Task-Oriented Dialog Systems for Dravidian Languages

Tushar Kanakagiri* **Karthik Radhakrishnan***
Language Technologies Institute
Carnegie Mellon University
{kradhak2, tkanakag}@cs.cmu.edu

Abstract

Task-oriented dialog systems help a user achieve a particular goal by parsing user requests to execute a particular action. These systems typically require copious amounts of training data to effectively understand the user intent and its corresponding slots. Acquiring large training corpora requires significant manual effort in annotation, rendering its construction infeasible for low-resource languages. In this paper, we present a two step approach for automatically constructing task-oriented dialogue data in such languages by making use of annotated data from high resource languages. First, we use a machine translation (MT) system to translate the utterance and slot information to the target language. Second, we use token prefix matching and mBERT based semantic matching to align the slot tokens to the corresponding tokens in the utterance. We hand-curate a new test dataset in two low-resource Dravidian languages and show the significance and impact of our training dataset construction using a state-of-the-art mBERT model - achieving a Slot F_1 of 81.51 (Kannada) and 78.82 (Tamil) on our test sets.

1 Introduction

With the surge in popularity of digital assistants (Google, Siri, Alexa) task-oriented dialog (TOD) systems have become commonplace in NLP research today. TOD systems are designed to complete a particular user goal by understanding requests from users and providing relevant information (Liu and Lane, 2018). They typically consist of four major components - Natural Language Understanding (Chen et al., 2016), Dialog State Tracking (Rastogi et al., 2020; Campagna et al., 2020), Dialog Policy Learning (Takanobu et al., 2019; Li et al., 2020), and Response Generation (Kummerfeld et al., 2019; Galley et al., 2019; Kale and Rastogi, 2020). In this work, we focus on NLU and its two subtasks - Intent Detection and Slot Filling. Intent Detection is typically cast as a sequence classification problem where the task is to classify the purpose or goal that underlies a user utterance into one of several predefined classes called intents (ex. CHECK-SUNRISE, SET-ALARM). The brevity and succinctness of the utterances coupled with the requirements to scale to different domains pose challenges to Intent Detection.

What's	temperature	in	New	York
O	B-weather	O	B-Loc	I-Loc
Intent : weather/find				

Table 1: Example of an utterance along with its slot in BIO (Beginning, Inside, Outside) notation and intent label.

Slot Filling involves identifying the intent arguments and is typically cast as a sequence labelling problem using the BIO notation. (see table 1 for an example). Prior research on the two tasks have mainly focused on English, reporting excellent performance owing to the availability of high quality and large amounts of annotated data (Schuster et al., 2018; Wu et al., 2020). But such performance has not been achieved in low-resource languages due to lack of such data which can be expensive to construct.

In this work, we present a simple but effective method to automatically create annotated training data for slot filling and intent detection in low resource languages making

[*] Equal contribution

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 85–93
April 20, 2021 ©2021 Association for Computational Linguistics

use of available English data. First, we independently translate utterances and annotated slots in English to the target language. Second, to align the slots with the utterance in the generated translation, we make use of morphology and semantics of the target language (§3). We conduct experiments on two Dravidian languages such as Tamil(TAM) and Kannada(KAN) and evaluate our methods on hand-annotated test sets of 600 utterances across the two languages. Tamil and Kannada belongs to south Dravidian (Tamil-Kannada) languages (Thavareesan and Mahesan, 2019, 2020a,b). By training using state-of-the-art Multilingual BERT : mBERT (Devlin et al., 2018) model for slot filling and intent detection, we show in §6 that our simple alignment heuristics outperform prior approaches relying on existing word alignment methods[1].

2 Related Work

TOD Systems - In the recent years, there has been a lot of work on building TOD systems broadly around two themes - Building out each component of the system independently (Chen et al., 2016; Campagna et al., 2020; Takanobu et al., 2019; Kummerfeld et al., 2019) and end-to-end TOD systems (Hosseini-Asl et al., 2020; Ham et al., 2020; Liang et al., 2020). Though end-to-end systems show better generalization, they usually require larger amounts of training data and may not be very feasible for low-resource settings. In this work, we primarily tackle the NLU component which performs Intent detection and Slot identification.

