Domain-specific Chinese Word Segmentation with Document-level Optimization

Qian Yan[1] , Chenlin Shen[1] , Shoushan Li[*1] , Fen Xia[2] , and Zekai Du[2]

[1] Natural Language Processing Lab, School of Computer Science and Technology, Soochow University, China
[2] Beijing Wisdom Uranium Technology Co., Ltd
{qyan,clshen}@stu.suda.edu.cn, lishoushan@suda.edu.cn
xiafen@ebrain.ai, 2656460787@qq.com

Abstract. Previous studies normally formulate Chinese word segmentation as a character sequence labeling task and optimize the solution in sentence-level. In this paper, we address Chinese word segmentation as a document-level optimization problem. First, we apply a state-of-the-art approach, i.e., long short-term memory (LSTM), to perform character classification; Then, we propose a global objective function on the basis of character classification and achieve global optimization via Integer Linear Programming (ILP). Specifically, we propose several kinds of global constrains in ILP to capture various segmentation knowledge, such as segmentation consistency and domain-specific regulations, to achieve document-level optimization, besides label transition knowledge to achieve sentence-level optimization. Empirical studies demonstrate the effectiveness of the proposed approach to domain-specific Chinese word segmentation.

Keywords: Chinese word segmentation, Document-level, LSTM, ILP.

1 Introduction

The task of word segmentation is to segment the continuous text into isolated words. As a fundamental task in Natural Language Processing (NLP) for those languages without word delimiters, e.g., Chinese [1], word segmentation has been applied as an essential pre-processing step for many NLP tasks, such as named entity recognition [2], event extraction [3], and machine translation [4].

In the literature, most of popular approaches to Chinese word segmentation (CWS) are machine learning-based. In the early years, CWS is treated as a character classification problem, i.e., classifying each Chinese character into a tag according to its position in a word. For instance, position tag *B* stands for the beginning of a word and *E* stands for the end of a word. For convenience, we refer to this approach as character-level word segmentation. As a representative, Xue [1] employs a maximum entropy model to CWS.

In recent years, CWS is modeled as a sequence labeling problem where not only

* Corresponding author

the characters are classified to different position tags but also the transition probability between two nearby position tags are employed to achieve the optimization in the sentence. For example, a segmentation containing the "*EE*" subsequence will be considered impossible in the sentence-level optimization because transition probability from one *E* to another *E* is zero due to the fact that a character with tag *E* always follows a character with tag *B* or *S*. For convenience, we refer to this approach as sentence-level word segmentation. As a representative, Tseng [5] employs the well-known conditional random fields (CRF) model to CWS.

However, although character-level and sentence-level word segmentation have achieved much success, they are incapable of handling challenges in domain-specific document-level CWS. Document-level segmentation approaches are especially necessary for domain-specific CWS, since domain-specific texts are often organized in document styles, such as judgments, patents, and scientific papers.

In this paper, we address document-level challenges in domain-specific CWS. First, we apply a long short-term memory (LSTM) classification model to perform character classification and obtain the posterior probabilities belonging to all position tags. Then, we propose a global objective function on the basis on character classification and achieve global optimization via a joint inference approach, named Integer Linear Programming (ILP). Besides various constrains on the label transition to achieve sentence-level optimization, various kinds of constrains on segmentation consistency and domain-specific textual regulation are proposed to achieve document-level optimization. Empirical studies demonstrate the effectiveness of these constraints in document-level word segmentation.

The remainder of this paper is organized as follows. Section 2 overviews related work on the Chinese word segmentation. Section 3 proposes the character-level and sentence-level approaches to CWS. Section 4 proposes our ILP-based approach to document-level CWS. Section 5 presents the experimental results. Finally, section 6 gives the conclusion and future work.

2 Related Work

There has been an enormous amount of work in the research fields of Chinese word segmentation. Existing word segmentation approaches can be mainly categorized into three groups: character-based [5], word-based [6], and both word-and-character-based approaches [7]. This paper mainly focuses on the character-based approaches to CWS.

The pioneer work by Xue [1] first models CWS as a character classification problem and subsequent studies further improve the tagging model into a character sequence labeling problem [5]. In the research line, many studies aim to improve the performance by various manners, such as feature expanding [8], active learning [9], and using different tag sets [10], with shallow learning models like CRF.

