Creating Spoken Dialog Systems in Ultra-Low Resourced Settings (Speech + NLP)

Geoffrey Kimani, Julienne Ayinkamiye, Arsène I. Muhire, Emmanuel N. Ndayisaba
{gnkimani, jayinkam, amuhire, enn}@andrew.cmu.edu

Carnegie Mellon University Africa

Abstract

In a world that is becoming increasingly automated, spoken dialogue systems have become an essential tool in improving human-computer interaction. Significant work has been done to create spoken dialogue systems, but there remains major constraints on data availability for low-resourced languages to develop such systems effectively. This is because most end-to-end automatic recognition systems that have been tested are data-intensive. A lot of research has gone into overcoming this constraint, one particularly promising is the acoustic-based intent classification which has shown great success. Our work aimed to reproduce the results of an acoustic-based intent classifier and improve its performance by exploring data augmentation techniques. We propose applying phoneme substitution and sequence generation as data augmentation techniques at different levels – in data preparation and in the model. We implemented the acoustic-based intent classifier with comparable results. Further, after applying augmentation techniques, we tested the model using the grabo dataset for one-to-seven speakers with one-to-seven recordings for the case of two, four, and six intents. We observed that there is a some improvement for some data points in the model's performance, but generally, data augmentation approaches we tried did not give us any significant improvement; thus, further investigations are needed to find data augmentation techniques on phonetic sequences that can be used to enhance the ability of intent classifier to solve the problem of insufficient data in low resource settings.

Index Terms— Intent, Low Resource, Data augmentation, Long short term Memory

1 Introduction

Speech is the most natural and efficient way of communicating between people and is becoming an integral part of human-computer interface [1]. Computer systems have also become a ubiquitous part of our lives, and there is a growing need to simplify human-computer interaction. [1] advocates for simple and effortless interfaces that can support unsophisticated users to access, process, and manipulate information. Human Language Technology (HLT) systems have proved vital in enabling this human-computer interaction. HLT systems have been used in creating conversational technologies like virtual assistants and chatbots, smart technologies like home automation systems, and voice and text analyzers [2]. Good examples of implementation of HLT systems are spoken dialogue systems like Alexa, Siri and Google Assistant, which allow users to interact with the system through spoken language [3] and have proved vital in communication, education, business, and decision-making. However, most of these systems support high-resource languages while leaving out low-resourced languages that are widely used in Africa and other parts of the world. [4] has shown that many communities that need Natural Language Processing systems are still under-served by it.

Implementing similar systems for a low-resource setting is vitally important as it has been shown that it can be useful in augmenting high-resource languages among illiterate people. [2] reviews an android based speech-to-speech application used to provide patient care. The application was used by healthcare workers to bridge the communication gap with patients and proved vital in breaking barriers between these two groups; one using a high resource language for work and the other low resource language to describe symptoms. This example gives a good illustration of problems faced by illiterate people in low resourced settings since most services are offered in high-resource languages. We also see how this idea can be extended to other vital services to allow better communication.

The biggest challenge, however, is data availability to build such systems. In this project, we take the approach using methodologies that are less data-intensive. We aim to implement a multilingual intent classifier that would aid in creating a spoken dialogue system for a low-resource setting. We start by replicating the work done by Akshat et. al in [5]. We then explore data augmentation techniques to improve the performance of the classifier more specifically we look into phonemes substitutions and sequence generation.

We find that though data augmentation has been shown to improve model performance in other classifiers [6], the techniques we used failed to improve the performance of the intent-based classifier that uses phonetic units.

This paper is structured as follows. Section 2 discusses a survey of the literature and related works. Section 3 describes the methodology including model description and data augmentation techniques. Sections 4 and 5 discuss the results obtained. Section 6 draws conclusions and proposes future work.

2 Literature review

Low-resourced languages are characterized by a lack of large and useful corpora that is sufficient to build NLP applications [7, 8]. Two important aspects of low-resourced-ness are language-specific and task-specific low-resourced-ness [9]. The first describes a scenario of insufficient resources to create robust, language-specific speech recognition systems. The second low-resourced-ness is task-specific which describes insufficient annotated data for a particular task unfolding either as a lack of enough speakers or lack of enough recordings per speaker to create a sufficient speech corpus.

