

1 Introduction

For the final project I implemented a sentence generator using NLP techniques and strategies I learned during the semester as well as new ones I implemented to help with the process. The application is backed by a database of tweets with 604,862 tweets spread out into 273,010 topics. When I came up with the idea for this project it was to see if it was possible to generate at least pseudo-sensible sentences using samples of tweets around a given topic.

1.1 Uses

While this application does not provide a high accuracy model of grammatical correctness, it gives some insight into the similarities of different individuals' way of speaking. By taking all of the sentences surrounding a certain topic, it's possible to generate a new sentence that can be passed off as a human generated sentence. This application gives an insight into the base layer of different AI techniques used in chat-bots and personal assistants. It's a good starting point for expanding towards more grammatically correct sentence generation and other uses.

1.2 Input and Output

Sentence generation is the process of outputting sentences in a human readable form given some input of some corpus of information that can be parsed and analyzed. To give the application a better form of structure it uses two separate inputs, a user chosen topic, and a precompiled list of sentences and topics in a server-side database. The details and structure of these inputs are discussed in more detail in Section 2: System Description. Using these two inputs, the application creates MLE based language models with which it randomly generates sentences using rejection sampling.

1.3 Example Outputs

To give an idea of how well the application can work here are a couple of examples of good output:

Input: food

Output: *Attempting to clean a magazine named Sirene. Exiting! Sooooo food! Haha.*

Input: school

Output: *Seriously, school tommorrow... nooooooooooooo!!!!!!!*

Some bad output:

Input: school

Output: *In I.T. Bored to start babysitting for school... all good mood. My good bye seniors*

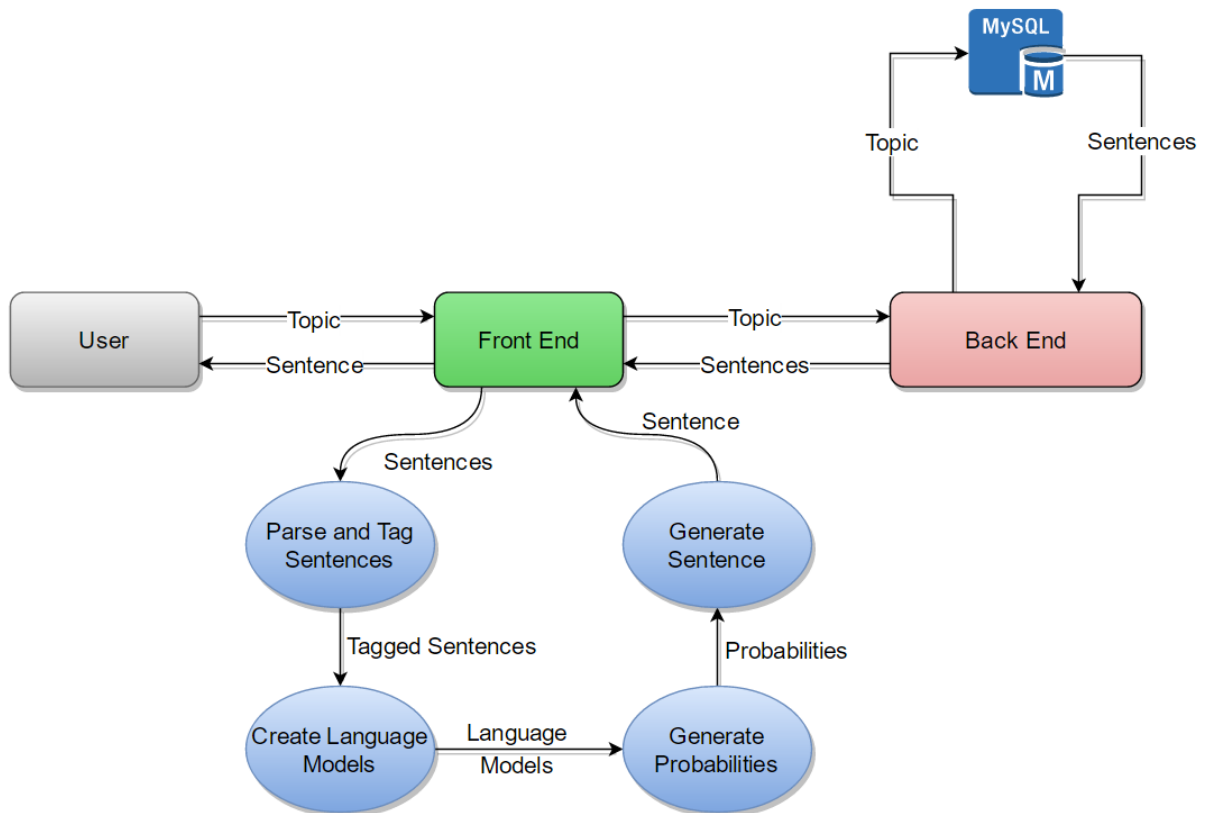
Input: rain

Output: *good want rain. Why rain - back in. I'm mad man Rhode Island tonight. . uh oh, it looks to wait to rain... going scuba diving tomorrow.*


2 System Description


2.2 System Diagrams

2.2.1 Overall Application Layout



2.2.2 Database Table Layouts

sentences	
	id INT PK
	sentences MEDIUMTEXT
Indexes	
PRIMARY id	

topics	
	id INT PK
	topics VARCHAR
	sentences LONGTEXT
Indexes	
PRIMARY id	
UNIQUE id	
UNIQUE topics	

2.1 System Overview

In order to use the application all the user needs to enter is a topic. As long as it's one word and shorter than 50 characters, the topic gets sent to the server. The server then checks the database for the topic and if it exists in the database, all of the sentences for that topic are pulled and sent back to the client front end.

The first step in the sentence generation is preparing the sentences. Each sentence is parsed, any empty tokens are removed, and starting and ending sentence tokens are added to it [`<s>`, `</s>`]. Once every sentence has been parsed, the list of sentences gets sent to the next routine to generate the language models. The language models generated are unigrams (single-word) with counts and bigrams (double-word) with counts. The language models are used to create an MLE based probability model.

The final routine takes the bigrams and the probabilities to generate a sentence using the probabilities as weights for rejection sampling. Starting with the start tag ('`<s>`'), a list of possible bigrams are generated and chosen from to create the sentence until it reaches the end of the sentence. This final routine will continue running until a sentence is generated before the upper limit on length and it contains the topic.

3 NLP Techniques Used

You would (must) have used some specific NLP technique in your project. Describe this in some detail in your own words. Be as technical you want to be. I want to be convinced that you have understood this technique clearly.

3.1 MLE Probability Model

3.2 Rejection Sampling

4 Evaluation

You should evaluate your system. Many of you would have a baseline version and would have improved it in some way. You need to do some experimental evaluation. This involves describing the data set you use, what constitutes good output, and how you would measure the quality of your output (e.g., precision/recall). Present your results succinctly. Describe why you think the results turned out the way they did.

5 Discussion and Conclusions

Summarize what you've learnt from this project. This should include, challenges you encountered, what you did to solve them, a succinct explanation of results, and what you would suggest as future work to improve the system.

Resources and Referenced Material

Tweet Data – Sentiment140: <http://help.sentiment140.com/for-students/>

Rejection Sampling - https://en.wikipedia.org/wiki/Rejection_sampling

Application Hosting for Frontend and Database – OpenShift: <https://www.openshift.com/>

Twitter API - <https://dev.twitter.com/>

Twitter API Exchange Wrapper - <https://github.com/J7mbo/twitter-api-php>