More recently, neural network approaches with deep learning models have attracted a great deal of attention. Some novel deep learning models have been adopted in CWS, such as, convolution neural network [11], tensor neural network [12], recursive neural network [13], long short-term memory (LSTM) [14], and gated recursive neural net-

work [15]. All these studies demonstrate that the deep learning models have achieved better segmentation results than the shallow learning models.

Different from all above studies, this study performs CWS in a document level. In the first step, our approach applies the label classification approach by LSTM model proposed by previous studies [14] which represents the state-of-the-art performing approach. However, in the following step, our approach aims to obtain a global optimization in the whole document, which has not been researched by any previous studies.

The closest work to ours is a recent work by Li and Xue [16] which deals Chinese patent word segmentation. However, their approach solves the problem by exploiting some document-level features and still applies sentence-level optimization learning model, i.e., CRF. In contrast, our approach optimizes the results with a document-level optimization learning model, i.e., ILP. Furthermore, their approach needs extra document-level labeled data in the specific domain to train the learning model while our approach does not need any labeled data.

3 Character-level and Sentence-level CWS

Our approach to sentence-level CWS mainly consists of two steps. In the first step, we apply the LSTM neural network to perform character classification, classifying each character to a position tag, i.e., {*B*, *M*, *E*, *S*}. Specifically, *B*, *M*, and *E* represent the *Begin*, *Middle*, and *End* of a multi-character segmentation and *S* represents a *Single* character segmentation. In the second step, we define a global objective function with the character classification results and achieve global optimization via Integer Linear Programming.

3.1 Character-level CWS with LSTM

In this subsection, we propose the LSTM classification model. Figure 1 shows the framework overview of the LSTM model for character classification. Formally, the input of the LSTM classification model consists of character unigram and bigram embeddings for representing the current character x_i, i.e.,

$$x_i = v_{c_i} \oplus v_{c_{i+1}} \oplus ... \oplus v_{c_{i+1},c_{i+2}} \quad (1)$$

Where $v_{c_i} \in \mathbf{R}^d$ is a *d*-dimensional real-valued vector for representing the character unigram c_i and $v_{c_i,c_{i+1}} \in \mathbf{R}^d$ is a *d*-dimensional real-valued vector for representing the character bigram c_i, c_{i+1}.

Through the LSTM unit, the input of a character is converted into a new representation h_i, i.e.,

$$h_i = LSTM(x_i) \quad (2)$$

Subsequently, the fully-connected layer accepts the output from the previous layer, weighting them and passing through a normally activation function as follows:

$$h^*_i = dense(h_i) = \phi(\theta^T h_i + b) \quad (3)$$

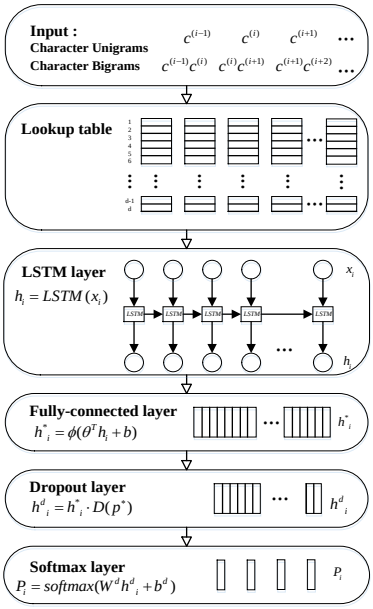

Fig. 1. The framework overview of the LSTM model for character-level CWS

Where ϕ is the non-linear activation function, employed "relu" in our model. h^*_i is the output from the fully-connected layer.

The dropout layer is applied to randomly omit feature detectors from network during training. It is used as hidden layer in our framework, i.e.,

$$h^d_i = h^*_i \cdot D(p^*) \tag{4}$$

Where D denotes the dropout operator, p^* denotes a tunable hyper parameter, and h^d_i denotes the output from the dropout layer.

