Capturing Author Style with LSTM: A New Perspective

Vasily Shcherbinin, University of Surrey

Introduction & Project Objectives

Generating text via Recurrent Neural Networks, such as LSTM or GRU, has been a common interest in natural language processing. Text generation is often done via prediction of next character in a sequence, although other approaches exist.

This project aims to provide another perspective to the problem of embedding author style into the machine learning process - to create a model that can be used to generate text in the stylistic of the author of the original texts being learned. The project objectives are:

- Capture the author "style" and generate text in similar stylistic.
- Use vocabulary from another source to mimic original author style, e.g. Trump writing Shakespearean sonnets.

This will be achieved by:

- Developing a new approach, encoding words from Text 1 by their morphological structure and representing words as tokens.
- Feeding above tokens into Long short-term memory Recurrent Neural Network (LSTM - RNN) for learning.
- Decoding tokens generated by model to keep original author style, but using another author vocabulary from Text 2.

Literature Review - Prior work

- Author style was successfully captured via the use of a Seq2Seq LSTM-based model - two switches with tensor product were used to control the transfer in the encoding/decoding processes [1].
- Yandex.Autopoet is an online website created by Yandex that generates poetry in style of various Russian poets using browser searches. Original author style (structure, rhythm, rhyme) are captured using a neural network (training takes "tens of days") [2].
- In [3], text was encoded via embedding ID's and fed into a GRU for identifying the author - project unsuccessful.
- General overview of using Seq2Seq LSTM for stylistic transfer has been provided in [4].

LSTM

For this project, the following LSTM-128 configurations were used:

Layer (type)	Output Shape	Param #
lstm_9 (LSTM)	(None, 128)	90112
dense_9 (Dense)	(None, 47)	6063

Layer (type)	Output Shape	Param #
bidirectional_1 (Bidirection	(None, 256)	180224
dense_10 (Dense)	(None, 47)	12079
activation_1 (Activation)	(None, 47)	0

References

[1] M. Han, O. Wu, and Z. Niu, "Unsupervised automatic text style transfer using lstm," in *Natural Language Processing and Chinese Computing* (X. Huang, J. Jiang, D. Zhao, Y. Feng, and Y. Hong, eds.), (Cham), pp. 281–292, Springer International Publishing, 2018.

[2] P. Kontsarenko, "How does yandex.autopoet work," Jul 2016.

[3] T. Edirisooriya and M. Tenney, "Applying artistic style transfer to natural language,"

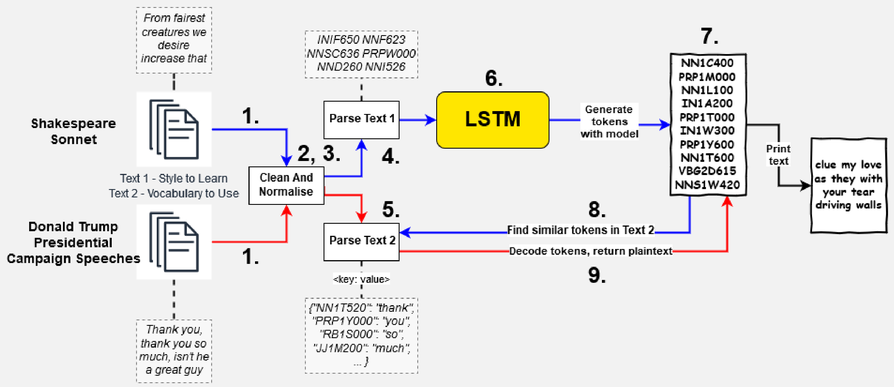
[4] J. Kabbara and J. C. K. Cheung, "Stylistic transfer in natural language generation systems using recurrent neural networks," 2016.

[5] D. E. Knuth, *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.

Problem Analysis

Two major problems to solve: how can we capture author style? How can we use author style to generate new text as someone else?

- Author style is made out of four components: word choice, sentence structure, sentence arrangement and figurative language.
- Therefore, in order to capture the author style, the four components above must be extracted from text and learned.
- Text generation is usually learned on a char-by-char basis - this method indirectly allows to capture word choice, but does not address other components above. Such approaches allow to generate text, but they do not provide any formal method of capturing the original authors writing style and embedding it into the text that is being generated by the machine. "Capturing style" is not easy. What ***is*** author style in the first place?
- Another challenge: how to use vocabulary from different source to generate text in the learned style? Common approaches do not allow this.
- Solution: Encode every word into a token containing information about morphological structure - this covers all four points above. Use this to learn the original text style, and then decode tokens using vocabulary from another text.



