Disaster-Bot: Tackling a Limited Corpus By Utilizing External Corpora in Imitative Dialogue Generation

Tristan Sparks
Department of Computer Science
McGill University
Qubec, Montreal H3A 0E9
tristan.sparks@mail.mcgill.ca

Mahyar Bayran
Department of Computer Science
McGill University
Qubec, Montreal H3A 0E9
mahyar.bayran@mail.mcgill.ca

Abstract

Markov Chains have gained notoriety as popular method for text generation in the style of a specific corpus. They make language modeling extremely simple, often with hilarious results. However, often when trying to imitate a specific style of text, such as a movie script, the small corpus size may limit the quality of results. In this paper we use a Markov model to generate text in the style of Tommy Wiseaus 2003 screenplay The Room, and we compare various techniques for utilizing external corpora to mitigate the effects of the small corpus.

1 Introduction

Text generation in the style of a specific corpus has become incredibly popular across the internet in recent years with well known projects such as, (Hracek, 2016; Justina Cho, 2017; Heaton, 2016). In discussing ideas for this project we were initially inspired by the now-Twitter-famous Automatic Donald Trump (Hracek, 2016). We decided on Tommy Wiseau's 2003 screenplay The Room due to its highly characteristic style of dialogue and limited lexicon. However, we quickly realized that the quality of our results would be limited by the same attributes which initially led us to choose this corpus, compounded by its short length.

Upon realizing the limitations of our chosen corpus, we decided to try and improve upon the basic Markov model, without resorting to the much more verbose approaches that one could take using RNNs (Graves, 2014), as these can quickly become complex and costly. As a baseline we will use the standard Markov Model as summarized in (Grzegorz Szymanski, 2018). We will expand upon this model by incorporating data from external corpora, selected to be similar in content. Our approach has the advantage of remaining relatively fast, and simple to understand.

The primary limitation of a Markov model is that its memory is intrinsically limited to the previous n words by the Markov assumption. When testing our baseline model, the generated text would often get stuck in loops, or generate text identical to text present in the original script. With these problems in mind we decided to try and supplement our data with similar data from different sources. The idea is that we can use these external weights to hone in on those ngrams which actually have high probability in natural speech, and lessen the influence of ngrams which are highly context specific, or which may not occur naturally in language. We also hoped that this strategy would partially resolve issues with grammaticality, although this is obviously infeasible to perfect in a model with memory limited to a finite window of words.

We will compare two methods of combining external data with our baseline. One which includes all external data, and one which cherry picks only the external data which is directly relevant to our primary corpus. We will also touch on the effects of smoothing and higher order Markov models.

2 Methods

As a primary corpus we selected the script from the 2003 movie The Room. Our baseline for modeling

will be a standard first order Markov model consisting of an initial word distribution and a transition matrix calculated using MLE estimates with optional add-δ smoothing. We will then look at incorporating data from an external corpus.

2.1 External Corpus Selection

The external data used will be a selection of screenplays taken from a list of IMDBs top 40 romance movies. We want the external data to be captured from scripts that are in a way similar to the primary corpus. We pick the union of 1-grams and 2-grams as the set of n-grams and we use ROUGE-N metric for nomination of external corpora being similar to the primary corpus to choose the ten most similar scripts and use these as our external corpora.

2.2 External Corpus Deployment

We will look at two models which combine data from our primary and external corpora. Note that in both methods we independently calculate both the primary and external transition matrices. In the first model (named *M-1*), we will combine primary and external distributions using a weighted average of all data from both the primary and external corpora, experimenting with the effects of various weightings. This yields a distribution calculated by the formula

$$T_{i,j} =$$

$$PCW * TPrimary_{i,j} + (1 - PCW)TExternal_{i,j}$$

over all tokens i, j in both corpora, where PCW refers to the primary corpus weight.

In the second model (named *M-2*), we will cherry pick only the information from the external corpora which corresponds to words and bigrams present in the primary corpus. For example, if the token "Oh hi" is present in the both the primary and external corpus, we will include $TExternal_{oh,hi}$ in the batch of external information to be used. We will then normalize the batch of external probabilities using the sumbasic function, and take a weighted average as was done in *M-1*, with the result that $T_{i,j}$ is now over only those tokens which exist in the primary corpus.

2.3 Evaluation

Evaluating the quality of generated text can be tricky (Xie, 2018) as the perceived quality of generated text may vary significantly by reader. In Machine Translation, one can measure quality by content overlap as there exists a gold-standard reference. In Auto-Summarization, quality is also much more heavily tied to content overlap. Thus in these tasks it makes sense to use ROUGE or BLEU metrics for evaluation. However, in generating imitative text, the definition of quality becomes murkier. Grammaticality is a reasonable metric, as well as similarity in tone, however we also wish to generate text which conveys semantic content not originally present in the source, in which case using similarity metrics like BLEU makes little sense. As far as we know there is no such metric that could qualify our work.