The softmax output layer is used to get the prediction probabilities, i.e.,

$$P_i = softmax(W^d h^d_i + b^d) \tag{5}$$

Where P_i is the set of predicted probabilities of the character classification, W^d is the weight vector to be learned, and the b^d is the bias term. Specifically, P_i consist of the posterior probabilities of the current character belonging to each position tag {*B*, *M*, *E*, *S*}, i.e.,

$$P_i = < p_{i,B}, p_{i,M}, p_{i,E}, p_{i,S} > \tag{6}$$

3.2 Sentence-level CWS with ILP

In this subsection, we optimize the obtained results from character classification with Integer Linear Programming (ILP). In the literature, ILP has been widely used in many NLP applications [17].

Specifically, the objective function in ILP is defined as follows:

$$\min \sum_{i=1}^{N} \{y_{i,B} \cdot (-\log p_{i,B}) + y_{i,M} \cdot (-\log p_{i,M}) + y_{i,E} \cdot (-\log p_{i,E}) + y_{i,S} \cdot (-\log p_{i,S})\} \quad (7)$$

Subject to:

$$y_{i,B} \in \{0,1\} \quad (8)$$

$$y_{i,M} \in \{0,1\} \quad (9)$$

$$y_{i,E} \in \{0,1\} \quad (10)$$

$$y_{i,S} \in \{0,1\} \quad (11)$$

$$y_{i,B} + y_{i,M} + y_{i,E} + y_{i,S} = 1 \quad (12)$$

Where N is total number of all characters in the sentence (or the document if applied in document-level word segmentation). $y_{i,B}$ is a Boolean label, denoting whether the final result of current character is *B* ($y_{i,B}$=1) or not ($y_{i,B}$=0). $y_{i,M}, y_{i,E}, y_{i,S}$ denote the same meaning as $y_{i,B}$.

For sentence-level CWS, the constraints implied in the label transition is proposed as following:

(C1): Label transition constraints

This type of constraints limits the position tags of two nearby characters. 4 cases are discussed according to the position tag of the current character.

- **Case (1.1):**

When the position tag of the current character is *B*, i.e., $y_{i,B} = 1$, the position tag of the next character could only be *M* or *E*, i.e., $y_{i+1,M} + y_{i+1,E} = 1$. Otherwise, when $y_{i,B} = 0$, the position tag of the next character could be anyone, i.e., $y_{i+1,M} + y_{i+1,E} = 0$ or 1. Therefore, we obtain the following constraint:

$$y_{i,B} - (y_{i+1,M} + y_{i+1,E}) \le 0 \quad (13)$$

- **Case (1.2):**

When the position tag of the character is *M*, we obtain the following constraint:

$$y_{i,M} - (y_{i+1,M} + y_{i+1,E}) \le 0 \quad (14)$$

- **Case (1.3):**

When the position tag of the character is *E*, we obtain the following constraint:

$$y_{i,E} - (y_{i+1,B} + y_{i+1,S}) \le 0 \quad (15)$$

- **Case (1.4):**

When the position tag of the character is *S*, we obtain the following constraint:

$$y_{i,S} - (y_{i+1,B} + y_{i+1,S}) \le 0 \quad (16)$$

4 Document-level CWS

As mentioned in Introduction, although character-level and sentence-level word segmentation have achieved much success, they are incapable of handling challenges in document-level CWS.

E1:
Segmentation result:
黄和昌 与 被告 签订 协议 。 约定 被告 所借 黄 和昌 500 万 元 借款 分 三次 还 清 。
Gold-standard segmentation:
黄和昌 与 被告 签订 协议 。 约定 被告 所借 黄和昌 500 万 元 借款 分 三次 还 清 。

(**English Translation**:
Hechang Huang signed an agreement with the defendant. They were agreed that 500 million Yuan that borrowed from Hechang Huang to the defendant would be paid back in three times.)

Fig. 2. An example with inconsistent segmentation frequently occurred in character-level or sentence-level segmentation

One challenge in document-level is how to achieve segmentation consistency in document-level. That is, two text fragments with the same Chinese character sequence should have the same segmentation result as much as possible. Figure 2 shows an instance frequently occurs in the character-level or sentence-level word segmentation where "黄和昌"*(Hechang Huang)* is recognized as a word in the first sentence but segmented into two words, i.e., "黄"*(Huang)* and "和昌"*(Hechang)* in the second sentence. Such segmentation inconsistency should and only can be avoided in document-level. To tackle this challenge, we propose some constrains as follows:

(C2): Segmentation consistency constraints

If two nearby Chinese characters c_i, c_{i+1} are the same as another two nearby Chinese characters c_j, c_{j+1} in a document, we constraint their position labels to be the same, i.e.,

When $c_i, c_{i+1} = c_j, c_{j+1}$, we have

$$y_{i,B} = y_{j,B},\ y_{i+1,B} = y_{j+1,B} \tag{17}$$

$$y_{i,M} = y_{j,M},\ y_{i+1,M} = y_{j+1,M} \tag{18}$$

$$y_{i,E} = y_{j,E},\ y_{i+1,E} = y_{j+1,E} \tag{19}$$

$$y_{i,S} = y_{j,S},\ y_{i+1,S} = y_{j+1,S} \tag{20}$$

Another challenge is the consideration of domain-specific textual regulations that are popular in document-level. Figure 3 shows an instance from a judgment (a decision law document of a court). In such text, plaintiffs and defendants are explicitly described in two lines in the front part. It is easy to capture such segmentation regulation from two textual patterns, i.e., "原告 *NAME1*, (*Plaintiff NAME1*,)" and "被告 *NAME2*, (*Defendant NAME2*,)" where "*NAME1*" or "*NAME2*" denotes a person or an organization name. To tackle this challenge, we propose some constrains as follows:

(C3): Textual regulation constraints

In this study, a textual pattern is defined as following:

Pattern="Tri1+NAME+Tri2"

E2:
民 事 裁 定 书
（2016）川1425民初404号
原告吕某某，男。委托代理人王刚，某律师事务所律师
被告黄某某，女。委托代理人张可，某律师事务所律师
本院……
(**English Translation**:
Civil Judgment
(2016) Chuan Civil Trial Num. 1425
Plaintiff Moumou Lv, Male. Attorney Gang Wang, lawyer of Some Lawyer Office.
Defendant Moumou Huang, Female. Attorney Ke Zhang, lawyer of Some Lawyer Office.
This Court……)

Fig. 3. An example from a judgment text

Where *Tri1* and *Tri2* are two trigger character sequences and *NAME* is a character sequence with variable length. We define that, in this pattern, *Tri1*, *Tri2*, and *NAME* are segmented to be three words.

This type of constrains is domain-specific and document-level. In judgments, we first use some rules to segment the whole document into several parts. Then, we focus on some popular textual patterns in the front part of a judgment. Specifically, we adopt some textual patterns as shown in Table 1.

Table 1. Some textual patterns in the front part of a judgment

Tri1	NAME	Tri2
被告（Defendant）	*NAME1*	,
原告（Plaintiff）	*NAME2*	,
代理人（Attorney）	*NAME3*	,
代表人（Attorney）	*NAME4*	,

These patterns are recognized with some regular expressions. Then, we obtain the begin index and end index of the character sequence of *Tri1*, *NAME*, or *Tri2*, which are denoted as q and r. The constraint to segment this character sequence to be word is given as following:

$$y_{q,B} = 1 \tag{21}$$

$$y_{r,E} = 1 \tag{22}$$

$$y_{k,M} = 1 \text{ where } q < k < r \tag{23}$$

5 Experimentation

In this section, extensive experiments are carried out to evaluate the proposed ILP-based approach to domain-specific CWS.

5.1 Experimental Settings

Data Sets: We use two data sets for evaluation. One is from OntoNotes 5.0 which contains six domains: *BN*, *BC*, *NW*, *MZ*, *TC*, and *WB* [18]. The data has been split into three data sets: training, development and test data. The other one is a domain-specific data set which is collected from (http://wenshu.court.gov.cn/) and annotated by ourselves according to the OntoNotes [18] word segmentation guideline. It contains two domains: *Contract* and *Marriage* and each domain contains 100 documents.
Embeddings: We use word2vec (http://word2vec.googlecode.com/) to pre-train character unigram and bigram embeddings using the two data sets.
Hyper-parameters: The hyper-parameter values in the LSTM model are tuned according to performances in the development data.
Evaluation Measurement: The performance is evaluated using the standard precision (*P*), recall (*R*) and *F* score.
Significance test: *T*-test is used to evaluate the significance of the performance difference between two approaches.

5.2 Experimental Results on Character-level and Sentence-level CWS

In this subsection, we test our LSTM with ILP-based approach to character-level and sentence-level CWS. The training and test data are both from OntoNotes5.0 [18].

