Expectation-Regulated Neural Model for Event Mention Extraction

Ching-Yun Chang and **Zhiyang Teng** and **Yue Zhang**
Singapore University of Technology and Design
8 Somapah Road, Singapore 487372
{chingyun_chang, yue_zhang}@sutd.edu.sg
zhiyang_teng@mymail.sutd.edu.sg

Abstract

We tackle the task of extracting tweets that mention a specific event from all tweets that contain relevant keywords, for which the main challenges include unbalanced positive and negative cases, and the unavailability of manually labeled training data. Existing methods leverage a few manually given seed events and large unlabeled tweets to train a classifier, by using expectation regularization training with discrete ngram features. We propose a LSTM-based neural model that learns tweet-level features automatically. Compared with discrete ngram features, the neural model can potentially capture non-local dependencies and deep semantic information, which are more effective for disambiguating subtle semantic differences between true event mentions and false cases that use similar wording patterns. Results on both tweets and forum posts show that our neural model is more effective compared with a state-of-the-art discrete baseline.

1 Introduction

A Distributed Denial of Service (DDoS) attack employs multiple compromised systems to interrupt or suspend services of a host connected to the Internet. Victims are often high-profile web servers such as banks or credit card payment gateways, and therefore a single attack may cause considerable loss. The aim of this paper is to build an automatic system which can extract DDoS event mentions from social media, a timely information source for events taking place around the world, so that the mined emerging incidents can serve as early DDoS warnings or signs for Internet service providers.

Ritter et al. (2015) proposed the first work to extract cybersecurity event mentions from raw Twitter stream. They investigated three different event categories, namely *DDoS attacks*, *data breaches* and *account hijacking*, by tracking the keywords *ddos*, *breach* and *hacked*, respectively. Not all tweets containing the keywords describe events. For example, the tweet "give me paypall or i will tell my mum and *ddos* u" shows a metaphor rather than a DDoS event. As a result, the event mention extraction task involves a classification task that filters out true events from all tweets that contain event keywords. Two main challenges exist for this task. First, the numbers of positive and negative examples are typically unbalanced. In our datasets, only about 22% of the tweets that contain the term *ddos* are mentions to DDoS attack events. Second, there is typically little manual annotation available. Ritter et al. (2015) tackled the challenges by weakly supervising a classification model with a small number of human-provided seed events.

In particular, Ritter et al. exploit expectation regularization (ER; Mann and McCallum (2007)) for semi-supervised learning from large amounts of raw tweets that contain the event keyword. They show that the ER approach outperforms semi-supervised expectation-maximization and one-class support vector machine on the task. They build a logistic regression classifier, using few human-labeled seed events and domain knowledge on the ratio between positive and negative examples for ER in training. Results show that the regularization method was effective on classifying unbalanced datasets.

Ritter et al. use manually-defined discrete features. However, the event mention extraction task is highly semantic-driven, and simple textual patterns may suffer limitations in representing subtle semantic differences between true event mentions and false cases with similar word patterns. Recently, deep learning received increasing research attention in the NLP community (Bengio, 2009; Mikolov et al., 2013; Pennington et al., 2014; Kalchbrenner et al., 2014; Vo and Zhang, 2015). One important advantage of deep learning is automatic representation learning, which can effectively encodes syntactic and information about words, phrases and sentences in low-dimensional dense vectors.

In this paper we exploit a deep neural model for event mention extraction, using word embeddings and a novel LSTM-based neural network structure to automatically obtain features for a tweet. Results on two human-annotated datasets show that the proposed LSTM-based representation yields significant improvements over Ritter et al. (2015).

2 Related Work

In terms of scope, our work falls into the area of information extraction from social media (Guo et al., 2013; Li et al., 2015). The proposed event mention extraction system is domain-specific, similar to works that aim at detecting categorized events such as disaster outbreak (Sakaki et al., 2010; Neubig et al., 2011; Li and Cardie, 2013) and cybersecurity events (Ritter et al., 2015). Such work typically trains semi-supervised classifiers to determine events of interest due to the limitation of annotated data. On the other hand, a few studies devote to open domain event extraction (Benson et al., 2011; Ritter et al., 2012; Petrović et al., 2010; Diao et al., 2012; Chierichetti et al., 2014; Li et al., 2014; Qiu and Zhang, 2014), in which an event category is not predefined, and clustering models are applied to automatically induce event types.

In terms of method, the proposed model is in line with recent methods on deep learning for neural feature representations, which have seen success in some NLP tasks (Collobert and Weston, 2008; Collobert et al., 2011; Chen and Manning, 2014). Competitive results have been obtained in sentiment analysis (Kalchbrenner et al., 2014; Kim, 2014; Socher et al., 2013b), semantic relation classification (Hashimoto et al., 2013; Liu et al., 2015), and question answering (Dong et al., 2015; Iyyer et al., 2014). In addition, deep learning models have shown promising results on syntactic parsing (Dyer et al., 2015; Zhou et al., 2015) and machine translation (Cho et al., 2014). Compared to syntactic problems, semantic tasks see relatively larger improvements by using neural architectures, possible because of the capability of neural features in better representing semantic information, which is relatively more difficult to capture by discrete indicator features. We consider event mention extraction as a semantic-heavy task and demonstrate that it can benefit significantly from neural feature representations.

