An Encoder-Memory-Decoder Framework for Sub-Event Detection in Social Media

Guandan Chen, Nan Xu, and Wenji Mao
[†]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, Beijing, China
{chenguandan2014,xunan2015,wenji.mao}@ia.ac.cn

ABSTRACT

Sub-event detection can help faster and deeper understanding of an event by providing human-friendly clusters, and thus has become an important research topic in Web mining and knowledge management. In existing sub-event detection methods, clustering based methods are brittle for using heuristic similarity metric to judge whether documents belong to the same sub-event, while topic model based methods are limited to the bag of words assumption. To overcome these drawbacks in previous research, in this paper, we propose an encoder-memory-decoder framework for sub-event detection. Our model learns document and sub-event representations suitable for the similarity metric in a data-driven manner, and transforms sub-event detection into selecting the most proper sub-event representation that can maximize text reconstruction probability. Considering the case of overfitting, we also apply transfer learning in our model. To the best of our knowledge, our model is the first to develop an unsupervised deep neural model for sub-event detection. We use Twitter as an examplar social media platform for our study, and experimental results show that our model outperforms baseline methods for sub-event detection.

CCS CONCEPTS

• **Information systems** → **Clustering**;

KEYWORDS

encoder-memory-decoder framework, sub-event detection, deep neural network

ACM Reference Format:
Guandan Chen, Nan Xu, and Wenji Mao. 2018. An Encoder-Memory-Decoder Framework for Sub-Event Detection in Social Media . In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269256

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269256

1 INTRODUCTION

The vigorous development of social media makes information cascade faster more than ever. Social events such as emergencies and political movements can spread widely, which makes social media a valuable information source. Take Twitter as an example, there are a large number of tweets related to a single event, and due to the complexity of an event, it often consists of several sub-events during its evolution (e.g. death toll increasing, hostage releasing in an attack event and battles in a war), which leads to different aspects of social effects. Thus, sub-event detection has drawn increasing research attention in recent years [1, 6–9], which provides human-friendly clusters for fast and deeper understanding of an event.

Sub-event detection aims to divide tweets about an event into significant and distinct sub-groups. It is a nontrivial task. The labeled data for sub-event detection is scarce because of the difficulty of event data annotation. Moreover, sub-events share a common background event, and sometimes the difference between sub-events is subtle.

Existing sub-event detection methods fall into two categories, namely clustering based methods [1, 6, 7] and topic model based methods [8, 9]. Clustering based methods mainly use shallow features such as tf-idf for tweets, and utilize heuristic similarity metric to judge whether tweets belong to the same sub-event. Topic model based methods assume a text generation process, and get sub-events by optimizing text generation probability, e.g. HDP [8] and MGe-LDA [9].

However, previous clustering based methods are brittle for using heuristic similarity metric to judge whether tweets belong to the same sub-event, while topic model based methods are limited to the bag of words assumption, i.e. the words are generated independently given the topic. To address these weaknesses in the related research, we propose an encoder-memory-decoder framework (*EMD*) for sub-event detection. Our model learns tweet and sub-event representations suitable for the similarity metric in a data-driven manner. We employ recurrent neural network to model the word sequence without bag of words assumption. Our model transforms sub-event detection into selecting the most proper sub-event representation that can maximize text reconstruction probability. Considering the case of over-fitting, we use transfer learning strategy in our model.

Our work makes several contributions.

- We propose a novel encoder-memory-decoder framework for sub-event detection, which converts the sub-event detection process into selecting the most proper

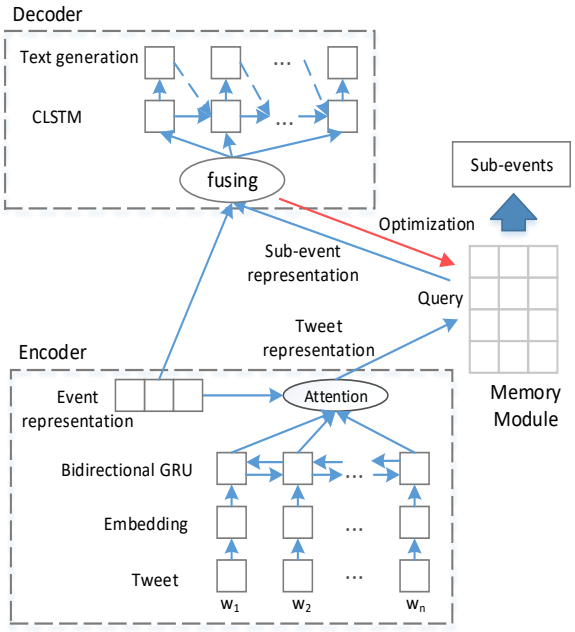

Figure 1: Encoder-Memory-Decoder Framework

sub-event representation to maximize text reconstruction probability.

