.86	0.88	0.83
Weighted F1	0.80	0.85	0.88	0.82

TABLE 7: Performance of various machine learning models on Topic intent type.

Metrics	CNNBiLSTM	CharCNNBiLSTM	XLNet	BERT
Accuracy	0.76	0.68	0.81	0.70
Weighted F1	0.76	0.68	0.81	0.70

TABLE 8: Message frequency averaged over sessions with specific design quality.

Quality	No. of Teams	Average Messages	Intent 1 Fraction	Intent 2 Fraction
1	8	198	0.88	0.76
2	9	183	0.86	0.75
3	11	214	0.86	0.70
4	12	156	0.84	0.66

TABLE 9: Correlation analysis of intent 1 and design quality.

Parameter	Value
Spearman's ρ	-0.369
p-value	0.019

TABLE 10: Regression analysis of intent 1 and design quality.

Parameter	Value
R-squared	0.01
Slope coefficient	-6.88
p-value	0.048

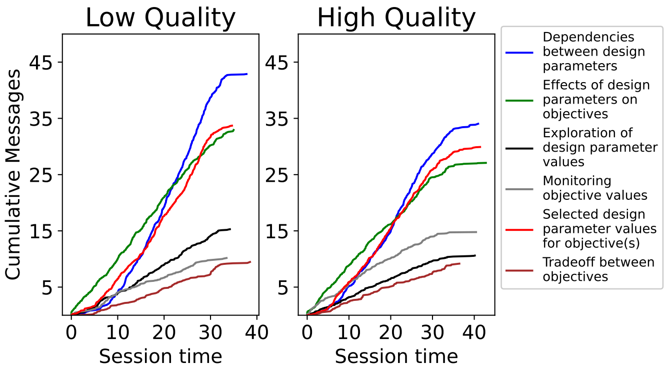


FIGURE 3: Time-series plot of Intent 2. 'Low Quality' teams are teams with quality = 1 or 2. 'High Quality' teams are teams with quality = 3 or 4.

of the messages.

The values of 'Intent 1 Fraction' ranged from 0.84 to 0.88. Therefore a majority of the messages are purposeful asking or giving information between the subsystem designers. The remaining messages, which neither ask nor provide information, are labeled as 'Other'. A few examples of messages classified as 'Other' include: 'Thank you', 'Sounds good', 'Okay', 'Got it. Most of these 'Other' type messages acknowledge receipt of the message, thereby giving feedback to the sender. Figure 1 shows the performance of individual teams against the fraction of messages classified as 'Other'. Tables 9 and 10 summarizes results from correlation studies and regression analysis for team performance vs intent type 'Other'. We observe that the higher-performing teams have a higher fraction (p-value of 0.02) of messages of the 'Other' type. This means that it is important to acknowledge and give feedback to other subsystems for efficient communication. Such feedback improves the shared mental state between subsystems.

We further studied time series of message characteristics to understand how the communication varied throughout the design session. Figure 2 shows the time series of information flow between the subsystems. Figure 2a shows the average cumulative messages of 'Provide' and 'Ask' type. From Figure 2a we observe that there is an equal number of messages providing and asking for information in the first 10-15% (around 5 minutes). This is possibly because the design process is in the exploratory stage and all the subsystems are eager to learn about other subsystems. An analysis of Intent 2 (Figure 3) shows that the main topic of communication in this exploratory stage is 'effects of design parameters on objectives'. The objectives depend on the design parameters controlled by all the subsystems. The design actors ask and provide information to understand the effect of design parameters by other subsystems on the system objectives.

Once the exploration phase passes, the messages primarily 'provide' information. Figures 2b and 2c compares the 'provide' and 'ask' information time-series of teams with each quality type. We observe that the slope of the cumulative message curve flattens in the last 15-20% (8 minutes) for the lower performing teams (Quality of 1 or 2). The lower-performing teams stall information exchange between subsystems towards the design deadline and focus on fine-tuning the local subsystem parameters. The higher-performing teams maintain information exchange even in the deadline phase, although at a much lower rate than in the main part of the experiment.

