Text Normalization for user-generated text

Ismini Lourentzou - Kabir Manghnani

March 26, 2018

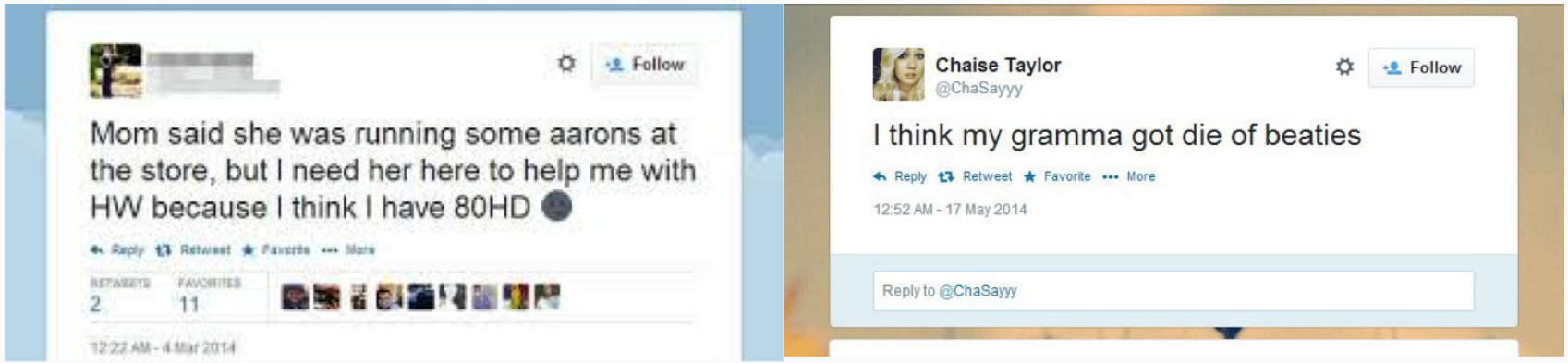
Nowadays...

- Most text is user-generated and online.
  - Vast quantities of online blogs and forums, social media platform posts, customer reviews, and other textual sources.
- User-generated text is the primary input for algorithms that:
  1. Understand user intent and preferences
  2. Predict trends
  3. Recommend items for purchase in targeted advertising
  4. ...

https://www.quora.com/Can-I-make-money-from-a-social-media-management-tool-for-Twitter-despite-a-crowded-market

Difficulties with noisy text

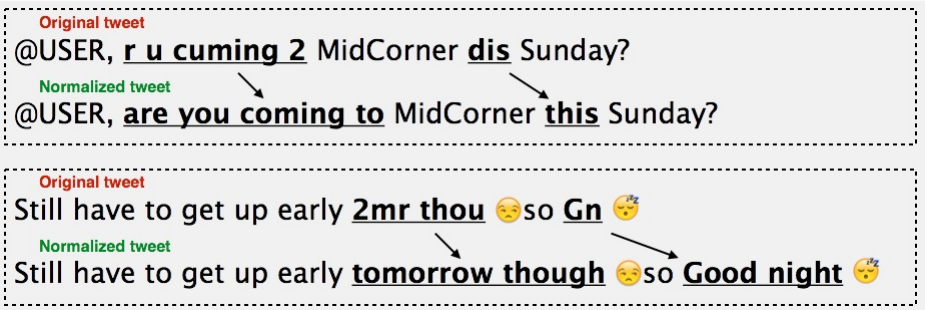
Social media usually deviates from standard language usage with high percentages of non-standard words, such as abbreviations, acronyms, phonetic substitutions, grammatical and spelling errors, etc.



- Problems in understanding the expressed content
  - Communication issues and confusion across multiple groups
  - Different group dialects (e.g. African American vernacular)
- NLP approaches struggle with noisy and informal language
  - Higher out-of-vocabulary (OOV) rates due to non-standard words
  - Learning long tail word representations requires enormous amounts of data

Task definition

Text Normalization identifies noisy parts of the text and substitutes with canonical forms



- correct spellings
  rite $\rightarrow$ **right**
- expand abbreviations
  tmrw $\rightarrow$ **tomorrow**
- phonetic substitutions
  4eva $\rightarrow$ **forever**

In several cases, the task is framed as mapping an OOV non-standard word to an IV standard one that preserves the meaning of the sentence.

Named entities, mentions and hashtages are considered OOV but do not need normalization (there is no appropriate IV word for them).