- To the best of our knowledge, our model is the first to develop an unsupervised deep neural model and adopt transfer learning technique within sub-event detection literatures.
- Experimental results on the publicly available dataset show that our model outperforms the state-of-the-art methods for sub-event detection.

2 PROPOSED MODEL

Given a set of tweets discussing an event, the goal of sub-event detection is to seperate them into several sub-groups and each sub-group consists of tweets discussing one aspect of the event.

As Figure 1 shows, our model consists of three components, namely encoder, memory module and decoder. The encoder maps each tweet into a vector representation, and the memory module holds the vector representations of sub-events. We use the encoded tweet representation as a query, and fetch a sub-event representation from the memory module. Then, the decoder tries to reconstruct the tweet using the sub-event representation and the event representation, where the event representation is a vector shared in the whole event and determined during optimization. Therefore, we convert the sub-event detection process into selecting the most proper sub-event representation for each tweet, which can maximize text reconstruction probability. We optimize sub-event representations, event representation and neural network parameters to maximize the text reconstruction probability for tweets until convergence. Finally, each tweet belongs to the sub-event whose representation can maximize text reconstruction probability. The detail of model structure is as follows.

2.1 Encoder-Memory-Decoder Framework

2.1.1 Encoder. A tweet is a sequence of words $[w_1, w_2, , w_n]$, where n is the length of the word sequence. We map each word w_t into a vector x_t using word embedding model. Then we use bidirectional Gated Recurrent Units (GRU) to encode the sequence further. It produces a sequence of states $[h_1, h_2, \ldots, h_n]$, where h_t is the t-th output of bidirectional GRU, which summarizes information of the t-th word and its context.

As not all words contribute equally for expressing the meaning of whole tweet, we introduce the attention mechanism [10] to get the vector representation of a tweet. The attention mechanism learns to allocate high weights for important elements in a sequence, so that it can emphasize informative parts and suppress less important ones in a tweet. However, it is hard to know which words can distinguish sub-events by only considering the tweet itself. Therefore, we incorporate an event representation in our attention mechanism to consider the background event information, which is shared within the whole event and determined during optimization. The attention mechanism maps a sequence of vectors into one single vector by weighted summing the sequence. Specifically,

$$u_t = tanh(W_h h_t + W_e z_e + b),\ t \in [1, n] \tag{1}$$

$$\alpha_t = \frac{exp(u_s^\top u_t)}{\sum_r exp(u_s^\top u_r)} \tag{2}$$

$$z = \sum_t \alpha_t h_t \tag{3}$$

where z is the vector representation. We first map each word's state h_t and event representation z_e into a hidden space to get the hidden representation u_t. We then compute its attention weight α_t from similarity between u_t and a context vector u_s, which indicates the informative word over whole tweet, and randomly initialized and jointly learned during optimization with other parameters W_h, W_e and b.

2.1.2 Memory module. The memory module consists of a matrix M. Each row M_j in the matrix corresponds to the representation of a sub-event. Given a tweet representation z as a query, we compute a score for each sub-event representation from

$$u_z = tanh(W_z z + b_z) \tag{4}$$

$$\beta_j = \frac{exp((M_j \cdot u_z)/\tau)}{\sum_j exp((M_j \cdot u_z)/\tau)} \tag{5}$$

where the tweet representation z is mapped to a hidden space to get u_z, β_j is the score for the j-th sub-event representation, W_z and b_z are parameters determined during optimization, τ is a hyper-parameter, and M_j is the j-th sub-event representation, which is refined during optimization. During testing, the output of the memory module z_{se} is the sub-event representation with the largest score. Then, z_{se} is the sub-event representation for the tweet. However, this operation is non-differentiable, so we use Monte Carlo sampling during optimization. At each time, we randomly draw a sample from a multinomial distribution with the parameter β, and output corresponding sub-event representation. We use the

logarithm of the text generation probability as reward to optimize encoder parameters, which will be introduced in the following content. Let R be the reward, the loss is

$$l_R = (z_{se} \cdot u_z - R)^2 \tag{6}$$

2.1.3 Decoder. The decoder takes the sub-event and the event representation as inputs, and tries to reconstruct the text sequence. Here, we assume that a good sub-event representation could keep text sequence information, and thus help the reconstruction of the text sequence. The event representation and the sub-event representation are fused using gates like operation, i.e.

$$r_{se} = \sigma(W_{re} z_e + W_{rse} z_{se} + b_r) \tag{7}$$

$$e = r_{se} \odot z_{se} + (1 - r_{se}) \odot z_e \tag{8}$$

where W_{re}, W_{rse}, and b_r are parameters to be optimized, σ is sigmoid function, $\odot$ is element-wise multiplication, and e is a vector used for reconstructing the text sequence. Here, r_{se} plays a role like gates controling the amount of information coming from z_{se} or z_e. We employ a Contextual Long Short Term Memory (CLSTM) [4] to generate the text sequence. Specifically, the outputs are computed from

$$[h_1^o, h_2^o, \ldots, h_n^o] = CLSTM([\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n], e) \tag{9}$$

