

COSC 73: COMPUTATIONAL LINGUISTICS

FALL 2014, DARTMOUTH COLLEGE

---

## Twitter Bot Final Project

---

**pome**

**[sic]**

*Authors:*

Ben PACKER

Byrne HOLLANDER

*Professor:*

Sravana REDDY

November 25, 2014

# Contents

<b>1</b>	<b>Description of the motivating problem</b>	<b>2</b>
1.1	Goal . . . . .	2
1.2	The Poem Composition Process . . . . .	2
<b>2</b>	<b>A brief survey of related research, citations to the implemented algorithms, and description of similar existing products</b>	<b>3</b>
2.1	Brief survey of related research . . . . .	3
2.2	Citations to the implemented algorithms . . . . .	3
2.2.1	Find Similar Pairs of Tweets - <i>word2vec</i> . . . . .	3
2.3	Description of similar existing products . . . . .	4
<b>3</b>	<b>User feedback</b>	<b>5</b>
<b>4</b>	<b>Analysis of the program's shortcomings and thoughts on future development</b>	<b>6</b>
4.1	Analysis of the program's shortcomings . . . . .	6
4.2	Thoughts on future development . . . . .	6
<b>5</b>	<b>Links</b>	<b>7</b>

# 1 Description of the motivating problem

## 1.1 Goal

The goal of the project was to make a Twitter bot capable of composing poems made up of Tweets related to a particular subject. The result should be a combination of existing Tweets into a couplet, where the two lines have the same number of syllables and rhyme. The program interface is used entirely through Twitter: a Twitter user may tweet @PomeSic -query query, where the query is in quotes and specified by a flag. Ideally, the poem that our program composed should both satisfy the basic poem requirements and be a small representation of what the Twitter sphere thinks about the chosen topic at the given time. This last specification, that the poem should not only be coherent but reflective of the Twitter discourse about a particular subject, has led us to prioritize algorithms that create a more native Twitter feel, including the use of Twitter vocabulary and Twitter-specific context similarity.

## 1.2 The Poem Composition Process

Since the goal of the project is to compose a functional poem of existing Tweets, we did not have to come up with a mechanism of generating text from scratch. Instead, our approach to the problem was to get existing Tweets that could be combined to make poems, and then modify the Tweets so that they met our poetic specifications. The overall process of selecting Tweets and then modifying them is presented below:

- 1: Find similar pairs of Tweets
- 2: Normalize the Tweets.
- 3: Make the Tweets rhyme.
- 4: Make the Tweets the same number of syllables.
- 5: De-normalize the Tweets.

## 2 A brief survey of related research, citations to the implemented algorithms, and description of similar existing products

### 2.1 Brief survey of related research

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### 2.2 Citations to the implemented algorithms

#### 2.2.1 Find Similar Pairs of Tweets - *word2vec*

Whenever we use the word similarity in this paper to describe the similarity between two words, the metric we have decided to use is the cosine similarity of context vectors using a skip-gram context model. We used the freely available software *word2vec*<sup>1</sup> in order to compute the context vectors, and the Python package Gensims *word2vec* interface<sup>2</sup> for computing the cosine similarity. Although we initially encountered long loading times (around 3.5 minutes) when trying to load the context vectors into Gensim, after we recomputed the context vectors and stored them in binary, the loading time went down to a little under 30 seconds. While *word2vec* gives the user the option to use either a skip-gram model or a continuous bag of words model, we decided to use a skip-gram model based on the *Efficient Estimation of Word Representations in Vector Space* paper<sup>3</sup>. In the paper, the authors demonstrate that the skip-gram model outperforms a continuous bag of words model on its semantic disambiguation abilities, while a continuous bag of words model better reproduces syntactic similarity. Since the syntactic requirements of Twitter are loose already and we are primarily interested in this as a measure of semantic similarity, we decided to use the skip-gram model.

---

<sup>1</sup><http://code.google.com/p/word2vec/>

<sup>2</sup><http://radimrehurek.com/gensim/models/word2vec.html>

<sup>3</sup><http://arxiv.org/pdf/1301.3781.pdf>

## **2.3 Description of similar existing products**

Several similar Twitter bots exist now. One of them is @Pentametron which looks for poetry in everyday musings. It retweets tweets that are written in perfect iambic pentameter and even posts unintentional sonnets on a website associated with the account. There is also an accidental haiku bot (@accidental575) that finds tweets in the 5-7-5 haiku form and tweets them formatted and with attribution.

### 3 User feedback

Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum.

## **4 Analysis of the program's shortcomings and thoughts on future development**

### **4.1 Analysis of the program's shortcomings**

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

### **4.2 Thoughts on future development**

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

## 5 Links

Link to Github page

<http://github.com/byrnehollander/pomesic>

Link to Twitter page

<http://www.twitter.com/pomesic>