Since these tasks are framed as sequence classification and tagging tasks, sentence encoder models have been used to tackle them. Prior works have utilized Recurrent Neural Networks (Liu and Lane, 2016; Goo et al., 2018; E et al., 2019) and more recently have used BERT (Chen et al., 2019).

Multilingual TOD - Though there currently exist multiple large-scale datasets for TOD (Byrne et al., 2019; Budzianowski et al., 2018; Hemphill et al., 1990), they are monolingual English Corpora. Schuster et al. (2018) released the Facebook TOD dataset which contains utterances from three languages - English, Spanish, Thai geared towards cross-lingual transfer of dialog. On this dataset, there have been works on zero and few-shot transfer using Latent Variable Transfer (Liu et al., 2019), Mixed Language Training (Liu et al., 2020) but they don't display consistent gains over augmentation with MT+Word alignment data.

On Indian languages, there have been works towards Indic and Code-Switched TOD systems (Jayarao and Srivastava, 2018; Banerjee et al., 2018; Mandl et al., 2020) but these datasets are manually constructed and are smaller in size. Furthermore, they're primarily for Hindi and there currently exists no dataset for languages like Kannada, Telugu, etc.

Auto-construction of datasets - Given the cost and difficulty in acquiring annotators, prior works have employed methods to create synthetic training data. Gupta et al. (2020) use Google Translate to create synthetic utterances in Indic languages towards the task of spoken intent detection but do not tackle the slot filling task. More recently López de Lacalle et al. (2020) employed a Seq2Seq translation and word alignment to project the Spanish slot tags to Basque. However, we show that given the flexible word ordering and rich morphology, word alignment systems do not work particularly well for Dravidian languages. Furthermore, since Dravidian languages do not have large open-source corpora to learn MT and alignment systems reliably, we utilize Google Translate API, morphology and semantics based heuristic slot aligner which does not require any parallel data.

3 Dataset Construction

The training dataset for the low-resource languages is constructed from the Facebook Multilingual Task-Oriented Dialog dataset (Schuster et al., 2018) which contains utterances in English, Spanish and Thai tagged with the corresponding intent and slot labels by hand.

3.1 Auto-construction from English data

Our dataset construction consists of two steps - Translation and slot assignment. The translation phase converts the English utterances

[1] Code and data available at `https://github.com/karthikradhakrishnan96/TOD-Dravidian`

to their target language equivalents and the slot assignment phase transfers slot labels from the English phrases to their target language equivalents. As seen in Figure 1, words often get translated to multiple words/don't have an equivalent in the translated language ('the' in the sentence is folded into 'groomer' in the target language), making the slot assignment problem non-trivial.

For the translation phase, we experiment with two methods - Google Translate and a transformer based Seq2Seq model.

Previous works have tackled slot transfer using Word Alignment (López de Lacalle et al., 2020; Schuster et al., 2018) by first predicting the word alignment between the translations and then copying over the labels from the source language words to their corresponding target language equivalents. This maximises sentence level alignment probabilities and is usually learned using a parallel sentence corpora. Furthermore, owing to the rich morphology and flexible word-ordering of Dravidian languages, these models typically have high alignment error rates. To transfer the slot spans from English to our target language, we only require the alignments of the slot spans and not the entire utterance. We tackle this by translating the annotated English spans to our target language and using various heuristics to align the slots to the utterance.

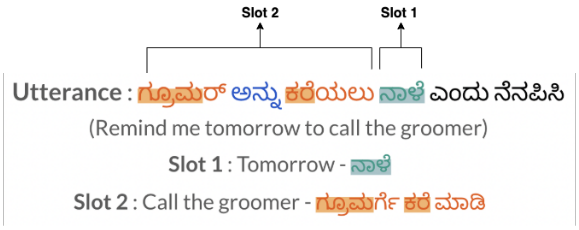

Slot 2
Slot 1
Utterance : ಗ್ರೂಮರ್ ಅನ್ನು ಕರೆಯಲು ನಾಳೆ ಎಂದು ನೆನಪಿಸಿ
(Remind me tomorrow to call the groomer)
Slot 1 : Tomorrow - ನಾಳೆ
Slot 2 : Call the groomer - ಗ್ರೂಮರ್ಗೆ ಕರೆ ಮಾಡಿ

Figure 1: Our prefix matching and span expansion technique. Highlighted text shows the matching prefix of tokens from slots and utterances. We can observe how span expansion helps label the ಅನ್ನು token correctly and obtain a contiguous span and account for noise in our alignment technique.

