Introduction

- **Keyphrase prediction**: distill salient information from massive posts
- **Challenges**:
  - Social media language is noisy and informal (data sparsity)
  - Prior work only extract keyphrases from the source post

Source post with keyphrase “super bowl”:
[S]: Somewhere, a wife that is not paying attention to the game, says “I want the team in yellow pants to win.”

Relevant tweets:
- [T1]: I been a steelers fan way before black & yellow and this super bowl!
- [T2]: I will bet you the team with yellow pants wins.
- [T3]: Wiz Khalifa song ‘black and yellow’ to spur the pittsburgh steelers and Lil Wayne is to sing ‘green and yellow’ for the packers.

Our solution: topic-aware keyphrase generation model

- **Topic-aware**: post-level latent topics learned from corpus can alleviate the data sparsity
- **Sequence generation**: create new keyphrases

Our Approach

- **Overall framework**

  ![Diagram of the overall framework](image)

  - **Sequence Encoder**
  - **BoW Encoder**
  - **BoW Decoder**
  - **GRU**
  - **Topic-aware Decoder**

  - Input: Post in word index $x_{seg}$
  - Keyphrase: $(y_1, y_2, ..., y_M)$
  - Output: Post in bag of words $x_{bow}$

- **Neural topic model (NTM)**

  - **BoW Encoder**
    - Prior latent variables
    - $\mu = f_\theta(f_\alpha(x_{bow}))$
    - $\log \sigma = f_\theta(f_\alpha(x_{bow}))$
  - **BoW Decoder**
    - For each word $w \in x$
    - Draw word $w \sim \text{softmax}(f_\theta(\theta))$

- **Keyphrase generation (KG) model**

  - Base model: standard seq2seq with copy mechanism
  - Advanced: topic-aware sequence decoder

  ![Diagram of the neural topic model](image)

  - Decoder state: $h_t = f_{\text{att}}(w_t; \theta_t; s_{t-1})$
  - Attention: $f_{\text{att}}(w_t) = \text{tanh}(W_w h_t; s_{t-1})$
  - Copy switch: $b_t = \sigma(W_w h_t; s_{t-1}; \theta_t) + b_{aux}$

- **Joint learning topics and keyphrases**

  \[ L_{\text{NTM}} = D_{KL}(p(z) \parallel q(z|x)) - \sum_{n} \log(P(y_n | x_n; \theta_n)), \]
  \[ L_{\text{KG}} = - \sum \log(P(y_n | x_n; \theta_n)), \]
  \[ L = L_{\text{NTM}} + \gamma \cdot L_{\text{KG}} \]

Experiment Results

- **Main results**

  - **Table showing performance metrics**

  - **Figure showing comparison between present vs. absent keyphrase**

  - **Figure showing topic modeling**

  - **Figure showing further discussions**

  - **Figure showing conclusion & future work**

Data Description

- **Source posts**
  - # of posts per post
  - # of KP per post vocab

  - Twitter: 44,113 vs. 19,52
  - Weibo: 46,296 vs. 33,07
  - StackExchange: 49,447 vs. 87,94

- **Target KP**
  - Avg len per KP
  - % of abs. KP vocab

  - Twitter: 4,347 vs. 1,92
  - Weibo: 2,136 vs. 2,55
  - StackExchange: 12,114 vs. 1,41

**Table showing performance metrics**

- **Baseline models**
  - MAJORITY
  - Seq2Seq
  - RecTreat

- **Performance metrics**
  - Accuracy
  - F1 score

Conclusion & Future Work

- **We are the first to propose a topic-aware keyphrase generation model that allows end-to-end training with latent topics**
- **We newly construct three social media datasets for this task**
- **Our solution: topic-aware keyphrase generation model is challenging**
- **Topic modeling models are effective**
- **Latent topics are consistently helpful for indicating keyphrases, especially for absent keyphrases**

Find our code & data