3 Baseline

We take the method of Ritter et al. (2015) as a baseline. Given a tweet containing the keyword *ddos*, the task is to determine whether a DDoS attack event is mentioned in the tweet. A logistic regression classifier is used, which is trained by maximum-likelihood with ER on unlabeled tweets, and automatically generated positive examples from a few seed events.

3.1 Seed Events

Ritter et al. (2015) manually pick seed events, represented as (ENTITY, DATE) tuples, and treated tweets published on DATE referencing ENTITY as positive training instances. For example, (*GitHub*, 2013 July 29)[1] is defined as a seed DDoS event, and the tweet "@amosie *GitHub* is experiencing a large DDos https://t.co/cqEIR6Rz6t" posted on 2013 July 29 is seen as an event mention since it contains the ENTITY *GitHub* as well as matches the DATE 2013 July 29. Those tweets with the word *ddos* but not matching any seed events are grouped as unlabeled data.

3.2 Sparse Feature Representation

Each tweet is represented by a sparse binary vector for feature extraction, where the features consist of bi- to five-grams containing a name entity or the event keyword. For better generalization, all

[1] https://status.github.com/messages/2013-07-29

NE: GitHub	keyword: DDoS
USR NE	JJ DDoS
NE is	DDoS URL
USR NE is	DT JJ DDoS
NE is experiencing	JJ DDoS URL
USR NE is experiencing	experiencing DT JJ DDoS
NE is experiencing DT	DT JJ DDoS URL
USR NE is experiencing DT	is experiencing DT JJ DDoS
NE is experiencing DT JJ	experiencing DT JJ DDoS URL

Table 1: Features of a tweet by Ritter et al. (2015).

words other than common nouns and verbs are replaced with their part-of-speech (POS) tags. Table 1 shows an example of contextual features extracted from the tweet "@amosie GitHub is experiencing a large DDos https://t.co/cqEIR6Rz6t". As can be seen from the table, the features contain shallow wording patterns from a tweet, which are local to a 5-word window. In contrast, the observed average tweet length is 16 words, with the longest tweet containing 48 words, which is difficult to fully represent using only a local window. Our neural model addresses the limitations by learning global tweet-level syntactic and semantic features automatically.

3.3 Logistic Regression Classification with Expectation Regularization

With the feature vector $\vec{f_s} \in \mathbb{R}^d$ defined for a given tweet s, the probability of s being an event mention is defined as:

$$p_\theta(y = 1|s) = \frac{1}{1 + e^{-\vec{\theta}\vec{f_s}}} \tag{1}$$

where $\vec{\theta} \in \mathbb{R}^d$ is a weight vector.

Given a set of event mentions $M = \langle m_1, m_2, ..., m_j \rangle$ and a set of unlabeled instances $U = \langle u_1, u_2, ..., u_k \rangle$, Ritter et al. (2015) train an ER model that maximizes the log-likelihood of positive data while keeping the conditional probabilities on unlabeled data consistent with the human-provided expectations. The objective function is defined as:

$$\begin{aligned} O(\theta; M, U) = & \underbrace{\sum_{m \in M} \log p_\theta(y = 1|m)}_{Log\ Likelihood} \\ & - \underbrace{\lambda^U \Delta(\tilde{p}, \hat{p}_\theta^U)}_{Expectation\ Regularization} \\ & - \underbrace{\lambda^{L^2} \|\theta\|^2}_{L^2\ Regularization} \end{aligned} \tag{2}$$

The expectation regularization term $\Delta(\tilde{p}, \hat{p}_\theta^U)$ is defined as the KL divergence between the model's posterior predictions on unlabeled data, $\hat{p}_\theta^U$, and the human-provided label expectation priors, $\tilde{p}$:

$$\begin{aligned} \Delta(\tilde{p}, \hat{p}_\theta^U) &= D(\tilde{p}||\hat{p}_\theta^U) \\ &= \tilde{p} \log \frac{\tilde{p}}{\hat{p}_\theta^U} + (1 - \tilde{p}) \log \frac{1 - \tilde{p}}{1 - \hat{p}_\theta^U} \end{aligned} \tag{3}$$

4 Distant Seed Event Extraction

We follow Ritter et al. (2015), using a set of seed events and large raw tweets for ER. However, we take a fully-automated approach to find seed events, since manual listing of seed DDoS events can be a costly and time consuming process, and requires a certain level of expert knowledge.

