A Hybrid Model to Classify Sudden Topic Change, Misunderstanding and Non-understanding in Human Chat-bot Interaction

Maitreyee Tewari, Monika Jingar, and Suna Bensch

Umeå University, Umeå, Sweden
{maittewa,monikaj,suna}@cs.umu.se

Abstract. In a natural dialogue, humans can handle misunderstanding, non-understanding, and sudden topic change integrally. An essential aspect of human-machine interaction is natural language understanding (NLU). This work proposes a hybrid model for NLU combining feature extraction with indicator classes (syntactic tokens and sequences) and semantic similarity for automatic labelling and a deep CNN learning model to integrally detect a sudden topic change, misunderstanding and non-understanding. The results report a significant improvement for the convolution model compared to the baseline multi-layer perceptron model for the classification task.

Keywords: Non-Understanding; Misunderstanding; Sudden Topic Change; Syntactic Tokens and Sequences; Cosine Similarity; Convolution Neural Network; Dependency Parsing; Miscommunication Detection; Hybrid Model

1 Introduction

Natural language dialogues play a significant role in exchanging meaningful information between individuals (human-human) during an interaction [10]. Natural language understanding (NLU) in dialogue management systems (DMS) enable machines to understand human language. Agreeing with [15], we view misunderstanding, non-understanding, and sudden topic change as natural events that occur in a dialogue. Thus, we assume that they are diversions, introduced by dialogue participants, that may prolong or change the course of a dialogue. However, if not managed and detected, it may lead to breakdown in dialogues. We assume that they can evidence themselves with syntactic and semantic features for pairs of dialogue utterances, among others. For example *clarification questions, repetitions, confirming* or asking for an *explanation*.

Hence, this work investigates features (syntax and semantics) in transcribed dialogues for the evidence of patterns for the three cases: *sudden topic change, misunderstanding, and non-understanding*. The novelty of our work includes an emphasis on patterns of 'sudden topic change', that are essential to detect, because they often lead to dialogue breakdown, misunderstanding and non-understanding.

The contribution of this work is the following: based on expert human heuristics and previous literature, determining the indicators (syntactic tokens and sequences)

and using cosine similarity for word2vec to automatically label misunderstanding, non-understanding, and sudden topic change. And a CNN classifier that detects the aforementioned miscommunication types.

The paper is structured as follows: Section 2 presents a summary of related work, while Section 3 provides some background on miscommunication types. Section 4 presents method with feature engineering and finding out cosine similarity for word2vec representation of specific syntactic tokens from utterances and highlights the architecture of the baseline and the CNN model. The baseline and the CNN model results can be found in Section 5. This article concludes with some discussion and final remarks in Section 6.

2 Related Work

In the research community, the DMS handles the miscommunication types with error or recovery mechanisms, often combining features from the NLU, DM and ASR [18,19]. Other works have explored object-based methods [12], state-based methods [22], or utilisation of dialogue acts [11]. Theoretical base in particular for *misunderstanding* was proposed by Fraser [4]. In [3], the authors analysed the different aspects to handle linguistic misunderstanding. Abductive theory [20] was used for identification and repairing of misunderstanding. In [8] authors proposed plan based models for handling non-understanding and misunderstanding.

This work falls close to following articles: in [1] the authors identify patterns in dialogues for predicting misunderstanding. Their in-depth manual analysis consisted of: agent's *inability* to adapt to the expectation, *evidence* that misunderstanding has occurred, *correction mechanisms* used and the *outcome* of the misunderstanding such as resolved or unresolved. In [21], authors proposed the exploration of topic transition, keyword extraction and communication function labels for the detection of different types of miscommunication types.

Authors in [13] presented a support vector machine (SVM) based classifier to detect misunderstanding. From manually annotated corpora, they extracted automatic word spotting and semantic similarity based features. In [6], authors presented an analysis of hot-spots of misunderstanding in DMS. Manually labelled dialogues (problematic and non-problematic) were used to label the utterances as speech errors, out-of-domain utterance, and back-end errors. More recently, researchers have focused on handling miscommunication with neural network models [9]. Authors in [16] provided computational models for miscommunication based on syntactic and phrasal patterns. This work also utilises the knowledge from previous works regarding syntactic patterns and semantic features to detect three miscommunication types: sudden topic change, misunderstanding and non-understanding, and the next section briefly explains them.