This uniqueness of low-resource settings presents a major challenge of data availability for researchers; thus, they must work around this hurdle to successfully build language systems. Some of the workarounds include using methodologies that are less data-intensive, using cross-lingual models, curating data, or collecting more data for a given language.

Different studies separately employed Automatic speech recognition(ASR) and Natural Language Understanding(NLU) systems to build spoken dialogue systems. For example, textual transcripts generated by ASR systems were formerly used as inputs for intent recognition in NLU systems. However, this pipeline of systems is prone to propagating errors from lower modules to higher and thus affecting the overall performance of the system consequently leading to the exploration of end-to-end spoken language understanding systems[10]. These end-to-end spoken language understanding systems are data-intensive [11] which is a major hurdle in the low-resourced settings.

In languages with limited resources, small datasets result in a small amount of annotated speech data. This hinders the advancement of robust ASR modules, affecting overall system performance. Initially, various researchers focused on improving ASR by accumulating a large corpus of the low resource language, training the ASR system on a high-resource language, or training the ASR models with a large number of small speech datasets of different languages to create universal or cross-linguistic ASR systems.

Another method is to focus on identifying the intent immediately from the linguistic units, referred to as phonemes, among the first papers to explore this idea was by Akshat Gupta et. al who generated phones from collected banking services audio using Allosaurus. Allosaurus is trained to perform universal phone recognition and is not a language-specific model. This means the phonetic transcriptions will be more accurate than English phonemic representations. The generated phones were used for intent classification employing the Naive Bayes model with absolute discounting which proved phonemes are applicable in real-life banking systems and multilingual systems[12].

In Akshat Gupta et. al work phoneme transcriptions were generated by allosaurus for low resource languages belonging to Romance and Indic language families. The intent classifier was a hybrid CNN + LSTM architecture with multilingual training yielding improved cross-lingual transfer and zero-shot performance on unknown languages under the same family umbrella. Furthermore, the authors showed that the multilingual model generally performs better than the uni-lingual model[5].

Similarly, [10] modified the CNN + LSTM architecture by incorporating self-attention mechanisms and feeding the output into a fully connected layer for intent classification. This yielded results competitive to high resource systems and outperformed the existing low resources systems. In paper [13], the study performed intent classification on the Mandarin-Chinese language, the first block is a phoneme recognizer, and the second is a transformer-based language model that takes the phonemes as input. The authors found that the transformer-based model performed worse than the CNN + LSTM architecture on the same task.

Data augmentation has also been shown to improve the performance of models. [14] notes that this technique has been shown to be effective in scenarios where data is scarce. While data augmentation has been successfully applied in image processing and computer vision though it remain under-explored in NLP applications [6]. [14] classifies data augmentation methods into three main categories; paraphrasing-based, noising-based, and sampling-based methods.

Paraphrasing methods generated paraphrases of the original data which is used as the augmentation data. Since paraphrases are an alternative way of conveying the same information, these methods offer limited semantic difference and thus limited performance improvement [14]. Example of these methods are semantic embedding, and machine translation.

Noising techniques add or remove continuous or discrete noise terms in the data which leads to more changes and in effect improving the robustness of the model [14]. Common techniques include insertion, deletion, swapping, mixup and substitution.

Sampling techniques use the data's original distribution to sample new data for augmentation. These techniques also improve the performance of the model as the sampled points are usually novel and increase the diversity of the data[14]. In our experiments we explore the noising and sampling techniques since they show better results and model robustness.

3 Methodology

3.1 Data Description

In this project, we used the Grabo dataset, which comprises commands recorded in both Flemish and English. The collection contains 11 speakers, each of whom has 36 recordings. Ten of the 11 speakers are Flemish and one is English. Each recording has 15 utterances, each of which corresponds to a single command but differs in how the speaker delivers it.

All utterances in single recording have a corresponding target. The targets are considered to be the intents of the command in utterances. In our project we employed 2 and 4 intents classifiers.