We leverage news articles to collect seed events, representing events as (ENTITY, DATE RANGE) tuples. The ENTITY in our seed events is defined as a name entity that appears in either the *assailant* or *victim* role of an *attack* event labeled by frame-semantic parsing, and the DATE RANGE is a date window around the news publication date. We use a date window rather than a definite news publication date because news articles are not always published on the day a DDoS attack happened. Some examples are given in Figure 1.

We parse DDoS attack news collected from http://www.ddosattacks.net[2] with a state-of-the-art frame-semantic parsing system (SEMAFOR; Das et al. (2010)). Tweets are gathered using the Twitter Streaming API[3] with a case-insensitive track keyword *ddos*. Name entities are extracted from both news articles and tweets using a Twitter-tuned NLP pipeline (Ritter et al., 2011).[4]

Table 2 shows two example DDoS attack news, where the ENTITY values are included in the victim roles, *RBS*, *Ulster Bank*, *GovCERT* and *FBI* in the first news, and *Essex* in the second. It is worth noting that the DDoS attack on RBS, Ulster Bank and Natwest was actually on 2015 July 31. The correlation between tweet mentions and news reports are shown in Figure 1, where each bar indicates the

[2] Most of the articles are about DDoS attack events, while a smaller number discusses the nature of DDoS attacks and related issues.

[3] https://dev.twitter.com/streaming/overview

[4] https://github.com/aritter/twitter_nlp

News Title	DDoS Attacks Take Down RBS, Ulster Bank, and Natwest Online Systems
Date	2015 August 02
Sentences	But as can be seen from the attacks *against* ***RBS****, NatWest, and* ***Ulster Bank****, and the warnings from* ***GovCERT****.ch and the* ***FBI***, these attacks are coming back into vogue again.
News Title	Bored Brazilian skiddie claims DDoS against Essex Police
Date	2015 September 04
Sentences	A teenager from Brazil has claimed responsibility for a distributed denial of service (DDoS) attack *on* ***Essex*** *Police's website*, following a similar attack on another force earlier this week. They added: "Officers investigating the suspected denial of service attack *on the* ***Essex*** *Police website* ... are liaising with other law enforcement agencies to identify any investigative leads"

Table 2: Example news sentences where victim roles are in italic and ENTITY is in bold.

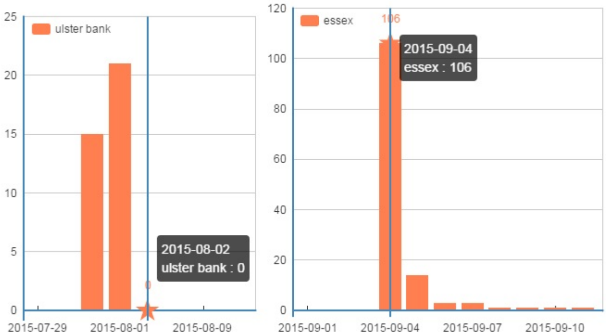


Figure 1: Visualization of the numbers of tweets mentioning Ulster bank (on the left) and Essex (on the right) around the news publication dates.

number of tweets (y-axis) containing a certain ENTITY posted on a certain DATE (x-axis). According to these, we used a 11-day (-3,7) window centered at the news publication date for extracting positive training instances. Experiments show that our method can find seed events with 97% accuracy.

5 Neural Event Mention Extraction

The overall structure of our representation learning model is shown in Figure 2. Given a tweet, two LSTM models (Section 5.1) are used to capture its sequential semantic information in the left-to-right and right-to-left directions, respectively. For deep semantic representation, each LSTM model can include multiple stacked layers. Neural pooling (Section 5.2) is performed on each LSTM layer to extract rich features. Finally, features from the left-to-right and right-to-left components are combined using neural tensors (Section 5.3), and the resulting features are used as inputs to a feed-forward neural network for classification (Section 5.4).

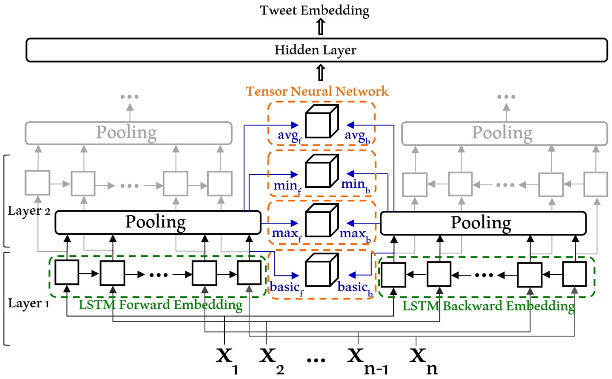


Figure 2: Architecture of the proposed neural tweet representation model.

