Simultaneous Fake News and Topic Classification via Auxiliary Task Learning

Tsun-hin Cheung, Kin-man Lam
Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
Email: tsun-hin.cheung@connect.polyu.hk, enkmlam@polyu.edu.hk

***Abstract*— Using social media, in particular, reading news articles, has become a necessary daily activity and an important way of spreading information. Classification of topics of new articles can provide up-to-date information about the current state of politics and society. However, this convenient way of sharing information can lead to the growth of falsification. Therefore, distinguishing between real and fake news, as well as fake-news classification, have become essential and indispensable. In this paper, we propose a new and up-to-date dataset for both fake-news classification and topic classification. To the best of our knowledge, we are the first to construct a dataset with both fake-news and topic labels, and employ multi-task learning for learning these two tasks simultaneously. We have collected 21K online news articles published from January 2013 to March 2020. We propose an auxiliary-task long short-term memory (AT-LSTM) neural network for text classification via multi-task learning. We evaluate and compare our proposed model to five baseline methods, via both single-task and multi-task learning, on this new benchmark dataset. Experimental results show that our proposed AT-LSTM model outperforms the single-task learning methods and the hard parameter-sharing multi-task learning methods. The dataset and codes will be released in the future.**

Keywords—web data mining, fake-news classification, topic classification, multi-task learning

I. INTRODUCTION

The spread of misinformation on the Internet is an influential and critical issue, especially in social media. Fake-news articles provide false information to the public and have a strong impact on both politics and society (an example is shown in Fig. 1). There is an increasing trend for fake news since the 2016 US Presidential election [1]. Automatic fake-news detection has raised public interest, since it is useful to reduce human effort in classification. Several ways of identifying online fake-news articles have been proposed in recent years. For example, there are tools for spotting domain names and IP addresses of fake-news sources. However, it is easy to change the domain names or dynamic IP addresses, so it is difficult to prevent fake news. This also leads to the need for a significant amount of human effort to maintain the list of the sources. Moreover, people may repost the fake-news articles on their social network sites without specifying the sources. This makes the tracing of fake-news sources more difficult. Due to the successful development of machine learning and natural language processing, several prediction models for fake-news classification have been developed in recent decades.

The first public dataset for fake-news classification was released by Vlachos and Riedel [2] in 2014. It is a small dataset, which contains about 200 sentences. Therefore, the dataset is not large enough to train deep neural models. A relatively recent study, by Wang [3] in 2017, collected 12.8K corpus for fact-checking classification through POLITIFACT.COM's API. The study considered the statements of a fact with several types of metadata, such as speakers, subject, history, etc. This dataset contains fact-like statements, which are different from the form of news articles. They proposed a hybrid convolutional neural network for fake statement classification by concentrating on the statements and their metadata features. Our proposed model was inspired by this method, but we employ the long short-term memory (LSTM) encoders and the classification of the meta-data.

The Kaggle challenge [4], developed by George McIntire, provides a dataset for classifying fake-news articles. In this challenge, the fake-news articles were collected from the websites listed in BS detector [5], while those real-news articles were from traditional news media websites, such as *New York Times*, *Bloomberg*, and *The Guardian*. Our data collection strategy is similar to this challenge. On top of this, we have further crawled the news meta-categories that are used for topic classification, as shown in Fig. 1.

Figure 1. A fake political article published in *newsthump.com*.

Topic classification, also called text categorization, has a longer history than fake-news classification. It has been studied since the early development of the World Wide Web (WWW). Some of the famous datasets for categorising news articles include 20 Newsgroup [6] released in 1995, Reuters-21578 [7] released in 1997, and AG News released in 2004. There have been intensive studies on automatic text classification based on these datasets. However, the news articles in the datasets were published more than 15 years ago, and written in traditional styles. In our studies, we have collected up-to-date online news articles, published from 2013 to 2020, on media websites. This makes our study suitable for

current real-world applications for fake-news and topic classification.

