Event Representation Learning Enhanced with External Commonsense Knowledge

Abstract

Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of commonsense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external commonsense knowledge about the intent and sentiment of the event. Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market.

Introduction

Events are a kind of important objective information of the world. Structuralizing and representing such information as machine-readable knowledge are crucial to artificial intelligence BIBREF0, BIBREF1. The main idea is to learn distributed representations for structured events (i.e. event embeddings) from text, and use them as the basis to induce textual features for downstream applications, such as script event prediction and stock market prediction.

Parameterized additive models are among the most widely used for learning distributed event representations in prior work BIBREF2, BIBREF3, which passes the concatenation or addition of event arguments' word embeddings to a parameterized function. The function maps the summed vectors into

an event embedding space. Furthermore, BIBREF4 ding2015deep and BIBREF5 weber2018event propose using neural tensor networks to perform semantic composition of event arguments, which can better capture the interactions between event arguments.

This line of work only captures shallow event semantics, which is not capable of distinguishing events with subtle differences. On the one hand, the obtained event embeddings cannot capture the relationship between events that are syntactically or semantically similar, if they do not share similar word vectors. For example, as shown in Figure FIGREF2 (a), “PersonX threw bomb” and “PersonZ attacked embassy”. On the other hand, two events with similar word embeddings may have similar embeddings despite that they are quite unrelated, for example, as shown in Figure FIGREF2 (b), “PersonX broke record” and “PersonY broke vase”. Note that in this paper, similar events generally refer to events with strong semantic relationships rather than just the same events.

One important reason for the problem is the lack of the external commonsense knowledge about the mental state of event participants when learning the objective event representations. In Figure FIGREF2 (a), two event participants “PersonY” and “PersonZ” may carry out a terrorist attack, and hence, they have the same intent: “to bloodshed”, which can help representation learning model maps two events into the neighbor vector space. In Figure FIGREF2 (b), a change to a single argument leads to a large semantic shift in the event representations, as the change of an argument can result in different emotions of event participants. Who “broke the record” is likely to be happy, while, who “broke a vase” may be sad. Hence, intent and sentiment can be used to learn more fine-grained semantic features for event embeddings.

Such commonsense knowledge is not explicitly expressed but can be found in a knowledge base such as Event2Mind BIBREF6 and ATOMIC BIBREF7. Thus, we aim to incorporate the external commonsense knowledge, i.e., intent and sentiment, into the learning process to generate better event representations.

Specifically, we propose a simple and effective model to jointly embed events, intents and emotions into the same vector space. A neural tensor network is used to learn baseline event embeddings, and we define a corresponding loss function to incorporate intent and sentiment information.

Extensive experiments show that incorporating external commonsense knowledge brings promising improvements to event embeddings, achieving 78% and 200% improvements on hard similarity small and big dataset, respectively. With better embeddings, we can achieve superior performances on script event prediction and stock market prediction compared to state-of-the-art baseline methods.

Commonsense Knowledge Enhanced Event Representations

The joint embedding framework is shown in Figure FIGREF3. We begin by introducing the baseline event embedding learning model, which serves as the basis of the proposed framework. Then, we show how to model intent and sentiment information. Subsequently, we describe the proposed joint model by integrating intent and sentiment into the original objective function to help learn high-quality event representations, and introduce the training details.

Commonsense Knowledge Enhanced Event Representations ::: Low-Rank Tensors for Event Embedding

The goal of event embedding is to learn low-dimension dense vector representations for event tuples $E=(A, P, O)$, where P is the action or predicate, A is the actor or subject and O is the object on which the action is performed. Event embedding models compound vector representations over its predicate and arguments representations. The challenge is that the composition models should be effective for learning the interactions between the predicate and the argument. Simple additive transformations are incompetent.

We follow BIBREF4 (BIBREF4) modelling such informative interactions through tensor composition. The architecture of neural tensor network (NTN) for learning event embeddings is shown in Figure FIGREF5, where the bilinear tensors are used to explicitly model the relationship between the actor and the action, and that between the object and the action.

