

Switch-LSTMs for Multi-Criteria Chinese Word Segmentation

Jingjing Gong,* Xinchi Chen,* Tao Gui, Xipeng Qiu†

Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
Shanghai Institute of Intelligent Electronics and Systems
{jjgong, xinchichen13, tgui16, xpqiu}@fudan.edu.cn

Abstract

Multi-criteria Chinese word segmentation is a promising but challenging task, which exploits several different segmentation criteria and mines their common underlying knowledge. In this paper, we propose a flexible multi-criteria learning for Chinese word segmentation. Usually, a segmentation criterion could be decomposed into multiple sub-criteria, which are shareable with other segmentation criteria. The process of word segmentation is a routing among these sub-criteria. From this perspective, we present Switch-LSTMs to segment words, which consist of several long short-term memory neural networks (LSTM), and a switcher to automatically switch the routing among these LSTMs. With these auto-switched LSTMs, our model provides a more flexible solution for multi-criteria CWS, which is also easy to transfer the learned knowledge to new criteria. Experiments show that our model obtains significant improvements on eight corpora with heterogeneous segmentation criteria, compared to the previous method and single-criterion learning.

Introduction

Chinese word segmentation (CWS) is a preliminary and important task for Chinese natural language processing (NLP). Currently, the state-of-the-art CWS methods are based on supervised machine learning algorithms and greatly rely on a large-scale annotated corpus whose cost is extremely expensive. Although several CWS corpora have been built with great efforts, their segmentation criteria are different. Since these segmentation criteria are from different linguistic perspectives, it remains to be a challenging problem on how to effectively utilize these resources.

Most existing methods focus on improving the performance of each single segmentation criterion. Recently, deep learning methods have been widely used in segmenting Chinese words and can effectively reduce the efforts of feature engineering (Zheng, Chen, and Xu 2013; Pei, Ge, and Baobao 2014; Chen et al. 2015a; 2015b; Cai and Zhao 2016; Zhang, Zhang, and Fu 2016; Yao and Huang 2016; Gong et al. 2017). However, it is a waste of resources if we fail to fully exploit all the corpora with different criteria.

*J. Gong and X. Chen contributed equally to this paper.

†Corresponding author.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Corpora	Lin	Dan	won	the championship	
CTB	林丹		拿了	总冠军	
PKU	林	丹	拿了	总	冠军
MSRA	林丹		拿了	总	冠军

Table 1: Illustration of the different segmentation criteria, which partially share some underlying sub-criteria.

Chen et al. (2017) proposed a multi-criteria learning framework for CWS. Specifically, they regard each segmentation criterion as a single task under the framework of multi-task learning (Caruana 1997), where a shared layer is used to extract the criteria-invariant features, and a private layer is used to extract the criteria-specific features. However, it is unnecessary to use a specific private layer for each criterion. These different criteria often have partial overlaps. As shown in Table 1, given a sentence “林丹拿了总冠军 (Lin Dan won the championship)”, the three commonly-used corpora, PKU’s People’s Daily (PKU) (Yu et al. 2001), Penn Chinese Treebank (CTB) (Fei 2000) and MSRA (Emerson 2005), use different segmentation criteria. The segmentation of “林丹 (Lin Dan)” is the same as in PKU and MSRA criteria, and the segmentation of “总|冠军 (the championship)” is the same as in CTB and MSRA criteria. All these three criteria have same segmentation for the word “拿了 (won)”.

Inspired by the above example, we propose a more flexible model for multi-criteria CWS. Each segmentation criterion can be separated into several sub-criteria, each of which is implemented by a long short-term memory neural network (LSTM) (Hochreiter and Schmidhuber 1997). When segmenting a sentence, these sub-criteria are automatically switched for different words. Therefore, specific segmentation criteria can be regarded as a combination of these sub-criteria.

Specifically, our model consists of several LSTM cells, each of which represents a segmentation sub-criterion, and a controller, called switcher, to switch the routing among the different LSTMs. These sub-criteria could be shared among different segmentation criteria.

The major merits of our proposed approach are simplicity and flexibility. Existing work on multi-task learning usually has private layers for each task, parameter, and scale of the

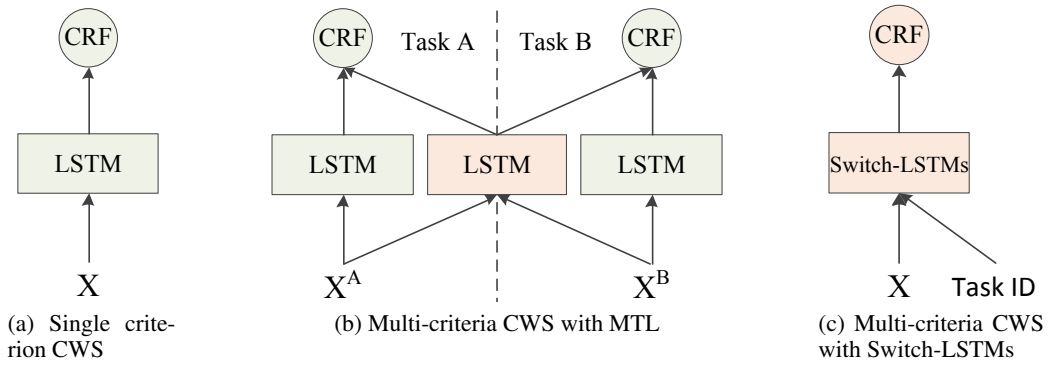


Figure 1: Architectures of single-criterion and multi-criteria Chinese word segmentation. Parameters of components in orange are shared, and those in green are private.

