Creation and Analysis of Telugu Conversational Corpus

Vamshi Krishna Srirangam[1], Koushik Reddy Sane[1], Sairam Kolla[2], and Manish Shrivastava[1]

[1] Language Technologies Research Centre (LTRC)
Kohli Center on Intelligent Systems (KCIS)
International Institue of Information Technology, Hyderabad, India.
{v.srirangam,koushikreddy.sane}@research.iiit.ac.in
{m.shrivastava}@iiit.ac.in
[2] Microsoft, Hyderabad, India.
{Sairam.Kolla}@microsoft.com

Abstract Understanding conversational data is an important task in dialogue systems. In this paper, we present a conversational corpus of Telugu, a low resource Indian Language. The corpus consists of dialogue exchanges between different characters portrayed in Telugu movies. Each dialogue in the data set has its own time stamp and character information, the time stamp consists of the start time and the end time of the dialogue which is extracted from the respective video of a movie. The novelty of this paper lies in creation of the Telugu conversational corpus. We also explain the importance of such low resource conversational resources in this paper. We try to understand the corpus by employing a sequence-to- sequence encoder with attention decoder model on our data where word perplexity is used as a measure. Interesting insights are drawn from the model's performance results.

Keywords: Conversational corpus · Dialogue systems · Telugu.

1 Introduction

Telugu is an Indian language which lacks digital conversational data. Conversational data such as Movie scripts are publicly available for English language, but for Telugu there are neither conversational data nor movie scripts data available online. Since the manual creation of such a corpus is time consuming and cost expensive, there is no Telugu conversational corpus. Here we created a corpus which is inspired from the Cornell Movie-Dialog corpus [5] in English. The movie dialogues in our corpus are extracted from Telugu movies by native Telugu speakers who are fluent in both Telugu and English.

The time stamp acts as a mapping between the dialogue and the audio visual data from the movie. This mapping helps in the analysis of many features from speech and vision in relation to the dialogue. The annotation of the character data adds subjectivity to dialogues and helps us to analyze them in relation

to the character, like in the analysis of personality features and social factors involved with the character.

2 Related work

The most popular work on dialogue corpus is done by Cornell University researchers where they have released the English movie dialogue corpus. There are also deep learning models experimented on this movie-dialogue data [12]. Earlier works on Telugu dialogue data include dialogue act recognition [6] and task dependent dialogue systems [14] using synthetic dialogue data. But there is no such movie-dialogue corpus or conversational dialogue resources for Telugu language available.

3 Data collection

The dialogue data is taken from Telugu movies spanning over two decades from 2000-2018. The movies from which we took data belong to different genres like action, drama, romance and musical in order to maintain the diversity in the data. Few movies in our corpus are "Mirchi", "Bhale Dongalu" and "Temper". The movie videos are publicly available on YouTube [3]. The data is collected by eight individuals who are fluent in both English and Telugu. Each individual collected data for a single movie, thus creating a dialogue corpus from eight movies. The Telugu dialogues are written using the English Roman Script with each dialogue following the format (*time_stamp*) $ *character* $ *dialogue*, here $ is a delimiter. The time stamp has start time and end time written in (hh:mm:ss) format. The corpus will be released online in near future.

Table 1. An example from the corpus

Time Stamp	**Character**	**Dialogue**
01:54:30- 01:59:40	villan	aa intlo oni la funcion ata e intlo evrni pilavaledhani edupu..
01:59:40- 02:00:45	lawyer	hospital opening twaralo vundhi..apudu mimmalni pilustaru…
02:00:45- 02:01:48	thataya	ma jivitham lo theruvaanu anukunna hospital ni malli theripicharu me runam ela thirchukovali.. goppa koduku ni kannaru andi….aaa adhrustam andhariki vundadhu…

3.1 Scope of the corpus

The Telugu conversational corpus has widespread use in areas such as speech recognition using the given dialogue and the respective audio file extracted from

[3] https://www.youtube.com/

the time stamp. It is useful in vision when juxtaposed with the dialogue and video file extracted from the corresponding time stamp. Applications like persona based or character dependent dialogue systems using the character data associated with the dialogue, task independent end-to-end dialogue systems and multi-party chat summarization systems can be built using this. The data set can further be enriched by annotating it with information like coherence relations between the dialogues and dialogue states. This data set can also be used in building the Telugu-English translation corpus as the respective English subtitles for the same are also available.

Due to its richness, the dialogue corpus helps us in understanding the differences between general underlying conversational structures between the movies which are scripted by directors and the regular social interaction between people. Opining mining on this data at sentence and scene level can be of great assistance in understanding the flow of emotions through the story. Given the dialogue data which is collected from different movies helps us in understanding the writing style of artists. The corpus makes the task-oriented systems more conversational rather than mere information delivery systems.