The inputs of NTN are the word embeddings of A, P and O, and the outputs are event embeddings. We initialized our word representations using publicly available d-dimensional ($d=100$) GloVe vectors BIBREF8. As most event arguments consist of several words, we represent the actor, action and object as the average of their word embeddings, respectively.

From Figure FIGREF5, $S_1 \in \mathbb {R}^d$ is computed by:

where $T^{[1:k]}_1 \in \mathbb {R}^{d\times d\times k}$ is a tensor, which is a set of k matrices, each with $d\times d$ dimensions. The bilinear tensor product $A^TT_1^{[1:k]}P$ is a vector $r \in \mathbb {R}^k$, where each entry is computed by one slice of the tensor ($r_i=A^TT_1^{[i]}P, i = 1, \cdots , k$). The other parameters are a standard feed-forward neural network, where $W \in \mathbb {R}^{k \times \it 2d}$ is the weight matrix, $b \in \mathbb {R}^k$ is the bias vector, $U \in \mathbb {R}^k$ is a hyper-parameter and $f=\it tanh$ is a standard nonlinearity applied element-wise. S_2 and C in Figure FIGREF5 are computed in the same way as S_1.

One problem with tensors is curse of dimensionality, which limits the wide application of tensors in many areas. It is therefore essential to approximate tensors of higher order in a compressed scheme, for example, a low-rank tensor decomposition. To decrease the number of parameters in standard neural tensor network, we make low-rank approximation that represents each matrix by two low-rank matrices plus diagonal, as illustrated in Figure FIGREF7. Formally, the parameter of the i-th slice is $T_{appr}^{[i]}=T^{[i_1]}\times T^{[i_2]}+diag(t^{[i]})$, where $T^{[i_1]}\in \mathbb {R}^{d\times n}$,

$T^{[i_2]}\in \mathbb {R}^{n\times d}$, $t^{[i]}\in \mathbb {R}^d$, n is a hyper-parameter, which is used for adjusting the degree of tensor decomposition. The output of neural tensor layer is formalized as follows.

where $[T_{appr}]_1^{[1:k]}$ is the low-rank tensor that defines multiple low-rank bilinear layers. k is the slice number of neural tensor network which is also equal to the output length of S_1.

We assume that event tuples in the training data should be scored higher than corrupted tuples, in which one of the event arguments is replaced with a random argument. Formally, the corrupted event tuple is $E^r=(A^r, P, O)$, which is derived by replacing each word in A with a random word w^r in our dictionary $\mathcal {D}$ (which contains all the words in the training data) to obtain a corrupted counterpart A^r. We calculate the margin loss of the two event tuples as:

where $\mathit {\Phi }=(T_1, T_2, T_3, W, b)$ is the set of model parameters. The standard L_2 regularization is used, for which the weight $\lambda $ is set as 0.0001. The algorithm goes over the training set for multiple iterations. For each training instance, if the loss $loss(E,E^r)=\max (0,1-g(E)+g(E^r))$ is equal to zero, the online training algorithm continues to process the next event tuple. Otherwise, the parameters are updated to minimize the loss using back-propagation BIBREF9.

Commonsense Knowledge Enhanced Event Representations ::: Intent Embedding

Intent embedding refers to encoding the event participants' intents into event vectors, which is mainly used to explain why the actor performed the action. For example, given two events “PersonX threw basketball” and “PersonX threw bomb”, there are only subtle differences in their surface realizations, however, the intents are totally different. “PersonX threw basketball” is just for fun, while “PersonX threw bomb” could be a terrorist attack. With the intents, we can easily distinguish these superficial similar

events.