3 Background

We assume that during natural dialogue, participants talk spontaneously about topics such as people, things, ideas, opinions etc. In those dialogues, a smooth transition is often due to the previous knowledge and cooperation among the participants. However,

it is not uncommon that the participants also change the topic of discussion *suddenly*. For instance, in Table 1 the top example is a *sudden topic change*, where the system (S), instead of responding to the topic of *humanity*, suddenly changes the topic to *Ebola fighters*.

A *non-understanding* occurs when one participant in a dialogue is unable to create any interpretation of the immediate previous utterance. The middle example in Table 1 is of a non-understanding where, a user (U) asks a question to which the system (S) indicates that it did not understand by asking a question, which (U) responds by a clarification.

A *misunderstanding*, can happen when 'a participant **misinterprets** one or more previous utterances of another participant'. A misunderstanding can be detected at different stages [3], or by the participant recognising its occurrence [8], namely *self or other* misunderstanding. The bottom example in Table 1 illustrates a misunderstanding, where (S) interprets *U*'s previous utterance as information instead of a question, and when *U* provides a negative feedback then *S* realises that it misunderstood *U* and indicates it by the keyword *ohh* and then provides the information that *U* wanted to know.

To formalise the patterns indicating such miscommunications, we explore the following research question: *How to model detection of miscommunication types for classification algorithms based on syntactic and semantic features?* The overall goal of this research is to build hybrid models for detecting and managing different miscommunication types integrally with-in a DMS. To this end, automatic detection of sudden topic change, misunderstanding and non-understanding has been done and is presented here. The detection is based on the extracted indicators of (syntax and semantics types) and the immediate history that is, adjacent pairs or *n*-grams of utterances. The ground truth for the selection of the indicator classes is based on empirical analysis by one of the authors on a subset of a publicly available corpora with dialogue breakdown and previous works by [21,6,13].

4 Methodology

4.1 Corpora

Data aggregation for dialogues is a time consuming process in general and specifically in case for miscommunication types is very challenging, because of its complexity and sparsity in human and chat-bot dialogues. This work used the publicly available corpora from 'Dialogue Breakdown Detection Challenge (DBDC3)' [7]. The data can be accessed here [1]. DBDC3 is an aggregation of 415 dialogues with close to $10,000$ utterances from following sources: Conversational Interface Challenge (CIC), a chatbot developed at Moscow Institute of Physics and Technology (YI) and Workshop and Session Series on on Chatbots and Conversational Agents (WOCHAT), which consisted of two different types of dialogue management systems (TickTock [23] and IRIS [2]).

DBDC3 contained 100 dialogues from **TickTock**, which is a conversational agent for casual dialogues designed for prolonged interaction. TickTock's DMS is based on engagement model (defined by a scale: strongly disengaged, disengaged, engaged,

[1] https://dbd-challenge.github.io/dbdc3/data/

Turn No.	Sudden topic change
1	U: Do you hate humanity?
2	S: These Ebola fighters are nothing short of an absolute best?
3	U: Ebola is not connected to the topic.
Turn No.	Non-understanding
1	U: Can you summarise the text?
2	S: What do you mean?
3	U: I mean telling the general idea from the texts.
Turn No.	Misunderstanding
1	U: Do you know what will be for lunch today?
2	S: What?
3	U: I do not know, that is why I am asking you.
4	S: Ohh, I guess some pasta.

Table 1: Illustration of sudden topic change (top example), non-understanding (middle example) and misunderstanding (bottom example).

strongly engaged) that assesses user engagement and modifies its response, for instance by shifting the topic if the user seems less involved. From **IRIS** DMS, there were again 100 dialogues, which is a chat oriented DMS. It's framework is based on vector space and lies in the category of example-based DMS. From CIC [2] there were 115 dialogue samples, which is a collection of human-chatbot conversations aggregated during human evaluation round of CIC in 2017. And from **YI** chatbot [3] 100 dialogues were collected for the DBDC3 challenge, where Amazon Mechanical Turk (AMT) workers were assigned to have conversations with YI chatbot. The aggregation of all the above dialogues from TickTock, IRIS, CIC, and YI is the DBDC3 corpora, and it came with dialogue breakdown labels, provided by the crowd-sourced workers from AMT or CrowdFlower[4].