Fake-news classification and topic classification are applications of text classification. In recent years, deep-learning models have been widely used for text classification, and have achieved great performance. Convolutional neural networks (CNNs) [8] and recurrent neural networks (RNNs) [9] are the two most popular deep neural models for natural language understanding. These deep neural models are considered the baseline methods used for comparison in our experiments.

Multi-task learning has been broadly studied in machine learning, across a number of fields, including computer vision (CV) and natural language processing (NLP). It is similar to transfer learning, and aims to learn several related tasks at the same time. In NLP, multi-task learning has been used in jointly learning the tasks, such as Part-of-Speech (POS) tagging and Named Entity Recognition (NER) [10], as well as sentiment and sarcasm classification [11]. This learning strategy has shown its powerful generalization capacity to train the deep neural models.

Our proposed method comes with the idea of auxiliary tasks in multi-task learning. The objective of auxiliary tasks is to supplement and support the learning of the main tasks in multi-task learning. Auxiliary tasks are mainly for learning a robust representation of input documents to boost the training of deep neural models. Liebel and Korner [12] studied the effect and performance of auxiliary tasks in CNN models for image classification. We employ this concept to form our proposed multi-task learning framework for text classification.

II. Methodology

In this section, we will first describe how data for our dataset was collected from the Internet, and then, how the collected data was labelled to construct the dataset. After that, we give a detailed presentation of our proposed deep neural model for fake-news and topic classification, and the training of the model.

A. Data collection

We developed our web crawlers and collected the fake-news articles from those websites listed in *mediabiasfactcheck.com*, as well as those real news articles from the *New York Times* and the *Guardian's* APIs. For each of the news article's web pages, we extract its meta-description tag as our input document, as illustrated in Table I. We also parsed the HTML web page and obtained the topics assigned to each news article based on the news websites, and then grouped them into five categories, with the labels as shown in Table II. We can see that the percentage of fake news available is the highest for "Politics", while the lowest is for "Sports".

TABLE I
DATA FIELDS - DESCRIPTION OF OUR NEW BENCHMARK DATASET.

FIELD	DESCRIPTION
URL	The unique identifier for each news article.
Title	The title of each news article.
Content	The snippet of the news article. It is used as the input document.
Reality	Whether the document comes from a fake or reliable news media.
Topic	The category that the article assigned to.

TABLE II
DATA DISTRIBUTION – NUMBER OF REAL AND FAKE DOCUMENTS FOR THE DIFFERENT CATEGORIES.

TOPIC	REALITY	SIZE
Politics	Real	2653
	Fake	2329
Science/Technology	Real	3633
	Fake	1193
Business	Real	2377
	Fake	1574
Health	Real	2542
	Fake	1100
Sports	Real	2432
	Fake	473

B. The proposed model

Since there are two output labels in the dataset, we aim to build a deep neural network that jointly learns the two classification tasks. We regard one of the tasks as the main task and the other one as the auxiliary task. The auxiliary task is responsible for improving the training of the main task. We propose the auxiliary-task long short-term memory (AT-LSTM) for jointly learning the two tasks. Fig. 2 shows the overview of our proposed AT-LSTM.

Word Embedding: The input of the text-classification model is a document $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$, where T is the number of words in the document. There are two output labels, denoted as $\boldsymbol{y}_i \in \mathbb{R}^2$, for each task i in our model. Word embedding is to map the words in a document into real-value vectors. Each word $\boldsymbol{x}_i$ is represented by a D-dimensional embedding vector, i.e. $\boldsymbol{x}_i \in \mathbb{R}^D$, which is then passed to the LSTM encoders.