All audio recordings were converted to phonemes by Allosaurus system before being used the intent classifier. Allosaurus system is a model pre-trained on more than 2000 languages to recognize audio and output phonetic transcripts. Using Allosaurus is particularly special because languages heavily share phones and that makes it easier to have speech recognition system for low-resourced languages without the need for Automatic Speech Recognition system that may require so much data to train.

Following the original paper, 7 speakers were used for training, 2 for validation, and 2 for testing. The model was trained on different combinations of speakers and utterances. For instance, where S=1 and K=1, the model was trained on one speaker and one utterance. The model was trained on the smallest data point feasible to reflect the most extreme low-setting circumstance.

3.2 Model Description

The first phase of our project was duplicating the work similar to one done in [5] of developing an end-to-end intent classifier for phonetic transcripts generated by Allosaurus system from auditory data. The figure 1 below shows the flow of the system.

The classifier (figure 2a) has an embedding layer that captures the semantic information in the dataset. The created embeddings are next passed through two Convolutional Neural Networks (CNN) and subsequently into a two layers Bidirectional Long-Short Term Memory (LSTM). The CNN layers extract high-level spatial and contextual features from phonemes while LSTM captures long-term relationships between sequences of phones. The embedding layer employs an embedding size of 128, whereas the first CNN layer employs a kernel size of three and the second employs a kernel size of five. After the CNN layers, there is a batch norm layer. The LSTM layer has two layers and is bi-directional. After LSTM, there is a full connected layer for classification. Figure 2 shows the full model architecture.