LexNorm dataset

2015 ACL-IJCNLP Workshop on Noisy User-generated Text (W-NUT) [1]

Dataset	Tweets	Tokens	Noisy	1:1	1:N	N:1	Overall
train	2950	44385	3942	2,875	1,043	10	3,928
test	1967	29421	2776	2,024	704	10	2,738

Table 1: LexNorm statistics

Number of non-standard word types unique to the training and test partitions was 777 and 488, respectively.

Related work: Word-level substitutions [2]

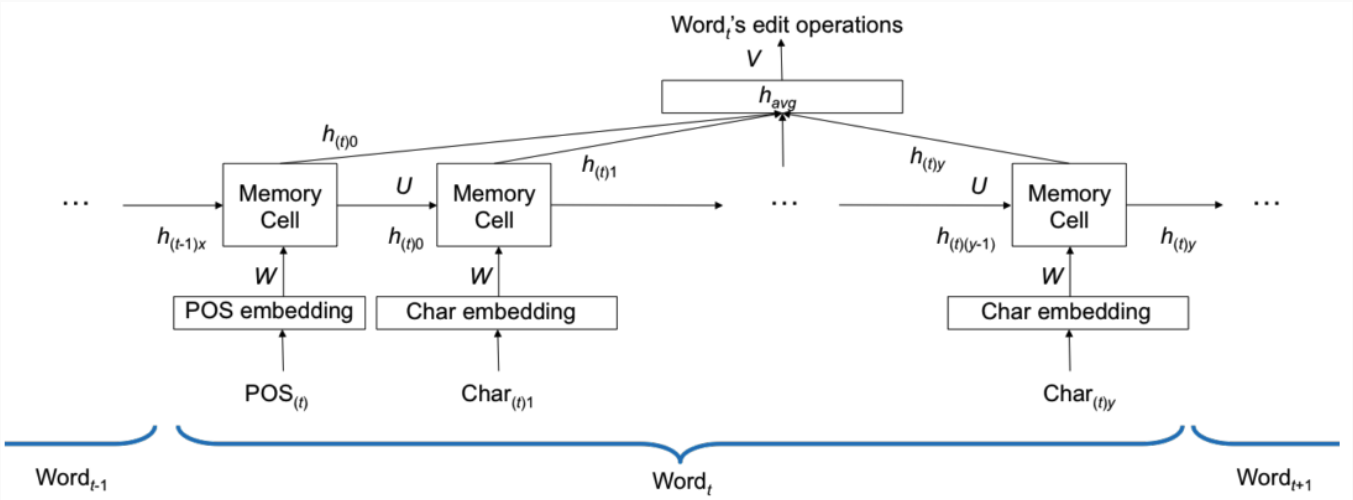
1. Generate set of candidate canonical forms for each token t_i
   - The token itself
   - Top-m most similar canonical forms found in training data
2. Given a tweet T, one of its token t_i and one of the token's candidate c, train a binary classifier that predicts whether c is the correct canonical form of t_i in the tweet T. Select the one with the highest confidence score as the canonical form of t_i.
   - Features include:
     i. String similarity between t_i and c
     ii. Frequency counts and POS of previous and next word
     iii. Percentage of times t_i is normalized to c

String	Similarity Feature Set
"love"	"$lo", "ov", "ve$", "l|v", "o|e"
"looove"	"$lo", "oo", "ov", "ve$", "l|o", "o|o", "o|v", "o|e"
"car"	"$ca", "ar$", "c|r"
"cat"	"$ca", "at$", "c|t"

Top-m most similar: Jaccard similarity on character substrings

Related work: Sequence to Edits LSTM [3]

1. Create a dictionary mapping every word to a list of normalized forms
2. Words with unique mapping are replaced **rite** $\rightarrow$ **right**
3. LSTM handles words with multiple mappings **ur** $\rightarrow$ {**your, you are**}
   - Model trained on **character-level edit operations**
     delete, replace, input character before current index, none
     ur $\rightarrow$ **you are :** *insert_y insert_o*, *insert_ _ insert_a*, *insert_e*