One challenge for incorporating intents into event embeddings is that we should have a large-scale labeled dataset, which annotated the event and its actor's intents. Recently, BIBREF6 P18-1043 and BIBREF7 sap2018atomic released such valuable commonsense knowledge dataset (ATOMIC), which consists of 25,000 event phrases covering a diverse range of daily-life events and situations. For example, given an event “PersonX drinks coffee in the morning”, the dataset labels PersonX's likely intent is “PersonX wants to stay awake”.

We notice that the intents labeled in ATOMIC is a sentence. Hence, intent embedding is actually a sentence representation learning task. Among various neural networks for encoding sentences, bi-directional LSTMs (BiLSTM) BIBREF10 have been a dominant method, giving state-of-the-art results in language modelling BIBREF11 and syntactic parsing BIBREF12.

We use BiLSTM model to learn intent representations. BiLSTM consists of two LSTM components, which process the input in the forward left-to-right and the backward right-to-left directions, respectively. In each direction, the reading of input words is modelled as a recurrent process with a single hidden state. Given an initial value, the state changes its value recurrently, each time consuming an incoming word.

Take the forward LSTM component for example. Denoting the initial state as $\overrightarrow{\mathbf {h}}^0$, which is a model parameter, it reads the input word representations $\mathbf {x}_0,\mathbf {x}_1,\dots ,\mathbf {x}_n$, and the recurrent state transition step for calculating $\overrightarrow{\mathbf {h}}^1,\dots ,\overrightarrow{\mathbf {h}}^{n+1}$ is defined as BIBREF13 (BIBREF13).

The backward LSTM component follows the same recurrent state transition process as the forward LSTM component. Starting from an initial state $\overleftarrow{\mathbf {h}}^{n+1}$, which is a model parameter,

it reads the input $\mathbf {x}_n,\mathbf {x}_{n-1},\dots ,\mathbf {x}_0$, changing its value to $\overleftarrow{\mathbf {h}}^n,\overleftarrow{\mathbf {h}}^{n-1},\dots ,\overleftarrow{\mathbf {h}}^0$, respectively. The BiLSTM model uses the concatenated value of $\overrightarrow{\mathbf {h}}^t$ and $\overleftarrow{\mathbf {h}}^t$ as the hidden vector for w_t:

A single hidden vector representation $\mathbf {v}_i$ of the input intent can be obtained by concatenating the last hidden states of the two LSTMs:

In the training process, we calculate the similarity between a given event vector $\mathbf {v}_e$ and its related intent vector $\mathbf {v}_i$. For effectively training the model, we devise a ranking type loss function as follows:

where $\mathbf {v}^{\prime }_i$ is the incorrect intent for $\mathbf {v}_e$, which is randomly selected from the annotated dataset.

Commonsense Knowledge Enhanced Event Representations ::: Sentiment Embedding

Sentiment embedding refers to encoding the event participants' emotions into event vectors, which is mainly used to explain how does the actor feel after the event. For example, given two events “PersonX broke record” and “PersonX broke vase”, there are only subtle differences in their surface realizations, however, the emotions of PersonX are totally different. After “PersonX broke record”, PersonX may be feel happy, while after “PersonX broke vase”, PersonX could be feel sad. With the emotions, we can also effectively distinguish these superficial similar events.

We also use ATOMIC BIBREF7 as the event sentiment labeled dataset. In this dataset, the sentiment of the event is labeled as words. For example, the sentiment of “PersonX broke vase” is labeled as “(sad, be

regretful, feel sorry, afraid)". We use SenticNet BIBREF14 to normalize these emotion words ($W=\lbrace w_1, w_2, \dots , w_n\rbrace $) as the positive (labeled as 1) or the negative (labeled as -1) sentiment. The sentiment polarity of the event P_e is dependent on the polarity of the labeled emotion words P_W: $P_e=1$, if $\sum _i P_{w_i}>0$, or $P_e=-1$, if $\sum _i P_{w_i}<0$. We use the softmax binary classifier to learn sentiment enhanced event embeddings. The input of the classifier is event embeddings, and the output is its sentiment polarity (positive or negative). The model is trained in a supervised manner by minimizing the cross entropy error of the sentiment classification, whose loss function is given below.

where C means all training instances, L is the collection of sentiment categories, x_e means an event vector, $p_l(x_e)$ is the probability of predicting x_e as class l, $p^g_l(x_e)$ indicates whether class l is the correct sentiment category, whose value is 1 or -1.

