Naturalization of Speech Using Filler Words

Ashwini M. Joshi, Parth Shah, Richa Sharma[1]

Abstract— **In this work, we present a set of methods to naturalize speech by transformation of text. The naturalization of speech is achieved by the addition of filler words or pauses. The first method is a bi-gram method that uses the frequency of the occurrences of desired bi-grams in input data as a basis to insert filler words or pauses in the given input sentence. The second method is a hybrid method that uses Recurrent Neural Networks with Long Short-Term Memory units to confirm the output of the previous method. On conduction of a blind survey, our methods are at par if not better than natural speech.**
Keywords: Naturalization of speech, filler words, uh, um, pauses, bi-grams, LSTM.

I. Introduction

Complex electronic systems that interact with humans are making their way into everyday life. From robots in the house to personal assistants on our phones, these systems aid us in completing both simple and complex tasks. Interfacing with these complex electronic systems has also evolved. The focus has shifted to voice based interactions over the more traditional text based interactions seen in legacy systems.[1] A recent example is Google's Duplex that can make calls on your behalf to book appointments or reserve tables.[2] Voice based commands and feedback provide a more natural way for us to interface with these systems. With the increase in voice based interaction, we propose methods to naturalize these interactions. As a part of the complex pipeline of posing a query to receiving a response from a system, we propose a module. This module takes the response text as input and outputs naturalized text which can then be passed to say, a text to speech (TTS) module. The goal is to arrive at a method which enables us to have a more natural conversation with a system.

Previous cognitive studies of natural human speech indicate the addition, deletion, modification of words, sentences of standard speech that are observed in natural conversation.[3] To implement the idea of naturalization of speech we analyze and implement the addition of two filler words and pauses. The filler words chosen are 'uh' and 'um'. In natural conversation, the above mentioned filler words and pauses are used when a conversationalist is either gathering their thoughts or emphasizing on a part of the conversation.[3] These filler words are an important part of natural conversations and unconsciously end up as a part of them. By inserting them in standard text, the text is transformed and as this study details, transformed to a more natural text. As an example, consider the following:

[1]Computer Science and Engineering, PES University, Bangalore. In alphabetical order.

Input Text: Let's see, Susan is 15. Aundrea is 9. Every stupid cliche you hear about kids, they change your life they make you a better person, they make you whole, it's all true! Now I get it. Output Text: (um) Let's see (pause) Susan is 15. Aundrea is 9. Every stupid cliche you hear about kids, (pause) they change your life, they make you a better person, they make you whole, (pause) it's all true! Now (pause) I get it.

The closest imitation of natural speech which is well recorded can be found in movie scripts as part of its dialogues. We have used the Baskin Engineering, UC Santa Cruz, Film Corpus 2.0[4] and the Cornell University, Movie Dialogue Corpus on Kaggle.[5] This is a large corpus of text consisting of approximately 2.2 million lines of text together. Out of this, 200 thousand plus lines have been identified with filler words of the three kinds, 'uh', 'um' and pauses. These identified lines were used as input data. This leads to a primary assumption in this work; the tasks of (a) identifying the correct position to insert a filler word and (b) identifying the correct filler word are sentence independent. Context limited to the sentence is sufficient.

As there are multiple ways of achieving natural speech, we aim to transform text in a way that sounds most natural. Our methods are evaluated using a medium sized sample of individuals using a blind test instead of traditional pattern matching or other concrete evaluation metrics.

Variation 1: This is (pause) the first time I am (uh) coming here. Variation 2: This is the (uh) first time (pause) I am coming here. Variation 3: This (uh) is the first time I am coming (pause) here.

Overheads of processing sound can be saved by adding this module before a TTS module and directly operating on say, response text. We have paid attention to the latency of the module as we realize it would be a part of a larger pipeline. A text based transformation is detailed in the following sections.