4.2 Feature Engineering

For the extraction of the indicators *syntactic tokens and sequences* and to compute semantic similarity between adjacent pairs, dependency parsing [5] was used. A dependency parser can generate syntactic structures (we refer as dependency graphs) of utterances based on dependencies between its words (entities). These dependencies are asymmetric relations such as noun-subject, direct-object, preposition and auxiliary verbs. The three main entities of the dependency graph are: *root*, *head* and *dependants*. Figure 1

[2] http://convai.io/data/
[3] http://ipavlov.ai/
[4] https://www.crowdflower.com/

illustrates a dependency graph, where *buy* is the *root* and the *head* while all the other words *should, we, groceries, tomorrow* are the dependants. On how we use dependency graphs and their entities is explained next.

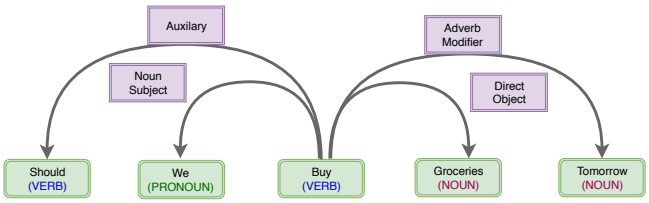


Fig. 1: A dependency graph for an utterance, representing dependency relations by text on the arcs and POS tags below each word.

Indicator classes : Most of the work [15,3,13,6] has already confirmed that explicit misunderstandings and non-understandings have sequences/patterns of keywords. Our method extracts syntactic features and categorising them under possible classes of: clarifications, contradictions, negative feedback, low topic similarity for adjacent pairs, and question-question pairs). Then a sequence of a combination of these classes of indicators is then used to determine one or the other miscommunication types.

For example, an utterance u is labelled as a sudden topic change when the similarity between topics for two adjacent utterances $(t, t+1$ or $t-1, t)$ is low and both of them are questions. An utterance u with a clarification question or a pair of $u, u+1$ has a clarification and or negative feedback can be considered as a non-understanding. A negative feedback preceded or succeeded by contradictory keywords can be a misunderstanding.

In order to determine the indicators following syntactic tokens (POS tags and dependency relations) from dependency parsing graphs were used: nouns, direct-object, indirect-object, interjection, adverb, adjective, auxiliary verb, coordinating-conjunctions, pronoun, adverb, and negation modifier. Dependency graphs also allows the extraction of sequences of syntactic tokens (auxiliary and tokens on the left or right, subject object verb, negative modifier and left or right tokens etc). A negative sequence is of the form $(token_1, right_{neigh1})$ where $token_1$ has **negative modifier** relation (not, donot, cannot, would not) with other words in an utterance, and $right_{neigh1}$ is the immediate right neighbour of the negative modifier. Examples include 'not know', 'donot believe', 'donot want' etc. and can be categorised under the indicator *negative feedback*.

We also extracted questions of type *(what, why)* using syntactic tokens (pronoun and adverb) from each utterance in a dialogue. Subject-object-verb (SOV) are of the form $(token_1, token_2, token_3)$, we use both complete and incomplete instances of it. For example 'are-you-talking', 'did you', 'can-you-repeat' can be categorised as *clarification* indicator class. We propose auxiliary sequences as three lexical tokens starting from the auxiliary verb and its two right neighbours $(aux_{verb}, right_{neigh1}, right_{neigh2})$ such as *i*) 'did not know', *ii*) 'did not get', *iii*) 'do you mean', *iv*) 'you were talking'. The *ii*) and *iii*) sequences combined with a question can be categorised under *clarification*, *iii*)

when combined with 'but' and $object - verb$ sequence of 'i thought' when combined with iv) is a *contradiction* indicator class.

The following extracted syntactic tokens (coordinating conjunctions, proper nouns, direct-object, indirect-object, interjection, adverb, adjective) were used to find semantic similarity using cosine similarity measure explained next. Table 2 presents a comparison between indicators that were manually determined and those extracted by dependency parsing.