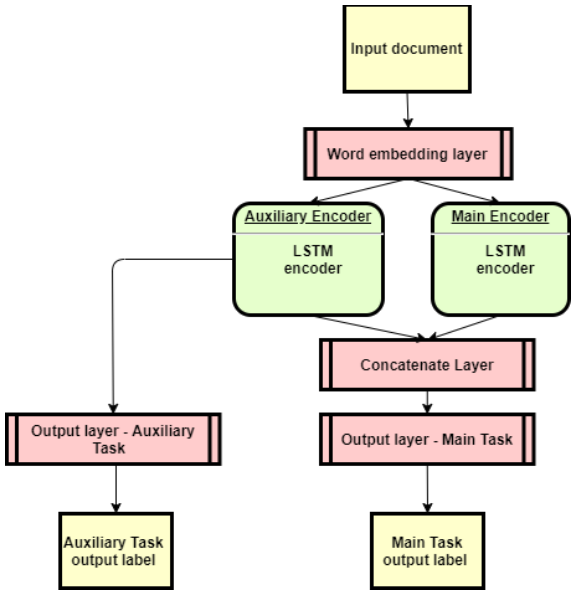

Figure 2. The proposed AT-LSTM model.

LSTM encoders: The length of the embedded vector for a document, depending on the number of words in the document, is not constant. The primary aim of the neural encoder is to represent the variable-length word embedding vectors as a fixed-length vector, say length M. There are two encoders in our proposed model: main encoder and auxiliary encoder. The main encoder is only responsible for the main task classification, while the auxiliary encoder is to generate a common feature for both the main and auxiliary tasks. Both encoders employ the LSTM unit. An LSTM unit can process an arbitrary-length sequence by recursively applying a transition function to form the hidden state vector. It consists of three gates – the forget gate $\boldsymbol{f}_t$, input gate $\boldsymbol{i}_t$, and output gate $\boldsymbol{o}_t$, and two memory states – cell state $\boldsymbol{c}_t$ and hidden state $\boldsymbol{h}_t$. The input gate $\boldsymbol{i}_t$ controls the amount of information from the current input to the cell state $\boldsymbol{c}_t$. The forget gate $\boldsymbol{f}_t$ tells the cell state $\boldsymbol{c}_t$ which information from the previous cell $\boldsymbol{c}_{t-1}$ to forget. The output gate $\boldsymbol{o}_t$ is responsible for selecting the information in the cell gate $\boldsymbol{c}_t$ to form the hidden state $\boldsymbol{h}_t$. All these vectors are in $\mathbb{R}^M$, where M is a hyperparameter. Given an input word embedding vector $\boldsymbol{x}_t$, we compute its representation $\boldsymbol{h}_t$ in the first layer of a LSTM encoder by using Equations (1) to (5):

$$\boldsymbol{i}_t = \sigma(W_i \cdot [h_{t-1}, x_t]) + b_i, \quad (1)$$

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t]\big) + b_f, \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) + b_o, \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (4)$$

$$h_t = o_t \odot \tanh\,(c_t), \quad (5)$$

where σ and $\odot$ denote the sigmoid function and elementwise multiplication, respectively. The weight matrix $\boldsymbol{W}_p$ and bias vector $\boldsymbol{b}_p$ are the learnable parameters of the gate or state p.

Output Layer: Since our objective is to represent an input document as a fixed-length vector, we regard the last hidden representation $\boldsymbol{h}_T$ as the feature extracted from the document. We use the last hidden vector in the auxiliary encoder for the auxiliary task classification. To achieve this, the feature vector is transformed into a K-dimensional vector by using a linear function, followed by the SoftMax function, to generate the output y_a, as follows:

$$y_a = \text{softmax}(W_s h_T), \quad (6)$$

where K is the number of possible labels in the auxiliary task.

Concatenation Layer: We concatenate the feature vectors from the two encoders to form the final feature representation of a document, which is used for the main classification task. Finally, we transform the final representation vector into a N-dimensional vector and pass it through the SoftMax function to generate the output y_m for the main classification task, as follows:

$$y_m = \text{softmax}(W_s' h_T'). \quad (7)$$

Loss function: There are two output layers in our model: the main task and the auxiliary task. Our goal is to jointly learn these two tasks simultaneously. The cross-entropy loss is used for both tasks. The total loss is a weighted sum of the cross-entropy loss of the main task and that of the auxiliary task, as follows:

$$Loss_{total} = -\,\alpha \sum \log(y_a) - (1-\alpha) \sum \log(y_m), \quad (8)$$

where α is a hyperparameter to control the relative weights of these two losses.