Here, $\tilde{x}_t$ is the input at t step, which can be the embedding vector of the word generated or the real word at $t-1$ step and it will be introduced in the following section. h_t^o is the t-th output of CLSTM. We then use a softmax function to get the word generation probability distribution

$$p_t = softmax(W_g h_t^o + b_g) \tag{10}$$

where p_t is the t-th word generation distribution, and W_g, b_g are parameters to be optimized.

We use cross entropy as the loss for text reconstruction, denoted as l_{gen}. The reward for optimizing the encoder is $R = -l_{gen}$, i.e. the logarithm of the text generation probability, as is mentioned above. In our model, entropy regularization [5] is used, which encourages the decision boundary lying in low-density regions. It is computed from

$$l_{sub} = -\sum_j \beta_j \log \beta_j \tag{11}$$

2.2 Learning Process

2.2.1 Scheduled Sampling Strategy. As mentioned above, in CLSTM, $\tilde{x}_t$ is is the input at t step, which can be the embedding vector of the word generated or the real word at $t-1$ step. It is hard to be optimized if we directly use greedy searching strategy to determine generated word. To solve this problem, we employ scheduled sampling strategy [2] similar to that in text generation field. That is, we use the embedding of the real previous word w_{t-1} instead of the generated word as the input for CLSTM with a probability p_{ss}, which gradually decays during optimization.

During optimization, we alternatively minimize l_R and $l_{gen} + \lambda l_{sub}$, where λ is a hyper-parameter. We adopt Adam as optimization algorithm. Each tweet belongs to the sub-event whose representation can maximize text reconstruction probability after convergence.

Table 1: Statistics of the Dataset

Event	#sub-event	#tweet	#hashtag
charliehebdo	21	23569	11
ferguson	8	7712	7
ottawashooting	11	16456	20
sydneysiege	14	12876	20

2.2.2 Transfer Learning Strategy. Considering scarcity of sub-event dataset, we adopt transfer learning in our model. Our encoder and decoder aim at text representation extraction and text generation. Thus, these two parts are not limited in the sub-event detection task. It is possible to train them using an external dataset and transfer parameter weights to our sub-event detection model. We adopt text reconstruction task to train the encoder and decoder. We map a tweet into a vector representation, and then try to reconstruct the text sequence using this representation. In text reconstruction task, the main difference to sub-event detection model is that we use z to replace e, i.e. we try to reconstruct the text sequence using the tweet representation.

We optimize text generation loss l_{gen} using an external dataset, then the trained parameters are reused in our sub-event detection model. During sub-event detection, we only need to optimize the attention mechanism parameters, and parameters related to the event as well as sub-event representations.

3 EXPERIMENT

3.1 Dataset

We use a publicly available dataset [11] to evaluate our model. This dataset was originally used for rumor detection. As it includes sub-event information, it is also used for evaluating sub-event detection in the related work [8]. When we recrawled tweets, we lose some tweets due to Twitter API limitation, and 5 events lose many tweets, so that we drop these 5 events. To train the text reconstruction model, we collect tweets using Twitter API from Aug. 9, 2016 to Dec. 10, 2016, with 171 million tweets.

3.2 Baseline Methods

We choose the following methods as baselines. (1) *tf-idf + SOM* [7] uses tf-idf as features, and then clusters tweets using Self Organizing Map. (2) *LDA* (Latent Dirichlet Allocation) [3] takes every tweet as a document. It assumes a document generation process. Then it gets topic distributions by optimizing document generation probability. It finally uses topic distributions to cluster tweets. (3) *HDP* [8] assumes document generation processes as Hierarchical Dirichlet Processes. It also clusters tweets using topic distributions. (4) *MGe-LDA* [9] is an extension of LDA by adding a layer about hashtags. During document generation process, it mutually generates hashtags and topics. (5) *Encoder + k-means* is a variation of our model. It uses the encoder from our text reconstruction model to extract tweet representations. Then, it uses k-means and cosine similarity to cluster tweets, taking the tweet representations as features.