We will use Perplexity to measure the information encoded in each model as a sort of baseline for quality, and supplement this with qualitative observations of the grammaticality, coherence, and originality of some samples of text generated by each model.

We will also compare the results of our modified Markov models with results obtained by training an RNN created by (Justina Cho, 2017). The authors have used LSTM-RNNs to generate Style-imitative text on corpora such as Sherlock Holmes stories and The Odyssey.

3 Results

3.1 Baseline Model

Some samples of text generated by our baseline standard Markov model are shown below. When a trigram model was used, the generated text tended to either get stuck in a loop due to the sparsity of the transition matrix, or generate sequences of text identical to sequences from the training corpus due to overfitting. A bigram model performed much better and will thus be used for subsequent tests.

Bigram Model Text Samples (Perplexity: 2.83):

– Itll be happy. Life you invited all he is time? you alright? tell me? whats happening with Johnny found somebody had.

– Why do their minds all outside? women out for you? I agree with her mind. But im just need a lot of course im fine how are you calling a princess

and I still love you want.

– Why? whats the interesting part of course im through with me crazy. Ill see Michelle. I know that.

Trigram Model Text Samples (Perplexity: 2.83)

– Thanks a lot of faith in Lisa. look. If you dont understand you. Were probably going to be married next month.

*– Youre not going to shake up our group of friends. Its going to get hurt. Youve got a little bit of a b**** told me he plans to buy you a ring clothes whatever you wanted and now you*

– Whats going on here? hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi hi

Additionally, shown below are sample results generated using different δ values for add-δ smoothing. Due to the small size of our corpus, any amount of smoothing tended to over-smooth the transition matrix resulting in near-random output. Thus zero smoothing will be used going forward.

$\delta = 1$:
– Times cleaned couldnt pecs come solution question lot pictures depressed no central pocket somebodys party actually concentrate ditch wanna part off bad upstairs cream ready princess son pants figure uh waiting pesto believe cheep future yesterday using breakers still someone harold

$\delta = 0.1$:
– Oh not pesto way dennys due till pizza stairs straight by tea few awesome belongs Lisa ordered goodnight our kitchen mikes pizza hurry work loans bro invited loyal hunh loves dream shake but nice need flatout upstairs but upstairs about needed

$\delta = 0$:
*– In the clock? wheres my f***ing minutes bitch. Youre right! I gotta work early. I brought the bank we will come on you should confront her.*

We will see that supplementing our data with data from external corpora will act as a sort of smoothing, as well as maintaining the grammatical structure, since it's assumed that the external corpora are grammatically and syntactically correct.

3.2 M-1 Model

Summarized below are samples of generated text along with the perplexity of the model over a range of PCW values.

$PCW = 0.9$ $(PX = 1.16)$:
– His zipper? I dont think youre hurting yourself. What are? golden gate park? who? whatd you.
– And he still going to borrow some fun. I had. I bought you want to make love?

$PCW = 0.7$ $(PX = 1.17)$:
– Good too? I want any disrespect? why?! have breast cancer. I dont show them. I think about?
– Whats with him. Im always bugging me. They like a promise they betray your promotion. Gave a sudden I married I want from me move before?

$PCW = 0.5$ $(PX = 1.175)$:
– Because im tired im still gonna marry the ice cream please. This creep! wheres the gun goes wrong?
– That? my toothbrush. For your most girls smiled at you very soon and near great get his zipper?

$PCW = 0.1$ $(PX = 1.22)$:
– Jimmy. Come up. its... No one hundred... Ill talk about it was poor bastard. When mickey ds.
– You mean no? im doing? come over tonight we gotta come on the shyest man does not good.

The text generated by *M-1* is interesting. The primary improvement over the standard Markov model is a slight smoothing effect, which improved the quality of the generated text. For lower values of PCW, the external corpus is given more weight and you can see that tokens not present in the primary corpus begin to appear, generating more unique text which nonetheless retains much of the primary corpus' tone. The perplexity of *M-1* increases with a higher weight of external information. This

makes sense as the external corpus is much larger, and therefore as we give it more weight, more information is encoded in *M-1*.

3.3 M-2 Model

Like *M-1*, we test on different values of PCW.