3.2 Statistics of the corpus

The corpus has a total of 8,590 dialogues and 780 unique movie characters spanning across eight different movies. There are 15,899 unique words out of which 14,265 are Telugu words and 1,634 are English words. These statistics depict the higher degree of code-mixing in recent Telugu movies.

4 Experiments on Data

The dialogues in the current Telugu movies are code-mixed where the dialogues contain words both from Telugu and English. In order to have a better understanding of the data, experiments like language identification and generating word representations from fastText [3] are done. FastText uses sub-word information and is capable of generating representations for unseen words. We also made an effort to design a task independent end-to-end dialogue system and analyze its performance on the data.

4.1 Task Independent End-to-End Dialogue System

Data creation The training data consists of the 14500 (Input, Output) pairs where the "Output", dialogue is a response to the "Input" dialogue where character and time stamp information are not included, the (Input, Output) pairs are extracted from the movies within the scenes. The scenes are marked by the annotators at the time of creating the corpus and are based upon metrics like change in the characters, location, topic of conversation. The extraction of (Input, Output) pairs is done inside a scene so that the (Input, Output) pairs remain coherent. Table 2 shows examples of (Input,Output) pairs.

Table 2. Example (Input,Output) pair

Input	Output
Dani valla problems kani, solve avvavu, kasepu open ga matlaudunkunte pani	vellaku entha dabbu echavo teliyadu gani vellu nekosam entha risk ayina chesetattu unnaru.mundhu ee china vadu vasthadu vediki dookudu ekkuva.
entee veerapratap office kaaa avunandi ma friend jai hello	ma owner veerapratap

4.2 Sequence to Sequence Encoder with Attention Decoder model on the Data

We have experimented with the Sequence to Sequence Encoder with Attention Decoder model on the following different types of data and analyzed the model's performance.

Conversational corpus Transliterated to Telugu

Data creation The (Input, Output) pairs are transliterated to Telugu using language Identification and Transliteration tool [2]. Table 3 shows examples of transliterated Telugu (Input, Output) pairs.

Table 3. Example Telugu Transliterated (Input, Output) pairs

Input	Output
చాదీవిందీ చాలూ కానీ రిపోర్ట్ తీసుకో	నిజాలుం చేబీథ్ ఇలానే ఉన్టుండీ
అధి రా చేప్పను గా థామ్ముదు బాథీకీ వున్దగా ఏ ఫాక్టరీ కత్తాదమ జారాగధు..	నేనుం చావీపోఇనా సరే జారాగనివ్వాను..
కుర్చండీ మే బాగ్ థేస్థీ కుర్చుంటామ	ఓక్కా నిమీషమ బాగ్ నాదీ కదండీ

Experiment In order to build an End-to-End Dialogue System, we chose a Sequence-to-Sequence Encoder with Attention Decoder model [1] to train our data. Perplexity [4] is used as a metric to evaluate the performance of the model. The tables 4 and 5 show the perplexity results, the perplexity increased when the fastText embeddings were used. The perplexity results are comparatively very low when the size of training data is increased to 14500 from 4000 samples and trained for 4 epochs. Table 4 shows the perplexity results of a Sequence to Sequence Encoder with Attention Decoder model on the Data Transliterated to Telugu, trained on 4000 pairs and tested on 1000 pairs. Table 5 shows the perplexity results of a Sequence-to-Sequence Encoder with Attention Decoder

Table 4. Transliterated Telugu 4000 training pairs, 1000 test pairs

Model	**Perplexity**
RNN	2900.4309
RNN with fastText	3438.7359

Table 5. Transliterated Telugu 14500 training pairs and 3500 test pairs

Model	**Perplexity**
RNN	32.3145
RNN with fastText	30.6573

model on the Data Transliterated to Telugu, trained on 14500 pairs and tested on 3500 pairs.

In order to have a better understanding of the data and the model, we ran the sequence-to-sequence encoder with attention decoder(RNN) on the Cornell movie dialog corpus by translating it to Telugu.

Cornell Movie-Dialogues Data translated to Telugu

Data creation We randomly sampled a data of 5000 pairs from Cornell Movie-Dialogues corpus and then we translated it using the Google Translate tool [4] available online.

Experiment The same Sequence-to-Sequence Encoder with Attention Decoder model was run on the data with and without using the fastText embeddings. The perplexity got reduced when run without the embeddings and the perplexity results are relatively higher with fastText embeddings compared to the initial experiment without it. Table 6 shows the perplexity results of a Sequence-to-Sequence Encoder with Attention Decoder model on Cornell Movie Dialog data translated to Telugu, trained on 4000 pairs and tested on 1000 pairs.

In order to understand the relation between the data and the model, we have ran the same model on the Cornell Movie Dialog corpus in English

Cornell English Movie-Dialog Data

Data creation We randomly sampled 5000 (Input, Output) pairs from the Cornell Movie Dialog corpus.

