Morphological segmentation with LSTM Networks

Presenter: Wazir Ali

10/26/2018

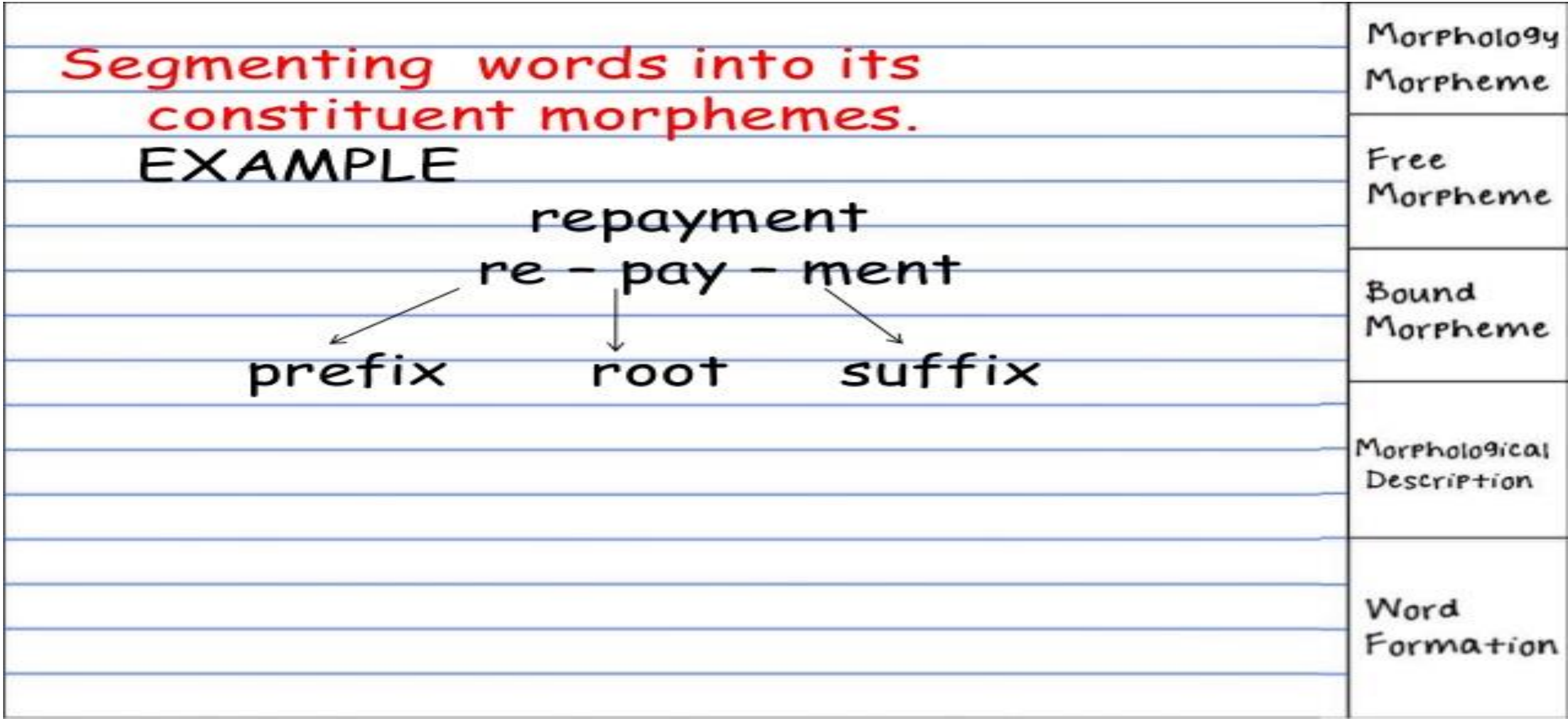
Gist of Presentation

- Introduction
- The regular LSTM
- Attention LSTM
- Discussion
- Conclusion
- References

