Joint Learning on Relevant User Attributes in Micro-blog

Jingjing Wang, Shoushan Li*, Guodong Zhou
Natural Language Processing Lab, School of Computer Science and Technology
Soochow University, Suzhou, 215006, China
djingwang@gmail.com, {lishoushan, gdzhou}@suda.edu.cn

Abstract

User attribute classification aims to identify users' attributes (e.g., *gender*, *age* and *profession*) by leveraging user generated content. However, conventional approaches to user attribute classification focus on single attribute classification involving only one user attribute, which completely ignores the relationship among various user attributes. In this paper, we confront a novel scenario in user attribute classification where relevant user attributes are jointly learned, attempting to make the relevant attribute classification tasks help each other. Specifically, we propose a joint learning approach, namely Aux-LSTM, which first learns a proper auxiliary representation between the related tasks and then leverages the auxiliary representation to integrate the learning process in both tasks. Empirical studies demonstrate the effectiveness of our proposed approach to joint learning on relevant user attributes.

1 Introduction

Social media, such as Twitter and Facebook, enables the users to post messages and share information in social networks, producing an unprecedented amount of user generated content (UGC) with rich facts about the users, including their personal attributes. Since then, UGC has been applied to various user attribute classification tasks, which recognizes user attributes, such as *gender* [Wang *et al.*, 2015a; Zhu *et al.*, 2015], *age* [Marquardt *et al.*, 2014] and *profession* [Tu *et al.*, 2015]. During the last few years, user attribute classification has drawn more and more attention due to its great potential influence to various applications, such as personality analysis, intelligent marketing and online advertising [O'Connor *et al.*, 2010; Volkova *et al.*, 2013; Preotiuc-Pietro *et al.*, 2015].

However, previous studies mainly focus on single attribute classification involving only one attribute, which ignores the relationship among various user attributes. Intuitively, the relationship among various user attributes may benefit different attribute classification tasks and should be considered. For instance, Figure 1 gives the true personal attributes of user **A** with an attached text. According to phrase "*writing C code*" in the message text, user **A** is very likely to have *profession* "*IT worker*", which is actually not true since his/her real *profession* attribute is "*Student*". However, if the *age* of user **A** is correctly classified to be "*19*" according to phrase "*19 years old*", we can easily adjust his/her *profession* attribute to be "*Student*" since a 19 year-old person is more likely to be a college student than an IT worker. Therefore, in some scenarios, a user's one attribute is helpful to infer his/her another attribute. Therefore, a feasible way to improve the performance of user attribute classification is to perform joint learning on relevant user attributes by capturing the relationship among various user attributes.

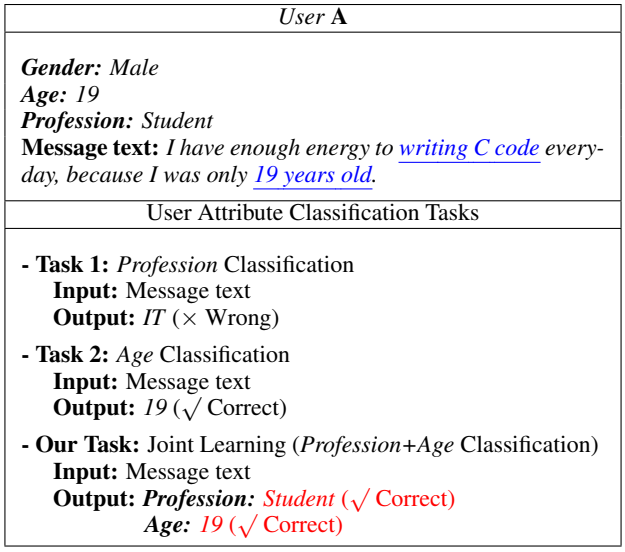

Figure 1: An example of joint learning on user attribute

In this paper, we address a novel scenario in user attribute classification, namely joint learning on relevant user attributes. Suppose there are two user attributes involving in our user attribute classification tasks, we first separate the twin user attribute classification task into a main task and an auxiliary task and then propose a joint learning approach to boost the performance of the main task with the help of the auxiliary task. In particular, our joint learning approach is based on a neural network architecture, namely

*Corresponding author

Aux-LSTM, which first learns an auxiliary representation from the auxiliary task with an auxiliary Long Short-Term Memory (LSTM) layer and then integrate the auxiliary representation into the main task for joint learning.