computational graph will linearly grow along with the number of tasks. Also, private layers will reduce the flexibility of the model, for example when a new task comes in, approaches with private layers will need to stop training and reconstruct the whole graph. While in our proposed method, all layers can be considered as the shared layer, which will perfectly avoid the growing number of parameter and graph scale. And more interestingly, we can pre-allocate many preserved task embeddings with negligible overhead. With extra task embeddings, we can effortlessly hot-plug a new task into the in-training model.

Finally, we experiment on eight Chinese word segmentation datasets. Experiments show that our models are effective to improve the performance of CWS.

Background

Single Criterion CWS with LSTM-CRF Framework

Chinese word segmentation task could be viewed as a character based sequence labeling problem (Zheng, Chen, and Xu 2013; Pei, Ge, and Baobao 2014; Chen et al. 2015a; 2015b). Specifically, each character in a sentence $X = \{x_1, \dots, x_n\}$ is labelled as one of $\mathcal{L} = \{B, M, E, S\}$, indicating the begin, middle, end of a word, or a word with single character. The aim of CWS task is to figure out the ground truth of labels $Y^* = \{y_1^*, \dots, y_n^*\}$:

$$Y^* = \arg \max_{Y \in \mathcal{L}^n} p(Y|X). \quad (1)$$

The popular architecture of neural CWS could be characterized by three components: (1) a character embedding layer; (2) feature extraction layers consisting of several classical neural networks and (3) a CRF tag inference layer. Figure 1a illustrates the general architecture of CWS.

Embedding layer In neural models, the first step is to map discrete language symbols into distributed embedding space. Formally, each character x_t is mapped as $\mathbf{e}_{x_t} \in \mathbb{R}^{d_e} \subset \mathbf{E}$, where d_e is a hyper-parameter indicating the size of character embedding, and \mathbf{E} is the embedding matrix.

Encoding layers Usually a bi-directional LSTM (Bi-LSTMs) (Hochreiter and Schmidhuber 1997) is a prevalent choice for the encoding layer. It could incorporate information from both sides of sequence

$$\mathbf{h}_t = \text{Bi-LSTMs}(\mathbf{e}_{x_t}, \vec{\mathbf{h}}_{t-1}, \overleftarrow{\mathbf{h}}_{t+1}, \theta), \quad (2)$$

where $\vec{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ are the hidden states at step t of the forward and backward LSTMs respectively. θ denotes all the parameters in Bi-LSTMs layer.

Inference Layer The extracted features are then sent to conditional random fields (CRF) (Lafferty, McCallum, and Pereira 2001) layer for tag inference. In CRF layer, $p(Y|X)$ in Eq (1) could be formalized as:

$$p(Y|X) = \frac{\Psi(Y|X)}{\sum_{Y' \in \mathcal{L}^n} \Psi(Y'|X)}. \quad (3)$$

Here, $\Psi(Y|X)$ is the potential function. In first order linear chain CRFs, we have:

$$\Psi(Y|X) = \prod_{t=2}^n \psi(X, t, y_{t-1}, y_t), \quad (4)$$

$$\psi(\mathbf{x}, t, y', y) = \exp(\delta(X, t)_y + \mathbf{b}_{y'y}), \quad (5)$$

where $\mathbf{b}_{y'y} \in \mathbf{R}$ is trainable parameters respective to label pair (y', y) . Score function $\delta(X, t) \in \mathbb{R}^{|\mathcal{L}|}$ calculates scores of each label for tagging the t -th character:

$$\delta(X, t) = \mathbf{W}_\delta^\top \mathbf{h}_t + \mathbf{b}_\delta, \quad (6)$$

where \mathbf{h}_t is the hidden state of Bi-LSTMs at step t ; $\mathbf{W}_\delta \in \mathbb{R}^{d_h \times |\mathcal{L}|}$ and $\mathbf{b}_\delta \in \mathbb{R}^{|\mathcal{L}|}$ are trainable parameters.

Multi-Criteria CWS using Multi-Task Learning

Annotations in Chinese word segmentation are valuable and expensive. Chen et al. (2017) tried to train on Chinese word segmentation annotations in multiple heterogeneous criteria to improve the performance. The multi-task learning framework is a suitable way to exploit the shared information among these different criteria as shown in Figure 1b.