From Figure 3 we also observe that 'dependencies between design parameters' is the main topic of text messages for both low and high performing teams. This means the teams use the text messages as a way to fill the knowledge gap between subsystems. The higher performing teams monitor objective values more frequently than lower performing teams.

4.3 Limitations

There are limitations on the applicability of NLP models. The predictive accuracy of such models depends on their hyperparameters. Therefore, the training process requires tuning certain hyperparameters to achieve good performance. Further, it is necessary that the intents being evaluated in the classification task are mutually independent and exhaustive to ensure a low error rate. The number of intents may also influence the model accuracy especially given that the training sample size is small. Accordingly, the training data needs to be validated by multiple human raters as a prerequisite. Ambiguity in the selected intents or errors in the training data can propagate through model training and can result in reduced model accuracy and invalid predictions on unseen data.

5 Implications for Design Practice

Intent 1 in our analysis captures the direction of information flow in pairwise communication. Digitalization affects design activities and behavior of design actors, particularly in multidisciplinary design [45]. Smooth information exchange is vital for effective coordination when designers have limited knowledge outside of their discipline [3]. Increased knowledge gaps between disciplines due to disciplinary advances further highlight the need for timely information exchange. Smooth information exchange involves asking for missing information and promptly answering information requested from other disciplines. We capture the information flow between design actors (Intent 1) using the XLNet model with 88% accuracy on an engine design task. Future design assistants can separate messages 'asking' for information from general messages to capture the attention of the discipline from whom such information is requested.

An analysis of the text messages exchanged by all the 40 teams from the engine design experiment shows that more than 75% of all messages either 'provide' or 'ask' information. The remaining messages, labeled as 'Other', include messages that give feedback to the message sender (e.g., 'Thank you', 'sounds good', 'Okay', etc.). We observe that a higher fraction of messages in the high-performing teams are of the 'Other' type. Therefore, it is important to acknowledge or offer feedback to other disciplines in collaborative design. Though such messages do not provide additional information, it improves the shared mental models and consensus between subsystem disciplines forming effective interdisciplinary teams. Future design assistants can ensure that such feedback is provided by tagging the text messages. The exact critical mass or fraction of 'Other' type messages is a topic for future studies. However, the observations presented in this paper are a step in that direction.

Intent 2 labels the topic of conversation in the messages exchanged. As evidenced from Figure 3, the dependencies between design parameters and effects of design parameter values on objectives are, on average, the most frequently discussed topic in design communication. Design as a social process emphasizes that dialog across boundaries helps to realize dependencies and resolve inconsistencies so that consensus is reached [26]. The results agree with this social process view of engineering design. Additionally, time-series studies show that high-performing teams maintain a higher consistency in information exchange at the beginning and closing states of the design cycle. Low-performing teams abstain from information exchange towards the design deadline as they are too focused on optimizing their subsystem discipline. Future design assistants can flag and notify design subsystems when there is a stall in information exchange. Intent 2 time-series study shows that most of the discussion by high-performing teams towards the end of the design activity is dependencies-related as last-minute efforts to drive their system quality higher. The co-evolution of interdisciplinary communication and system quality is evident from this plot [7].

6 Conclusions

This paper studied content analysis in text-based engineering design communication, representing as an intent classification task and training natural language processing (NLP) models on small labeled dataset. The characterization of communication is necessary for identifying practical topics and interaction patterns. Identifying predefined intents and topics can support directed research efforts and analysis of legacy design records. The recent advancement in NLP capabilities offers an opportunity to automate content analysis from known topics/intents. This study demonstrates the applicability of state-of-the-art NLP models to design dialogs. The reliability testing on the engineering students' design communication reveals the high predictive accuracy despite small sample size of labeled data. The XLNet offers the highest average accuracy across different intents. This model is pretrained using a generalized autoregressive method and learns bidirectional contexts in a dialog [17].

Future research can further test supervised learning algorithms for improved accuracy, e.g., calibrating the number and diversity of intent classes and the optimal number of labeled samples. Additionally, future work requires understanding how pre-trained NLP models can accurately predict intents for other data sources. Semi-supervised learning algorithms may improve transferability by incorporating expert knowledge with available data for effective content analysis [46]. Establishing the reliability of NLP methods for characterizing design communication will help build text-based design assistants in the future. Tagged messages from the past interaction [12] can enable textual suggestions to individual designers.