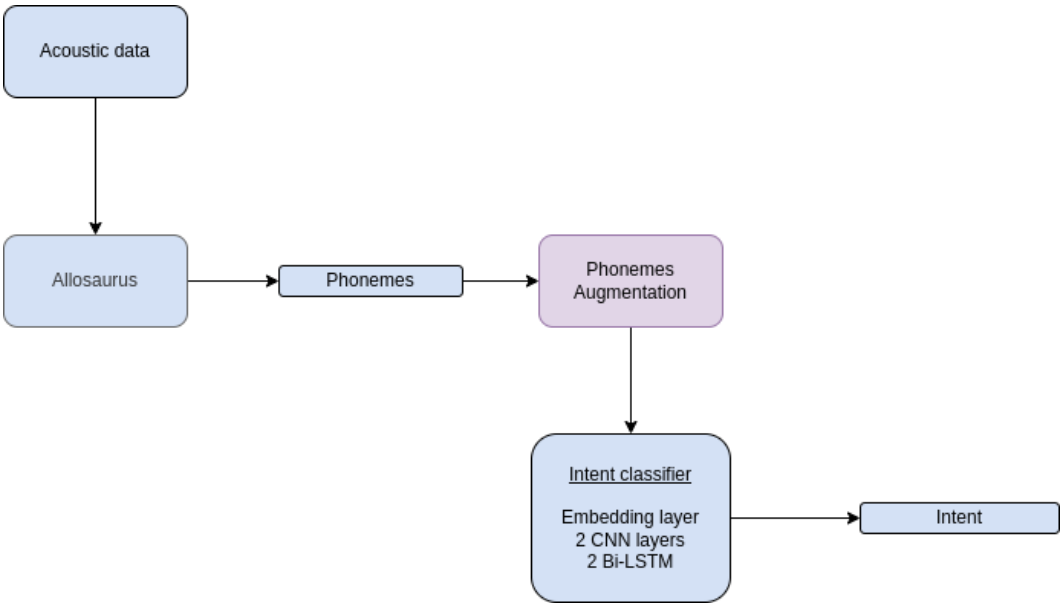


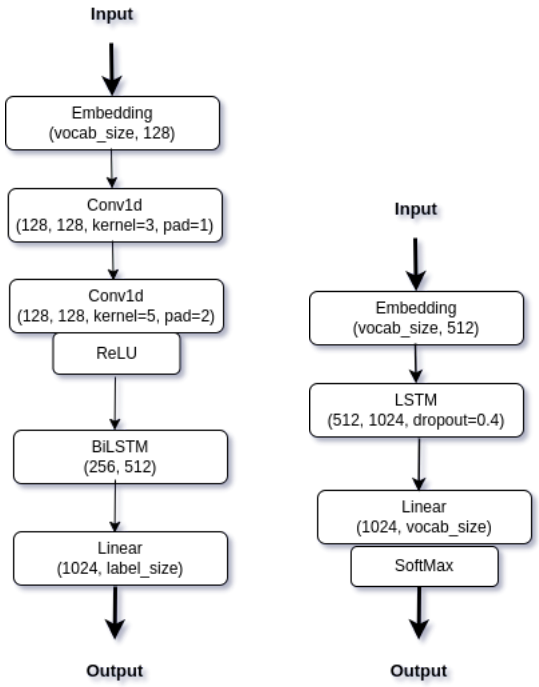
Figure 1: Data flow from auditory data through Allosaurus to intent classifier

The second phase of the study focused on attempting to improve the aforementioned model's performance by experimenting with various data augmentation techniques. We experimented with strategies that are often used for augmenting text data on phoneme sequences. We began by trying to increase data by employing phoneme sequence-to-sequence creation. The goal of the method was to generate a variety of utterances with the same semantics [15]. We expected that the more diverse utterances added to the dataset, the better the intent classifier model would be. The model for sequence-to-sequence generation has an embedding layer and one LSTM layer as figure 2b shows. While using the model, in each utterance, the next phoneme is replaced by generated phoneme with a certain probability and the new modified utterance is added to the dataset with a corresponding label as the original utterance. The resulting utterances had slightly different sequence but with very similar semantics to the original. In additional to sequence generation model, we also explored substitution method. We used the phoneme representation of words and computed the edit distance between a given word A and every other word constituted in the dataset. From this we selected the top three words (their phoneme representations) that were closed to the true word. We did this as certain words sound acoustically similar thus the dataset may not contain the true representation of the word. We carried out random selection for which words would be fed into our model for a given target outcome.

3.3 Evaluation metrics

We utilized accuracy as an evaluation metric to assess each model's performance in this study. True Positive and True Negative are used to calculate accuracy. When correctly recognized, an activity can be classed as True Positive (TP) or True Negative (TN), however, when incorrectly classified, it can be labeled as False Positive (FP) or False Negative (FN).

Given:
· TP: denotes the number of all true positive samples



(a) Classifier Model Architecture

(b) Model Architecture for Sequence Generation

Figure 2: Pipeline models

· TN: denotes the number of all true negative samples
· FP: denotes the number of false positive samples
· FN: denotes the number of False Negative samples.

Accuracy measures the model's performance by adding the number of true positive and true negative samples and dividing it by the total number of samples (TP, FP, TN, and FN), as shown in the following equation.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

4 Baseline Implementation Results

Table 1 and table 2 below show the results of the model on the 2 and 4 intents, respectively. Each cell was produced from using S number of speakers along the rows and K recordings along the columns for training. That is to mean where S = 1 and K = 1, the model was trained on one utterance from one speaker. Overall, although there is high variance in results, the model performed better as the number of speakers and records increased. However, when the number of records exceed 5, the rate by which the performance improves starts to decrease. This suggests, in case one is faced with low-resource setting situation, that using 2 utterances in case of two intents and 3 utterances in case of 4 intents would be enough to train our model and trust that the results are going to be good.

	k=1	k=2	k=3	k=4	k=5	k=6	k=7
S=1	62.52	89.58	76.25	80.62	84.37	88.88	79.54
S=2	68.75	85.41	77.51	84.37	80.46	91.66	87.51
S=3	65.62	90.66	82.53	90.62	84.37	90.97	89.22
S=4	68.95	91.16	82.55	90.65	79.68	94.44	87.75
S=5	68.78	89.58	81.25	89.58	85.15	90.27	85.79
S=6	68.82	91.62	82.57	89.59	86.15	91.66	85.79
S=7	68.45	92.57	85.03	86.45	88.28	92.36	84.09