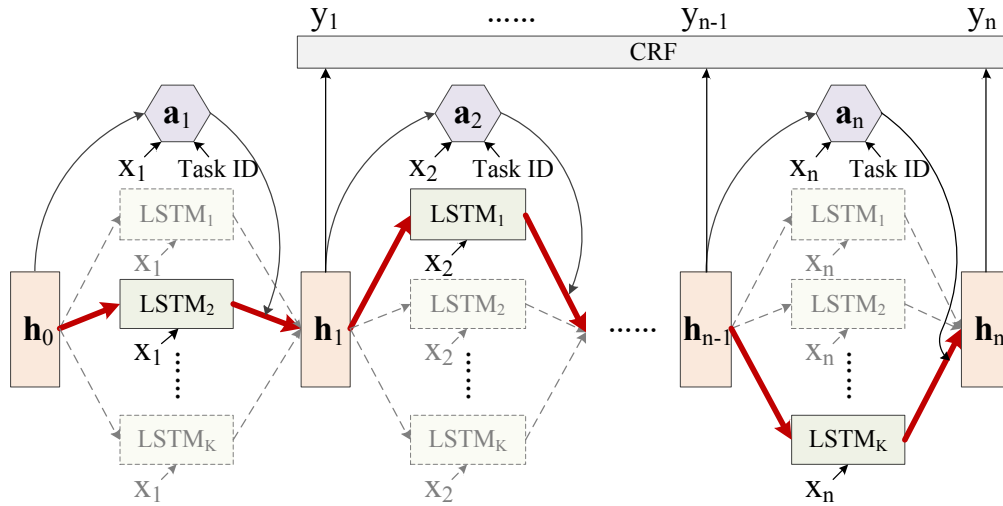


Figure 2: Switch-LSTMs for multi-criteria Chinese word segmentation. \mathbf{a}_t denote switches taking \mathbf{h}_{t-1} , x_t and task ID m as inputs, and output distributions over k -way LSTMs.

Formally, assuming that there are M corpora with heterogeneous segmentation criteria, we refer \mathcal{D}_m as corpus m with N_m samples:

$$\mathcal{D}_m = \{(X_i^{(m)}, Y_i^{(m)})\}_{i=1}^{N_m}, \quad (7)$$

where X_i^m and Y_i^m denote the i -th sentence and the corresponding label in corpus m respectively.

To exploit information across multiple datasets, the feature layer additionally introduce an shared Bi-LSTMs, together with the original private Bi-LSTMs as in Eq (2). Concretely, for corpus m , the hidden states of shared layer and private layer are calculated as:

$$\mathbf{h}_t^{(s)} = \text{Bi-LSTMs}(\mathbf{e}_{x_t}, \overrightarrow{\mathbf{h}}_{t-1}^{(s)}, \overleftarrow{\mathbf{h}}_{t+1}^{(s)}; \theta_h^{(s)}), \quad (8)$$

$$\mathbf{h}_t^{(m)} = \text{Bi-LSTMs}(\mathbf{e}_{x_t}, \overrightarrow{\mathbf{h}}_{t-1}^{(m)}, \overleftarrow{\mathbf{h}}_{t+1}^{(m)}; \theta_h^{(m)}). \quad (9)$$

The score function $\delta(X, t)$ in Eq (6) is replaced by:

$$\delta(X, t) = g(\mathbf{h}_t^{(s)}, \mathbf{h}_t^{(m)}; \theta_g^{(m)}), \quad (10)$$

where g is a feed-forward neural network, taking shared and private features as inputs.

Objective The objective is to maximize the log likelihood of true labels on all the corpora:

$$\mathcal{J}_{seg}(\Theta^m, \Theta^s) = \sum_{m=1}^M \sum_{i=1}^{N_m} \log p(Y_i^{(m)} | X_i^{(m)}; \Theta^m, \Theta^s), \quad (11)$$

where $\Theta^m = \{\theta_h^{(m)}, \theta_g^{(m)}\}$ and $\Theta^s = \{\mathbf{E}, \theta_h^{(s)}\}$ denote all the private and shared parameters respectively.

Switch-LSTMs for Multi-Criteria CWS

Multi-task learning framework separates its parameters into private ones and shared ones. There are two main drawbacks. First, the model architecture should be manually designed. Second, it is very difficult to generalize to a new tiny

dataset. Unlike the multi-task learning framework, switch LSTM doesn't have any private parameters as shown in Figure 1c. Figure 2 gives the architecture of the proposed switch-LSTMs.

Switch-LSTMs

Switch-LSTMs consist of K independent LSTM cells and a switcher. At each time step t , switch-LSTMs will switch to one of K LSTM cells according to the switcher state $\mathbf{a}_t \in \mathbb{R}^K$. Thus, the switch-LSTMs could be formalized as:

$$\mathbf{s}_{t,k} = \text{LSTM}(\mathbf{e}_{x_t}, \mathbf{h}_{t-1}; \theta_k^{(s)}), \forall k \in [1, K] \quad (12)$$

$$\mathbf{a}_{t,k} = \text{Switch}(\mathbf{e}_{x_t}, \mathbf{s}_{t,k}, \mathbf{e}_m), \forall k \in [1, K] \\ \triangleq \text{softmax}(\mathbf{W}[\mathbf{e}_{x_t}, \mathbf{s}_{t,k}, \mathbf{e}_m]), \quad (13)$$

$$\mathbf{h}_t = \sum_{k=1}^K \mathbf{a}_{t,k} \mathbf{s}_{t,k}, \quad (14)$$

where $\theta^{(s)} = \{\theta_1^{(s)}, \theta_2^{(s)}, \dots, \theta_K^{(s)}\}$ denotes the parameter of the corresponding LSTM. Since the switcher picks layers according to the task property as well, we introduce a task embedding $\mathbf{e}_m \in \mathbf{E}^m$ for task ID m .