Commonsense Knowledge Enhanced Event Representations ::: Joint Event, Intent and Sentiment Embedding

Given a training event corpus with annotated intents and emotions, our model jointly minimizes a linear combination of the loss functions on events, intents and sentiment:

where $\alpha , \beta , \gamma \in [0,1]$ are model parameters to weight the three loss functions.

We use the New York Times Gigaword Corpus (LDC2007T07) for pre-training event embeddings. Event triples are extracted based on the Open Information Extraction technology BIBREF15. We initialize the word embedding layer with 100 dimensional pre-trained GloVe vectors BIBREF8, and fine-tune initialized word vectors during our model training. We use Adagrad BIBREF16 for optimizing the parameters with initial learning rate 0.001 and batch size 128.

Experiments

We compare the performance of intent and sentiment powered event embedding model with state-of-the-art baselines on three tasks: event similarity, script event prediction and stock prediction.

Experiments ::: Baselines

We compare the performance of our approach against a variety of event embedding models developed in recent years. These models can be categorized into three groups:

Averaging Baseline (Avg) This represents each event as the average of the constituent word vectors using pre-trained GloVe embeddings BIBREF8.

Compositional Neural Network (Comp. NN) The event representation in this model is computed by feeding the concatenation of the subject, predicate, and object embedding into a two layer neural network BIBREF17, BIBREF3, BIBREF2.

Element-wise Multiplicative Composition (EM Comp.) This method simply concatenates the element-wise multiplications between the verb and its subject/object.

Neural Tensor Network This line of work use tensors to learn the interactions between the predicate and its subject/object BIBREF4, BIBREF5. According to the different usage of tensors, we have three baseline methods: Role Factor Tensor BIBREF5 which represents the predicate as a tensor, Predicate Tensor BIBREF5 which uses two tensors learning the interactions between the predicate and its subject, and the predicate and its object, respectively, NTN BIBREF4, which we used as the baseline event embedding model in this paper, and KGEB BIBREF18, which incorporates knowledge graph information in NTN.

Experiments ::: Event Similarity Evaluation ::: Hard Similarity Task

We first follow BIBREF5 (BIBREF5) evaluating our proposed approach on the hard similarity task. The goal of this task is that similar events should be close to each other in the same vector space, while dissimilar events should be far away with each other. To this end, BIBREF5 (BIBREF5) created two types of event pairs, one with events that should be close to each other but have very little lexical overlap (e.g., police catch robber / authorities apprehend suspect), and another with events that should be farther apart but have high overlap (e.g., police catch robber / police catch disease).

The labeled dataset contains 230 event pairs (115 pairs each of similar and dissimilar types). Three different annotators were asked to give the similarity/dissimilarity rankings, of which only those the annotators agreed upon completely were kept. For each event representation learning method, we obtain the cosine similarity score of the pairs, and report the fraction of cases where the similar pair receives a higher cosine value than the dissimilar pair (we use Accuracy $\in [0,1]$ denoting it). To evaluate the robustness of our approach, we extend this dataset to 1,000 event pairs (similar and dissimilar events each account for 50%), and we will release this dataset to the public.

Experiments ::: Event Similarity Evaluation ::: Transitive Sentence Similarity

Except for the hard similarity task, we also evaluate our approach on the transitive sentence similarity dataset BIBREF19, which contains 108 pairs of transitive sentences: short phrases containing a single subject, object and verb (e.g., agent sell property). It also has another dataset which consists of 200 sentence pairs. In this dataset, the sentences to be compared are constructed using the same subject and object and semantically correlated verbs, such as `spell' and `write'; for example, `pupils write letters' is compared with `pupils spell letters'. As this dataset is not suitable for our task, we only evaluate our approach and baselines on 108 sentence pairs.

