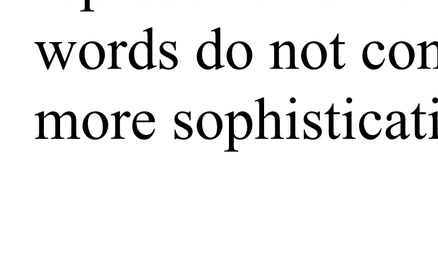
Sentiment Analysis for Twitter

DeepX: Trapit Bansal, Kate Silverstein, Jun Wang

Problem



Who are people supporting?
What are their attitudes towards each topic?
Twitter will tell you!

Topic

- Given some tweets, we generally classify them as "positive"or "negative".

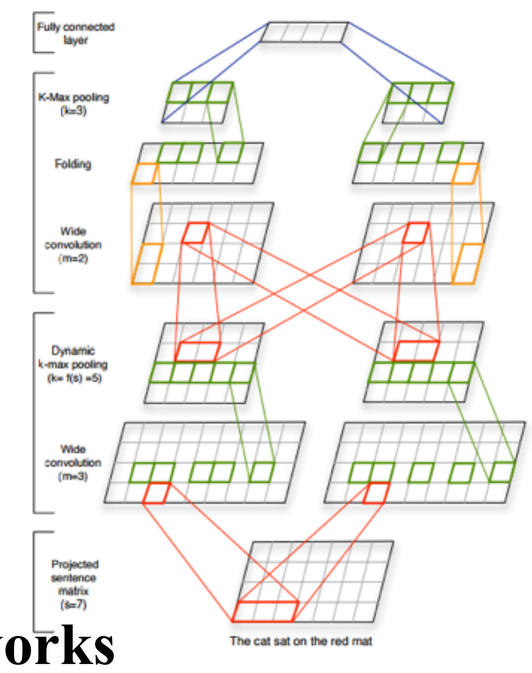
Challenges

- Noisy text, wrong spellings (like coool), emoticons or slangs, etc.
- Huge vocabulary
- Sentiment classification for a tweet needs a representation of the entire tweet, since single words do not convey the full sentiment, so needs more sophistication than word embeddings

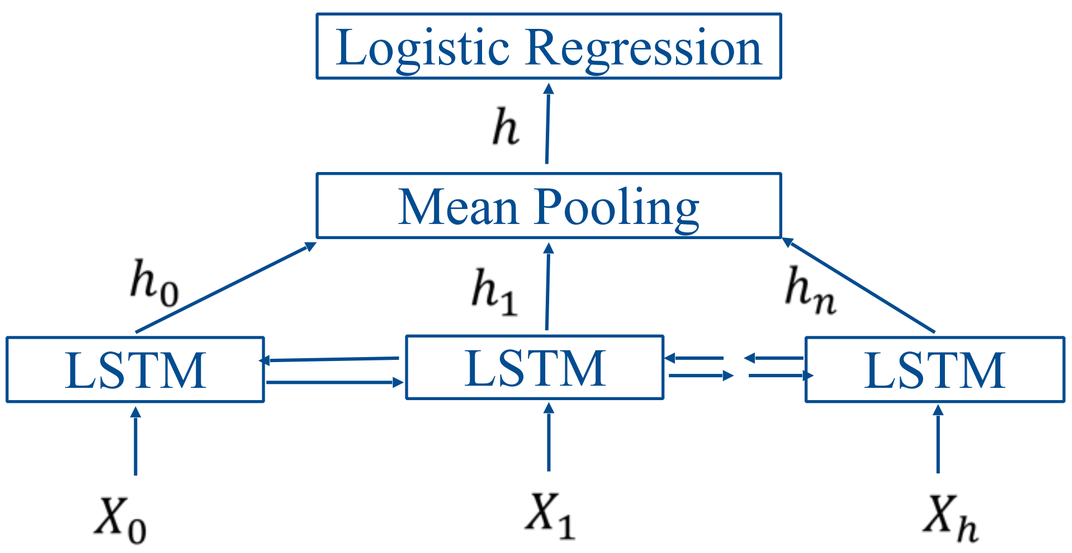
Solution

CNN for Modeling Sentences

- Word vector: learning
- Supervisor depends on tasks
- Dynamic k-Max Pooling
  - $k_l = \max\left(k_{top}, \left\lceil \frac{L-l}{l} - s \right\rceil\right)$
- Folding
  - Sums every two rows
- Multiple Feature Maps
  - Different features



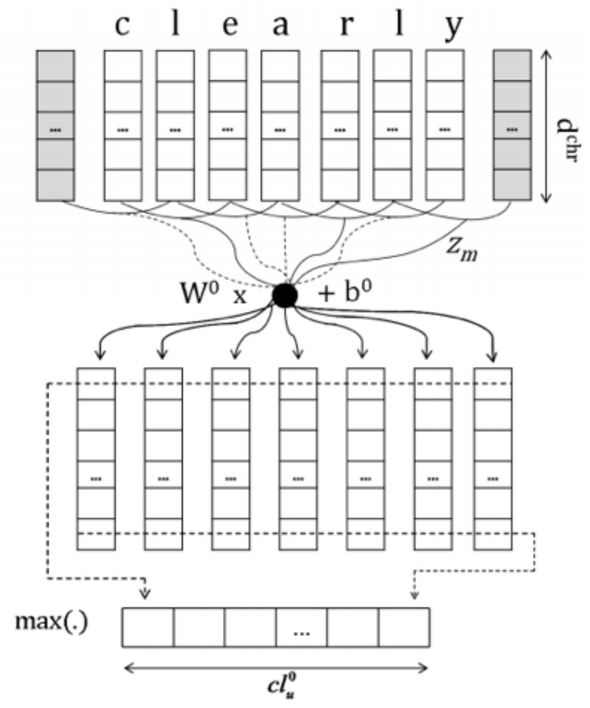
Bi-directional LSTM Networks



Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. **A convolutional neural network for modelling sentences**. ACL, 2014

Character-level Feature Extraction

- Local features around each character of the word
- Combined using a max operation to create a fixed-sized character-level embedding of the word



Results

Training: 1.6M distantly-supervised tweets collected by Go et. al. (2009)
Evaluation: SEMEVAL 2016

	1.6M (test)	*SEMEVAL16*
WORD		
random	NA	NA
GLOVE	83.01	NA
SSWE	80.45	26.88
DCNN	75.0	**81.3**
CHAR		
lstm/ascii	**84.4**	77.73
lstm/utf8	49.03	61.09
char-DCNN	NA	NA

- Results show accuracy (# correct/total)
- Binary classification ('neg' vs. 'pos')

Future

Low-level:

- Compare max vs. mean pooling
- Hyperparameter optimization
- Rerun experiments in multiclass setting

High-level:

- Learn joint character-word embeddings?
- Tweet generation?