We use a simple word match as our baseline, where we identify words from utterances that match words from the annotated slots. This baseline achieves a poor Slot F_1 score owing to the variation in translation of certain words when translated independently and as part of a sentence. This can be attributed to the morphologically rich nature of Indian languages, where the context of the inflection can lead to different suffixes during translation for each word. To tackle this, as seen in Figure 1, we use a technique that matches a word from the annotated slot to a word from the utterance that has the maximum prefix overlap. This helps account for a wide variety of variations and greatly helps in the automatic construction of the low-resource dataset.

Finally, we apply a span expansion technique where we assign the labels of the aligned slot tokens to those intra-span tokens that did not align with any slot. This helps us obtain a contiguous span which is a requirement of sequence tagging models. Additionally, this also helps account for errors in our alignment approach when certain intra-span tokens do not get aligned with any utterance token.

3.2 Including Semantic Matching

Upon analyzing the unmatched words from our aligner, we noticed that the words get translated differently when used in a complete sentence and when used individually. These variations could be due to transliteration (Word gets transliterated when used in slot phrase but translated when used in the utterance), synonyms (Phrase and utterance translations are synonyms) etc. Since the words are completely different, the prefix matching does not find any matching candidates. We use mBERT to obtain word embedding in-phrase and word embeddings in-utterance and cosine similarity to find the closest matching word to align to.

3.3 Manual tagging for test set

To quantitatively evaluate the quality of our automatically constructed training data, we require a gold test set in the target low-resource language. As this is unavailable for many Dravidian languages, we manually tag a test set for two of them (Kannada and Tamil), to obtain the gold alignments in the low-resource languages between annotated slots and the utterances. We first translate utterances to the low-resource language using Google Translate. Next, we remove examples that contain incorrect utterance translations. We finally obtain a sample of 300 examples for

Ex. 1	அடுத்த வாரம் வெப்பநிலை என்ன? (What is the temperature next week?)
Slots	B-DT I-DT B-WN O
Ex. 2	Necesitaré un suéter en Denver mañana? (WillIneed a sweater in Denver tomorrow?)
Slots	O O B-WA O B-LOC B-DT
Ex. 3	ಸಾಂತಾ ಬರಾಬರಾದಲ್ಲಿ ಈ ವಾರಾಂತ್ಯದಲ್ಲಿ ತಾಪಮಾನ ಹೇಗಿರುತ್ತದೆ (Santa **Barbara this** weekend temperature what)
Slots	B-LOC **B-LOC I-DT** I-DT B-WN
Ex. 4	Necesito un informe *meteorológico* para el viernes (needed a report weather for Friday)
Slots	O O B-WN *I-WN* B-DT I-DT I-DT

Table 2: Some qualitative examples from our dataset (With word-by-word ENG translations) and predictions by our mBERT model. (DT - DateTime, WN - Weather Noun, LOC - Location, WA - Weather Attribute)

each language. For each language and each example, the utterance tokens corresponding to the slots are tagged, by two graduate students who are native speakers of the language (We obtain an inter-annotator Cohen Kappa score of 0.9303 for KAN and 0.9606 for TAM).

To aid with annotation, we make use of Doccano (Nakayama et al., 2018) to help provide the annotators an easy to use interface (Figure 3).

4 Model

We learn the intent and slot filling tasks jointly with BERT (Devlin et al., 2018), more specifically, the multilingual variant - mBERT which is pre-trained on 104 languages. We pass the [CLS] vector representations onto a Multi-Layer perceptron to classify the intent onto one of the predefined intent classes. We pass the token representations produced by the mBERT model onto another Multi-Layer perceptron to identify the slots. In the case of a word consisting of multiple tokens, we only use the first sub-word token and ignore the others during training and test. Figure 2 shows the architecture of our system. Since mBERT is pre-trained across many languages, we follow the same architecture for our zero-shot and few-shot experiments.