III. EXPERIMENTS, RESULTS

A. Data preprocessing

We split our collected corpus into three subsets: 10.6K, 5K, and 5K documents used for training, validation, and testing, respectively. Each document is assigned two labels, one for fake-news classification and another one is the topic category. The special characters, punctuation marks, etc. are removed from the input documents. Then, we tokenize the documents into a sequence of words using the Natural Language Toolkit (NLTK) [13].

B. Experimental settings

The pretrained 200-dimensional Glove word embeddings [14] are used as input for the deep models evaluated in our experiments. For single-task learning, we trained and evaluated the baseline results for both tasks using fastText [15], textCNN [8], LSTM [9], AttenLSTM [16] separately. We also compare our methods to the hard parameter sharing LSTM via multi-task learning [17]. For the textCNN model, the filter sizes of the CNN model are 3, 4, and 5. The number of filters in each layer is 100. For LSTM and AttenLSTM, the number of layers is 2, while the dimension of each hidden layer in LSTM is 256. For our proposed AT-LSTM model, the number of layers is 2, and the dimension of each hidden layer in LSTM is 128. For all the experiments, the maximum number of epochs is 100, and the batch size is 32. We used the Adam optimizer to train our models, and the learning rate used is 0.001. To avoid overfitting, we used a dropout rate of 0.5 and weight decay of 1e-6. We implemented the deep model with PyTorch, and reported the performance on the testing datasets with the best model achieved, when the validation accuracy, as below, is the highest during training.

$$error\ rate = \frac{\text{number of mispredicted samples}}{\text{number of samples}}, \quad (9)$$

$$accuracy = 1 - error\ rate\,. \quad (10)$$

C. Overall results

Table 3 shows the overall accuracy of the two classifications tasks trained and evaluated on several models via single-task learning and multi-task learning methods.

TABLE III
OVERALL COMPARISON OF THE BASELINE METHODS AND OUR PROPOSED MULTI-TASK LSTM.

Type	Methods	Fake-news Classification Accuracy (%)	Topic Classification Accuracy (%)
Single Task	fastText [15]	76.82	73.79
Single Task	textCNN [8]	87.52	74.24
Single Task	LSTM [9]	91.51	76.59
Single Task	AttenLSTM [16]	92.95	76.82
Multi-task	FS-LSTM [17]	92.01	76.68
Multi-task	**AT-LSTM (ours)**	**93.19**	**77.23**

IV. Discussion

A. Comparison of single-task learning

For single-task learning, experiment results show that the LSTM encoders outperform the CNN encoders, i.e. textCNN, and fastText. fastText is based on bag-of-words features, in both the fake news classification and topic classification tasks. In Table III, we can see that textCNN and the LSTM encoder achieve a 10.7% and 14.69% improvement, in terms of accuracy, when compared to fastText for fake news classification. For topic classification, the accuracy of textCNN and LSTM encoder is 0.45% and 2.8%, respectively, higher than that of fastText. This means that the neural encoders can capture more advanced and useful information than the bag-of-words features in fastText for both tasks. The LSTM encoder has a 4% improvement in fake news classification and a 2% improvement in topic classification, compared to textCNN. This shows that classifying fake news requires more long-term dependent context information than topic classification, rather than the local and position-invariant features in a sentence.

B. Comparison of single-task learning and multi-task learning

As LSTM outperforms textCNN, we evaluate our multi-task learning framework on the LSTM encoders. It is worth noting that any deep neural networks can be used as the encoders in our proposed multi-task learning framework. The experimental results in Table III show that our proposed multi-task learning framework outperforms all the single-task learning methods. The fully shared LSTM (FS-LSTM) and our proposed auxiliary-task LSTM (AT-LSTM) increase the accuracy by 0.5% and 1.68%, respectively, compared to LSTM for fake-news classification, and 0.09% and 0.64%, respectively, for topic classification. This shows that the two tasks have common features, which can be learnt by the LSTM encoders.