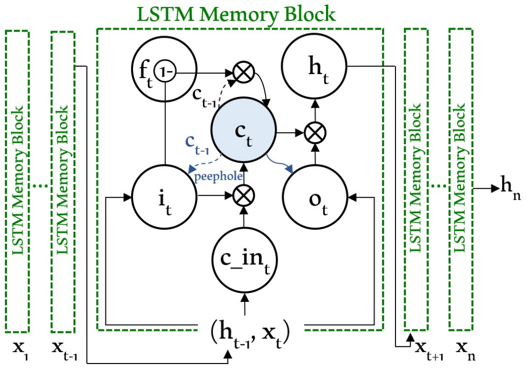


Figure 3: LSTM-based text embedding for word vectors $x_1, x_2, \ldots, x_n$.

5.1 LSTM Models

The main goal of our neural model is to find dense vector representations for tweets, which are effective features for event mention extraction. Starting from word embeddings (Mikolov et al., 2013; Pennington et al., 2014), a natural way of modeling a tweet is to treat it as a sequence and use a recurrent neural network (RNN) structure (Pearlmutter, 1989). LSTM (Hochreiter and Schmidhuber, 1997) is a variant of RNNs, which is better at exploiting long range context thanks to purpose-built units called *memory blocks* to store history information.

LSTM has shown improvements over conventional RNN in many NLP tasks (Jozefowicz et al., 2015; Graves et al., 2013b; Cho et al., 2014).

A typical LSTM memory block consists of three gates (i.e. *input*, *forget* and *output*), which control the flow of information, and a *memory cell* to store the temporal state of the network (Gers et al., 2000). While traditionally the values of gates are decided by the input and hidden states in a RNN, we take a variation with *peephole connections* (Gers and Schmidhuber, 2000), which allows gates in the same memory block to learn from the current cell state. In addition, to simplify model complexity, we use coupled *forget* and *input* gates (Cho et al., 2014).

Figure 3 illustrates the memory block used for our tweet representation. The network unit activations for input x_t at time step t are defined by the following set of equations:

Gates at step t:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (4)$$

$$f_t = 1 - i_t \quad (5)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \quad (6)$$

Cell:

$$\begin{aligned} c_t &= f_t \otimes c_{t-1} \\ &\quad + \ i_t \otimes tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_{c\_in}) \end{aligned} \quad (7)$$

Hidden State:

$$h_t = o_t \otimes tanh(c_t) \quad (8)$$

The W terms in Equations 4–7 are the weight matrices (W_{ic} and W_{oc} are diagonal weight matrices for peephole connections); the b terms denote bias vectors; σ is the logistic sigmoid function; and $\otimes$ computes element-wise multiplication of two vectors. i_t, f_t and o_t are *input*, *forget* and *output* gates, respectively; c_t stores the cell state, and h_t is the output of the current memory block.

Inputs For the inputs $x_1, x_2, ..., x_n$, we learn 50-dimension word representations using the skip-gram algorithm (Mikolov et al., 2013). The training corpus was collected from the tweet archive site, and a total of 604,926,764 tweets were used. Each tweet was tokenized using a tweet-adapted tokenizer (Owoputi et al., 2013), and stopwords and punctuations are removed. The trained model contains 5,251,332 words.

Layers Recent research has shown that both RNNs and LSTMs can benefit from depth in space (Graves et al., 2013a; Graves et al., 2013b; Sak et al., 2014; Sak et al., 2015). A deep LSTM is built by stacking multiple LSTM layers, with the output sequence of one layer forming the input sequence for the next, as shown in Figure 2. At each time step the input goes through multiple non-linear layers, which progressively build up higher level representations from the current level. In our tweet representation model, we embody a deep LSTM architecture with up to 3 layers.

5.2 Pooling

Given a LSTM and an input sequence $x_1, x_2, ..., x_n$, using the last state h_n as features is a *basic* representation strategy for the sequence. Apart from this approach, another common feature extraction strategy is to apply pooling (Boureau et al., 2011) over all the states $h_1, h_2, ..., h_n$ to capture the most characteristic information. Pooling extracts fixed dimensional features from $h_1, h_2, ..., h_n$, which has variable length. In our model we consider different pool strategies, including *max*, *average* and *min* poolings. For convenience of writing, we refer to the *basic* feature strategy also as *basic* pooling in later sections. When there are multiple LSTM layers, the features consist of the pooling results from each layer, concatenated to give a single vector.