Methodology

1. Import datasets/texts as plain text - Text 1 will be used to learn the author style, Text 2 will be used as vocabulary/dictionary source for converting LSTM result back into human-readable text.
2. Clean and normalise Text 1 - remove redundant white space, punctuation, non-English words; convert to lower case. This can greatly enhance training times.
3. Clean and normalise Text 2. Split sentences into individual words.
4. Parse Text 1, generating different style corpuses (see code).
5. Parse Text 2, generating key-value vocabulary pairs.
6. Feed the Text 1 style corpus into LSTM-128 / Bidirectional LSTM-128.
7. Generate characters from model, e.g. 2000 characters. Split by white space.
8. Use Levenshtein Distance to find closest match for encoding from generated result in Text 2 - effective for situation if token for word in Text 1 is not present in Text 2. Optionally, impose additional constraints, e.g. syllable count, rhythm, rhyme.
9. Retrieve plain text word associated with encoding token and format as required.

Result evaluation

- Experiments carried out on two corpus: **Collection of Shakespeare sonnets** and **Donald Trump Presidential Campaign Speeches**.
- Using Multinomial Naive Bayes to validate results (42 test samples) generated by LSTM returns a 100% classification accuracy - clear distinction in author styles between Shakespeare and Trump texts generated by both LSTM and Bidirectional LSTM. **Proof that author style has been captured**.
- Mostly grammatically correct sentence structure with no absurd words. LSTM-128 and Bidirectional LSTM-128 performed equally well.

```
Accuracy :  1.0
              precision    recall  f1-score   support

 shakespeare       1.00      1.00      1.00        21
       trump       1.00      1.00      1.00        21

   micro avg       1.00      1.00      1.00        42
   macro avg       1.00      1.00      1.00        42
weighted avg       1.00      1.00      1.00        42
```

Multinomial Naive Bayes Text Classification Result

```
Trump:          he was bad they knew it thank you thank you seriously with liberal now I dont
                know how can with the people when i want to look china for trump for anybody
                before their fences in our country then so many people truly one of the hacks
                for fight one leadership having education she will awfully make itself to
                happen like i right the hell wanted all radio in iowa lawsuit even it wanted...

Shakespeare:    me thee lips to kiss the weakness and to we might live now the very foul
                what back you be him for you before me own that delayed shape doth cool in
                prize of the outward sooner dull where not for love  my pursuit love ruin
                how write where dig alone directed not some shadow mistress my moan will
                not born and such all not so painted side a power to see i more glass what...

Shakespeare-to-Trump:  Those fear clue fences could drew thus more clue my love as then they
                       with your tear driving walls will mine by some why you martyrs when
                       thinks more fourth off choices confront how i am northern dew by tea
                       crazy liked wears to know an going their brand weakness tea sun spending
                       confront water never kill my much more kill they when to say shut that
```

Text by LSTM-128: Trump, Shakespeare, Trump as Shakespeare sonnet.

- Attempts to capture the syllable structure of Shakespearean sonnets, but no understanding of rhyme - needs to be added separately.
- Author style substitution is ineffective - loss of context and original meaning, but some attempts at adhering to structure present.

New Encoding

The text is encoded via a synthesis of tokenisation and Soundex, a phonetic algorithm for indexing and comparing words by sound. In order to encode a word via Soundex[5]:

1. Keep the words first letter and remove all occurrences of a, e, i, o, u, y, h, w.
2. Substitute following letters with digits as per table below :

Code	Letters
0	A, E, H, I, O, U, W, Y
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R

3. Condense all adjacent identical digits into one digit (e.g. 11 -> 1).
4. If a digit of a letter is the same as the first digit, remove the digit but keep the letter.
5. If the result contains less than 3 digits, append up to three zeros to fulfil token length requirement.

For example, *"hello"* will be **H400**. Next, using NLTK.tokenise package, every word is tokenised by part of speech, e.g. NN (noun), JJ (adjective), etc. Finally, information regarding syllable count, phonetic stress and rhyme can be retrieved via Pronouncing library, an interface to the CMU Pronouncing Dictionary. This poster's project code can accommodate for different needs, e.g. for cases where syllable count is important (e.g. poetry) or where phonetic stress and rhyme are more important (songs). The final word encoding for "hello" with syllables thus looks like this: NN2H400. The encoding above enables the capturing of author style through capturing word choice, word length and sentence structure. It allows the LSTM to learn new types of information e.g. the correct order of parts of speech, or that a specific author uses basic or complex language (language complexity is often determined by word length). This encoding also allows to substitute one author style with another authors vocabulary, e.g. Trump writing Shakespearean sonnets - the style of Shakespearean sonnets is learned by the LSTM, whilst Trump vocabulary is used to fit the style learned.

Conclusion

The new approach:

- Allows to capture style of chosen author.
- Allows to effectively generate text in style of chosen author.
- Successfully captures sentences structure and allows to create close to grammatically correct sentences with proper words.
- LSTM-128 and Bidirectional LSTM-128 performed equally well.
- Needs improvement in generating structured poetry and rhyme.
- Needs improvement in transferring author styles.
- Needs further work focusing on rhythm, rhyme, phonetic stresses.