Utterance-Sequences	Manual-Features	Extracted-Features	Indicator-classes
(1.U,2.S)	(who, Lorri), (you-talking-abut)	(you, who, Lorri), (you-talking, actress-Lorri)	Clarification, Question
(2.S,3.U)	(you-talking, actress-Lorri) (was-not-talking, about-Lorri)	(you-talking, actress-Lorri), (not-talking, i-Lorri-talking, you)	Negative-Feedback, Contradict

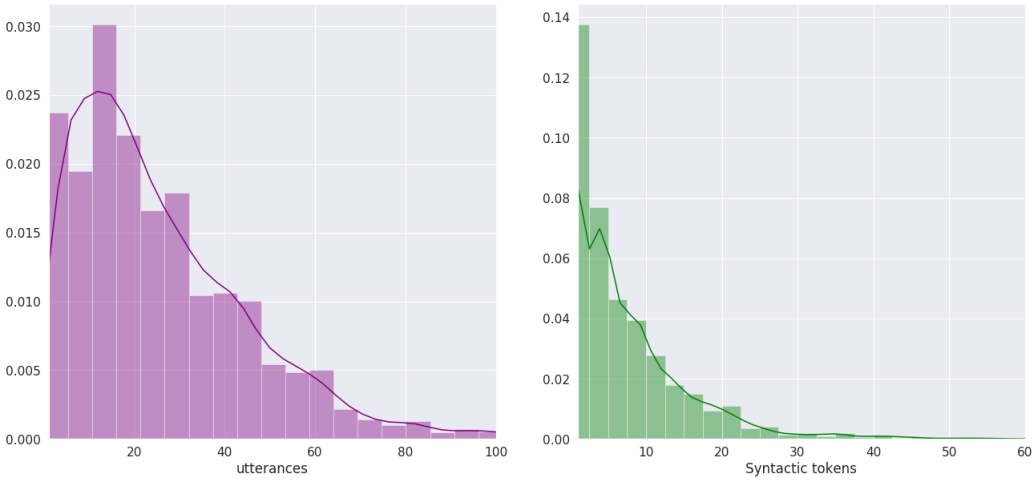
Table 2: The first column gives the adjacent pairs that contribute to a miscommunication types. The second column gives the indicators detected during manual analysis. The third column gives the indicators extracted from graphs generated by dependency parsing and the fourth column gives the class of indicators.

Figure 2 shows the distribution of utterances and their length, and the extracted syntactic tokens. Figure 3 provides the distribution for the extracted sequences: subject-object-verb, auxiliary verb sequences and negative token sequences.

Cosine Similarity with syntactic tokens : To automate the labelling of the three miscommunication types, one criteria especially for labelling an utterance with a sudden topic change is the dissimilarity between topics in adjacent turns. With dependency parse graphs following (coordinating conjunctions, nouns, direct-object, indirect-object, proper noun, interjection, adverb, adjective) syntactic tokens are extracted and then transformed to word2vec [14] space. For more than one vector for an utterance the vectors are averaged and a cosine similarity for a pair of such vectors is computed. A cosine similarity gives the distance between the projection of the vectors in the word2vec space.