Every pair is annotated by a human with a similarity score from 1 to 7. For example, pairs such as (design, reduce, amount) and (company, cut, cost) are annotated with a high similarity score, while pairs such as (wife, pour, tea) and (worker, join, party) are given low similarity scores. Since each pair has several annotations, we use the average annotator score as the gold score. To evaluate the cosine similarity given by each model and the annotated similarity score, we use the Spearman's correlation ($\rho \in [-1,1]$).

Experiments ::: Event Similarity Evaluation ::: Results

Experimental results of hard similarity and transitive sentence similarity are shown in Table TABREF23. We find that:

(1) Simple averaging achieved competitive performance in the task of transitive sentence similarity, while performed very badly in the task of hard similarity. This is mainly because hard similarity dataset is specially created for evaluating the event pairs that should be close to each other but have little lexical overlap and that should be farther apart but have high lexical overlap. Obviously, on such dataset, simply averaging word vectors which is incapable of capturing the semantic interactions between event arguments, cannot achieve a sound performance.

(2) Tensor-based compositional methods (NTN, KGEB, Role Factor Tensor and Predicate Tensor) outperformed parameterized additive models (Comp. NN and EM Comp.), which shows that tensor is capable of learning the semantic composition of event arguments.

(3) Our commonsense knowledge enhanced event representation learning approach outperformed all baseline methods across all datasets (achieving 78% and 200% improvements on hard similarity small and big dataset, respectively, compared to previous SOTA method), which indicates that commonsense

knowledge is useful for distinguishing distinct events.

Experiments ::: Event Similarity Evaluation ::: Case Study

To further analyse the effects of intents and emotions on the event representation learning, we present case studies in Table TABREF29, which directly shows the changes of similarity scores before and after incorporating intent and sentiment. For example, the original similarity score of two events “chef cooked pasta” and “chef cooked books” is very high (0.89) as they have high lexical overlap. However, their intents differ greatly. The intent of “chef cooked pasta” is “to hope his customer enjoying the delicious food”, while the intent of “chef cooked books” is “to falsify their financial statements”. Enhanced with the intents, the similarity score of the above two events dramatically drops to 0.45. For another example, as the event pair “man clears test” and “he passed exam” share the same sentiment polarity, their similarity score is boosted from -0.08 to 0.40.

Experiments ::: Script Event Prediction

Event is a kind of important real-world knowledge. Learning effective event representations can be benefit for numerous applications. Script event prediction BIBREF20 is a challenging event-based commonsense reasoning task, which is defined as giving an existing event context, one needs to choose the most reasonable subsequent event from a candidate list.

Following BIBREF21 (BIBREF21), we evaluate on the standard multiple choice narrative cloze (MCNC) dataset BIBREF2. As SGNN proposed by BIBREF21 (BIBREF21) achieved state-of-the-art performances for this task, we use the framework of SGNN, and only replace their input event embeddings with our intent and sentiment-enhanced event embeddings.

BIBREF22 (BIBREF22) and BIBREF21 (BIBREF21) showed that script event prediction is a challenging problem, and even 1% of accuracy improvement is very difficult. Experimental results shown in Table TABREF31 demonstrate that we can achieve more than 1.5% improvements in single model comparison and more than 1.4% improvements in multi-model integration comparison, just by replacing the input embeddings, which confirms that better event understanding can lead to better inference results. An interesting result is that the event embeddings only incorporated with intents achieved the best result against other baselines. This confirms that capturing people's intents is helpful to infer their next plan. In addition, we notice that the event embeddings only incorporated with sentiment also achieve better performance than SGNN. This is mainly because the emotional consistency does also contribute to predicate the subsequent event.

