Evaluation of Generative Models on the Basis of Tweet Data

Zachary Bebbington
zbebbington@umass.edu

Quan Bui
qbui@umass.edu

Frick Shao
fshao@umass.edu

1 Introduction

Twitter is a social media platform utilized by millions every day. And some of those users tweet incredibly often, possibly multiple times a days every day of the week. What this provides us with is a larger amount of textual data pertaining to the writing style, as well as the writing content, of those users. The problem that we would like to address is: does a generative model exist that can effectively mimic the tweeting style of a user when trained on their past tweets?

Using a computer to write tweets for you will only interest those that don't understand the social aspect of twitter. But the true interest in addressing this problem is what it means about the models themselves. To discover a model that can accurately mimic the tweeting of a human would be to discover a model capable of achieving two incredible feats.

First off, in order to mimic the writing of a twitter user the model would have to learn the writing style that they utilize. It would have to learn all the mannerisms that the individual uses while they write. And for a model to achieve this level of text emulation would be incredible.

Secondly, the model would be required to categorize the different subjects that the user tweets about. Twitter users are allowed to write about anything. Which means in one post they could be talking about their lovely pets. But in another they could be discussing how they feel about the current political climate of the US. If the model is unable to recognize that these two topics are heavily dissimilar we could see it creating a tweet that mashes these two fields together which would result in a rather head scratching read.

And because no two people write the same way it would be achieving both of these feats using the limited data set that comes from that one twitter user.

2 Related work

Although, there are likely not any works that are exactly like ours, there are ones that have the same concepts. In a system called KBNL or Knowledge Based Natural Language Processing System, Jim Barnett, Kevin Knight, Inderjeet Mani, and Elaine Rich (1990) attempted to divide language into two kinds of information (Barnett, 1990). The goal of this work is to train a model to learn knowledge and rules about the language. This is similar to our first step except we want to train the model over one specific persons language. The first type of information are the facts about the world such as bunnies are under the broader category of animals. The second are linguistic rules. The example given by the authors is the sentence David ate the turtle (53). The word ate implies that there must be something being eaten and something doing the eating. It doesnt make sense to classify turtle under the category of food so that this sentence makes sense as they usually are not considered food. In the approach taken by the authors, they will instead derive two rules. The first is that almost any animal can be considered food. The second, an entity can potentially be referred to by the most common product used to make it. As a result, there does not need to be an explicit rule for every combination of words. For each animal, there is no need to give it a secondary meaning of meat. On the other hand, since this system is heavily based on knowledge of the world and rules, these need to be accurate. If the system did not know that turtle is a type of animal, it might interpret the sentence differently and produce a wrong conclusion. This means that there needs to be a lot of data for the system to train over. A second work that applies similar concepts is by Elena Not, Massimo Zancanaro, Mark T. Marshall, Daniela Petrelli, and Anna Pisett (Not, 2017). Their goal was to create personalized postcards given a set of logs for a specific place like a museum. It is similar to our goal as given some information, we output a tweet that is stylized to resemble a specific person. However, there is a difference in that the postcard is not attempting to emulate a person, and rather is just outputting information in a informal and natural way. The approach used is shallow template-based text generation. This means that given a specific museum, the system will use a specific set of pre prepared sentences with gaps that are to filled in with information from the logs. The sentences are assembled dynamically in order to make it sound more natural (4). This approach is obviously very limited and if we were to apply it to our model, our tweets would not sound like the intended target as people are not limited to pre-set sentences. However, it is suitable for the original intended task, as the goal was only to make the postcards sound natural. This type of model also has the benefits that it is very fast to train. It really only needs to know given some data, which gaps can the data fill in the pre-prepared sentences.

3 Your approach

Several generative models already exist that can generate text based on input text. Our approach to finding an answer to the proposed problem is to examine those already existing generative models. We will collect tweets from specific Twitter users through the Twitter API. Then train the models on those tweets to generate tweets that mimic the user's vocabulary and writing style.

We will find the best generative model for mimicking a twitter user. Best will be defined as tweets that have a logical/grammatical flow to them and can be distinctly identified as belonging to the user. Meaning if we take two users with drastically different tweeting mannerisms (grammar, spelling, word-choice, topic, etc), it will not difficult to tell which person the fake tweet was mimicking. Once we find the best one, we will optimize it for the task.

3.1 Milestones & Schedule

1. Find generative models to use, collect tweets from specific users, and generate tweets based on those users. (1 Week)

2. Analyze and compare the results to find the best model. (1 Week)

3. Then we will optimize the best model for the task. Changing aspects of the model, such as the loss function used. (2 Weeks)

4. Write progress report (Nov 16)

5. Examine how well the final model mimics a twitter users writing style. Do error analysis. (2 Weeks)

6. Write final report and prepare presentation. (2 Weeks)

4 Data

We will be utilizing tweets from specific twitter users as the data for our project. The exact users that we plan to use is still undetermined. But we want to pick at least one user that has made a large number of tweets, to act as a data set with a large training set, as well as someone who has tweeted relatively infrequently, to see how the models can perform on smaller sets of data. In addition, it may be good to find one user that tweet about various topics to see how well the models handle that, and another user that tweets about one topic to provide some kind of baseline.

This is data that can be easily accessed thanks to the Twitter API which allows us to pull the tweet history of any public twitter account.

Because our goal is to discover the generative model that can best emulate the tweeting style of a twitter user, this is the exact data that we want to work with.

5 Tools

We will be using any number of existing generative models that we can find available online. And from those, we will be using any libraries/toolkits that were involved with them. To gather and parse the data, we will be accessing the Twitter API through Tweepy for Python.

References

Barnett, Jim, e. a. (1990). *Communications of the ACM*, 33(8):50–71.

Not, Elena, e. a. (2017). *Proceedings of 12th Biannual Conference of the Italian SIGCHI Chapter*, pages 1–9.