ACKNOWLEDGMENT

The authors acknowledge the contribution of Bhavika P Jain and Joseph Thomas in preparing the training dataset and preliminary analysis.

REFERENCES

[1] Eppinger, S. D., 2002. Patterns of product development interactions. MIT Engineering Systems Division Working Papers. ESD-WP-2003-01.05.

[2] Sosa, M. E., Eppinger, S. D., and Rowles, C. M., 2004. "The misalignment of product architecture and organizational structure in complex product development". *Management science,* ***50***(12), pp. 1674–1689.

[3] Kvan, T., 2000. "Collaborative design: what is it?". *Automation in construction,* ***9***(4), pp. 409–415.

[4] McGowan, A.-M. R., Seifert, C. M., and Papalambros, P. Y., 2012. "Organizational influences on interdisciplinary interactions during research and design of large-scale complex engineered systems". In 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 17 - 19 September, Indianapolis IN, AIAA/ISSM.

[5] Tenopir, C., and King, D. W., 2004. *Communication patterns of engineers*. John Wiley & Sons, Hoboken, New Jersey.

[6] Meluso, J., Austin-Breneman, J., and Shaw, L., 2019. "An agent-based model of miscommunication in complex system engineering organizations". *IEEE Systems Journal,* ***14***(3), pp. 3463–3474.

[7] Chaudhari, A. M., Gralla, E. L., Szajnfarber, Z., and Panchal, J. H. "Co-Evolution of Communication and System Performance in Engineering Systems Design: A Stochastic Network-Behavior Dynamics Model". Vol. Volume 6: 33rd International Conference on Design Theory and Methodology (DTM). V006T06A048.

[8] Snider, C., Škec, S., Gopsill, J., and Hicks, B., 2017. "The characterisation of engineering activity through email communication and content dynamics, for support of engineering project management". *Design Science,* ***3***, p. e22.

[9] Piccolo, S. A., Maier, A. M., Lehmann, S., and McMahon, C. A., 2019. "Iterations as the result of social and technical factors: empirical evidence from a large-scale design project". *Research in Engineering Design,* ***30***(2), pp. 251–270.

[10] Cagan, J., Dinar, M., Shah, J. J., Leifer, L., Linsey, J., Smith, S., and Vargas-Hernandez, N., 2013. "Empirical studies of design thinking: Past, present, future". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 55928, American Society of Mechanical Engineers, p. V005T06A020.

[11] Pilny, A., McAninch, K., Slone, A., and Moore, K., 2019. "Using supervised machine learning in automated content analysis: An example using relational uncertainty". *Communication Methods and Measures,* ***13***(4), pp. 287–304.

[12] Chatterjee, A., and Sengupta, S., 2020. "Intent mining from past conversations for conversational agent". *arXiv preprint arXiv:2005.11014*.

[13] Perkins, H., and Yang, Y., 2019. "Dialog intent induction with deep multi-view clustering". *arXiv preprint arXiv:1908.11487*.

[14] Goyal, A., Metallinou, A., and Matsoukas, S., 2018. "Fast and scalable expansion of natural language understanding functionality for intelligent agents". *arXiv preprint arXiv:1805.01542*.

[15] Pennington, J., Socher, R., and Manning, C. D., 2014. "Glove: Global vectors for word representation". In Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

[16] Devlin, J., Chang, M., Lee, K., and Toutanova, K., 2018. "BERT: pre-training of deep bidirectional transformers for language understanding". *CoRR,* ***abs/1810.04805***.

[17] Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V., 2019. "Xlnet: Generalized autoregressive pretraining for language understanding". *CoRR,* ***abs/1906.08237***.

[18] Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T., 2019. "Transfer learning in natural language processing". In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18.

[19] Sundararajan, V., 2006. "Constructing a design knowledge base using natural language processing". In ASME International Mechanical Engineering Congress and Exposition, Vol. 47748, pp. 401–407.