2 Related Work

In the last decade, many researches have devoted their efforts on user attribute classification (e.g., *profession*, *gender* and *age* classification) in several research communities, such as natural language processing and social network analysis. Related studies differ primarily in focusing on different styles of texts and extracting different types of features for one user attribute.

For the *gender* classification task, Schler *et al.,* [2006] exploit the differences in writing style and content between *male* and *female* bloggers to determine an unknown author's *gender* on the basis of a blog vocabulary. Mohammad and Yang [2013] show that there are marked differences across genders in how they use emotion words in work-place email. Ciot *et al.,* [2013] conduct the first assessment of latent attribute inference in various languages beyond English, focusing on *gender* inference of Twitter users. Li *et al.,* [2015] aim to identify the genders of two interactive users on the basis of micro-blog text. Some other studies, such as [Mukherjee and Liu, 2010], [Peersman *et al.*, 2011] and [Gianfortoni *et al.*, 2011], focus on exploring more effective features to improve the performance.

For the *age* classification task, most studies are devoted to explore efficient features in blog and social media. Schler *et al.,* [2006] focus on textual features extracted from the blog text, such as word context features and POS stylistic features. Peersman *et al.,* [2011] apply a text categorization approach to *age* classification with textual features extracted from the text in social media. More recently, Marquardt *et al.,* [2014] propose a multi-label classification approach to predict both the *gender* and *age* of authors from texts adopting some sentiment and emotion features.

For the *profession* classification task, there is less related studies. Tu *et al.,* [2015] extract the features from the verification descriptions in the users' personal information to achieve a significant performance improvement due to the fact that all the users in their collected data set are verified by officials of Sina Wiebo. However, most of users in social media are unverified and without any verification descriptions about them.

Different from all above studies, we focus on the classification task involving two or more user attributes. To the best of our knowledge, this is the first attempt to use the classification task of one user attribute to help the classification task of another user attribute.

3 Data Collection

We collect our data set from Tencent Micro-blog[1], which is one of the most popular SNS websites in China. From this website, we crawl each user's homepage containing user information (e.g. *name*, *profession*, *age*, *gender*) and the posted messages. The data collection process starts from some randomly selected users, and then crawl the data of their followers and friends. To get a more reliable data, we remove these users if they publish less than 50 messages. In total, we collect the user information and messages of 25000 users as our data set.

Gender Category	User Number
male	7236
female	8798

Table 1: User distribution of *male* and *female*

Profession Category	User Number
Student	2224
IT	1798
Government	1230
Finance	1087
Education	822
Services	743
Art	664

Table 2: User distribution of different *profession*

Table 1 shows the number of *male* and *female* users who have public *gender* information in the data set collected by us. We select these users as the data set for the *gender* classification task.

Table 2 shows the number of the users, whose *profession* information is available, in top-seven available *profession* categories accounting for the most proportions. We select the users from these seven *profession* categories as the data set for the *profession* classification task. We select the users from these seven *profession* categories as the data set for the *profession* classification task.

Figure 2 shows the *age* distribution of the users whose *age* attribute information can be obtained from their homepages in our data set. From this figure, we can see that most users are young and their *ages* distribute in the range of 18-28 years old. To enforce the task difficulty, we select the users whose *ages* range from 18 to 28 as the data set for the *age* classification task, totally 11 *age* categories.