II. Methods

A. Lazy Lookup

In this method, the insertions and their relative order are based on the frequency of occurrences in the input data. We name this method 'Lazy Lookup' for the purpose of this study. This method was adopted from a publication which built an FSA network for the same purpose.[6] The Lazy

Lookup takes an input sentence that needs to be naturalized. It is configurable with parameters such as percentage of naturalization, probability distribution. It outputs the transformed text as a sentence with the filler words inserted. This can be easily refactored to take multiple sentences as inputs. Lazy lookup consists of the following steps:

1) Create bi-grams with frequency count for input text. As prepossessing, the start of each sentence is appended with a special token. This is to not rule out the possibility of insertions of filler words at the beginning of a sentence.
   a) Create all possible bi-grams. Let the first word in the bi-gram be position 1 and second position be position 2.
   b) Discard bi-grams that do not have a filler word as part of the pair.
   c) Count and store the frequency of each bi-gram created in descending order. Top entries in the pre-formed bi-grams are detailed in Table I.
2) Clean the input sentence. A basic cleanup is performed where case is modified, quotes are dropped, special characters and numbers are dropped. Punctuation's are treated as their own words.
3) Load the pre-formed bi-grams with frequency count. Once computed, they can be loaded into memory upon an input text query.
4) Find possible insertions in the input sentence and perform a lookup in the pre-formed bi-grams
   a) Split the input sentence into bi-grams
   b) Iterate through pre-loaded bi-grams to find all possible insertions. This means, either position 1 or position 2 can match with an entry in the pre-loaded bi-grams
   c) All possible insertions are returned
5) Draw from the possible insertions based on a percentage and probability distribution
   a) The model allows for a 'naturalization percentage'. This translates to the number of insertions of filler words in an input sentence of a fixed length. Higher the percentage, more filler words are inserted.
   b) The probability distribution can be selected to bias the draw from the possible insertions. A uniform probability distribution is the default. The probabilities assigned to set draw, D are detailed in Equation 1.

$$D = \{\forall x \in D, P(x) = \frac{1}{len(D)}\} \qquad (1)$$

6) Construct output by adding the drawn possible insertions to the output sentence

A clear observation here is that the some bi-grams occur in natural speech more often than others and this is a property of natural speech that we are exploiting.

TABLE I
STORED BI-GRAMS AND THEIR FREQUENCIES IN INPUT TEXT.

Position 1	Position 2	Frequency
(pause)	I	13062
(pause)	and	8364
you	(pause)	4533
and	(uh)	100
(uh)	I	88

Recognizing that having a valid draw for all input sentences is not possible, we have a fallback method. This method is called upon if none of the words in the input sentence match our input data. It is identical to the method detailed above but a layer of abstraction is added. Instead of using the words of the input sentences, the model uses the Parts of Speech (POS) of the respective words in the input sentence. This is generated using a POS tagger.[7] Following is an example of a converted sentence. There is a significant performance overhead as the input sentence has to be tagged with the respective POS tags.

Input Sentence: I do not understand the final question.
POS tagged: NNS VBP RB VB DT JJ NN.

The merits of this method include (a) fast lookup for high latency as the bi-grams of the input text can be pre-formed and (b) diversity in inserting filler words as it involves a randomize function. The performance of this method is detailed in the results section.

B. Hybrid

In this method, a combination of the bi-gram method and an additional method is used. This additional method is a Recurrent Neural Network (RNN) composed of units of Long Short Term Memory (LSTM)[8] units to predict the next word given a seed phrase. LSTMs are popularly used as units of an RNN when a particular text generation problem is at hand. We name this method 'Hybrid' for the purpose of this study. Instead of using a draw function to choose a subset of possible insertions in the bi-gram method, all possible insertions are stored. These possible sentences are passed onto the RNN model to confirm the insertion. If the predicted word from the RNN matches the possible insertion, the sentence is confirmed. The hybrid consists of the following steps:

1) Clean the input sentence. A basic cleanup is performed where case is modified, quotes are dropped, special characters and numbers are dropped. Punctuation is treated as its own word.
2) The input sentence is sent to the bi-gram method.
3) A tokenizer[9] is instantiated