5 Experiment Setup

We use the BASE model of BERT and mBERT in all our experiments with a batch size of 64 and a learning rate of 1e-5 with Adam (Kingma and Ba, 2014) optimizer. On zero and few shot transfer experiments, the model is trained for about 10 epochs before evaluation/fine-tuning on few shot data.

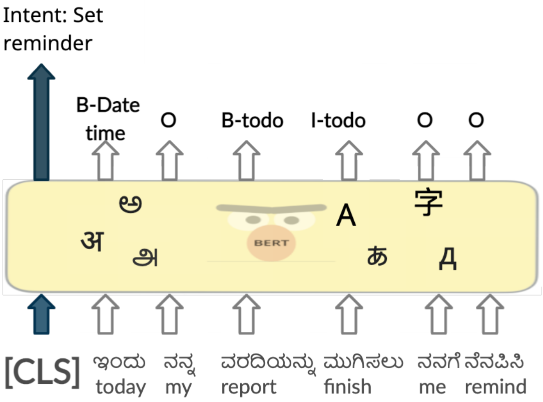


Figure 2: Model Architecture showcasing a KAN utterance passed to mBERT model. (word-by-word ENG translations are included for readability). Sub-word tokens omitted for brevity. Slot labels and intent are predicted from word and [CLS] representations respectively

6 Analysis and Discussion of Results

6.1 Qualitative examples

Table 2 shows some qualitative examples from our dataset and the predictions made by our model. The first two examples demonstrate successful predictions on TAM and ES when trained on 5K auto-tagged examples. The third example demonstrates an error due to local normalization during optimization. Since we don't explicitly model that B-LOC (Beginning Location) does not occur immediately after a B-LOC and that I-LOC is more probable, the model predicts B-LOC for both slots. Utilising global normalization over BERT (CRF) could help resolve this issue.

The fourth example showcases the efficacy of using mBERT during alignment. On our training set, we noticed that the word 'weather' gets translated into 'meteorológico' when used in a sentence but gets translated to 'clima' when used individually. Since these words have no common prefix, the words are not usually tagged by our aligner. But when mBERT is used, since these terms are very close semantically, the slots are assigned correctly, subsequently resulting in better predictions.

6.2 Effect of Dataset construction heuristic

Without any models, we first evaluated the efficacy of our dataset tagger by autotagging and manually tagging the 300 test examples and comparing the slot F_1 between them. On our hand-annotated set of 300 examples, we show that the slot F_1 obtained by our tagging method (§3) outperforms simple word match and existing word alignment baselines in Table 3.

Method	Lang	Slot F1
Word Alignment	TAM	31.17
Translate+Overlap	TAM	46.82
Ours	TAM	80.70
Ours	KAN	80.76

Table 3: Performance of different slot alignment methods

6.3 Effect of number of training samples

We conduct experiments with varying amounts of data to show the effect of training data size in Table 4. We can see that even with 300 examples from our auto-tagged dataset, the model is able to achieve significant boost as compared to the zero shot setting of training on ENG and testing on KAN. Initial training on ENG followed by a few shot adaptation on KAN provides an even higher boost in slot performance. Upon varying the number of training samples to 10K and 20K for KAN, we see some minor improvement for 10K for the slot but see a 1.5 F1 boost when using 20K auto-annotated samples.

Given that KAN and TAM are more closely related, we can see that the zero shot transfer from KAN to TAM achieves 62.22 & 46.16, outperforming the transfer from ENG to TAM. We also experimented with training a single model jointly on KAN and TAM but observed similar but slightly reduced performance as compared to using 5K examples from just one language (Rows 8, 13 and 6,14) which could be due to interference when optimizing for multiple languages. We also observe that using 300 examples of KAN and 5K of TAM achieves better performance on the TAM further corroborating the hypothesis that larger number of related language datapoints cause interference, leading to lower scores.