For comparison, we implement following approaches to CWS:

- **CRF-Char (Character-level)**: This is a shallow learning approach which employs conditional random fields (CRF) as the classification algorithm. In the implementation, we apply the tool of CRF++ (http://crfpp. sourceforge. net) and both character unigrams and bigrams are used. The length of the character context window is 2. Note that both the training and development are merged as training data for CRF learning.
- **CRF-Sen (Sentence-level)**: This is similar to CRF-Char except adding a label transition feature which is employed to optimize the segmentation results of each sentence. In the implementation, a special feature named "*B*" is added in the feature template. It is exactly the approach by Tseng [5] which represents the most popular one to CWS before the deep learning approaches appear.
- **LSTM-Char (Character-level):** This is a deep learning approach which employs LSTM as the classification algorithm. This approach is illustrated in Section 4.1 and it is similar to the state-of-the-art approach to CWS, namely LSTM-1, by Chen [14] but missing the sentence inference step.
- **LSTM-Sen (Sentence-level)**: This is exactly the state-of-the-art approach to CWS, namely LSTM-1, by Chen [14].
- **LSTM-ILP(C1) (Sentence-level)**: This is our approach to CWS which employs ILP to optimize the results of each sentence. Since the optimization is performed in a sentence rather than a document, only the first type of constrains, namely C1, is leveraged.

Due to the space limitation, we report the results of three domains including *BN*, *BC*, and *NW*. Table 2(a) - Table 2(c) show the performances of different approaches to CWS

Table 2. Performances of different approaches to CWS (Tested on OntoNotes 5.0)

(a) Performances of different approaches to CWS (Test domain: *BN*, OOV rate: 0.065)

	P	*R*	*F*	R_{OOV}
CRF-Char	0.934	0.934	0.934	0.669
CRF-Sen	0.950	0.948	0.949	0.755
LSTM-Char	0.939	0.940	0.939	0.682
LSTM-Sen	**0.952**	**0.955**	**0.953**	**0.757**
LSTM-ILP(C1)	**0.952**	**0.955**	**0.953**	**0.757**

(b) Performances of different approaches to CWS (Test domain: *BC*, OOV rate: 0.075)

	P	*R*	*F*	R_{OOV}
CRF-Char	0.928	0.936	0.932	0.674
CRF-Sen	0.944	0.949	0.946	0.750
LSTM-Char	0.935	0.942	0.938	0.707
LSTM-Sen	**0.949**	**0.951**	**0.950**	**0.773**
LSTM-ILP(C1)	0.948	0.948	0.948	0.768

(c) Performances of different approaches to CWS (Test domain: *NW*, OOV rate: 0 .084)

	P	*R*	*F*	R_{OOV}
CRF-Char	0.928	0.925	0.927	0.677
CRF-Sen	0.952	0.947	0.949	**0.799**
LSTM-Char	0.934	0.939	0.936	0.682
LSTM-Sen	**0.956**	**0.956**	**0.956**	0.791
LSTM-ILP(C1)	0.955	0.954	0.954	0.785

in these three domains. From these tables, we can see that the OOV rates in the three domains are in the range of 0.065-0.084, which are all less than 0.1. The low OOV rate in these three domains. In all domains, the segmentation performances are more than 90% in terms of *F* score.

In character-level CWS, LSTM-Char generally performs better than CRF-Char in terms of *F* score and OOV Recall.

In sentence-level CWS, LSTM-Sen performs better than CRF-Sen in three domains in terms of *F* score and in two domains in terms of OOV Recall. Averagely, LSTM-Sen outperforms CRF-Sen in terms of *F* score, although the improvement is slight. Our approach LSTM-ILP(C1) performs comparable to LSTM-Sen. This result confirms that, in sentence-level CWS, our approach achieves the state-of-the-art performances.

5.3 Experimental Results on Domain-specific Document-level CWS

In this subsection, we test our LSTM with ILP approach to CWS in document-level

CWS. The training data is from OntoNotes 5.0 while the test data is from the judgments.