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 2, 10)             17340
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               44400
_________________________________________________________________
dense_1 (Dense)              (None, 1734)              175134
=================================================================
Total params: 236,874
Trainable params: 236,874
Non-trainable params: 0
```

Fig. 1. Description of the model used as part of Hybrid

a) Data from an input file is loaded and fed into the tokenizer.
b) The tokenizer converts the text to sequences.
c) Vocabulary size, total sequences are calculated. Sequence length is set to three. The model considers sequences of two words in predicting one word.
d) The sequences are converted to categorical data for training.

4) An RNN is setup with parameters as detailed in Figure II-B. The input data consists of the sequences loaded from the input file and passed through the tokenizer. The hyper parameters for the model are:
   - Number of layers: 1
   - Units: 100 LSTMs
   - Loss function: Categorical Cross-entropy[10]
   - Optimizer: Adam[11]
   - Epochs: 300
5) The input sentence with a possible insertion is passed to this model to predict the next word. If it is the same word as the selected word, the insertion is confirmed and the entire sentence is saved. This is repeated for all possible insertions. Based on the 'naturalization percentage' from a set of all confirmed sentences, sentences are selected and flattened to apply more than one filler word to a particular input sentence.
6) Construct output by printing the flattened sentence as the output sentence

The input data for the RNN is a concise file of approximately 500 lines only. The length of each line is also limited. These lines represent the best subset of training data. This is done to ensure a high accuracy level. For larger input data, an accuracy drop is observed as this simple model fails to fit a large sample space.

An additional step was performed on the output of the above mentioned model. Since the input data was heavily biased towards 'pause' as the filler word, the output sentences mostly consisted of this as the filler word as they are more likely to get confirmed. However, according to a cognitive study of natural human speech,[3] pauses in natural conversations can be replaced by other filler words. Long sentences are also more likely to begin with filler words such as 'uh' and 'um'. Hence, a last stage that replaces the 'pauses' with 'uh's and 'um's was incorporated.

An example iteration of the above model is detailed below:

Input Sentence: Let us try this one more time.
Word index: Let
Next word from the bi-gram method: (pause)
Generated word from the hybrid method: me
Predicted output: Let me
Incorrect Prediction

The merits of this method are it's correctness. There is no probabilistic element associated with the final output. However, this method over-fits for the input data and is highly sensitive to the same.

III. Results

As discussed previously, there is no one correct way of naturalizing speech. Filler words and pauses can be inserted at multiple positions in a sentence. Therefore, to test our models, we present a blind survey. The survey conducted consisted of clips which were excerpts from interviews found on the internet. These interviews were first transcribed with the inclusion of filler words and pauses. This was then used as inputs for our methods by obscuring the filler words and pauses. As a result, we obtained the output with the insertion of filler words and pauses. A voice actor voiced the original interview as well as the transformed interview for uniformity. Two interviews from each of the methods were added to the survey.
Each section on the survey had the original interview and the respective transformed interview. The sample of individuals were asked to pick one from each set that sounded more natural.The options included were either of the interviews and a third option, 'cannot determine'. We consider the option 'cannot determine' as a win as this would mean that our method is at par with natural speech.