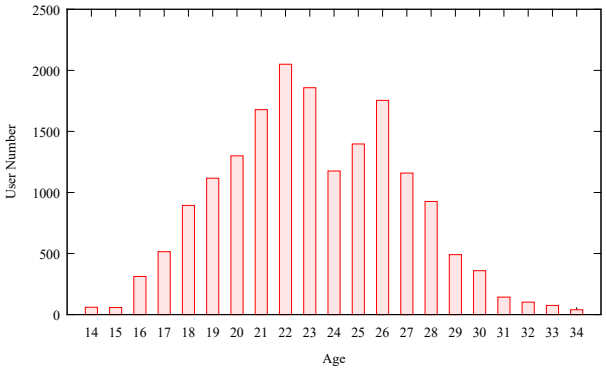

Figure 2: User distribution of different *ages*

[1]http://t.qq.com

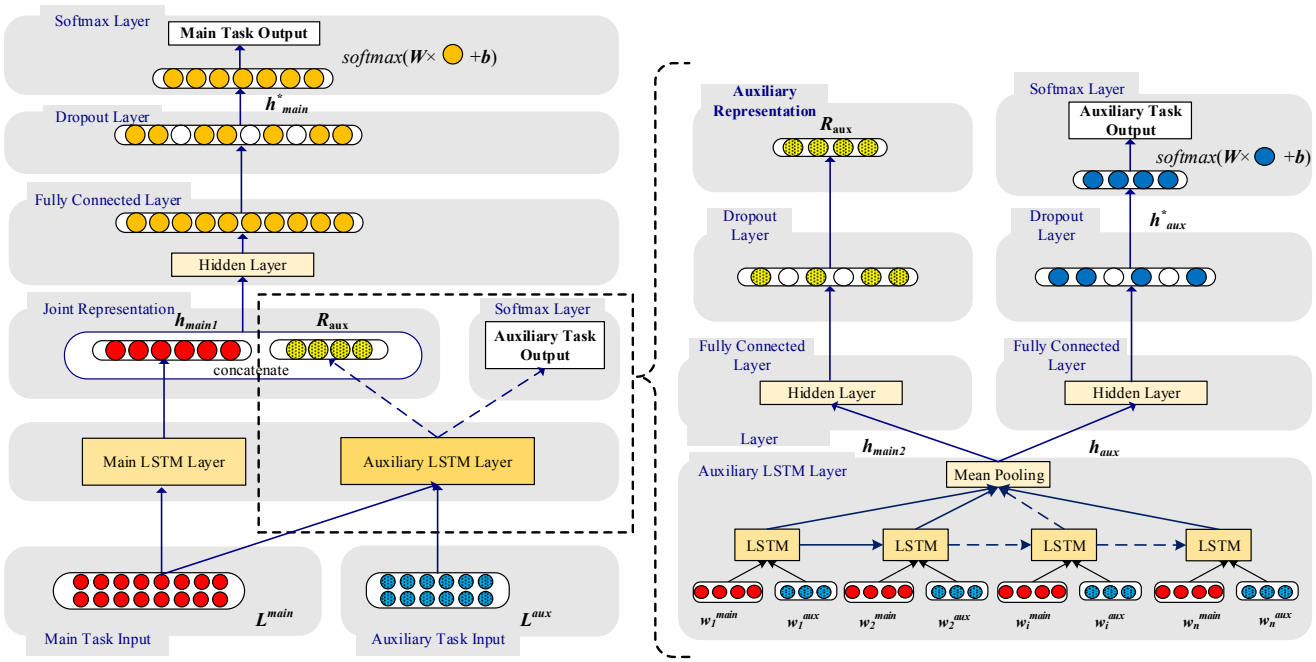

Figure 3: The overall architecture of Aux-LSTM model

4 Joint Learning on Relevant Attributes via Aux-LSTM

In this section, we describe our Aux-LSTM approach to joint learning on two user attributes in detail. The first subsection introduces the basic LSTM network. The second subsection introduces the LSTM model for user classification. Finally, we present the joint learning approach to user classification.

4.1 Basic LSTM Network

Long short-term memory network (LSTM) is proposed by [Hochreiter and Schmidhuber, 1997] to deal with gradient explosion or disappearance. LSTM maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary. In order to map the input sequence of main task to a fixed-sized vector, we adopt the standard LSTM layer used by [Graves, 2013], which consists of four components formalized through Equation (1) - (6), i.e., an input gate i_t, an output gate o_t, a forget gate f_t, and a memory cell c_t:

$$i_t = \sigma(W^{(i)}w_t + U^{(i)}h_{t-1} + b^{(i)}) \quad (1)$$

$$f_t = \sigma(W^{(f)}w_t + U^{(f)}h_{t-1} + b^{(f)}) \quad (2)$$

$$o_t = \sigma(W^{(o)}w_t + U^{(o)}h_{t-1} + b^{(o)}) \quad (3)$$

$$g_t = \tanh(W^{(g)}w_t + U^{(g)}h_{t-1} + b^{(g)}) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

Where σ denotes a sigmoid function, w_t is the current input at time step t and $\otimes$ denotes the elementwise multiplication. The candidate memory cell g_t is computed by Equations (4). c_t computed by Equations (5) defines the calculation in each memory cell at each time step t , and the output h_t computed by Equations (6) is the hidden state of LSTM units.