5.3 Neural Tensor Network for Feature Combination

Given the pooling methods, we extract features r_f and r_b for the forward and backward multi-layer LSTMs, respectively. Inspired by Socher et al. (2013a), we use a neural tensor network (NTN) to combine the bi-directional r_f and $r_b \in \mathbb{R}^d$. The network can be formalized as follows:

$$V = tanh(r_f^T \, T^{[1:q]} \, r_b + W_{ntn} \begin{bmatrix} r_f \\ r_b \end{bmatrix} + b_{ntn}) \quad (9)$$

where $T^{[1:q]} \in \mathbb{R}^{d \times d \times q}$ is a tensor, $W_{ntn} \in \mathbb{R}^{q \times 2d}$ and $b_{ntn} \in \mathbb{R}^q$ are the weight matrix and bias vector, respectively, as that in the standard form of a neural network. The bilinear tensor product $r_f^T \, T^{[1:q]} \, r_b$ is a vector $v \in \mathbb{R}^q$, where each entry is computed by one slice of the tensor:

$$v_i = r_f^T \, T^{[i]} \, r_b \qquad (i = 1, 2, \ldots, q) \quad (10)$$

1. **NSA** site went down due to ‘internal error’, not DDoS attack, agency claims http://t.co/B7AzoLPsKf < isn’t that the same thing
2. **NSA** denies DDOS attack took place on website, claims internal error http://t.co/WW7uFM4Xk5
3. @HostingSocial True Shikha,**Enterprises** are at a greater risk with increased DDoS attacks & #cloud solns need to take measures for prevention

Table 3: The three false positives in the 100 automatically extracted mentions, where EVENT ENTITIES are in bold.

The NTN combined features are concatenated, and fed into a *tanh* hidden layer. The output of the layer, $\vec{f_s}$, becomes the final representation of a tweet, and is used to compute the probability of the tweet being an event mention, as shown in Equation 1.

5.4 Classification

The final classifier of the neural network model is Equation 1, consistent with the baseline model. As a result, ER is applied in the same way as Equation 2. The main difference between our model and the baseline is in the definition of $\vec{f_s}$, the former being a deep neural network and the latter being manual features. Consequently, Equation 1 can be regarded as a softmax layer in our deep neural model, for which all the features and parameters are trained automatically and consistently.

For training, the parameters are initialized uniformly within the interval $[-a,a]$, where

$$a = \sqrt{\frac{6}{H_k + H_{k+1}}} \qquad (11)$$

H_k and H_{k+1} are the numbers of rows and columns of the parameter, respectively (Glorot and Bengio, 2010). The parameters are learned using stochastic gradient descent with momentum (Rumelhart et al., 1988). The model is trained by 500 iterations, in each of which unlabeled instances are randomly sampled so that the same numbers of positives and unlabeled data are used.

6 Experiments and Results

6.1 Data

We streamed tweet with the track kayword *ddos* for five months from April 13 to September 13, 2015. In addition, we extracted tweets containing the word *ddos* from a tweet archive[5] in the period from September 2011 to September 2014. Using the distant seed event extraction scheme described in Section 4, a total number of 930 mentions covering 45 ENTITY were automatically derived. In order to examine whether the automatically-collected instances are true positives and hence form a useful training set, an author of this paper annotated 100 extracted mentions finding that that 3 are false positives, as listed in Table 3. The result suggests that the automatically extracted mentions are reliable.

	Training	Dev	Test	Dark Web Test
Positive	930	43	160	82
Negative	–	157	640	318
Unlabeled	127,774	–	–	–

Table 4: Statistics of the datasets.

The remaining tweets were randomly split into a 200-instance development set, a 800-instance test set, and an unlabeled training set.[6] Both the development and test sets were annotated by a human judge and an author of this paper. The inter-annotator agreement on the binary labeled 1000 instances was measured by using Fleiss’ kappa (Fleiss et al., 2013), and the score, which is 0.85 for the data, represents *almost perfect agreement* according to Landis and Koch (1977). There were 47 out of the 1,000 tweets that received different labels, for which another human judge made the final decision.

To test the applicability of the proposed mention extraction system on other domains, we collected 400 sentences containing the keyword *ddos* from Dark Web Monitor (TNO, 2015), a platform gathering dark web data. Again each sentence was annotated by two human judges, and the third person made the final decision on conflicting cases. The inter-annotator agreement kappa score on this dataset is 0.85, consistent with the tweet annotation. Table 4 presents the statistics of the datasets.

6.2 Evaluation

We follow Ritter et al. (2015) and evaluate the performance by the area under the precision-recall curve (AUC), where *precision* is the fraction of re-

[5] https://archive.org/details/twitterstream

[6] http://people.sutd.edu.sg/ yue_zhang/pub/naacl16.cyc.zip

	basic	*max*	*avg*	*min*
1-LSTM-layer+concat	0.41	0.39	0.38	0.39
1-LSTM-layer+NTN	0.43	0.43	0.44	0.42
2-LSTM-layer+NTN	0.44	0.42	0.44	0.44
3-LSTM-layer+NTN	0.45	**0.47**	0.46	0.46

Table 5: AUCs of different model architectures.

Pooling	AUC
max+avg	0.48
max+min	0.50
max+basic	**0.51**
max+basic+min	0.50
max+basic+min+avg	0.47

Table 6: AUCs of different pooling methods.

trieved instances that are event mentions, and *recall* is the fraction of gold event mention instances that are retrieved. Precision-recall (PR) curves offer informative pictures on the classification of unbalanced classes (Davis and Goadrich, 2006).