We abbreviate the above equations of switch-LSTMs to

$$\mathbf{h}_t = \text{Switch-LSTMs}(\mathbf{e}_{x_t}, \mathbf{h}_{t-1}; \theta^{(s)}). \quad (15)$$

Similar to Bi-LSTMs, we use bi-directional switch LSTMs (Bi-Switch-LSTMs) for multi-criteria Chinese word segmentation. The feature \mathbf{h}_t extracted by Bi-Switch-LSTMs could be formalized as:

$$\mathbf{h}_t = \text{Bi-Switch-LSTMs}(\mathbf{e}_{x_t}, \overrightarrow{\mathbf{h}}_{t-1}, \overleftarrow{\mathbf{h}}_{t+1}), \quad (16)$$

where $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ are forward and backward adaptive LSTMs respectively as in Eq (15). Thus, the forward and backward switch gate statuses are $\overrightarrow{\mathbf{A}} = \{\overrightarrow{\mathbf{a}}_1, \dots, \overrightarrow{\mathbf{a}}_n\}$ and $\overleftarrow{\mathbf{A}} = \{\overleftarrow{\mathbf{a}}_1, \dots, \overleftarrow{\mathbf{a}}_n\}$ respectively.

Corpora			# of Tokens	# of Chars	Dict Size	Char Types	# of Sents	OOV Rate
Sighan05	MSRA	Train	2.4M	4.1M	88.1K	5.2K	86.9K	-
		Test	0.1M	0.2M	12.9K	2.8K	4.0K	2.60%
	AS	Train	5.4M	8.4M	141.3K	6.1K	709.0K	-
		Test	0.1M	0.2M	18.8K	3.7K	14.4K	4.30%
Sighan08	PKU	Train	1.1M	1.8M	55.2K	4.7K	47.3K	-
		Test	0.2M	0.3M	17.6K	3.4K	6.4K	3.33%
	CTB	Train	0.6M	1.1M	42.2K	4.2K	23.4K	-
		Test	0.1M	0.1M	9.8K	2.6K	2.1K	5.55%
	CKIP	Train	0.7M	1.1M	48.1K	4.7K	94.2K	-
		Test	0.1M	0.1M	15.3K	3.5K	10.9K	7.41%
	CITYU	Train	1.1M	1.8M	43.6K	4.4K	36.2K	-
		Test	0.2M	0.3M	17.8K	3.4K	6.7K	8.23%
	NCC	Train	0.5M	0.8M	45.2K	5.0K	18.9K	-
		Test	0.1M	0.2M	17.5K	3.6K	3.6K	4.74%
	SXU	Train	0.5M	0.9M	32.5K	4.2K	17.1K	-
		Test	0.1M	0.2M	12.4K	2.8K	3.7K	5.12%

Table 2: Details of the eight datasets.

The score function $\delta(X, t)$ in Eq (6) is replaced by:

$$\delta(X, t) = \sum_{i=1}^K \vec{\mathbf{a}}_{t,i} \vec{\mathbf{W}}_i \vec{\mathbf{h}}_t + \sum_{i=1}^K \overleftarrow{\mathbf{a}}_{t,i} \overleftarrow{\mathbf{W}}_i \overleftarrow{\mathbf{h}}_t, \quad (17)$$

where $+$ denotes the element-wise addition operation, $\overleftarrow{\mathbf{W}}_i$ and $\vec{\mathbf{W}}_i$ are parameters for i th LSTM cells in *left* and *right* Switch-LSTMs respectively.

Objective for Multi-task Learning

The objective is to maximize the log likelihood $\mathcal{J}_{seg}(\Theta)$:

$$\mathcal{J}_{seg}(\Theta) = \sum_{m=1}^M \sum_{i=1}^{N_m} \log p(Y_i^{(m)} | X_i^{(m)}, m; \Theta), \quad (18)$$

where $\Theta = \{\mathbf{E}, \theta^{(s)}, \mathbf{E}^m, \mathbf{W}, \vec{\mathbf{W}}, \overleftarrow{\mathbf{W}}\}$ are all shared trainable parameters.

Experiments

Datasets

We experiment on eight CWS datasets from SIGHAN2005 (Emerson 2005) and SIGHAN2008 (Jin and Chen 2008). Table 2 gives the details of the eight datasets. AS, CITYU and CKIP are in traditional Chinese. MSRA, PKU, CTB, NCC and SXU are in simplified Chinese. We randomly pick 10% instances from training set as the development set for all the datasets.

Experimental Configuration

The character embedding size d_e is set to 100, task embedding size is set to 20, the hidden size d_h for our proposed Switch-LSTMs are set to 100, the number of choices K in K -way switch is set to one of $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. As a common approach to alleviate overfitting, we dropout our embedding with a probability of 0.2. Other than embedding, we use Xavier uniform initializer for all trainable parameters in our model. All traditional Chinese characters are mapped

to the simplified Chinese character, and all experiments are conducted upon pre-trained character embedding with bi-gram feature, embeddings are pre-trained with word2vec toolkit (Mikolov et al. 2013) on Chinese Wikipedia corpus. Character embeddings are shared across tasks.