4.2 LSTM Model for User Classification

LSTM takes word embeddings as input. Each word in a text is represented by a real-valued row vector. Given a text with n words $T = \{w_1, w_2...w_n\}$, all the word vectors are stacked in a word embedding matrix $L \in \mathbb{R}^{d\times|V|}$, where d is the dimension of word embeddings, V is the vocabulary. Then, we feed L into the LSTM layer. Through the LSTM layer, the input of word embedding matrix is converted into a new representation h, i.e.,

$$h = \text{LSTM}(L) \quad (7)$$

After calculating the hidden vector of each step, we feed the representation h, which obtained by averaging the outputs of LSTM over all T time steps, into a fully-connected layer to obtain the representation h' as follows:

$$h' = dense(h) = \phi(\theta^T h + b) \quad (8)$$

Where ϕ is the "ReLU" activation function and $dense(\cdot)$ denotes the output of the fully-connected layer.

Subsequently, a dropout layer is used to obtain less interdependent network units and achieve better performance. The computing function is given as follows:

$$h^* = h' \cdot D(p^*) \quad (9)$$

Where D denotes the dropout operation and p^* denotes the dropout probability.

Finally, in user classification task, we feed h^* to a softmax layer to get the prediction probabilities p, i.e.,

$$p = \text{softmax}(Wh^* + b) \quad (10)$$

Where w is the weight vector to be learned and b is the bias term.

4.3 Joint Learning for User Classification

Figure 3 delineates the overall architecture of Aux-LSTM model which contains a main LSTM layer and an auxiliary LSTM layer. In our study, we consider one user attribute classification task as the main task and another user attribute classification classification task as the auxiliary task. The goal of the approach is to employ the auxiliary representation to assist the performance of the main task. The main idea of our Aux-LSTM approach lies in that the auxiliary LSTM layer is shared by both the main and auxiliary tasks so as to leverage the learning knowledge from two user attribute classification tasks.

(1) The Main Task:

Formally, the representation of the main task is generated from both the main LSTM layer and the auxiliary LSTM layer respectively:

$$h_{main1} = \text{LSTM}_{main}(L^{main}) \tag{11}$$

$$h_{main2} = \text{LSTM}_{aux}(L^{main}) \tag{12}$$

Where h_{main1} means the representation for the classification model via the main LSTM layer, while h_{main2} means the representation for the classification model via the auxiliary LSTM layer. Note that, the auxiliary LSTM layer is a shared LSTM layer bridging the two user attribute classification tasks.

Subsequently, we feed h_{main2} into a fully-connected layer followed by a dropout layer to get the **auxiliary representation** R_{aux}, i.e.,

$$R_{aux} = dense(h_{main2}) \cdot D(p^*) \tag{13}$$

We can obtain a novel representation after concatenating above two representation h_{main1} and R_{aux} and use them as the input of a fully-connected layer followed by a dropout layer in the main task:

$$h^*_{main} = dense(h_{main1} \oplus R_{aux}) \cdot D(p^*) \tag{14}$$

Where h^*_{main} denotes the output of the dropout layer in the main task and $\oplus$ denotes the concatenate operator.

(2) The Auxiliary Task:

The representation of the auxiliary task is generated from the auxiliary LSTM layer:

$$h_{aux} = \text{LSTM}_{aux}(L^{aux}) \tag{15}$$

Subsequently, a fully-connected layer followed by a dropout layer is utilized to obtain a feature vector for classification:

$$h^*_{aux} = dense(h_{aux}) \cdot D(p^*) \tag{16}$$

Where h^*_{aux} denotes the output of the dropout layer in the auxiliary task.