6.3 Development Experiments

For the proposed model, we empirically set the LSTM output vector h_t, the NTN output V, and the size of the hidden layer to 32.[7] For the ER model, the human-provided label expectation prior $\tilde{p}$ is set to 0.22 since the percentage of positives in the development set is 22%, and the parameter λ^U is set to one-tenth of the positive training data.[8]

6.3.1 Feature Combination

We first test whether using a NTN to combine the bi-directional representations can give a better performance compared to simply concatenating the two representation vectors. Table 5 gives AUCs of one-layer *basic*, *max*, *avg* and *min* pooling strategies tested on the tweet development set. We can see that all the four different pooling strategies perform better when the NTN combination is used. As a result, for the following experiments we only consider using NTNs to combine bi-directional representations.

Next we observe the effect of using different numbers of LSTM layers in our model. AUCs of *basic*, *max*, *avg* and *min* pooling strategies with respect to 1, 2 and 3 LSTM layers are presented in Table 5. In most of the cases, the performance of the model increases when the LSTM architecture goes deeper, and we build our final models using 3 LSTM layers.

6.3.2 Pooling Strategies

In the previous experiments, *max* pooling achieves the highest AUC with the architecture 3-LSTM-layer+NTN, we are interested in whether combining *max* with other pooling strategies would further increase the performance. Table 6 summarizes the AUC of various combinations, according to which we choose *max+basic* for final tests.

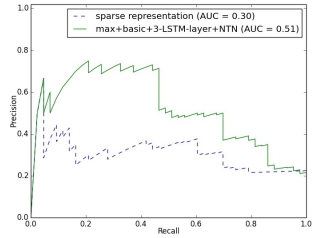


Figure 4: Development PR curves.

Finally, we test the performance of sparse feature representations as used in the model of Ritter et al. (2015). Figure 4 shows the PR curves of the sparse representation and the best setting *max+basic* evaluated on the development set. The AUC of using sparse representation is 0.30 while that of the *max+basic* model is 0.51. The runtime performances of training with sparse feature representations and neural feature representations are 276.17 and 1137.87 seconds, respectively, running on a single thread of an Intel Core i7-4790 3.60GHz CPU.

6.4 Final Test

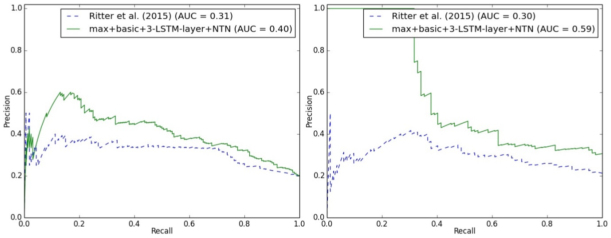
Figure 5 presents the PR curves of the baseline sparse feature representation and the final neural model evaluated on the datasets, and Table 7 gives the AUC for these test-set evaluations. From the curves we can see that the sparse representation is comparatively less efficient in picking out negative examples, since at a lower recall the model does not gain a higher precision. In contrast, LSTM-based representation demonstrates a better trade-off between recall and precision. We do not have a strong intuition on why the performance on dark web test set is better than that on tweet test set for the proposed model.

[7] The hidden layer size is chosen by comparing development test scores using the sizes of 16, 32 and 64.

[8] Mann and McCallum (2007) found that λ^U does not require tuning for different data set.