Related work results on LexNorm

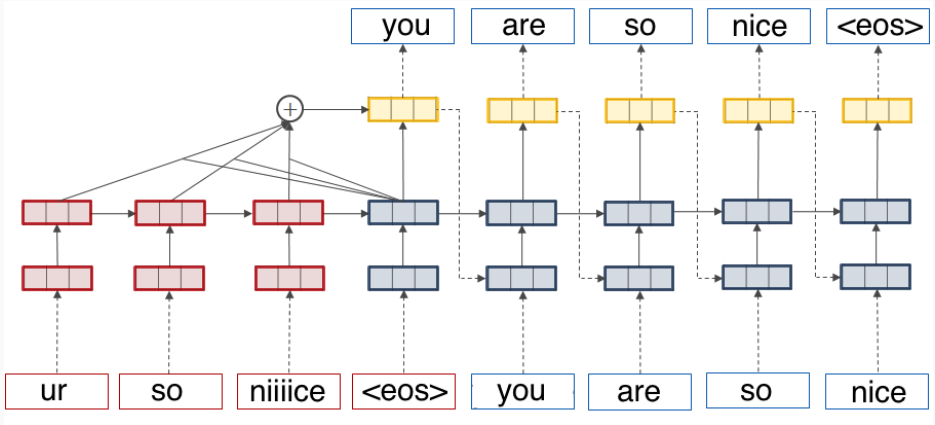
Team name	Precission	Recall	F1	Method highlights
NCSU SAS NING	0.9061	0.7865	0.8421	Random Forest
NCSU SAS WOOKHEE	0.9136	0.7398	0.8175	Lexicon +LSTM
NCSU SAS SAM	0.9012	0.7437	0.8149	ANN
IITP	0.9026	0.7191	0.8005	CRF + Rule
DCU-ADAPT	0.8190	0.5509	0.6587	Generalized Perceptron
LYSGROUP	0.4646	0.6281	0.5341	Spanish Normalization Adaption

Table 2: Results of the constrained systems for WNUT Shared Task

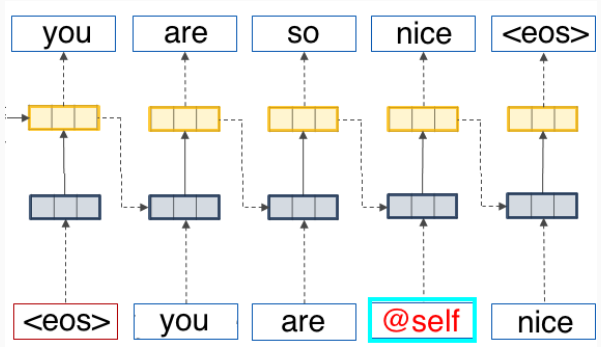
Why Seq2Seq?

- Where did context go? Would adding more context help?
- Can we build end-to-end models despite the smaller data size?
- How to balance trade-off between precision and recall?
- How to force the model to distringuish between words/phrases that need change and the ones that do not?
- How to handle OOV/UNK words?
- What is the best way to blend dictionary-based methods with seq2seq models?

Seq2Seq models for Text Normalization



Figures modified from https://github.com/OpenNMT/OpenNMT-py



special token **@self** to indicate that the input is to be left alone.

- ✓ Scheduled sampling (teacher forcing)
- ✓ Copy Mechanism
- ✓ Actor-Critic model for tuning F1 score directly
- \+ Dual Character+Word Encoder for OOV/UNK words
- \+ Mask decoding to choose only tokens from the input sequence or a specific token set, i.e. dictionary
  binary mask to the models softmax output

Seq2Seq preliminary results on LexNorm

Model	Precission	Recall	F1
Seq2Seq+Attn	0.458	0.671	0.545
Bi-Seq2Seq+Attn	0.425	0.678	0.523
Bi-Seq2Seq+Attn+sharedVocab	0.477	0.710	0.570
Bi-Seq2Seq+Attn+@self	0.298	0.245	0.269

Table 3: Results of some Seq2Seq models for WNUT Shared Task

Conclusion

Remaining tasks

1. Parameter tuning
2. Evaluate actor-critic model and copy mechanism
3. Incorporate the [+] items
4. Perform error analysis
5. Handle "unknowns" that need normalization

The problem of the relatively small data size and the unseen words that need to be normalized remain unsolved, and will be explored later on.

Qihao Shao is building a larger normalization dataset!

References i

T. Baldwin, M.-C. de Marneffe, B. Han, Y.-B. Kim, A. Ritter, and W. Xu.

Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition.

In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135, 2015.

N. Jin.

Ncsu-sas-ning: Candidate generation and feature engineering for supervised lexical normalization.

In *Proceedings of the Workshop on Noisy User-generated Text*, pages 87–92, 2015.

References ii


S. Leeman-Munk, J. Lester, and J. Cox.
Ncsu_sas_sam: deep encoding and reconstruction for normalization of noisy text.
In *Proceedings of the Workshop on Noisy User-generated Text*, pages 154–161, 2015.