[4] https://translate.google.com/

Table 6. Translated Cornell English-Movie Dialogs 4000 training pairs, 1000 test pairs

Model	Perplexity
RNN	1654.5184
RNN with fastText	2177.2824

Experiment The same Sequence-to-Sequence Encoder Decoder Model with attention was run on the data with and without using the fastText embeddings, interestingly the perplexity results are low as shown in table 8. Table 8 shows the perplexity results of a Sequence-to-Sequence Encoder with Attention Decoder model on Cornell English Movie-Dialog Data, trained on 4000 pairs and tested on 1000 pairs

Table 7. Example of (Input,Output) pairs from Cornell Movie-Dialog data

Input	Output
That must be him. Water taxi. Get us one	Too late, they won't come back out till morning
Look, I can't help you with Quincey if that's what you're after. This has nothing to do with him	So you're just attracted to me, is that it?

Table 8. Cornell English Movie-Dialog 4000 training pairs, 1000 test pairs

Model	Perplexity
RNN	1144.6391
RNN with fastText	1067.9600

Combined Telugu Dialogue Data and Cornell English Movie-Dialog Data

Data creation A mixed vocab of Telugu-English data is created by randomly choosing 5000 pairs from Cornell English Movie-Dialog data and 5000 pairs from our dialogue corpus. The training data and testing data consists of 10000 (Input, Output) pairs where 8000 (Input, Output) pairs are used for training and 2000 (Input, Output) pairs for testing.

Table 9. Telugu-English combined 8000 training pairs and 2000 test pairs

Model	**Perplexity**
RNN on combined Telugu and English (Input, Output) pairs	2340.7413
RNN on Telugu (Input, Output) pairs	3635.1392

Experiment When the Sequence-to-Sequence based encoder with attention decoder model is trained and tested, the perplexity of the model reduced. But when the trained model is tested on only (Input, Output) pairs take from Telugu language the perplexity increased, see table 9. Table 9 shows the perplexity results of a Sequence to Sequence Encoder with Attention Decoder model on combined Telugu Dialogue data and Cornell English Movie-Dialogues Data, trained on 8000 pairs and tested on 2000 pairs.

Word Ordering

Data Creation Difference in the word ordering of Telugu and English language is one of the reasons for higher perplexity, here we changed the word order of English language to align with those of Indian languages [10, 7, 11].

Experiment The word ordered 5000 English (Input,Output) pairs are combined with 5000 Telugu (Input, Output) pairs. 8000 pairs are sampled randomly from the total 10000 for training data and the rest 2000 pairs are used for testing. Table 10 shows the perplexity results of a Sequence to Sequence Encoder with Attention Decoder model on combined Telugu Dialogue data and word ordered Cornell English Movie-Dialogues Data, trained on 8000 pairs and tested on 2000 pairs.

Table 10. Telugu- word ordered English combined 8000 training pairs, 2000 test pairs

Model	**Perplexity**
RNN on combined Telugu and English (Input, Output) pairs	887.5810
RNN on Telugu (Input, Output) pairs	3889.5080

5 Result analysis

Perplexity

We can clearly see that the perplexity of model decreased when word embeddings from fastText are used in case of Cornell English Movie-Dialog corpus. The perplexity of Cornell Movie-dialog data translated to Telugu is comparatively less than that of our transliterated data.

The perplexity of the model initially decreased in case of training data consisting of both Telugu and English pairs because the testing data also has English pairs along with Telugu pairs. The perplexity of the model increased when tested on only Telugu pairs. Though the data is created in order to take leverage from the English data, the knowledge gained from English data could not be used because the word order in English is different from that of Telugu.

Size of the Corpus

We can say from the above observation that, size of the training data is one of the factors that is affecting the perplexity of the system. The model is only trained on 4000 (Input, Output) pairs which is low for a deep learning sequence-to-sequence model and the perplexity decreased with the increase of training data to 14500 pairs. The other important thing is word embeddings, having more training data helps in creating richer word embeddings.

Transliteration

Since the training data used for our system is transliterated from English, the errors in transliteration could be responsible for the higher perplexity of the system. This also explains why perplexity of translated data is less than that of transliterated data.

Word order

The perplexity of the combined Telugu and English pairs decreased when the English data is word ordered in accordance to that of Indian languages. Perplexity decreased when testing data consists of both Telugu and word ordered English pairs. Perplexity increased when tested only on Telugu pairs. Word order is one of the important factor to reduce the perplexity. The increased perplexity is due to the errors in changing the word order of English to Indian Languages and transliteration.