(3) Joint Learning:

Once we obtain the representation of the main task and the auxiliary task, i.e., h^*_{main} and h^*_{aux}, we feed them into the softmax layers to predict the probability of label y given the inputs L as follows:

$$\widehat{p}_\theta(y^{main}|L^{main}) = \text{softmax}(Wh^*_{main} + b) \tag{17}$$

$$\widehat{p}_\theta(y^{aux}|L^{aux}) = \text{softmax}(Wh^*_{aux} + b) \tag{18}$$

Where $\widehat{p}_\theta(y^{main}|L^{main})$ is the output of the main task and $\widehat{p}_\theta(y^{aux}|L^{aux})$ is the output of the auxiliary task.

To learn the parameters of our Aux-LSTM model, we define our joint cost function as a weighted linear combination of the cost functions of both the main and auxiliary tasks, i.e.,

$$\begin{aligned} J(\theta) = &-\lambda \cdot \sum_{i=1}^{N}\sum_{j=1}^{C} y_j^{main} \cdot \log \widehat{p}_\theta(y_j^{main}|L_i^{main}) \\ &-(1-\lambda)\sum_{i=1}^{N}\sum_{j=1}^{C} y_j^{aux} \cdot \log \widehat{p}_\theta(y_j^{aux}|L_i^{aux}) + \frac{l}{2}||\theta||_2^2 \end{aligned} \tag{19}$$

Where the y_j^{main} and y_j^{aux} are the ground-truth label; N is the number of training samples; C is the category number and l is a L_2 regularization to bias parameters.

In the above equation, λ is the weight parameter between the main task and auxiliary task. In our Aux-LSTM model, λ is set to be 0.75 in order to reduce the influence of noisy information from auxiliary task. Besides, the model parameters are optimized by using Adagrad [Duchi *et al.*, 2011]. All the matrix and vector parameters are initialized with uniform distribution in $[-\sqrt{6/(r+c)}, \sqrt{6/(r+c)}]$, where r and c are the numbers of rows and columns in the matrices [Glorot and Bengio, 2010]. In order to avoid over-fitting, the dropout strategy is used in both the LSTM layer and auxiliary LSTM layer.

5 Experimentation

5.1 Experimental Settings

- **Data Settings:** The users are collected from Tencent Micro-blog, one of the most popular SNS websites in China. For each kind of user attribute (i.e., *profession*, *gender* and *age*) classification task, we randomly split the users into a training set (80% users) and a test set (20% users). We also set aside 10% users from the training as the validation data which is used to tune learning algorithm parameters.
- **Word Embeddings:** We use word2vec[2] (Skip-gram model is used) to pre-train word embeddings using our crawled data set containing 25000 users and word embedding matrices are not updated during model training. The dimensionality of word vector is set to be 200. The window size is set as 5.
- **Hyper-parameters:** The hyper-parameter values in the LSTM model are tuned according to performances in the development data.
- **Evaluation Metrics and Significance test:** The performance is evaluated using Macro-F1 (F), which is calculates as $F = \frac{2PR}{P+R}$, where the overall precision P and recall R are averaged on the precision/recall scores from all categories. Furthermore, t-test is used to evaluate the significance of the performance difference between two approaches [Yang and Liu, 1999].

[2]http://word2vec.googlecode.com/

Method	*Profession*			*Age*			*Gender*		
	P	R	F	P	R	F	P	R	F
SVM	0.395	0.262	0.315	0.305	0.219	0.255	0.825	0.903	0.862
ME	0.402	0.270	0.323	0.297	0.236	0.263	0.811	0.902	0.854
CNN	0.402	0.273	0.325	0.306	0.234	0.265	0.827	0.904	0.864
PCNN	0.405	0.281	0.332	0.310	0.242	0.273	0.831	0.902	0.865
LSTM	0.405	0.291	**0.339**	0.312	0.254	**0.280**	0.837	0.904	**0.869**

Table 3: Experimental results of five approaches on individual classification for each user attribute

5.2 Experimental Results on Individual Classification

For thorough comparison, we implement several kinds of classifiers for individual classification for each user attribute:

- **SVM:** SVM classifier with four types of textual features including word unigram and two kinds of complex features, i.e., F-measure, POS-pattern. These kinds of textual features yield the state-of-the-art performance for user classification proposed by [Mukherjee and Liu, 2010].
- **ME:** ME classifier with the same features settings as SVM. This is exactly the approach proposed by [Li *et al.*, 2015].
- **CNN:** Basic bow-CNN[3] proposed in [Johnson and Zhang, 2015]. Each user is represented by a bag of features consisting of above four types of textual features.
- **PCNN:** CNN with two or more convolution layers in parallel so as to complement each other to improve model performance (i.e., the extension of bow-CNN), proposed in [Johnson and Zhang, 2015].
- **LSTM:** The standard LSTM model including a LSTM layer, a fully connected layer and a dropout layer. The representation model is **Word Embeddings** described in section 5.1.