Table 1: Validation accuracy on two intents

	K=1	K=2	K=3	K=4	K=5	K=6	K=7
S=1	62.50	59.37	66.67	56.25	52.50	57.29	64.28
S=2	62.50	68.75	60.41	76.56	63.75	75.00	61.60
S=3	75.00	75.00	77.08	75.33	77.50	70.83	85.71
S=4	73.33	68.75	83.33	75.33	77.50	83.33	75.00
S=5	75.20	75.55	81.25	71.87	80.00	83.33	83.92
S=6	87.50	78.12	79.16	89.06	88.75	84.37	86.60
S=7	81.25	84.37	87.50	87.50	83.75	95.83	86.60

Table 2: Validation accuracy on four intents

5 Data augmentation results

The tables below containing results for sequence to sequence generation and phoneme substitution, indicate that generally our augmentation techniques do not improve on the performance of the baseline. Sequence generation show improvements on some combinations of speakers and utterances and is mostly comparable to the baseline, but one can not rely on it for better performance. Phoneme substitution on the hand, performed mostly worse especially on on very few utterances.

	k=1	k=2	k=3	k=4	k=5	k=6	k=7
S=1	57.81	65.14	71.59	75.44	74.65	67.26	77.45
S=2	64.06	69.64	65.99	70.08	76.38	65.17	64.04
S=3	60.93	81.25	68.75	72.76	72.56	72.32	78.36
S=4	62.35	84.82	71.51	89.73	86.85	86.01	80.57
S=5	67.18	87.71	77.72	83.03	90.62	90.77	88.25
S=6	65.62	80.23	74.41	86.16	82.29	80.05	83.24
S=7	67.87	83.03	84.09	92.85	85.41	84.82	89.32

Table 3: Validation accuracy on two intents sequence generation

	k=1	k=2	k=3	k=4	k=5	k=6	k=7
S=1	56.25	53.75	55.83	54.65	52.78	56.875	64.22
S=2	68.75	59.37	56.25	57.85	63.75	73.95	52.25
S=3	62.05	65.62	64.58	76.56	61.25	62.45	65.17
S=4	71.25	78.15	85.41	73.45	83.75	82.29	73.21
S=5	75.44	75.16	93.75	65.62	77.57	83.33	91.97
S=6	62.56	81.85	95.83	81.25	75.01	83.33	89.14
S=7	81.25	84.85	91.66	89.06	81.25	82.296	68.75

Table 4: Validation accuracy on four intents sequence augmentation

	k=1	k=2	k=3	k=4	k=5	k=6	k=7
S=1	25.00	50.00	37.50	59.38	43.75	50.00	51.56
S=2	43.75	56.25	53.12	50.00	66.67	64.58	67.19
S=3	43.75	93.75	65.62	81.25	72.92	93.75	75.00
S=4	25.00	62.50	59.38	68.75	60.42	81.25	73.44
S=5	50.00	75.00	75.00	81.25	66.67	81.25	71.88
S=6	43.75	87.50	71.88	68.75	70.83	83.33	76.56
S=7	50.00	93.75	68.75	84.38	68.75	85.42	71.88

Table 5: Validation accuracy on two intents substitution augmentation

	k=1	k=2	k=3	k=4	k=5	k=6	k=7
S=1	37.50	43.75	39.58	35.94	40.00	35.42	36.61
S=2	62.50	56.25	52.08	56.25	57.50	56.25	41.96
S=3	75.00	65.62	47.92	60.94	68.75	58.33	66.96
S=4	56.25	59.38	54.17	71.88	53.75	67.71	63.39
S=5	68.75	62.50	68.75	68.75	68.75	69.79	74.11
S=6	56.25	71.88	58.33	59.38	71.25	61.46	76.79
S=7	68.75	59.38	68.75	51.56	68.75	76.04	69.64

Table 6: Validation accuracy on four intents substitution augmentation

6 Conclusion

The above results suggest that an end-to-end classifier built on phonemes produced by Allosaurus can be used to deal with issues that arise when building speech recognition systems of low resource settings. This might be very useful for building speech recognition systems for many low resource languages, especially in Africa where no advanced Automatic Speech Recognition systems have yet been built mostly due to lack of enough data.