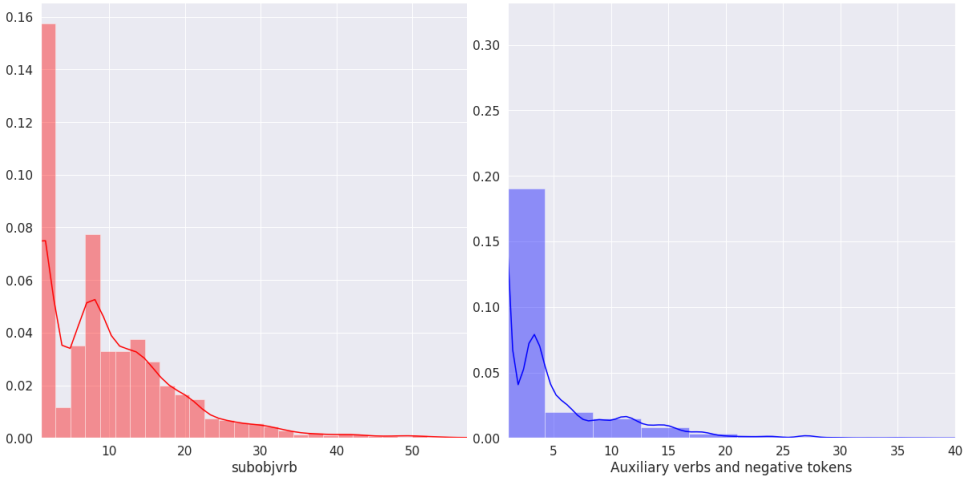
The cosine similarity between two vectors $\mathbf{w_x}$ and $\mathbf{w_y}$ is given as: $Sim(\mathbf{w_x}, \mathbf{w_y}) = \frac{\mathbf{w_x} \cdot \mathbf{w_y}}{|\mathbf{w_x}||\mathbf{w_y}|}$ where, $\mathbf{w_x}$ and $\mathbf{w_y}$ are m-dimensional vectors over the set of words $W = \{w_1, \ldots, w_m\}$. A cosine similarity is between $[-1, 1]$ where -1 indicates vectors of words are opposite and 1 indicates the same vectors. For finding out cosine similarity between adjacent utterances we use the syntactic tokens (nouns, direct-object, indirect-object, proper noun, interjection, adverb, adjective) for two adjacent utterances. Table 3 illustrates the cosine similarity computed for vectors of syntactic tokens for adjacent pairs from the DBDC3 corpora that this work uses. First, the syntactic tokens are extracted (name, bot, you, people, mars, comment), then transformed to vectors, which are

Fig. 2: (a) Distribution of dialogue utterances. (b) Distribution of syntactic tokens with adverbs, interjections, coordination conjunctions



(a) (b)

Fig. 3: (a) Distribution of subject-object-verb (SOV) sequences. (b) Distribution of auxiliary sequences and negative tokens



(a) (b)

then averaged for each pair if there was more than one token. Finally, cosine similarity on transformed vectors is computed.

4.3 Corpora Labelling

The above feature engineering with indicator classes and topic shifts is the basis for labelling of the corpora. The indicator classes consist of; contradiction, clarification, what/why questions, and negative feedback. Cosine similarity for adjacent pairs is categorised as; 'topic opposite' for $(<= -1)$, 'topic none' for $(= 0)$, 'topic not similar' for $(> 0.0, <= 0.3)$, 'topic similar' for $(> 3, < 7)$, 'topic same' for $>= 7$. This categorisation was done after a manual analysis by one of the author on a random sample of the corpora with the cosine scores and we found that the cosine similarity scores should be close to 1 for the words to be the same, topic none indicates that there were no topics available. After extracting the indicator classes and determining the cosine similarity between the vectors of adjacent turns, the labels are assigned to each turn with following patterns of features:

- Sudden topic change: ($bi-$grams) or adjacent pairs of indicator classes (of type question) and cosine similarity ('topic opposite' and 'topic not similar').
- Non-understanding: ($tri-$grams) of indicator classes (question and or clarification and or negative feedback) and cosine similarity (topic none)
- Misunderstanding ($five-$grams) of utterances with all or some indicator classes (contradiction and or negative feedback and or question) and cosine similarity ('topic similar' and or 'topic none')

Dialogue Utterances	Syntactic Tokens	Cosine Similarity
S1: My name is bot U1: Nice to meet you bot.	name, bot you, bot	**0.79** (same topic)
U1: When will people go to mars S1: Your comment is irrelevant.	People,Mars Comment	**0.27** (Different topic)

Table 3: An excerpt of adjacent utterances and the cosine similarity between their syntactic tokens (nouns, direct object and indirect object).