Table 3 shows the results of five approaches to three user attribute classification tasks respectively. From the table, we can see that, **SVM** and **ME** yield similar results, while **CNN** and **PCNN** achieve better performances than **SVM** and **ME**. This result implies that deep learning approaches are more appropriate for the task of user attribute classification. Among the five approaches, **LSTM** performs best in all three tasks. The success might due to the fact that LSTM model can capture sequence information in the context and be more effective in learning representations with a flexible compositional structure [Wang *et al.*, 2015b]. Therefore, it is a good choice to pick **LSTM** as the basic classification algorithm in our joint learning approach. Experimental results show that our LSTM method outperforms the other four approaches.

5.3 Experimental Results on Joint Learning

For thorough comparison, we implement three approaches to joint learning on two user attributes:

- **Meta-Learning:** This approach is a two-stage classification algorithm. In the first stage, we train a **SVM** classifier on the auxiliary task and obtain the posterior probabilities of the training samples in the main task. The posterior probabilities are prepared as meta-features for the classifier in the second stage. In the second stage, we train another **SVM** classifier on the main task with the extra meta-features obtained in the first stage. This approach is a straightforward strategy to use the information of the other user attribute.
- **Multi-task LSTM:** This baseline is inspired by the DNN-based multi-task learning framework with shared word representations proposed by [Collobert *et al.*, 2011]. Two LSTM-based sequence embedding model are jointly trained with their specific word embeddings and a set of shared word embeddings. All the embeddings are initialized with pre-trained word2vec embeddings, and are dynamically updated during model training. Besides, unlike the input of **LSTM**, the specific word embeddings are pre-trained separately on the data set of each user attribute classification task, while the shared embeddings are pre-trained on the entire data set containing 25000 users.
- **Aux-LSTM:** This is our approach which performs joint learning on two user attribute classification tasks. The representation model is **Word Embeddings**.

(a) Main Task: *Gender* Classification

In this section, we report the classification results of different approaches where the main task is *gender* classification and the auxiliary task is *age* or *profession* classification. Table 4 shows the results of different approaches to *gender* classification where **LSTM** is an individual classification approach while the other approaches are all joint learning approaches.

From Table 4, we can see that:

(1): Joint learning on *gender* and *age* classification is not helpful, no matter what joint learning approaches are applied. The failure of the joint learning approaches is due to the fact that there might be not much correlation between user's *age* and *gender* information. Fortunately, our approach yields a performance loss by a very small margin (from 0.869 to 0.867).

Method	*Gender*		
	P	R	F
LSTM	0.837	0.904	0.869
Meta-Learning (*Gender+Age*)	0.820	0.889	0.853
Multi-task LSTM (*Gender+Age*)	0.822	0.891	0.855
Aux-LSTM (*Gender+Age*)	0.834	0.903	0.867
Meta-Learning (*Gender+Prof*)	0.837	0.908	0.871
Multi-task LSTM (*Gender+Prof*)	0.842	0.911	0.875
Aux-LSTM (*Gender+Prof*)	**0.885**	**0.922**	**0.903**

Table 4: Experimental results of joint learning for *gender* classification

[3]http://riejohnson.com/cnn_download.html

Method	*Age*		
	P	R	F
LSTM	0.312	0.254	0.280
Meta-Learning (*Age+Gender*)	0.297	0.239	0.265
Multi-task LSTM (*Age+Gender*)	0.301	0.243	0.269
Aux-LSTM (*Age+Gender*)	0.310	0.254	0.279
Meta-Learning (*Age+Prof*)	0.314	0.261	0.285
Multi-task LSTM (*Age+Prof*)	0.320	0.269	0.292
Aux-LSTM (*Age+Prof*)	**0.335**	**0.295**	**0.314**