We also saw that adding data augmentation techniques in an attempt to create diverse utterances of similar semantics and hence increase the dataset did not improve the performance of the classifier used. This could be because the techniques used are not suitable for the task or they need improvement implementation-wise. As such, future works could focus on using more advanced sequence generation approaches such as Variational Auto-encoders or Generative Adversarial Networks to generate diverse sequences for data augmentation. Lastly, it would be better to experiment the model on collected data and build a model that is trained to recognize languages that belong to the same family.

7 Github Repository

https://github.com/GeoffreyKimani/IDLProject

References

[1] V. W. Zue, "Human computer interactions using language based technology," in Proceedings of ICSIPNN '94. International Conference on Speech, Image Processing and Neural Networks, Apr. 1994, p. I-VII vol.1. doi: 10.1109/SIPNN.1994.344982.

[2] D. K. Calteaux, "Human language technologies for African languages," p. 27.

[3] T. Nthite and M. Tsoeu, "End-to-End Text-To-Speech synthesis for under resourced South African languages," in 2020 International SAUPEC/RobMech/PRASA Conference, Jan. 2020, pp. 1–6. Doi: 10.1109/SAUPEC/RobMech/PRASA48453.2020.9041030.

[4] M. Doumbouya, L. Einstein, and C. Piech, "Using Radio Archives for Low-Resource Speech Recognition: Towards an Intelligent Virtual Assistant for Illiterate Users," arXiv:2104.13083 [cs], Apr. 2021, Accessed: Feb. 26, 2022. [Online]. Available: http://arxiv.org/abs/2104.13083

[5] A. Gupta, X. Li, Rallabandi, Sai Krishna, and A. W. Black, "Acoustics based intent recognition using discovered phonetic units for low resource languages," 2021, pp. 7453–7457. doi: 10.1109/ICASSP39728.2021.9415112.

[6] S. Y. Feng et al., "A Survey of Data Augmentation Approaches for NLP," arXiv:2105.03075 [cs], Dec. 2021, Accessed: Mar. 05, 2022. [Online]. Available: http://arxiv.org/abs/2105.03075

[7] A. Magueresse, V. Carles, and E. Heetderks, "Low-resource Languages: A Review of Past Work and Future Challenges," arXiv:2006.07264 [cs], Jun. 2020, Accessed: Feb. 26, 2022. [Online]. Available: http://arxiv.org/abs/2006.07264

[8] Sciforce, "NLP for Low-Resource Settings," Sciforce, Oct. 11, 2019. https://medium.com/sciforce/nlp-for-low-resource-settings-52e199779a79 (accessed Feb. 26, 2022). [9] A. Gupta et al., "Intent Recognition and Unsupervised Slot Identification for Low Resourced Spoken Dialog Systems," arXiv:2104.01287 [cs], Sep. 2021, Accessed: Apr. 02, 2022. [Online]. Available: http://arxiv.org/abs/2104.01287

[10] M. Faruqui and D. Hakkani-Tür, "Revisiting the boundary between ASR and NLU in the age of conversational dialog systems," Computational Linguistics, vol. 48, Art. no. 1, Apr. 2022, doi: 10.1162/coli$_{a0}$0430.

[11] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, and E. Cambria, "Recent advances in deep learning-based dialogue systems," May 2021.

[12] A. Gupta, Sai Krishna Rallabandi, and A. W. Black, "Mere account me kitna balance hai? - On building voice enabled Banking Services for Multilingual Communities," CoRR, vol. abs/2010.16411, 2020, [Online]. Available: https://arxiv.org/abs/2010.16411.

[13] Z. Guo, Y. Li, G. Chen, X. Chen, and A. Gupta, "Word-free spoken language understanding for mandarin-chinese," 2021, doi: 10.48550/ARXIV.2107.00186.

[14] B. Li, Y. Hou, and W. Che, "Data Augmentation Approaches in Natural Language Processing: A Survey," arXiv:2110.01852 [cs], Nov. 2021, Accessed: March 06, 2022. [Online]. Available: http://arxiv.org/abs/2110.01852

[15] Hou, Y., Liu, Y., Che, W. and Liu, T., 2022. Sequence-to-Sequence Data Augmentation for Dialogue Language Understanding. [online] arXiv.org. Available at: https://arxiv.org/abs/1807.01554