Training

For each training step, we sample 6 tasks from the task pool, each with a batch size of 128, the probability of being sampled is proportional to the total number of samples in the corresponding task. Then we feed 6 batches of data into 6 computational graphs and then update the global parameter synchronously. For parameter selection, we keep track of parameters that perform the best on the development set. The training process is terminated after 7 epochs without improvement on development set, and note that we use an averaged F scores over all tasks to measure how good our model is.

Overall Results

Switch-LSTMs could work on both single-criterion learning and multi-criteria learning scenarios. Table 3 gives the overall results of the proposed model. Here, Switch-LSTMs have 4-way switches. When applying Switch-LSTMs to single-criterion learning scenario, we omit the \mathbf{e}_m term in Eq (13) and the action \mathbf{a}_t is only related to the current state \mathbf{s}_t and the input $\mathbf{e}_{x_{t+1}}$.

In *single-criterion learning* scenario, we compare Switch-LSTMs with prevalent LSTM and Bi-LSTMs model. As we can see, Bi-LSTMs outperform LSTM and Switch-LSTMs outperform previous LSTM and Bi-LSTMs models. Quantitatively speaking, Switch-LSTMs obtain 94.76 in average F-value, while LSTM and Bi-LSTMs obtain 93.97 and 94.14 in average F-value respectively. It shows that the capability of Switch-LSTMs is greater than conventional LSTM.

In *multi-criteria learning* scenario, we compare Switch-LSTMs with Bi-LSTMs and multi-task learning framework (MTL) proposed by Chen et al. (2017). Notably, Bi-

Models		MSRA	AS	PKU	CTB	CKIP	CITYU	NCC	SXU	Avg.
Single-Criterion Learning										
LSTM	P	95.13	93.66	93.96	95.36	91.85	94.01	91.45	95.02	93.81
	R	95.55	94.71	92.65	85.52	93.34	94.00	92.22	95.05	92.88
	F	95.34	94.18	93.30	95.44	92.59	94.00	91.83	95.04	93.97
	OOV	63.60	69.83	66.34	76.34	68.67	65.48	56.28	69.46	67.00
Bi-LSTMs (Chen et al. 2017)	P	95.70	93.64	93.67	95.19	92.44	94.00	91.86	95.11	93.95
	R	95.99	94.77	92.93	95.42	93.69	94.15	92.47	95.23	94.33
	F	95.84	94.20	93.30	95.30	93.06	94.07	92.17	95.17	94.14
	OOV	66.28	70.07	66.09	76.47	72.12	65.79	59.11	71.27	68.40
Stacked Bi-LSTM (Chen et al. 2017)	P	95.69	93.89	94.10	95.20	92.40	94.13	91.81	94.99	94.03
	R	95.81	94.54	92.66	95.40	93.39	93.99	92.62	95.37	94.22
	F	95.75	94.22	93.37	95.30	92.89	94.06	92.21	95.18	94.12
	OOV	65.55	71.50	67.92	75.44	70.50	66.35	57.39	69.69	68.04
Switch-LSTMs This Work	P	96.07	93.83	95.92	97.13	92.02	93.69	91.81	95.02	94.44
	R	96.86	95.21	95.56	97.05	93.76	93.73	92.43	96.13	95.09
	F	96.46	94.51	95.74	97.09	92.88	93.71	92.12	95.57	94.76
	OOV	69.90	77.80	72.70	81.80	71.60	59.80	55.50	67.30	69.55
Multi-Criteria Learning										
Bi-LSTMs	P	94.64	93.54	93.24	92.87	93.26	91.41	89.30	92.61	92.61
	R	93.20	94.06	91.94	91.75	93.41	90.64	88.04	92.42	91.93
	F	93.91	93.80	92.59	92.31	93.33	91.02	88.66	92.51	92.27
	OOV	65.60	89.20	64.90	75.40	80.00	74.80	64.00	68.50	72.80
Multi-Task Framework (Chen et al. 2017)	P	95.76	93.99	94.95	95.85	93.50	95.56	92.17	96.10	94.74
	R	95.89	95.07	93.48	96.11	94.58	95.62	92.96	96.13	94.98
	F	95.82	94.53	94.21	95.98	94.04	95.59	92.57	96.12	94.86
	OOV	70.72	72.59	73.12	81.21	76.56	82.14	60.83	77.56	74.34
Switch-LSTMs This Work	P	97.69	94.42	96.24	97.09	94.53	95.85	94.07	96.88	95.85
	R	97.87	96.03	96.05	97.43	95.45	96.59	94.17	97.62	96.40
	F	97.78	95.22	96.15	97.26	94.99	96.22	94.12	97.25	96.12
	OOV	64.20	77.33	69.88	83.89	77.69	73.58	69.76	78.69	74.38

Table 3: Results of the proposed model on the test sets of eight CWS datasets. There are two blocks. The first block consists of single-criterion learning models. LSTM and Bi-LSTMs are baselines and the results on them are reported in Chen et al. (2017). The second block consists of the multi-criteria learning model. Multi-task framework for multi-criterion Chinese word segmentation is proposed by Chen et al. (2017). Here, P, R, F, OOV indicate the precision, recall, F value and OOV recall rate respectively. The maximum F values are highlighted for each dataset.