Compared to the LSTM network with hard parameter sharing, our proposed AT-LSTM model achieves better performance. When topic classification is the main task and fake-news classification is the auxiliary task, the accuracy of the main and auxiliary tasks is 77.23% and 92.44%, respectively. When the two tasks are interchanged, the accuracy of the main and auxiliary tasks is 93.19% and 76.35%, respectively. This shows that the task assigned as the main task can achieve a relatively higher accuracy. It is worth noting that the auxiliary task has a regularization effect to avoid overfitting in the main LSTM encoder. The information in the auxiliary encoder is helpful to the main task. This explains why AT-LSTM has a better generalization capacity when the topic classification and fake-news classification are learned simultaneously by using multi-task learning.

C. The hyperparameter of the weights of the loss function

We evaluate our results for different values of α in the loss function. It is found that the optimal value of α depends on the specific task. In the experiment, we found that using a larger value α can achieve a better performance when topic classification is the main task, i.e. $\alpha = 0.7$, as shown in Fig. 3. By contrast, a smaller value α is required for achieving a better performance for fake-news classification, i.e. $\alpha = 0.5$. For each of the two tasks, when the range of value α is between 0.2 and 1, the classification task assigned as the main task can always achieve a better performance. This is due to the fact that the auxiliary encoder can generate common features across two tasks and help to boost the performance of the main classification task.

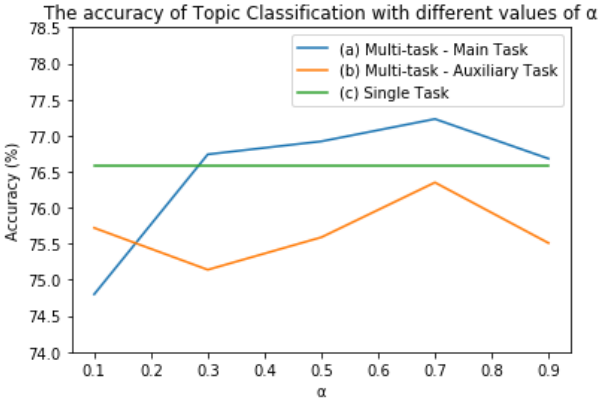

Figure 3. The effect of different values of α for topic classification. (a) AT-LSTM: Topic classification is the main task. (b) AT-LSTM: Topic classification is the auxiliary task. (c) Single-task LSTM.

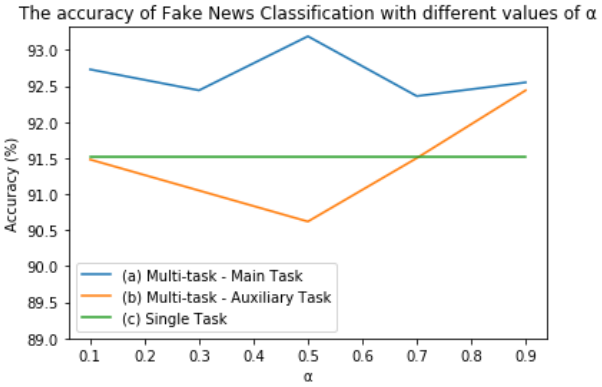

Figure 4. The effect of different values of α for fake news classification (a) AT-LSTM: Fake news classification is the main task. (b) AT-LSTM: Fake news classification is the auxiliary task. (c) Single-task LSTM.

D. Error Analysis

From Fig. 5, the accuracy of correctly classifying real and fake news articles is 96% and 88%, respectively. This means that classifying fake-news articles is more difficult than real-news articles. One of the possible reasons for this is that the database is imbalanced, so that there are more real-news articles than that of fake-news articles in the dataset.

From Fig. 6, we can see that the category "Sports" has the highest accuracy, which is 86%. It means that the sports news articles are more discriminative than the other categories. On the other hand, those business news articles have the lowest accuracy, which is 68%. These news articles are more difficult to classify, because they are close to the politics and science/

technology categories. We can see that the business and science/technology categories are most confused. This is reasonable and acceptable, because a news article may have more than one topic, and the news websites do not label them separately.