4.4 Supervised Model

The above mentioned indicator classes and cosine similarity were used as features to classify adjacent pairs under (*sudden topic change (stc), non-understanding (non-understand), or misunderstanding (misunderstand)* and ones without any indicator classes and comparatively high semantic similarity as *normal*. As input to the classifiers,

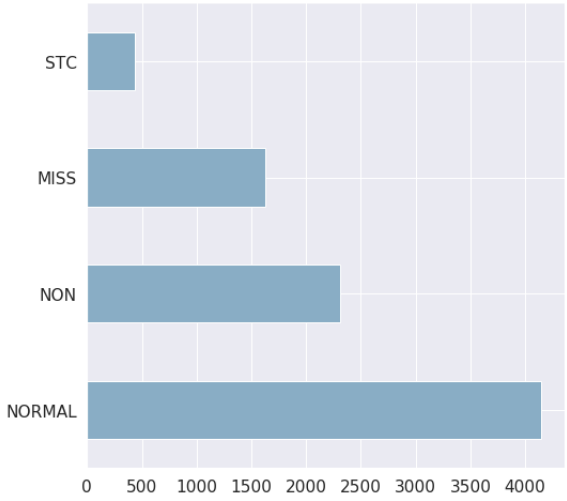


Fig. 4: Number of utterance samples for each miscommunication types (sudden topic change, misunderstanding, non-understanding) after the automatic labelling of the corpora with combination of indicator classes and topic shift categories.

we use several combinations of features. 1) *uni*-gram of utterances. 2) *uni*-gram of utterances, indicator classes classes and cosine similarity categories. 3) *bi*-gram of utterances. 4) *bi*-gram of utterances, indicator classes classes, cosine similarity categories.

As an output, the model predicts one of the four classes: *stc, non-understand, misunderstand, and normal*. We run experiments with above features and selected (*bi*-gram of utterances, indicators and cosine similarity) because of their high density and performance compared to the rest of the above combinations. We consider a multi-layer perceptron as our baseline model and compare it to our CNN model. Multi-layer perceptron was used as the baseline for two reasons, first it has less training time, and second it's simplistic nature. Initial experiment on the sequential model gave poor performance, so we ran another set of experiments with PCA [17], which performed better. Before moving ahead, we present the architecture for the PCA based baseline and the CNN model, for further research and easy reproduce-ability. The architecture of the sequential model consists of linearly stacked, fully connected 100 dense units for input layer and hidden layer. We perform over and under sampling to the input features. It allows a normal distribution and avoids bias towards the most frequent class of labels. The selected features are split with $60:40$ ratio for training (around $11,000$ utterances) and validation (around 7900 utterances) sets. We train the model with 100 epocs, batch-size 10 and early stopping rate of 0.5.

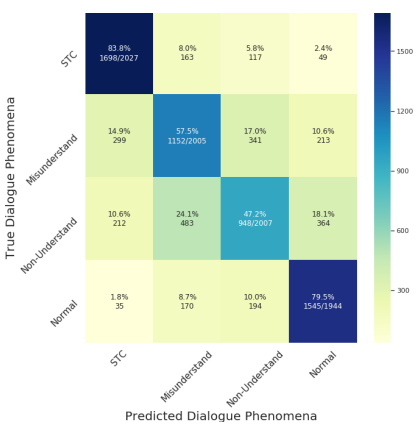
CNN Model: are regularised by convolving filters and applied to local features, meaning that they learn complex patterns from the features and transform them to simpler forms. Our CNN model has the following architecture: as input, we take the embedding space of the words in utterances, followed by three hidden layers: 1-D convolution layer with 100 *convolving filters*, *global maxpool* and a dense output layer

with 100 units. The input features are transformed to embedding space, tokenized based on a selection of vocabulary tokens from the corpora, and transformed to sequences. The selected features are split with $60:40$ ratio for training (around $11,000$ utterances) and validation (around 7900 utterances) sets. We present results of the convolution model for 100 epocs, batch size of 128 and early stopping rate of 0.5. We use *categorical cross entropy* and *Softmax* as loss functions and *adam* for optimising both the models.

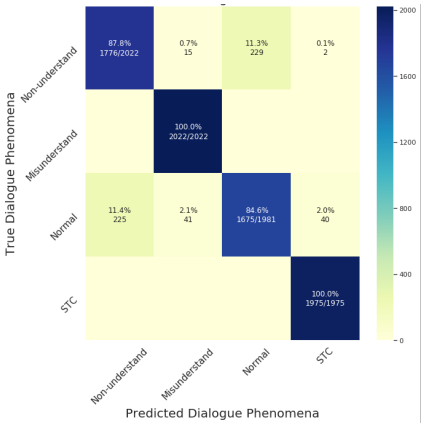
5 Results

This section presents the results of the PCA based baseline model and the CNN model. Both the models use the same features, loss function and optimisation function. The training loss of PCA based baseline model descended till (0.9). On the other hand, the CNN model achieved significant drop in the loss value of up-to (0.2).