Table 5: Experimental results of joint learning for *age* classification

(2): Joint learning on *gender* and *profession* classification is beneficial for performance improvement, no matter what joint learning approaches are applied. Specifically, our **Aux-LSTM (*gender+prof*)** approach achieves a 3.4% promotion on Macro-F1 (F) compared to individual **LSTM** approach. Our approach also performs better than **Meta-Learning (*gender+prof*)** and **Multi-task LSTM (*gender+prof*)**. Significance test shows that the improvement of our approach over the other two joint learning approaches is significant ($p - value < 0.05$).

(b) Main Task: *Age* Classification

In this section, we report the classification results of different approaches where the main task is *age* classification and the auxiliary task is *gender* or *profession* classification. Table 5 shows the results of different approaches to *age* classification.

From Table 5, we can see that:

(1): Similar to the results in the previous experiments, joint learning on *age* and *gender* is also not helpful, no matter what joint learning approaches are applied. Fortunately, our approach still yields a performance loss by a very small margin (from 0.280 to 0.279).

(2): Joint learning on *age* and *profession* classification is beneficial for performance improvement, no matter what joint learning approaches are applied. Specifically, our **Aux-LSTM (*age+prof*)** approach achieves a 3.4% promotion on Macro-F1 (F) compared to individual **LSTM** approach. Our approach also performs better than **Meta-Learning (*age+prof*)** and **Multi-task LSTM (*age+prof*)**. Significance test shows that the improvement of our approach over the other two joint learning approaches is significant ($p - value < 0.05$).

(c) Main Task: *Profession* Classification

In this section, we report the classification results of different approaches where the main task is *profession* classification and the auxiliary task is *age* or *gender* classification. Table 6 shows the results of different approaches to *profession* classification.

From Table 6, we can see that:

(1): Joint learning on *profession* and *age* classification is beneficial for performance improvement, no matter what joint learning approaches are applied. Specifically, our **Aux-LSTM (*prof+age*)** approach achieves a 3.4% promotion on Macro-F1 (F) compared to individual LSTM approach. Our approach also performs better than **Meta-Learning (*prof+age*)** and **Multi-task LSTM (*prof+age*)**. Significance test shows that the improvement of our approach over the other two joint learning approaches is significant ($p - value < 0.05$).

Method	*Profession*		
	P	R	F
LSTM	0.405	0.291	0.339
Meta-Learning (*Prof+Age*)	0.410	0.293	0.342
Multi-task LSTM (*Prof+Age*)	0.415	0.303	0.350
Aux-LSTM (*Prof+Age*)	**0.430**	**0.329**	**0.373**
Meta-Learning (*Prof+Gender*)	0.413	0.302	0.349
Multi-task LSTM (*Prof+Gender*)	0.423	0.309	0.357
Aux-LSTM (*Prof+Gender*)	**0.450**	**0.329**	**0.380**

Table 6: Experimental results of joint learning for *profession* classification

(2): Joint learning on *profession* and *gender* classification is beneficial for performance improvement, no matter what joint learning approaches are applied. Specifically, our **Aux-LSTM (*prof+gender*)** approach achieves a 4.1% promotion on Macro-F1 (F) respectively compared to individual **LSTM** approach. Our approach still performs better than **Meta-Learning (*prof+gender*)** and **Multi-task LSTM (*prof+gender*)**, which certifies the stronger capability of our Aux-LSTM model on leveraging sharing information across two relevant tasks. Significance test shows that the improvement of our approach over the other two joint learning approaches is significant ($p - value < 0.05$).

Overall, our experimental results show that the two user attributes of *gender* and *age* are not related and joint learning on them is not helpful. The user attribute *profession* is related to both *age* and *gender*, and joint learning on them is consistently effective for performance improvement.

6 Conclusion

In this paper, we propose a joint learning approach, namely Aux-LSTM, to perform the task of user attribute classification when relevant user attributes exist. Our approach well incorporates the relationship among the relevant user attributes. Specifically, we employ an auxiliary LSTM layer to learn the auxiliary representation for the main user attribute classification task. Experiments on three different user attribute classification tasks demonstrate that our proposed method significantly boosts the performance of the main task with the help of the auxiliary representation when the two user attributes are related.