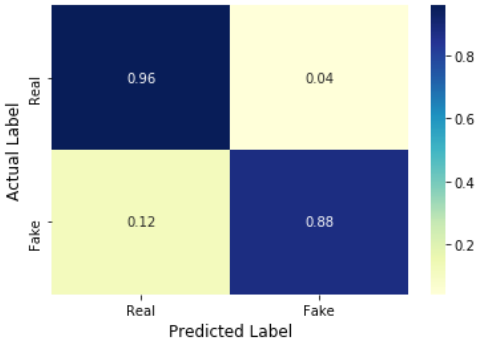

Figure 5. The confusion matrix of fake-news classification using AT-LSTM.

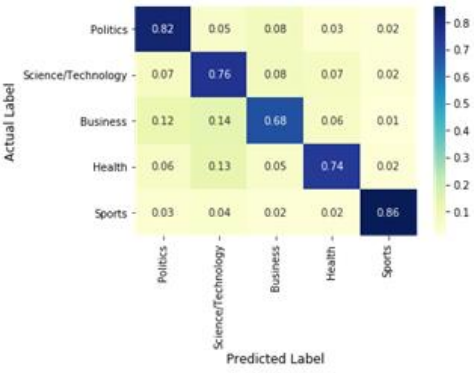

Figure 6. The confusion matrix of topic classification using AT-LSTM.

V. Conclusion

In this paper, we have constructed a new benchmark dataset for both the research of fake-news classification and topic classification. Our experiment results have shown that using the LSTM encoders can achieve a better performance than the textCNN encoders, for both tasks trained by single-task learning. In order to train the two tasks simultaneously, we proposed an auxiliary-task long short-term memory (AT-LSTM) framework to jointly learn these two tasks. Our model can achieve better performance than the multi-task LSTM network with hard parameter sharing, due to the generalization capacity of the auxiliary task. In our future work, we will consider more tasks, such as sentiment analysis, for text classification via multi-task learning.

References

[1] Alexandre Bovet, Hernan A. Makse, "Influence of fake news in Twitter during the 2016 US presidential election," *Nature Communications* 10, 2019.

[2] Andreas Vlachos, Sebastian Riedel, "Fact Checking: Task definition and dataset construction, " *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, 2014, pp. 18–22.

[3] William Yang Wang, "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection, " *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 422–426.

[4] https://github.com/GeorgeMcIntire/fake_real_news_dataset, accessed 20 October 2018.

[5] B.S. Detector. A browser extension that alerts users to unreliable news sources. [Online]. Available: http://bsdetector.tech/.

[6] Ken Lang, "Newsweeder: Learning to filter netnews," *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 331-339.

[7] D. D. Lewis. Reuters-21578 text Categorization test collection. Distribution 1.0. README file (version 1.2). Manuscript, September 26, 1997.

[8] Yoon Kim, "Convolutional Neural Networks for Sentence Classification," *Conference on Empirical Methods in Natural Language Processing*, 2014.

[9] Sepp Hochreiter, Jurgen Schmidhuber, "Long Short-term memory," *Neural Computation*, 1997, pp.1735-1780.

[10] Victor Sanh, Thomas Wolf, Sebastian Ruder, "A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks," *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[11] Navonil Majumder, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, Alexander Gelbukh, "Sentiment and Sarcasm Classification with Multitask Learning," *IEEE Intelligent Systems* 34(3), 2019.

[12] Liebel, Lukas and Marco Körner, "Auxiliary Tasks in Multi-task Learning." *ArXiv abs/1805.06334*, 2018.

[13] Bird, Steven, Edward Loper and Ewan, "Natural Language Processing with Python,". *O'Reilly Media Inc*, 2009.

[14] Jeffrey Pennington , Richard Socher , Christopher D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[15] Joulin, Armand and Grave, Edouard and Bojanowski, Piotr and Mikolov, Tomas, "Bag of Tricks for Efficient Text Classification," *arXiv preprint arXiv:1607.01759*, 2016.

[16] GangLiu, JiabaoGuo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, 2019, pp. 325-338.

[17] Pengfei Liu, "Recurrent Neural Network for Text Classification with Multi-Task Learning," *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.