Spelling Variations

One more important thing regarding the vocabulary of our data is that few words are written in different spellings as the dialogue corpus is collected by different individuals where they have written Telugu data in English roman script. For example the word “ekkada” meaning “where” in English is written as (“ekada”, “ekkada”, “ekkadda”,“ekadaa”), leading to a wide variety of representations for few words in the vocabulary and therefore increasing the perplexity.

6 Measures to normalize the vocabulary

The dialogue corpus has to be normalized in order to minimize multiple representations of a single word, we have worked using a couple of algorithms in

order to clean the data. The first one is the edit distance between the words, where you replace a set of words with a single word where the cost function for the edit distance is calculated as the minimum number of character edits or removals required to convert a particular string to the other. But this algorithm did not work well because the cost function replaced words with minimum distance which were not supposed to be replaced in cases like “cat” , “mat” and failed to replace words like “ekada” , “ekkadaa” because the edit distance was higher.

The other algorithm is cleaning words based upon the semantic distance between the word vectors. The word vectors are created from the corpus using fastText, but this approach failed too as the corpus was too small and the similar or neighboring words did not have similar representations and were placed apart leading to the incorrect representations of words. We did not use it due to poor representation of word vectors.

7 Conclusion and Future work

Word representations

We are working on normalization of the data by taking the word representation for transliterated Telugu words from publicly available fasText representations for Telugu [5] to calculate the edit distance, these vectors might be helpful as these vectors are created from a huge dump of data from online sources like Wikipedia. The other techniques for word normalization like clustering and semantic modeling [13] are to be experimented and implemented for richer word representations.

POS Tagging

The perplexity of the model can also be reduced by creating richer and better representations aided by POS tagging [9] and language identification.

Morph analysis

Telugu being an agglutinative language multiple words are combined together as a single word like (eedukuntuvelladu = eedu + kuntu + vellal + du) which means ”he went away swimming”, using morphological analyzers to split such words to morphemes level helps in generating a better vocabulary and creating rich word representations which can capture information at root level and helps in better performance in cases like domain transfer.

Word order

In order to take leverage of English data, the word order of the English language has been changed to word oder similar to that of Indian languages. We can also try this technique with any language other than English whose word order is similar to that of Telugu or by changing its word order to that of similar to Telugu and use the knowledge obtained from training on that data and apply it on our model.

Future works on the corpus such as sentiment analysis [8] for opining mining, understanding the differences between dialogue and monologue data like

[5] https://fasttext.cc/docs/en/crawl-vectors.html

turn taking behaviour, convergence and dialogue act recognition encourage us to continue this field of research.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bhat, I.A., Mujadia, V., Tammewar, A., Bhat, R.A., Shrivastava, M.: Iiit-h system submission for fire2014 shared task on transliterated search. In: Proceedings of the Forum for Information Retrieval Evaluation. pp. 48–53. FIRE '14, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2824864.2824872, http://doi.acm.org/10.1145/2824864.2824872
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
4. Chen, S.F., Beeferman, D., Rosenfeld, R.: Evaluation metrics for language models (1998)
5. Danescu-Niculescu-Mizil, C., Lee, L.: Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In: Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics. pp. 76–87. Association for Computational Linguistics (2011)
6. Jitta, D.S., Chandu, K.R., Pamidipalli, H., Mamidi, R.: "nee intention enti?" towards dialog act recognition in code-mixed conversations. In: Asian Language Processing (IALP), 2017 International Conference on. pp. 243–246. IEEE (2017)
7. Kunchukuttan, A., Mishra, A., Chatterjee, R., Shah, R., Bhattacharyya, P.: Sata-anuvadak: Tackling multiway translation of indian languages. pan **841**(54,570), 4–135 (2014)
8. Mukku, S.S., Choudhary, N., Mamidi, R.: Enhanced sentiment classification of telugu text using ml techniques. In: SAAIP@ IJCAI. pp. 29–34 (2016)
9. Nelakuditi, K., Jitta, D.S., Mamidi, R.: Part-of-speech tagging for code mixed english-telugu social media data. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 332–342. Springer (2016)
10. Patel, R.N., Gupta, R., Pimpale, P.B., et al.: Reordering rules for english-hindi smt. arXiv preprint arXiv:1610.07420 (2016)
11. Ramanathan, A., Hegde, J., Shah, R.M., Bhattacharyya, P., Sasikumar, M.: Simple syntactic and morphological processing can help english-hindi statistical machine translation. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I (2008)
12. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Hierarchical neural network generative models for movie dialogues. CoRR, abs/1507.04808 (2015)
13. Singh, R., Choudhary, N., Shrivastava, M.: Automatic normalization of word variations in code-mixed social media text. arXiv preprint arXiv:1804.00804 (2018)
14. Sravanthi, M.C., Prathyusha, K., Mamidi, R.: A dialogue system for telugu, a resource-poor language. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 364–374. Springer (2015)