For comparison, besides the approaches in the above subsection, we implement following approaches to CWS:

- **LSTM-Sen+Regulation-Rule**: a straight-forward approach to incorporating domain-specific regulation knowledge for CWS. In this approach, we first perform LSTM-Sen to CWS and then we apply simple rules to recognize words with all textual patterns to refine the results from LSTM-Sen.
- **LSTM-ILP(C2) (Document-level)**: This is our approach to CWS which employs ILP to optimize the results in the whole document. Only the second type of constrains, i.e., segmentation consistency constraints, is used.
- **LSTM-ILP(C3) (Document-level)**: In the same spirit to LSTM-ILP(C2), only the third type of constrains, i.e., textual regulation constraints, is used.
- **LSTM-ILP(C*i*+C*j*) (Sentence-level and Document-level):** This is our approach to CWS which employs ILP to optimize the results in the whole document. Both the *i*-th and the *j*-th types of constrains are used.
- **LSTM-ILP(C1+C2+C3) (Sentence-level and Document-level):** This is our approach to CWS which employs ILP to optimize the results in both the sentence-level and document-level. That is to say, all types of constrains are used.

Table 3 and Table 4 show the performances of different approaches to CWS tested on the second data set. From these two tables, we can see that the OOV rates in the two domains are in the range of 0.165-0.186, which is much higher than those in the last experiment. The high OOV rate is due to the fact that the training and test data are from different domains.

When only one type of document-level constrains is employed, the LSTM model with ILP, i.e., LSTM-ILP(C2) or LSTM-ILP(C3), performs significantly better than LSTM-Char in terms of *F* score (p-value<0.001), which verifies the effectiveness of using the document-level constrains, i.e., i.e., C2 or C3.

When two types of document-level constrains are employed, the LSTM model with ILP, i.e., LSTM-ILP (C1+C2), LSTM-ILP (C1+C3) and LSTM-ILP (C2+C3), perform significantly better than that of using only one (p-value<0.001).

When both the sentence-level and document-level constrains are employed, our approach, i.e., LSTM(C1+C2+C3), performs best. Especially, in the *Contract* domain, our approach achieves a gain of 8.9% over LSTM-Char in terms of *F* score, which is rather impressive. Even compared to the state-of-the-art approach LSTM-Sen, our approach significantly improves the *F* score from 0.841 to 0.915 (with a gain of 7.4%, p-value<0.001). Moreover, our approach outperforms LSTM-Char or LSTM-Sen in terms of OOV Recall by a wide margin.

It is worthy to note that LSTM-Sen+Regulation-Rule is a very strong baseline, which implies that regulation rules are very effective for segmenting judgment text. Nevertheless, our approach based on ILP with all constrains significantly outperforms LSTM-Sen+Regulation-Rule in both domains in terms of *F* score (p-value<0.001). Especially, in the *Contract* domain, the improvement of our approach over LSTM-Sen+Regulation-Rule is rather impressive, reaching 4.4% in terms of *F* score.

Table 3. Performances of different approaches to CWS
(Test domain: *Contract*, OOV rate: 0.186)

	P	*R*	*F*	R_{OOV}
LSTM-Char	0.816	0.837	0.826	0.735
LSTM-Sen	0.816	0.867	0.841	0.777
LSTM-Sen+ Regulation-Rule	0.856	0.887	0.871	0.804
LSTM-ILP(C1)	0.837	0.874	0.855	0.795
LSTM-ILP(C2)	0.828	0.843	0.835	0.747
LSTM-ILP(C3)	0.845	0.860	0.853	0.756
LSTM-ILP (C1+C2)	0.845	0.880	0.862	0.809
LSTM-ILP (C1+C3)	0.867	0.896	0.881	0.814
LSTM-ILP (C2+C3)	0.888	0.891	0.889	0.820
LSTM-ILP (C1+C2+C3)	**0.909**	**0.922**	**0.915**	**0.866**

Table 4. Performances of different approaches to CWS
(Test domain: *Marriage*, OOV rate: 0.165)

	P	*R*	*F*	R_{OOV}
LSTM-Char	0.879	0.875	0.877	0.804
LSTM-Sen	0.896	0.898	0.897	0.866
LSTM-Sen+ Regulation-Rule	0.909	0.900	0.905	**0.894**
LSTM-ILP(C1)	0.891	0.887	0.889	0.847
LSTM-ILP(C2)	0.883	0.874	0.878	0.806
LSTM-ILP(C3)	0.884	0.878	0.881	0.816
LSTM-ILP (C1+C2)	0.910	0.903	0.906	0.873
LSTM-ILP (C1+C3)	0.903	0.894	0.898	0.868
LSTM-ILP (C2+C3)	0.894	0.882	0.888	0.834
LSTM-ILP (C1+C2+C3)	**0.919**	**0.908**	**0.913**	**0.894**