In our future work, we would like to extend our proposed Aux-LSTM model to make it be capable of learning the auxiliary representation across three or more tasks. Moreover, we will make our efforts to incorporate more user attributes, e.g., locations, to perform joint learning.

Acknowledgments

This research work has been partially supported by three NSFC grants, No.61331011, No.61672366 and No.61375073.

References

[Ciot *et al.*, 2013] Morgane Ciot, Morgan Sonderegger, and Derek Ruths. Gender inference of twitter users in non-

english contexts. In *Proceedings of EMNLP-2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA*, pages 1136–1145, 2013.

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

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[Gianfortoni *et al.*, 2011] Philip Gianfortoni, David Adamson, and Carolyn P Rosé. Modeling of stylistic variation in social media with stretchy patterns. In *Proceedings of EMNLP-2011*, pages 49–59. Association for Computational Linguistics, 2011.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS-2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 249–256, 2010.

[Graves, 2013] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

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

[Johnson and Zhang, 2015] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of NAACL-2015, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 103–112, 2015.

[Li *et al.*, 2015] Shoushan Li, Jingjing Wang, Guodong Zhou, and Hanxiao Shi. Interactive gender inference with integer linear programming. In *Proceedings of IJCAI-2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2341–2347, 2015.

[Marquardt *et al.*, 2014] James Marquardt, Golnoosh Farnadi, Gayathri Vasudevan, Marie-Francine Moens, Sergio Davalos, Ankur Teredesai, and Martine De Cock. Age and gender identification in social media. In *Proceedings of CLEF-2014, Sheffield, UK, September 15-18, 2014.*, pages 1129–1136, 2014.

[Mohammad and Yang, 2013] Saif Mohammad and Tony Yang. Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of ACL-HLT-2011, Portland, Oregon, USA, 24 June, 2011*, pages 70–79, 2013.

[Mukherjee and Liu, 2010] Arjun Mukherjee and Bing Liu. Improving gender classification of blog authors. In *Proceedings of EMNLP-2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA*, pages 207–217, 2010.

[O'Connor *et al.*, 2010] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of ICWSM-2010, Washington, DC, USA, May 23-26, 2010*, 2010.

[Peersman *et al.*, 2011] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In *Proceedings of CIKM-2011, Glasgow, United Kingdom, October 28, 2011*, pages 37–44, 2011.

[Preotiuc-Pietro *et al.*, 2015] Daniel Preotiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. In *Proceedings of ACL-2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1754–1764, 2015.

[Schler *et al.*, 2006] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. Effects of age and gender on blogging. In *Proceedings of AAAI-2006, Stanford, California, USA, March 27-29, 2006*, pages 199–205, 2006.

[Tu *et al.*, 2015] Cunchao Tu, Zhiyuan Liu, and Maosong Sun. Prism: Profession identification in social media with personal information and community structure. In *Chinese National Conference on Social Media Processing*, pages 15–27. Springer, 2015.

[Volkova *et al.*, 2013] Svitlana Volkova, Theresa Wilson, and David Yarowsky. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of EMNLP-2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA*, pages 1815–1827, 2013.

[Wang *et al.*, 2015a] Jingjing Wang, Yunxia Xue, Shoushan Li, and Guodong Zhou. Leveraging interactive knowledge and unlabeled data in gender classification with co-training. In *Proceedings of DASFAA-2015 International Workshops, SeCoP, BDMS, and Posters, Hanoi, Vietnam, April 20-23, 2015, Revised Selected Papers*, pages 246–251, 2015.

[Wang *et al.*, 2015b] Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of ACL-2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1343–1353, 2015.

[Yang and Liu, 1999] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-1999, August 15-19, 1999, Berkeley, CA, USA*, pages 42–49, 1999.

[Zhu *et al.*, 2015] Zhu Zhu, Jingjing Wang, Shoushan Li, and Guodong Zhou. Interactive gender inference in social media. In *Database Systems for Advanced Applications - DASFAA 2015 International Workshops, SeCoP, BDMS, and Posters, Hanoi, Vietnam, April 20-23, 2015, Revised Selected Papers*, pages 252–258, 2015.