LSTMs is a special case of Switch-LSTM with K set to 1 (single-criterion learning). By concatenating all datasets, Bi-LSTMs performs poorly in multi-criteria learning scenario (the worst). Experimental results show that Switch-LSTMs outperform both Bi-LSTMs and multi-task learning framework on all the corpora. In average, Switch-LSTMs boost about +1% (96.12 in F-value) compared to multi-task learning framework (94.86 in F-value), and boosts +3.85% compared to Bi-LSTMs model (92.27 in F-value).

We could also observe that the performance benefits from multi-criteria learning, since, in this case, the model could learn extra helpful information from other corpora. Concretely, in average F-value, Switch-LSTMs for multi-criteria learning boosts +1.36% (96.12 in F-value) compared to Switch-LSTMs for single-criterion learning (94.76 in F-value).

Model Selection

Figure 4 shows the relationship between switch number and performance in the multi-criteria learning scenario. As we

can see, models with more than 2 switches are better than 1-switch-LSTM with a considerable margin, and the case with 4-way switches is slightly better than other settings. So we employ 4-way Switch-LSTMs for the following experiments. 1-way Switch-LSTMs are the traditional LSTM. So, LSTM could be viewed as a special case of the proposed Switch-LSTMs.

Scale of Parameter Set

Table 4 gives the results of multi-task framework and Switch-LSTMs on the test sets of eight datasets for multi-criteria learning. For 8 datasets, the multi-task framework contains 8 private Bi-LSTMs and 1 shared Bi-LSTMs, whereas Switch-LSTMs do not have any private parameters, consisting of K LSTM cells associated with one switch for control. As we could observe, the parameter set size of multi-task framework is 25K, while the parameter set size of Switch-LSTMs ranges from 4K to 36K with respect to various number of switches. However, as mentioned, Switch-LSTMs perform great when we have more than 2 switches.

Models		# of Param.	MSRA	AS	PKU	CTB	CKIP	CITYU	NCC	SXU	Avg.
Multi-Task Framework		25K	95.82	94.53	94.21	95.98	94.04	95.59	92.57	96.12	94.86
Switch-LSTMs # of switches	1	4K	93.91	93.80	92.59	92.31	93.33	91.02	88.66	92.51	92.27
	2	7K	95.74	94.61	95.82	97.17	95.33	95.48	93.16	96.91	95.53
	3	11K	96.70	94.94	95.70	97.21	94.47	95.04	92.94	96.57	95.45
	4	15K	96.71	95.03	96.22	97.55	95.57	95.38	93.37	96.91	95.84
	5	18K	96.47	95.84	95.89	97.63	95.15	95.33	93.50	96.73	95.82
	6	22K	96.52	94.93	95.79	97.54	95.23	94.50	93.64	95.85	95.50
	7	25K	96.44	94.80	96.23	97.60	95.44	95.15	93.47	96.96	95.76
	8	29K	96.38	95.12	96.22	97.51	94.84	95.31	93.30	96.39	95.63
	9	33K	96.46	94.98	95.91	97.47	94.66	95.27	93.56	97.25	95.69
	10	36K	96.88	94.80	96.00	97.80	94.63	94.82	93.18	96.29	95.55

Table 4: Results of multi-task framework and Switch-LSTMs with various numbers of switches on the test sets of eight datasets on multi-criteria learning scenario. Only F-values are reported. # of parameters only takes parameters of the model into account (input embedding matrix \mathbf{E} is not included).

Concretely, 2-way Switch-LSTMs obtain 95.53 on F-value averagely, outperforming the multi-task framework (94.86 on F-value). But the parameter set size of 2-way Switch-LSTMs is only 7K. Therefore, Switch-LSTMs could outperform multi-task framework with fewer parameters.

Visualization

To better understand how switches affect and contribute to the performance, we visualize the statuses of switches for eight datasets as shown in Figure 3. Concretely, each switch gives a distribution over k-way LSTM on all step t for all the instances. Thus, we calculate the global switch distribution by a normalization operation for each dataset. As we observe, switch distributions are diverse, and datasets with similar criteria obtain similar switch distributions. Due to space limitation, we only plot the switch statuses of forwarding Switch-LSTMs.

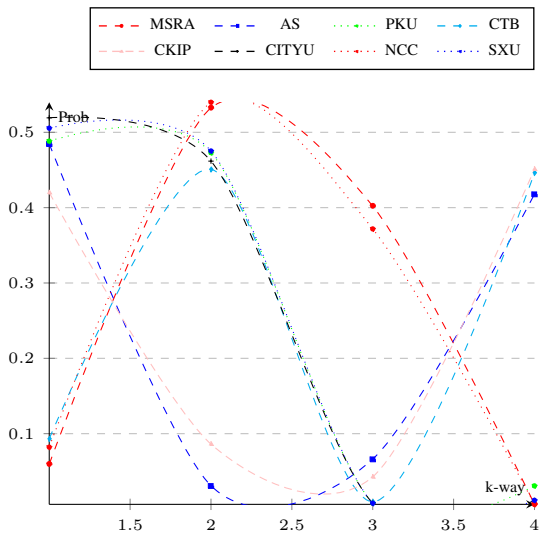


Figure 3: Distributions of switch statuses of forward Switch-LSTMs on test sets of eight datasets for multi-criteria learning. X-axis denotes every individual switch state, and Y-axis is the corresponding probability that state occurs.