6 Conclusion

This paper proposes a novel approach to domain-specific CWS which adopts Integer Linear Programming to optimize the character-classification results from the LSTM model. One major advantage of our approach is its convenience to incorporate various kinds of sentence-level and document-level segmentation knowledge, such as label transition, segmentation consistency, and domain-specific textual regulations, by formulating them as mathematical constrains in ILP. Empirical studies show that our ILP-based approach with each kind of constrains consistently improves the segmentation performance. Moreover, when two or three kinds of constrains are leveraged, the segmentation performance could be further apparently improved.

In our future work, we would like to improve our approach by trying some other kinds of constrains to leverage more kinds of segmentation knowledge to correct the segmentation errors. Furthermore, we would like to test our approach in much more other specific domains, such as scientific and patent documents.

Acknowledgments

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

References

1. Xue, N.W.: Chinese Word Segmentation as Character Tagging. Computational Linguistics and Chinese Language Processing 8(1), 29-48 (2003)
2. Gao, J.F., Li, M., Wu, A., Huang, C.N.: Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. Computational Linguistics 31(4), 531-574 (2005)
3. Chen, C., Ng, V.I.: Joint Modeling for Chinese Event Extraction with Rich Linguistic Features. In: Proceedings of COLING, pp. 529-544 (2012)
4. Zhang, R.Q., Yasuda, K., Sumita, E.: Improved Statistical Machine Translation by Multiple Chinese Word Segmentation. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 216-223 (2008)
5. Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.: A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, pp. 168-171 (2005)
6. Andrew, G.: A Hybrid Markov/Semi-Markov Conditional Random Field for Sequence Segmentation. In: Proceedings of EMNLP, pp. 465-472 (2006)
7. Zhang, M., Zhang, Y., Fu, G.: Transition-Based Neural Word Segmentation. In: Proceedings of ACL, pp. 421-431 (2016)
8. Shi, Y.X., Wang, M.Q.: A Dual-layer CRFs Based Joint Decoding Method for Cascaded Segmentation and Labeling Tasks. In: Proceedings of IJCAI, pp. 1707-1712 (2007)
9. Li, S.S., Zhou, G.G., Huang, C. R.: Active Learning for Chinese Word Segmentation. In: Proceedings of COLING, pp. 683-692 (2012)
10. Zhao, H., Huang, C.N., Li, M., Lu, B.L.: Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling. In: Proceedings of PACLIC, pp. 87-94 (2006)
11. Zheng, X.Q., Chen, H.Y., Xu, T.Y.: Deep Learning for Chinese Word Segmentation and POS Tagging. In: Proceedings of EMNLP, pp. 647-657 (2013)
12. Pei, W., Ge, T., Chang, B.: Max-Margin Tensor Neural Network for Chinese Word Segmentation. In: Proceedings of ACL, pp. 293-303 (2014)
13. Chen, X.C., Qiu, X.P., Zhu, C.X., Huang, X.J.: Gated Recursive Neural Network for Chinese Word Segmentation. In: Proceedings of ACL, pp. 1744-1753 (2015)
14. Chen, X.C., Qiu, X.P., Zhu, C.X., Liu, P.F., Huang, X.J.: Long Short-Term Memory Neural Networks for Chinese Word Segmentation. In: Proceedings of EMNLP, pp. 1197-1206 (2015)
15. Xu, J., Sun, X.: Dependency-based Gated Recursive Neural Network for Chinese Word Segmentation. In: Proceedings of ACL, pp. 567-572 (2016)
16. Li, S., Xue, N.: Effective Document-Level Features for Chinese Patent Word Segmentation. In: Proceedings of ACL, pp. 199-205 (2013)
17. Barzilay, R., Lapata, M.: Aggregation via Set Partitioning for Natural Language Generation. In: Proceedings of ACL, pp. 359-366 (2006)
18. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: OntoNotes: The 90% Solution. In: Proceedings of NAACL, pp. 57-60 (2006)