[20] He, Y., Camburn, B., Liu, H., Luo, J., Yang, M., and Wood, K., 2019. "Mining and representing the concept space of existing ideas for directed ideation". *Journal of Mechanical Design,* ***141***(12).

[21] Hou, T., Yannou, B., Leroy, Y., and Poirson, E., 2019. "Mining changes in user expectation over time from online reviews". *Journal of Mechanical Design,* ***141***(9).

[22] Suryadi, D., and Kim, H. M., 2019. "A data-driven approach to product usage context identification from online customer reviews". *Journal of Mechanical Design,* ***141***(12).

[23] Chen, C., Mullis, J., and Morkos, B., 2021. "A topic modeling approach to study design requirements". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 85383, American Society of Mechanical Engineers, p. V03AT03A021.

[24] Ball, Z., and Lewis, K., 2020. "Predicting design performance utilizing automated topic discovery". *Journal of Mechanical Design,* ***142***(12), p. e4.

[25] Thekinen, J., and Grogan, P. T., 2021. "Information exchange patterns in digital engineering: An observational study using web-based virtual design studio". *Journal of Computing and Information Science in Engineering,* ***21***(4).

[26] Bucciarelli, L. L., 1994. *Designing engineers*. MIT press, Cambridge, MA.

[27] Dong, A., Hill, A. W., and Agogino, A. M., 2004. "A document analysis method for characterizing design team performance". *J. Mech. Des.,* ***126***(3), pp. 378–385.

[28] Gyory, J. T., Cagan, J., and Kotovsky, K., 2019. "Are you better off alone? mitigating the underperformance of engineering teams during conceptual design through adaptive process management". *Research in Engineering Design,* ***30***(1), pp. 85–102.

[29] Chaudhari, A. M., Gralla, E. L., Szajnfarber, Z., Grogan, P. T., and Panchal, J. H., 2020. "Designing representative model worlds to study socio-technical phenomena: A case study of communication patterns in engineering systems design". *Journal of Mechanical Design,* ***142***(12), p. 121403.

[30] Suh, N. P., 1998. "Axiomatic design theory for systems". *Research in engineering design,* ***10***(4), pp. 189–209.

[31] Webster, J., and Kit, C., 1992. "Tokenization as the initial phase in nlp". pp. 1106–1110.

[32] Bird, S., Klein, E., and Loper, E., 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[33] Wei, J. W., and Zou, K., 2019. "EDA: easy data augmentation techniques for boosting performance on text classification tasks". *CoRR,* ***abs/1901.11196***.

[34] Yin, T. Translate.

[35] Ma, E., 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

[36] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J., 2020. "Deep learning based text classification: A comprehensive review". *CoRR,* ***abs/2004.03705***.

[37] Sezerer, E., and Tekir, S., 2021. "A survey on neural word embeddings". *CoRR,* ***abs/2110.01804***.

[38] Zhou, C., Sun, C., Liu, Z., and Lau, F. C. M., 2015. "A C-LSTM neural network for text classification". *CoRR,* ***abs/1511.08630***.

[39] Zhang, X., Zhao, J. J., and LeCun, Y., 2015. "Character-level convolutional networks for text classification". *CoRR,* ***abs/1509.01626***.

[40] Londt, T., Gao, X., Xue, B., and Andreae, P., 2020. "Evolving character-level convolutional neural networks for text classification". *CoRR,* ***abs/2012.02223***.

[41] Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R., 2019. "Transformer-xl: Attentive language models beyond a fixed-length context". *CoRR,* ***abs/1901.02860***.

[42] Wolf, T. Thomas wolf.

[43] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M., 2020. "Transformers: State-of-the-Art Natural Language Processing". Association for Computational Linguistics, pp. 38–45.

[44] Kudo, T., and Richardson, J., 2018. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing". *CoRR,* ***abs/1808.06226***.

[45] Zhang, G., Raina, A., Cagan, J., and McComb, C., 2021. "A cautionary tale about the impact of ai on human design teams". *Design Studies,* ***72***, Jan, p. 100990.

[46] Van Engelen, J. E., and Hoos, H. H., 2020. "A survey on semi-supervised learning". *Machine Learning,* ***109***(2), pp. 373–440.