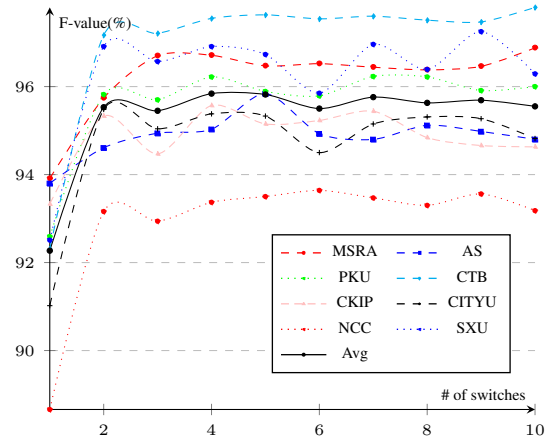


Figure 4: Results (on F-value) of Switch-LSTMs with various numbers of switches on development sets of eight datasets on multi-criteria learning scenario. X-axis denotes the # of switches that the model employs, and Y-axis reports the corresponding results.

Effects of Switch

To evaluate how the inputs of switch affect the performance, we do some input ablation experiments. As shown in Table 5, all the inputs of switch contribute to the model performance, and the task embedding e_m is the most effective part. Concretely, without task embedding e_m , the average F-value over eight datasets drops by 3.33% compared to the full model (95.84 on F-value). Besides, states s_t are also crucial to switch. The performance further drops by 0.81% in average F-value (from 92.51 to 91.70) by additionally ablating state information s_t .

We also tried two switch strategies to see how the switch affect the performance: randomly switch in the training phase and randomly switch in the testing phase. Former one always randomly picks up a switch status at each time step during training, and when at test phase, a normal switch is employed. The later one train the switch in a normal way but randomly picks up a switch status at each time step at the testing phase. By randomly pick up a switch at each time, we could test that if the model works well without the switch.

Models	MSRA	AS	PKU	CTB	CKIP	CITYU	NCC	SXU	Avg.
w/o states s_t	95.68	95.03	95.71	97.60	94.73	94.84	93.34	96.58	95.44
w/o input embeddings e_{x_i}	96.28	94.82	95.98	97.18	94.91	95.14	93.46	96.54	95.54
w/o task embedding e_m	93.36	93.49	92.90	92.65	93.81	91.61	89.25	93.02	92.51
w/o e_{x_i} & e_m	93.26	93.50	92.60	93.31	94.23	91.74	88.74	93.10	92.56
w/o s_t & e_m	93.39	91.97	91.66	92.51	92.09	91.22	88.43	92.33	91.70
Randomly switch in training	92.64	92.90	92.84	93.51	94.20	91.63	89.45	93.05	92.53
Randomly switch in testing	85.56	93.63	90.16	97.16	93.53	93.22	85.56	91.49	91.29
All (Full model)	96.71	95.03	96.22	97.55	95.57	95.38	93.37	96.91	95.84

Table 5: Results of the proposed model with different switch configurations on test sets of eight CWS datasets for multi-criteria learning. Only F-values are reported. Term “w/o” denotes “without”. The maximum F values are highlighted for each dataset.

# of instances	Models	MSRA	AS	PKU	CTB	CKIP	CITYU	NCC	SXU	Avg.
100	Single	75.71	68.51	78.87	79.65	65.59	77.18	71.75	81.29	74.82
	Transfer	89.08	91.93	91.50	92.08	94.02	93.62	88.97	93.34	91.82
300	Single	81.50	72.77	83.95	88.46	73.76	82.12	78.20	85.97	80.84
	Transfer	88.99	92.05	91.97	91.92	94.52	93.43	89.94	94.03	92.11
500	Single	83.52	75.29	86.02	90.33	75.12	86.43	79.89	87.12	82.97
	Transfer	89.17	92.23	91.90	94.94	94.61	93.72	89.98	94.19	92.59
700	Single	83.55	76.29	87.70	91.59	77.76	87.97	81.72	88.33	75.25
	Transfer	89.41	92.10	91.74	95.49	94.49	93.87	90.12	93.67	92.61
1000	Single	85.75	78.78	88.75	91.82	78.51	88.70	82.52	90.09	76.30
	Transfer	89.04	92.35	92.51	95.39	94.64	93.82	90.45	93.85	92.76

Table 6: Results of transfer learning of the proposed model on test sets of eight CWS datasets for single-criteria learning. Only F-values are reported. “# of instances” denotes how many instances involved for training. Single model is the conventional Bi-LSTM model. “Transfer” denotes the proposed Switch-LSTMs by fixing all the parameters learned from other 7 datasets except the new involved task embedding.

As we observe, random strategies on switch lead to poor performance (performance drops by 3% on F-value). It implies that a normal switch is crucial to our model, and switch mechanism contributes a lot in boosting performance.