The CNN model achieved 98% accuracy on the training set and 93% on the test set, compared to PCA based baseline model that had 76% accuracy on training set and 73% on the test set. Figure 5a and Figure 5b presents the confusion matrix for the PCA based baseline and CNN model respectively. A confusion matrix gives true (in the corpora) versus predicted classes (by the model). The PCA based baseline model can predict the sudden topic change with 83%, misunderstanding with 57%, non-understanding with 42% and normal with 79% accuracy. The CNN model predicts with higher accuracy all the miscommunication types (non-understanding 87%, misunderstanding 100%, sudden topic change 100%, and normal with 84%).



(a) Confusion matrix illustrating the performance of baseline model in predicting the miscommunication types and normal class labels.



(b) Confusion matrix illustrating performance of the CNN model in predicting the miscommunication types and normal class labels.

Fig. 5: Confusion Matrices comparing the results of the baseline baseline model with the CNN model.

The CNN model outperforms the baseline model by a high margin of 20% in precision, recall and F-score. The precision score gives the total number of *true positives* from all the predicted positive values. The recall of a model is the total amount of relevant instances that it can retrieve. The F-score is the harmonic mean of precision and recall. The precision score for baseline is 70.5%, while for CNN is **93.03**%, the Recall for baseline is 70.9% and CNN is **93.09**%, and F-score of baseline is 70.6% while for CNN is **93.01%**.

6 Discussion and Future Work

This work presented a hybrid NLU for building intrinsic understanding in machines to differentiate between a sudden topic change, misunderstanding and non-understanding. Manual empirical analysis of a sub-set of the corpora resulted into a set of indicator classes (syntactic tokens and sequences) and categorisation of topics based on cosine similarity for word2vec of syntactic tokens (nouns, adverb, adjective, coordinating conjunctions, interjections, direct object, indirect object) of two adjacent utterances. The extracted indicator classes classes and topic categorisation was used to automatically label the dialogue utterances. Then, a CNN multi-class classification model was used for automatically detecting them. The results indicated a good performance of the CNN model compared to the baseline multi-layer perceptron model.

The generic features (syntactic tokens and sequences, word2vec representations) provide a framework that can be easily modified and extended for other miscommunication types and natural language understanding tasks. Our approach is among the first to automate the entire flow to detect miscommunication types and is only a small contribution towards answering the complex problem of making language understanding more inclusive for dialogue management systems. Even though the model outperformed the baseline (multi-layer perceptron) by a large margin 20%, the challenges during and also after are many. One of them was to precisely encode the human heuristics derived from the expert analysis to syntactic and semantic features that could be extracted automatically. This was done because manual annotation of the data is cumbersome, so this work attempted to automate that process such that research community can focus on building better models rather than investing more than half of the time on labelling the corpora. For labelling we utilised $n-$grams of dialogue utterances (to capture the sequential nature of miscommunication types especially misunderstanding), but the range of $n-$grams is fixed while the sequences can be of flexible length and can lead to erroneous miscommunication types labels. Feature engineering has been discontinued for most parts of machine learning research by the scientific community, however for natural language understanding we believe it is essential if the aim is to fully automate the process of miscommunication types detection and management, along with providing to a certain degree explainable solutions with neural networks. Figure 5b indicates 100% accuracy for misunderstanding and sudden topic change, which is quite misleading because a manual analysis of prediction on the validation data contradicted it. Hence, the model's accuracy and the human perception of the utterances with miscommunication is still not completely aligned.

An immediate future work is to integrate dialogical theory [15] based formalism along with current indicator classes classes for labelling misunderstanding and non-understanding. In the near future, the results and the labelling will undergo an in-depth analysis and fine-tuning to improve the credibility of the model prediction. We also want to integrate the CNN classification model with a sequence to sequence generative neural network for integrally managing the miscommunication types with-in the dialogue management system.