$PCW = 0.9$ ($PX = 5961.48$):
*– You tell me to give me stupid b****. You know that. Now you to examine your slide out real good yeah.*
*– You take a b****. Are using end of a ring clothes whatever whatever. Thats ridiculous! hahaha.*

$PCW = 0.7$ ($PX = 2451.20$):
– Well at it. I have to establish these guidelines tramp! I yours. And he can Mike grand central station!
– Here. Aha wash up mixed up. thanks. Their homework. now. What are you have after all.

$PCW = 0.5$ ($PX = 1644.38$):
*– Dont love remember bank whenever you cup ever get killing alright? I gotta hysterical? story when is very soon could you f***.*
– Its going on?! how confuse me awesome owe me. I till fresh married hang on run you? ruin treat you have any more girls games.

$PCW = 0.1$ ($PX = 1090.72$):
– Use whoa there killed breast maybe treat excites takes hows saw it thought boyfriend marry signs soninlaw. Stairs days learn give well tramp Susan years human strong live lecture hot ive over giving sometimes shes cup manipulative would be sure
– Throw happening wife possible head to uhhuh ganged turned delivery asking eventually giving him help fresh fool lots head oh hey lots test miss plans few expected your story when movie known up qualified forgive is drive if right spoil work

The text generated by *M-2* exhibited minimal if any improvement over the baseline model.

3.4 LSTM-RNN

We used the LSTM-RNN used by (Justina Cho, 2017) and trained a model on the characters in the primary corpus with a sliding window size. Some of the trained LSTM-RNNs parameters are as follows: $optimizer = Adam$, $learning\ rate = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e{-}8$. We used the LSTM-RNN to generate a text with 500 characters:

*-any hica i don'one amy se moie lel, i move you, you jast dy toi bacyeruu? you're not my f***ing mother! you listee to me, boy no! somebody had better do something around here. are you okay, denny? i'm okay, are you okay? i'm okay! what's okay? he's taking drugs. come on, stop, it was a mistake. a mistake, that he takes drugs. let's go home. come on, it's clear. what's clear? i am going to call the police. mom, stop, it was denny's mistake, just stop! let's go. why did you do this? you know pe*

As can be seen, the results are not that interesting! This is due to the fact that the corpus is just too small for the RNN. The results would get better if the RNN is trained on the words of the corpus rather than the chars but as it has been said before, there is just not enough words in the corpus.

4 Discussion and Conclusion

In this project we compared four different models for dialogue generation, a standard Markov model baseline, *M-1* a model utilizing all information from external corpora, *M-2* which utilized selective external information, and a pre-programmed LSTM-RNN.

M-1 was the most successful model, with a noticeable qualitative improvement over the baseline. Including external information but maintaining a primary corpus weighting of greater than 0.7 contributes to an improvement in scope of generated text, virtually eliminating the issue of simply regurgitating text directly from the corpus, while still maintaining the tone of the primary corpus. Including external information also improved the expressivity of generated text. However, the grammaticality of the text remained relatively unchanged from the baseline.

The benefits described above were mostly lost by excluding uniquely external information as done in *M-2*. Although it produced an inordinately high perplexity score, the generated text was not of noticeably better quality than the baseline.

The issue of repeating text from the corpus was one which plagued the LSTM-RNN model, it transitioned quickly from nonsense, to verbatim lines from the corpus. This is a good sign for *M-1*, as it implies that with a small corpus a modified Markov model may actually be optimal. However, it is unlikely our model would beat an LSTM-RNN given sufficient training data.

In conclusion, utilizing external corpora in imitative text generation can produce much more expressive text than a standard Markov model, however it is still insufficient to produce entirely grammatical text, or text with a long term degree of coherancy due to the Markov assumption implicit in it's generation.

5 Statement of contributions

Both authors have contributed to the every stage of this project, such as design of the project, research, programming and implementation and writing the paper. For a more detailed account of the contributions please check out the GitHub repository of the project: *github.com/tristansparks/DisasterBot*

References

[Graves2014] Alex Graves. 2014. *Generating Sequences With Recurrent Neural Networks*. arXiv:1308.0850v5.

[Grzegorz Szymanski2018] Zygmunt Ciota Grzegorz Szymanski. 2018. *Hidden Markov Models Suitable for Text Generation*. researchgate.net.

[Heaton2016] Mike Heaton. 2016. *Trump-Bot*. mikeheaton.com/markov-trump-67711d895b34.

[Hracek2016] Filip Hracek. 2016. *Automatic Donald Trump*. filiph.github.io/markov/.

[Justina Cho2017] Andrew Zhou Justina Cho, Madhav Datt. 2017. *Style-Imitative Text Generation*. Harvard University, github.com/madhav-datt/textgen.

[Xie2018] Ziang Xie. 2018. *Neural Text Generation: A Practical Guide*. cs.stanford.edu/ zxie/textgen.pdf.