Knowledge Transfer

Switch-LSTMs could also be easily transferred to other new datasets. To evaluate the transfer capability of our model, we leave one dataset out and train Switch-LSTMs on other 7 datasets. Then, we fixed all the parameters except the newly introduced task embedding when training on instances of the leave out dataset. As shown in Table 6, Switch-LSTMs could obtain excellent performance when only 100, 300, 500, 700, 1000 training instances are available. The single model (conventional LSTM) cannot learn from such few instances. For instance, when we train with 1000 training examples, the single model only obtains 76.30 in average F-value, whereas Switch-LSTMs could obtain 92.76 (boosts 16.46 averagely). It shows that Switch-LSTMs could adapt to a new criterion by only learning a new task embedding, and the newly learned task embedding leads to a new switch strategy for the new criterion.

Related Work

It is a common practice for utilizing annotated data from different but similar domain to boost the performance for each task. Many efforts have been made to better utilizing the homogeneous factor in various tasks to help improve multiple

tasks especially those barren tasks with few examples.

Recently, some efforts have been made to transfer knowledge between NLP Tasks. Zoph and Knight; Johnson et al. (2016; 2017) have been jointly training translation models from and to different languages, it is achieved simply by jointly train encoder or both encoder and decoder. (Jiang, Huang, and Liu 2009; Sun and Wan 2012; Qiu, Zhao, and Huang 2013; Li et al. 2015; 2016; Chen, Zhang, and Liu 2016) adopted the stack-based model to take advantage of annotated data from multiple sources, and show that tasks can indeed help improve each other.

Chen, Zhang, and Liu (2016) adopted two neural models based on stacking framework and multi-view framework respectively, which boosts POS-tagging performance by utilizing corpora in heterogeneous annotations. Chen et al. (2017) have proposed a multi-criteria learning framework for CWS. Using a similar framework as in Caruana (1997), there are private layers for each task to extract criteria-specific features, and a shared layer for the purpose of transferring information between tasks, to avoid negative transfer, they pose an adversarial loss on the shared layer to impose source indistinguishability thus make it criteria-invariant.

Conclusions

In this paper, we propose a flexible model, called Switch-LSTMs, for multi-criteria CWS, which can improve the performance of every single criterion by fully exploiting the underlying shared sub-criteria across multiple heterogeneous

criteria. Experiments on eight corpora show the effectiveness of our proposed model. In future works, we are planning to experiment Switch-LSTMs on other multi-task problems such as transferring information between different document categorization datasets and further investigate a discrete version of Switch-LSTM via reinforcement learning.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. The research work is supported by Shanghai Municipal Science and Technology Commission (No. 17JC1404100 and 16JC1420401), and National Natural Science Foundation of China (No. 61672162 and 61751201).

References

- Cai, D., and Zhao, H. 2016. Neural word segmentation learning for chinese. In *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Caruana, R. 1997. Multitask learning. *Machine learning*.
- Chen, X.; Qiu, X.; Zhu, C.; and Huang, X. 2015a. Gated recursive neural network for chinese word segmentation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Chen, X.; Qiu, X.; Zhu, C.; Liu, P.; and Huang, X. 2015b. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Chen, X.; Shi, Z.; Qiu, X.; and Huang, X. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Chen, H.; Zhang, Y.; and Liu, Q. 2016. Neural network for heterogeneous annotations. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Emerson, T. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.
- Fei, X. 2000. The part-of-speech tagging guidelines for the penn chinese treebank (3.0). URL: <http://www.cis.upenn.edu/~chinese/segguide.3rd.ch.pdf>.
- Gong, C.; Li, Z.; Zhang, M.; and Jiang, X. 2017. Multi-grained chinese word segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.
- Jiang, W.; Huang, L.; and Liu, Q. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging: a case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*.
- Jin, G., and Chen, X. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Sixth SIGHAN Workshop on Chinese Language Processing*.
- Johnson, M.; Schuster, M.; Le, Q. V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F. B.; Wattenberg, M.; Corrado, G.; Hughes, M.; and Dean, J. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*.
- Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.
- Li, Z.; Chao, J.; Zhang, M.; and Chen, W. 2015. Coupled sequence labeling on heterogeneous annotations: Pos tagging as a case study. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.
- Li, Z.; Chao, J.; Zhang, M.; and Yang, J. 2016. Fast coupled sequence labeling on heterogeneous annotations via context-aware pruning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.
- Pei, W.; Ge, T.; and Baobao, C. 2014. Maxmargin tensor neural network for chinese word segmentation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Qiu, X.; Zhao, J.; and Huang, X. 2013. Joint chinese word segmentation and pos tagging on heterogeneous annotated corpora with multiple task learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Sun, W., and Wan, X. 2012. Reducing approximation and estimation errors for chinese lexical processing with heterogeneous annotations. In *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Yao, Y., and Huang, Z. 2016. Bi-directional lstm recurrent neural network for chinese word segmentation. In *International Conference on Neural Information Processing*.
- Yu, S.; Lu, J.; Zhu, X.; Duan, H.; Kang, S.; Sun, H.; Wang, H.; Zhao, Q.; and Zhan, W. 2001. Processing norms of modern Chinese corpus. Technical report, Technical report.
- Zhang, M.; Zhang, Y.; and Fu, G. 2016. Transition-based neural word segmentation. *Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL*.
- Zheng, X.; Chen, H.; and Xu, T. 2013. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*.
- Zoph, B., and Knight, K. 2016. Multi-source neural translation. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*.