References

1. Aberdeen, J., Ferro, L.: Dialogue patterns and misunderstandings. In: ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems. p. 5. ISCA, Switzerland (2003)
2. Banchs, R.E., Li, H.: Iris: A chat-oriented dialogue system based on the vector space model. In: Proceedings of the ACL 2012 System Demonstrations. pp. 37–42. Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
3. Bazzanella, C., Damiano, R.: The interactional handling of misunderstanding in everyday conversations. Journal of Pragmatics **31**(6), 817 – 836 (1999)
4. Bruce, F.: No conversation without misrepresentation, pp. 143–153. Walter.D.Gruyter Co, Germany (1993)
5. Covington, M.A.: A fundamental algorithm for dependency parsing. In: Proceedings of the 39th annual ACM southeast conference. pp. 95–102. Citeseer, Athens (2001)
6. Georgiladakis, S., Athanasopoulou, G., Meena, R., Lopes, J., Chorianopoulou, A., Palogiannidi, E., Iosif, E., Skantze, G., Potamianos, A.: Root cause analysis of miscommunication hotspots in spoken dialogue systems. In: inInterspeech. pp. 1156–1160. ISCA, India (2016)
7. Higashinaka, R., Funakoshi, K., Inaba, M., Tsunomori, Y., Takahashi, T., Kaji, N.: Overview of dialogue breakdown detection challenge 3. In: Proc. of Dialogue System Technology Challenge. vol. 10, p. 14. ELRA, USA (2017)
8. Hirst, G., McRoy, S., Heeman, P., Edmonds, P., Horton, D.: Repairing conversational misunderstandings and non-understandings. Speech communication **15**(3-4), 213–229 (1994)
9. Kraljevski, I., Hirschfeld, D.: Classification of correction turns in multilingual dialogue corpus. In: Interspeech. pp. 591–595. ISCA, Hyderabad, India (2018)
10. Leite, I., Martinho, C., Paiva, A.: Social robots for long-term interaction: A survey. International Journal of Social Robotics **5**, 291–308 (2013)
11. Malte, G.: Clarification in spoken dialogue systems. In: Proc. of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue. pp. 28–35. AAAI, California (2003)
12. McTear, M., O'Neill, I., Hanna, P., Liu, X.: Handling errors and determining confirmation strategies-an object-based approach. Speech Communication **45**, 249 – 269 (2005)
13. Meena, R., Lopes, J., Skantze, G., Gustafson, J.: Automatic detection of miscommunication in spoken dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 354–363. ACL, Czech Republic (2015)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. pp. 3111–3119. NIPS'13, Curran Associates Inc., USA (2013)
15. Per, L.: Troubles with Mutualities: Towards a Dialogical Theory of Misunderstanding and Miscommunication, pp. 176–208. Arbetsrapporter från Tema K, Cambridge University Press (1993), https://books.google.se/books?id=QgKGGwAACAAJ

16. Purver, M., Hough, J., Howes, C.: Computational models of miscommunication phenomena. Topics in cognitive science **10**(2), 425–451 (2018)
17. Shlens, J.: A tutorial on principal component analysis (2014)
18. Skantze, G.: Error Handling in Spoken Dialogue Systems-Managing Uncertainty, Grounding and Miscommunication. Gabriel Skantze, Stockholm (2007)
19. Skantze, G.: Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems. In: Recent Trends in Discourse and Dialogue, pp. 155–189. Springer, Lisbon, Portugal (2008)
20. Susan, M., Graham, H.: Abductive explanation of dialogue misunderstandings. In: Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics. pp. 277–286. Association for Computational Linguistics, Netherlands (1993)
21. Tewari, M., Bensch, S.: Natural language communication with social robots for assisted living. IROS Workshop in Robots for Assisted Living **1**, 1–6 (2018)
22. Wu, C.H., Su, M.H., Liang, W.B.: Miscommunication handling in spoken dialog systems based on error-aware dialog state detection. EURASIP Journal on Audio, Speech, and Music Processing **2017**, 1–17 (2017)
23. Yu, Zhou, A.P., Rudnicky, A.: Ticktock: A non-goal-oriented multimodal dialog system with engagement awareness. In: 2015 AAAI Spring symposium series. p. 4. AAAI, Palo Alto, California (2015)