Train	Test	Intent	Slot
ENG-30K	KAN	51.96	21.46
KAN-300	KAN	86.27	62.24
ENG-30K, KAN-300	KAN	81.04	65.82
ENG-30K	TAM	30.50	27.81
TAM-300	TAM	82.07	59.67
TAM-5K	TAM	93.08	78.68
ENG-30K, TAM-300	TAM	79.55	65.01
KAN-5K	KAN	93.13	79.87
KAN-10K	KAN	91.80	79.95
KAN-20K	KAN	93.79	81.51
KAN-5K	TAM	62.22	46.16
TAM-5K	KAN	59.47	37.24
KAN-5K, TAM-5K	KAN	93.46	79.75
TAM-5K, KAN-5K	TAM	92.76	78.21
KAN-300, TAM-5K	TAM	93.71	78.82

Table 4: Zero and Few shot transfer performance

6.4 Effect of the MT system

Given the key role of the MT system in the dataset construction, we compare the performance of two approaches - Google Translate and our own Seq2Seq Transformer MT system. On the EnTam corpus (Ramasamy et al., 2012), our MT system achieved a BLEU score of 10.89 (the best reported score on this dataset is 9.39 (Kumar and Singh, 2019)). The low performance of the mBERT model could be attributed to the low quality of the translations for the training dataset resulting from domain gap in the MT system. Since the EnTam corpus contains articles on Cinema, News, and Bible, it doesn't translate TOD utterances accurately (We examined frequent words in our training corpus - Remind, Alarm, etc and observed that these words didn't exist or were very infrequent in the EnTam corpus) Furthermore, owing to the differences in style (EnTam contains statements while TOD contains questions) we observed that some translations had changes in their meanings. We hence observe that 300 high quality training examples lead to similar performance as compared to 25K noisy training examples (Table 5)

MT System	Lang	Intent	Slot
TAM-300-Google MT	TAM	82.07	59.67
TAM-5K-Google MT	TAM	93.08	78.68
TAM-5K-Seq2Seq	TAM	80.50	51.32
TAM-25K-Seq2Seq	TAM	81.76	56.56

Table 5: Performance of different MT systems

6.5 Effect of Semantic Matching

We now present results when mBERT based semantic matching is included in the aligner in Table 6. We noticed a small increase in F1 score for KAN but a small drop for TAM. This could be attributed to two reasons: 1) Our alignment heuristic already tags spans accurately and the inclusion of mBERT only provides minor improvements and 2) Languages like KAN and TAM are underrepresented in the mBERT training set leading to lower performance on these languages.

On a non-Dravidian language Spanish, we notice a larger boost in performance (6 F1), indicating the strength of the mBERT representations for ES. We present a more detailed analysis of the performance on ES in the next section.

Lang	Slot F1	+mBERT
KAN-5K	79.87	80.63
KAN-20K	81.51	82.01
TAM-5K	78.68	76.24
ES-5K	71.85	77.66

Table 6: Performance difference due to the inclusion of mBERT

6.6 Performance on non-Dravidian Languages

Though our aligner was designed for Dravidian languages, there are languages in other families which also exhibit suffix-based morphology. We evaluate the performance of our system on Spanish by auto-creating the training data from ENG using our method and evaluating on the hand-annotated ES data provided in the Facebook dataset.

We can see from Table 7 that our auto-created dataset of 5K examples performs only 12 F_1 worse than training on the entire hand-annotated ES dataset (Consisting of 3.6K examples) further showcasing the quality of training examples produced by our method.

Train	Intent	Slot F1
ES-5K-auto	97.63	71.85
ES-5k-mBERT	97.13	77.66
ES-20K-mBERT	97.46	79.47
ES-FB	98.78	89.39

Table 7: Performance on a Romance language ES

7 Conclusion & Future Work

In this work, we demonstrated techniques to project training data from high resource languages to low-resource settings, thus efficiently obtaining large scale synthetic data for training TOD systems. We also showcased the efficacy of the dataset creation on a manually curated test set on KAN and TAM.