3.3 Experimental Results

We use B-Cubed and Normalized Mutual Information (NMI) as evaluation metrics, which are widely used for evaluating

Table 2: Experimental Results of Different Methods

Event	charliehebdo		ferguson		ottawashooting		sydneysiege	
Metric	B-Cubed	NMI	B-Cubed	NMI	B-Cubed	NMI	B-Cubed	NMI
tf-idf+SOM	0.492	0.728	0.382	0.579	0.539	0.645	0.515	0.678
LDA	0.518	0.764	0.379	0.610	0.465	0.653	0.433	0.648
HDP	0.520	0.729	0.427	0.595	0.442	0.610	0.472	0.703
MGe-LDA	0.540	0.702	0.465	0.629	0.435	0.643	0.403	0.655
Encoder + k-means	0.558	0.774	0.370	0.589	0.453	0.644	0.452	0.702
EMD	0.572	0.824	0.482	0.632	0.552	0.684	0.525	0.722

event detection and sub-event detection when the ground-truth is known. Table 2 shows the experimental results of different methods. It shows that our method outperforms all the baseline methods.

Among the baseline methods, tf-idf+SOM uses tf-idf as features. Surprisingly, this simple method gets better performance than the more complex topic model based methods in *ottawashooting* and *sydneysiege*. It is possibly due to the strong hypothesis of document generation processes in these methods. MGe-LDA performs best in *charliehebdo* and *ferguson*, which adds a hashtag generation layer to LDA. By further analysis of event hashtags, we found that these two events contain many hashtags related to sub-events. The other two events only contain hashtags related to the event itself, thus it is hard to recognize sub-events using these hashtags. So the performance of MGe-LDA relies on high quality hashtags.

In addition, there is no significant best baseline method that outperforms other baselines in all events. It shows that it is hard to get a robust sub-event detection model only considering tweets in the sub-event dataset itself. In contrast, *EMD* employs knowledge transferred from a large external dataset, making our model more robust.

The variation of our model, encoder + k-means only reaches better performance in *charliehebdo* event compared to baselines. It does not beat baselines in other three events, which shows that our model effectively fine tunes the results. Encoder + k-means directly uses the encoder from the text reconstruction model to extract features. In contrast, our model not only fine tunes parameter weights of the attention mechanism, it also gets sub-event representations and the event representation by optimizing text reconstruction probability.

To further verify the effectiveness of our attention mechanism, we provide an example of qualitative analysis using *charliehebdo* event, which is an attack event. In Table 3, we highlight three words with the largest attention weights in tweets. We found that the attention weights are reasonable. For example, in *charliehebdo* event, there is a sub-event about death toll increasing. By allocating more attention weights on the number, we can easily identify the sub-event. Meanwhile, we also find that the encoder can handle phrases like "at large". In the third tweet, "hostage" and "free" are also useful for identifying corresponding sub-event.

4 CONCLUSIONS

In this paper, we propose a novel encoder-memory-decoder framework for sub-event detection in social media, which transforms sub-event detection into selecting the most proper sub-event representation for each tweet that can maximize text reconstruction probability. Different from existing sub-event detection work using heuristic similarity metric or taking bag of words assumption, our model learns tweet and sub-event representations suitable for similarity comparison in a data-driven way. Considering the case of over-fitting, we employ transfer learning in our model. Experimental results show that our model outperforms baseline methods for sub-event detection.

Table 3: Words with High Attention Weights

BREAKING: At least 10 killed in shooting at French satirical newspaper Charlie Hebdo, Paris prosecutor's office says.
Charlie Hebdo shooting latest: 12 dead and gunmen still at large.
Several hostages freed at Jewish supermarket in Paris. Photo Thomas Samson.

ACKNOWLEDGMENTS

This work is supported in part by MSTC Grant 2016QY02D 0305, CAS Grant ZDRW-XH-2017-3, NSFC Grant 71621002, 7170218 and 61671450.

REFERENCES

[1] Dhekar Abhik and Durga Toshniwal. 2013. Sub-event detection during natural hazards using features of social media data. In *WWW*. ACM, 783–788.
[2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*. 1171–1179.
[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR* 3, Jan (2003), 993–1022.
[4] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291* (2016).
[5] Yves Grandvalet and Yoshua Bengio. 2006. Entropy regularization. *Semi-supervised learning* (2006), 151–168.
[6] Polykarpos Meladianos, Christos Xypolopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. 2018. An optimization approach for sub-event detection and summarization in twitter. In *European Conference on Information Retrieval*. Springer, 481–493.
[7] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In *WWW*. ACM, 683–686.
[8] PK Srijith, Mark Hepple, Kalina Bontcheva, and Daniel Preotiuc-Pietro. 2017. Sub-story detection in Twitter with hierarchical Dirichlet processes. *Information Processing & Management* 53, 4 (2017), 989–1003.
[9] Chen Xing, Yuan Wang, Jie Liu, Yalou Huang, and Wei-Ying Ma. 2016. Hashtag-Based Sub-Event Discovery Using Mutually Generative LDA in Twitter. In *AAAI*. 2666–2672.
[10] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*. 1480–1489.
[11] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one* 11, 3 (2016), e0150989.