A. Lazy Lookup

A few examples from our test corpus are detailed:

Original Interview: When we actually started recording the album we had this beautiful place, when we like rented this kind of beach (pause) shack match and (um) that's the only thing I asked for in the budget though.
Lazy Loading Output: When we actually started recording the (pause) album we had this beautiful place when we like rented this kind of beach shack match (uh) and that's the (pause) only thing I asked for in the (pause) budget though.

The readings in Table II represent the percentage of people who found that particular interview sample to sound most natural. Here we can notice that the results of the Lazy Lookup model is convincingly natural.

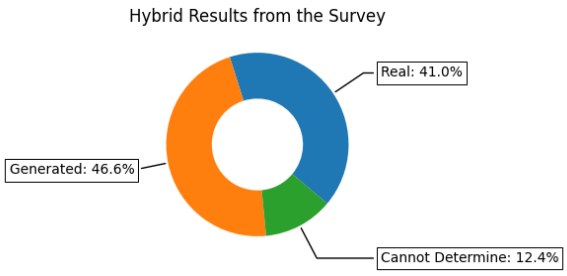

Fig. 2. Lazy Loading Results from the Survey

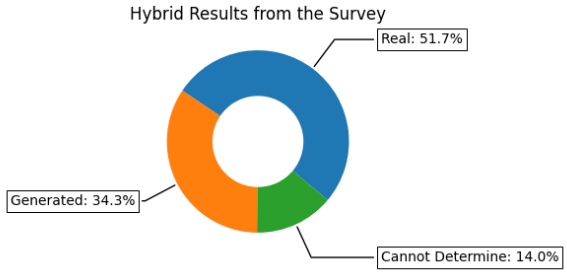

Fig. 3. Hybrid Results from the Survey

B. Hybrid

A few examples from our test corpus are detailed:

Original Interview: I spend a week in a year where I just go off and (uh) read (pause) people's PhD theses and new things that are going on in the field. Hybrid Output: I spend a week in a year where I just (uh) go off and read people's PhD theses and new things (uh) that are (pause) going on in the field.

The readings in Table III represent the percentage of people who found that particular interview sample to sound most natural. Here we can notice that the results of the Hybrid model is at par with the original interview.

TABLE II
RESULTS FOR LAZY LOOKUP SECTIONS IN THE SURVEY.

Sample	Original	Generated	Cannot Determine
Sample A	32.4%	52%	15.6%
Sample B	51.4%	40.5%	8.1%

TABLE III
RESULTS FOR HYBRID SECTIONS IN THE SURVEY.

Sample	Original	Generated	Cannot Determine
Sample A	54%	36.2%	9.8%
Sample B	49.4%	32.8%	17.8%

IV. DISCUSSION

This paper proposes different models for transforming text to a more natural spoken language and discusses the viability of these models. We then tested the results on a medium sized sample of people to ensure that the filler words were not being inserted in places that did not make sense. Studies have shown that humans tend to use filler words and pauses to communicate in a better manner. This helps the listener to understand what is being said more effectively. With an increase in the interaction between humans and electronic systems via voice in our day to day lives, natural speech can be used to make machine-human interactions seem more natural and familiar.

REFERENCES

[1] Voice Assistant Timeline: A Short History of the Voice Revolution https://voicebot.ai/2017/07/14/timeline-voice-assistants-short-history-voice-revolution/ Accessed: January 3, 2020
[2] Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html Accessed: January 3, 2020
[3] Clark, Herbert & Fox Tree, Jean. (2002). Using uh and um in spontaneous dialog. Cognition. 84. 73-111. 10.1016/S0010-0277(02)00017-3.
[4] Marilyn A. Walker, Grace I. Lin, Jennifer E. Sawyer. "An Annotated Corpus of Film Dialogue for Learning and Characterizing Character Style." In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey, 2012.
[5] Movie Dialog Corpus https://www.kaggle.com/Cornell-University/movie-dialog-corpus Accessed: January 30, 2020
[6] Sundaram, Shiva & Narayanan, Shrikanth (2003). An empirical text transformation method for spontaneous speech synthesizers.
[7] Categorizing and Tagging Words http://www.nltk.org/book/ch05.html Accessed: February 2, 2020
[8] Long short-term memoryhttps://en.wikipedia.org/wiki/Long_short-term_memory Accessed: February 2, 2020
[9] Encoder-Decoder Recurrent Neural Network Models for Neural Machine Translationy https://machinelearningmastery.com/encoder-decoder-recurrent-neural-network-models-neural-machine-translation Accessed: February 4, 2020
[10] Long Short-Term Memory https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy Accessed: February 4, 2020
[11] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980 [cs.LG]