Discrete Baseline Model (Ritter et al., 2015)	LSTM-based Model
Top 5:	Top 5:
N|0.9|0.0 They dealt with the ddos attacks with grace and confidence.	P|0.7|1.0 Until we have proof I don't think we can say who is responsiblemaybe it wasnt tor market who did the ddos, but check this out:http://silkroad5v7dywlc.onion/index.php?topic=8598.0maybe they did initate the ddos in the hopes of proving that their site is superior because they "fended off" a ddos attack faster than SRTM is super sketchy!
P|0.9|1.0 Thank you.And now, this is my hypothesis, only is a personal thinking, my thought of what happening (or something similar, at least): I think that Agora is under DDOS attacks constantly, maybe for another markets (probably Nucleus if I had to bet for one: right now they have the monopoly, practically, it's one of the three and more knowns and used DM's now (Agora, Nucleus, and Middle-Earth, at least this is my thought) all the vendors of Agora are going to Nucleus too and all publishing their listings there.	P|0.7|1.0 what's the status?you find it in the first post i set it to GREEN/ORANGE as the site is still under DDOS attack but temporarily accessable.greets
N|0.9|0.8 But it was basically explaining how the DDOS attacks on SR earlier in the year were the NSA triangulating its position by measuring PING return times and likes.	P|0.7|1.0 It seems their idea of a "hack" is a DDoS attack on the server (which does indeed go on right now, and as all DoS attacks, can result in denial of service) and a brute-force attack on the login system to try to find out users' passwords.
N|0.9|0.1 unforgiven I remember from sr2, many of the sr2 fanboys were all for DDOS attacks on Agora and tormarket if people remember.	P|0.6|1.0 One of the other markets (Nucleus) is paying some blackhat to DDOS most of the other markets, it's all over Reddit.Support here is asleep, I don't know how you can run a market with a daily uptime of 25%.I agree with OP.
N|0.8|0.2 you know things be stressful for admins and dev team right now :/keep your heads up guys, the work you do is the front line of our revolution for personal freedoms being regained.everyone here is a freedom fighter, you guys are our captainsthank you ALL for this wonderful community and sense of freedom you have brought us!so get this DDoS attack under control and keep on truckin!!!	P|0.7|1.0 He also said he was involved in helping DPR hack into Tormarket's database and launch the DDoS against the Russian cyberattackers.
Bottom 5:	Bottom 5:
N|0.5|0.0 I only words I could understand were "DDoS" and "Bastard".	N|0.5|0.0 I only words I could understand were "DDoS" and "Bastard".
N|0.5|0.9 In general, it seems like they have set the site up to accommodate all parties: escrow, vendor ratings, buyer ratings, quick wallet transactions, etc.Guess we'll see how they deal with the growing pains, DDOS, & hack attempts that will certainly come their way in the near future.	N|0.9|0.0 They dealt with the ddos attacks with grace and confidence.
N|0.5|0.0 Please ddos him.	N|0.5|0.0 DDOS IS PURE BULLSHIT.
N|0.5|0.0 Next fucking day, ddos dildos and damage....LEGs wares hit my drop while the market was still floundering like guppies on hot concrete, yeah, that's why.	N|0.5|0.0 can you guys ddos this guy?
N|0.5|0.2 child pornography, spamm, DDOS etc.	N|0.5|0.0 The DDoS has nothing to do with this problem.

Table 8: Top 5 and bottom 5 ranked dark web sentences as determined by the baseline and the proposed LSTM-based model. Format: class label|baseline score|neural score.



(a) Tweet test set. (b) Dark web test set.

Figure 5: Final PR curves.

	Tweet	Dark Web
Ritter et al. (2015)	0.31	0.30
max+*basic*+3-LSTM-layer+NTN	0.40	0.59

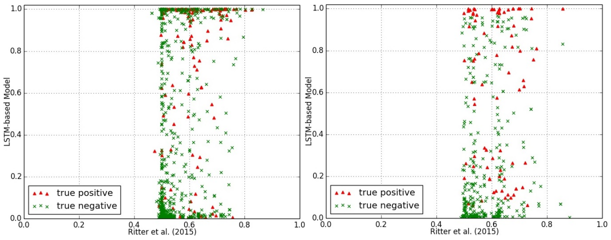
Table 7: Final AUCs.

6.5 Analysis

Table 8 shows the top 5 and bottom 5 ranked dark web sentences[9] as scored by the baseline and the proposed LSTM-based model, respectively. For each sentence, the human judgment (*P* for event mentions and *N* for non-event mentions) is given, followed by the probability values output by the baseline and the proposed system.

Only one of the top five most probable event-mentioning sentences as decided by the baseline is true positive. On the other hand, all of the top five sentences indicated by the proposed model are true positives. We investigate the contextual features that contribute to the false positive case "They dealt with the ddos attacks with grace and confidence." determined by the baseline, and find that the patterns "DT ddos", "ddos attack|NN", "DT ddos attack|NN IN" and "IN DT ddos" are ranked 2nd, 18th, 111th, 127th among the 15,355 contextual patterns, respectively, which have relatively high weights but only carry

[9] The sentence boundary was detected by NLTK PunktSentenceTokenizer.



(a) Tweet test set.

(b) Dark web test set.

Figure 6: Probability distributions on the test sets.

limited information. In contrast, the LSTM-based model can capture global syntactic and semantic features other than words surrounding *ddos* to distinguish mentions from non-mentions. From the table we can see that those high-confidence sentences determined by the LSTM-based model are more informative compared with those lower ranked sentences.

Figure 6 presents the probability distributions of positive and negative test cases as obtained by the baseline (x-axis) and the LSTM-based model (y-axis), respectively. It can be seen from the figures that the probabilities determined by the LSTM-based model are scattered between 0.0 and 1.0, while those by the baseline are gathered between 0.5 and 0.9, which shows that the proposed neural model can achieve better confidence on classifying event mentions. This demonstrates its stronger differentiating power as compared with discrete indicator features, as hypothesized in the introduction. In addition, for the proposed model a large portion of true positives (▲) are close to the top in both test sets, while more negatives (×) gather at the bottom of the dark web test set plot, compared to that in the tweet test set. As for the baseline model, many negatives locate around the horizontal centre, with a probability of 0.5, in the tweet test set, which explains why the baseline is relatively less effective on the precision-recall trade-off.