In the future, we hope to add more Dravidian languages - Telugu and Malayalam. We also hope to utilize social media data from these languages to generate code-switched/more natural utterances.

References

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. *arXiv preprint arXiv:1806.05997*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Yun-Nung Chen, Dilek Hakkani-Tür, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding.

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

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and B. Dolan. 2019. Grounded response generation task at dstc7.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Akshat Gupta, Xinjian Li, Sai Krishna Rallabandi, and Alan W Black. 2020. Acoustics based intent recognition using discovered phonetic units for low resource languages. *arXiv preprint arXiv:2011.03646*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Pratik Jayarao and Aman Srivastava. 2018. Intent detection for code-mix utterances in task oriented dialogue systems. In *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 583–587. IEEE.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Amit Kumar and Anil Kumar Singh. 2019. NLPRL at WAT2019: Transformer-based Tamil – English indic task neural machine translation system. In *Proceedings of the 6th Workshop on Asian Translation*, pages 171–174, Hong Kong, China. Association for Computational Linguistics.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros Polymenakos, and Walter S. Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Maddalen López de Lacalle, Xabier Saralegi, and Iñaki San Vicente. 2020. Building a task-oriented dialog system for languages with no training data: the case for Basque. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2796–2802, Marseille, France. European Language Resources Association.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Julia Kiseleva, Maarten de Rijke, Shahin Shayandeh, and Jianfeng Gao. 2020. Guided dialogue policy learning without adversarial learning in the loop. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2308–2317, Online. Association for Computational Linguistics.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. Moss: End-to-end dialog system framework with modular supervision. In *AAAI*, pages 8327–8335.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. *arXiv preprint arXiv:1911.04081*.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8433–8440.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. 2012. Morphological processing for english-tamil statistical machine translation. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pages 113–122.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at dstc8. *arXiv preprint arXiv:2002.01359*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

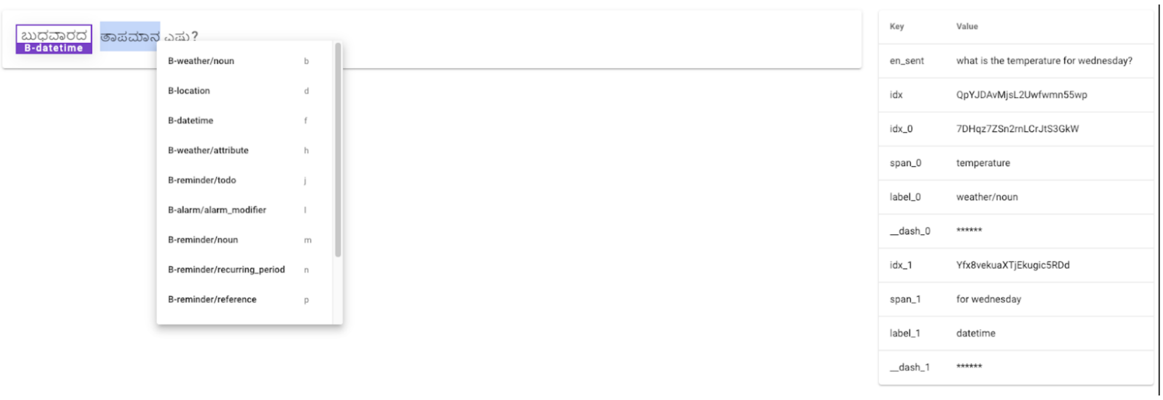
Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.

Appendix

A Annotation Tool



ಬುಧವಾರದ ತಾಪಮಾನ ಎಷ್ಟು?

Key	Value
en_sent	what is the temperature for wednesday?
idx	QpYJDAvMjsL2Uwfwmn55wp
idx_0	7DHqz7ZSn2rnLCrJtS3GkW
span_0	temperature
label_0	weather/noun
__dash_0	******
idx_1	Yfx8vekuaXTjEkugic5RDd
span_1	for wednesday
label_1	datetime
__dash_1	******

Figure 3: Annotation User Interface for manual creation of test dataset. The slot spans in English are shown in the right and the annotators were required to mark the corresponding slots in the low-resource language utterance on the left