Experiments ::: Stock Market Prediction

It has been shown that news events influence the trends of stock price movements BIBREF23. As news events affect human decisions and the volatility of stock prices is influenced by human trading, it is reasonable to say that events can influence the stock market.

In this section, we compare with several event-driven stock market prediction baseline methods: (1) Word, BIBREF23 luss2012predicting use bag-of-words represent news events for stock prediction; (2) Event, BIBREF24 ding-EtAl:2014:EMNLP2014 represent events by subject-predicate-object triples for stock prediction; (3) NTN, BIBREF4 ding2015deep learn continues event vectors for stock prediction; (4) KGEB, BIBREF18 ding2016knowledge incorporate knowledge graph into event vectors for stock prediction.

Experimental results are shown in Figure FIGREF33. We find that knowledge-driven event embedding is a competitive baseline method, which incorporates world knowledge to improve the performances of

event embeddings on the stock prediction. Sentiment is often discussed in predicting stock market, as positive or negative news can affect people's trading decision, which in turn influences the movement of stock market. In this study, we empirically show that event emotions are effective for improving the performance of stock prediction (+2.4%).

Related Work

Recent advances in computing power and NLP technology enables more accurate models of events with structures. Using open information extraction to obtain structured events representations, we find that the actor and object of events can be better captured BIBREF24. For example, a structured representation of the event above can be (Actor = Microsoft, Action = sues, Object = Barnes & Noble). They report improvements on stock market prediction using their structured representation instead of words as features.

One disadvantage of structured representations of events is that they lead to increased sparsity, which potentially limits the predictive power. BIBREF4 ding2015deep propose to address this issue by representing structured events using event embeddings, which are dense vectors. The goal of event representation learning is that similar events should be embedded close to each other in the same vector space, and distinct events should be farther from each other.

Previous work investigated compositional models for event embeddings. BIBREF2 granroth2016happens concatenate predicate and argument embeddings and feed them to a neural network to generate an event embedding. Event embeddings are further concatenated and fed through another neural network to predict the coherence between the events. Modi modi2016event encodes a set of events in a similar way and use that to incrementally predict the next event – first the argument, then the predicate and then next argument. BIBREF25 pichotta2016learning treat event prediction as a sequence to sequence problem

and use RNN based models conditioned on event sequences in order to predict the next event. These three works all model narrative chains, that is, event sequences in which a single entity (the protagonist) participates in every event. BIBREF26 hu2017happens also apply an RNN approach, applying a new hierarchical LSTM model in order to predict events by generating descriptive word sequences. This line of work combines the words in these phrases by the passing the concatenation or addition of their word embeddings to a parameterized function that maps the summed vector into event embedding space. The additive nature of these models makes it difficult to model subtle differences in an event's surface form.

To address this issue, BIBREF4 ding2015deep, and BIBREF5 weber2018event propose tensor-based composition models, which combine the subject, predicate and object to produce the final event representation. The models capture multiplicative interactions between these elements and are thus able to make large shifts in event semantics with only small changes to the arguments.

However, previous work mainly focuses on the nature of the event and lose sight of external commonsense knowledge, such as the intent and sentiment of event participants. This paper proposes to encode intent and sentiment into event embeddings, such that we can obtain a kind of more powerful event representations.

Conclusion

Understanding events requires effective representations that contain commonsense knowledge. High-quality event representations are valuable for many NLP downstream applications. This paper proposed a simple and effective framework to incorporate commonsense knowledge into the learning process of event embeddings. Experimental results on event similarity, script event prediction and stock prediction showed that commonsense knowledge enhanced event embeddings can improve the quality of event representations and benefit the downstream applications.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the National Key Research and Development Program of China (SQ2018AAA010010), the National Key Research and Development Program of China (2018YFB1005103), the National Natural Science Foundation of China (NSFC) via Grant 61702137.