7 Conclusion

We investigated LSTM-based text representation for event mention extraction, finding that automatic features from the deep neural network largely improve the sparse representation method on the task. The model performance can further benefit by exploiting deep LSTM structures and tensor combination of bidirectional features. Results on tweets and dark web forum posts show the effectiveness of the method.

Acknowledgments

We would like to thank Mr. Geoffrey Williams for data annotation, Mr. Lin Li for data processing, anonymous reviewers for their informative comments, and TNO for sharing dark web data. This research was funded by MINDEF Singapore, through the project CISDeM. Yue Zhang is the corresponding author.

References

Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proceedings of the Annual Meeting of the ACL*, pages 389–398, Portland, Oregon.

Y-Lan Boureau, Nicolas Le Roux, Francis Bach, Jean Ponce, and Yann LeCun. 2011. Ask the locals: multi-way local pooling for image recognition. In *ICCV, IEEE International Conference on*, pages 2651–2658.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on EMNLP*, pages 740–750.

Flavio Chierichetti, Jon Kleinberg, Ravi Kumar, Mohammad Mahdian, and Sandeep Pandey. 2014. Event detection via communication pattern analysis. In *International AAAI Conference on Weblogs and Social Media*.

Kyunghyun Cho, Bart Van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on EMNLP*, pages 1724–1734, Doha, Qatar.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th ICML*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: The Annual Conference of NAACL*, pages 948–956, Los Angeles, California.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd ICML*, pages 233–240, Pittsburgh, Pennsylvania.

Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. 2012. Finding bursty topics from microblogs. In *Proceedings of the Annual Meeting of the ACL*, pages 536–544, Jeju Island, South Korea.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of the Annual Meeting of the ACL and the 7th International Joint Conference on NLP*, pages 260–269.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the Annual Meeting of the ACL and the 7th International Joint Conference on NLP*, pages 334–343.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 2013. *Statistical methods for rates and proportions*. Wiley-Interscience, 3 edition.

Felix Gers and Jürgen Schmidhuber. 2000. Recurrent nets that time and count. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, pages 189–194.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.

Alan Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013a. Hybrid speech recognition with deep bidirectional LSTM. In *ASRU, 2013 IEEE Workshop on*, pages 273–278.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013b. Speech recognition with deep recurrent neural networks. In *ICASSP, 2013 IEEE International Conference on*, pages 6645–6649.

Weiwei Guo, Hao Li, Heng Ji, and Mona T Diab. 2013. Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the Annual Meeting of the ACL*, pages 239–249.

Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Simple customization of recursive neural networks for semantic relation classification. In *EMNLP*, pages 1372–1376.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the Conference on EMNLP*, pages 633–644.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd ICML*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on EMNLP*, pages 1746–1751, Doha, Qatar.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Jiwei Li and Claire Cardie. 2013. Early stage influenza detection from Twitter. *arXiv preprint arXiv:1309.7340*.

Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major life event extraction from twitter based on congratulations/condolences speech acts. In *Proceedings of the Conference on EMNLP*, pages 1997–2007, Doha, Qatar.

Hao Li, Heng Ji, and Lin Zhao. 2015. Social event extraction: Task, challenges and techniques. In *Proceedings of the IEEE/ACM International Conference on ASONAM*, pages 526–532.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *Proceedings of the Annual Meeting of the ACL and the 7th International Joint Conference on NLP*, pages 285–290, Beijing, China.

Gideon S Mann and Andrew McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th ICML*, pages 593–600.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Graham Neubig, Masato Hagiwara, and Koji Murakami. 2011. Safety information mining — what can NLP do in a disaster —. In *Proceedings of the 5th International Joint Conference on NLP*, pages 965–973.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013.

Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the Conference of NAACL*.

Barak A Pearlmutter. 1989. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on EMNLP*, pages 1532–1543.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The Annual Conference of NAACL*.

Likun Qiu and Yue Zhang. 2014. ZORE: A syntax-based system for Chinese open relation extraction. In *Proceedings of the Conference on EMNLP*, pages 1870–1880, Doha, Qatar.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on EMNLP*, pages 1524–1534.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of ACM SIGKDD*, pages 1104–1112.

Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. 2015. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, pages 896–905.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5:3.

Hasim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proceedings of the Annual Conference of INTERSPEECH*.

Hasim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk. 2015. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *ICASSP, 2015 IEEE International Conference on*, pages 4280–4284.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the international conference on WWW*, pages 851–860. ACM.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013a. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on EMNLP*.

TNO. 2015. *Dark Web Monitor; Innovation platform of the Dark Web Solutions Programme.* https://www.tno.nl.

Duy-Tin Vo and Yue Zhang. 2015. Deep learning for event-driven stock prediction. In *Proceedings of IJCAI*, BueNos Aires, Argentina, August.

Hao Zhou, Yue Zhang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the Annual Meeting of the ACL*, pages 1213–1222.