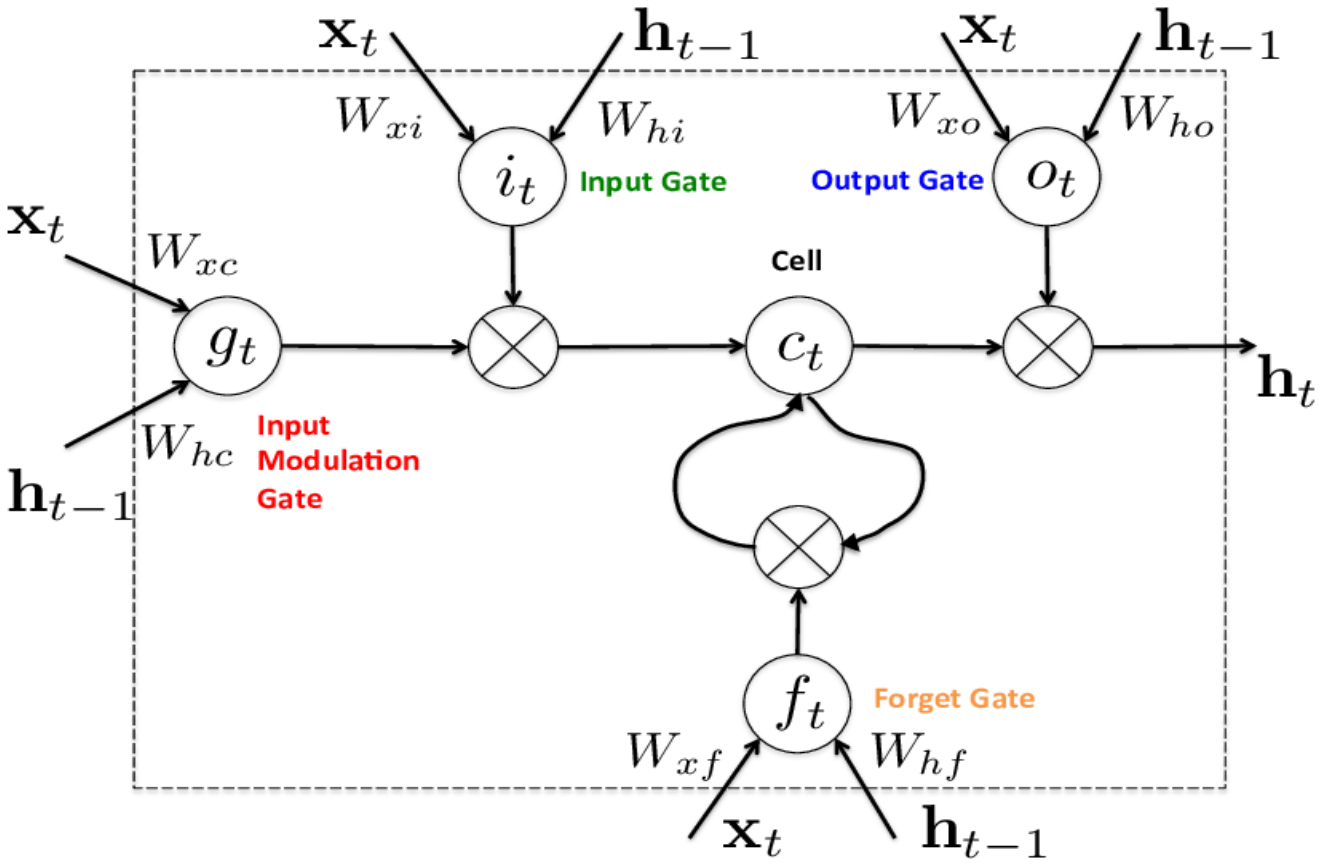
Introduction

- Morphology in natural language is the study of words, their formation and relationship to other words in same language.
- The segmentation is breaking of words into meaningful morphemes.
- It is fundamental task in NLP, specially for rich morphological languages.
- It is important to overcome the problem of data sparseness.
- The LSTM was invented specifically to avoid the vanishing gradient problem.

Examples of Segmentation



Regular LSTM Network



$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1})$$
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1})$$
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot H(W_{cx}x_t + W_{ch}h_{t-1})$$
$$h_t = o_t \odot c_t$$

Window LSTM network

1. The Window LSTM automatically learns structure of sequences and predict morphological boundaries of raw words.
2. Three language independent new architectures.
3. The proposed models provided good results even with limited amount of training data.
4. The models performed well even for complex morphological languages.

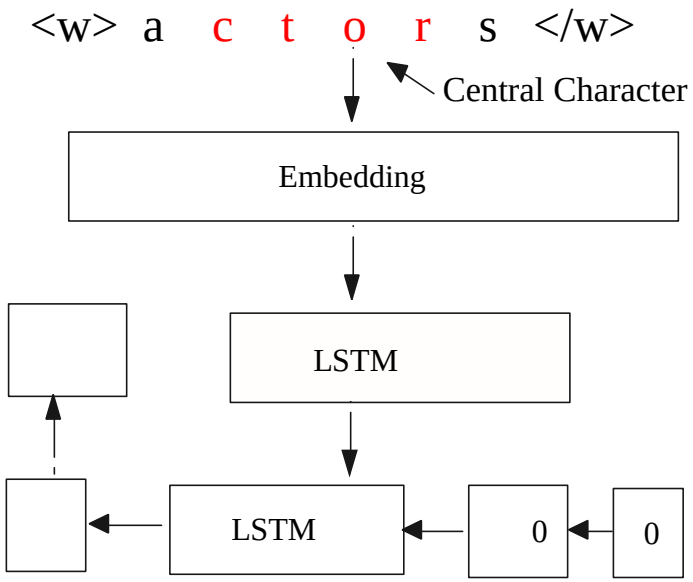
Single window LSTM

act+or+s

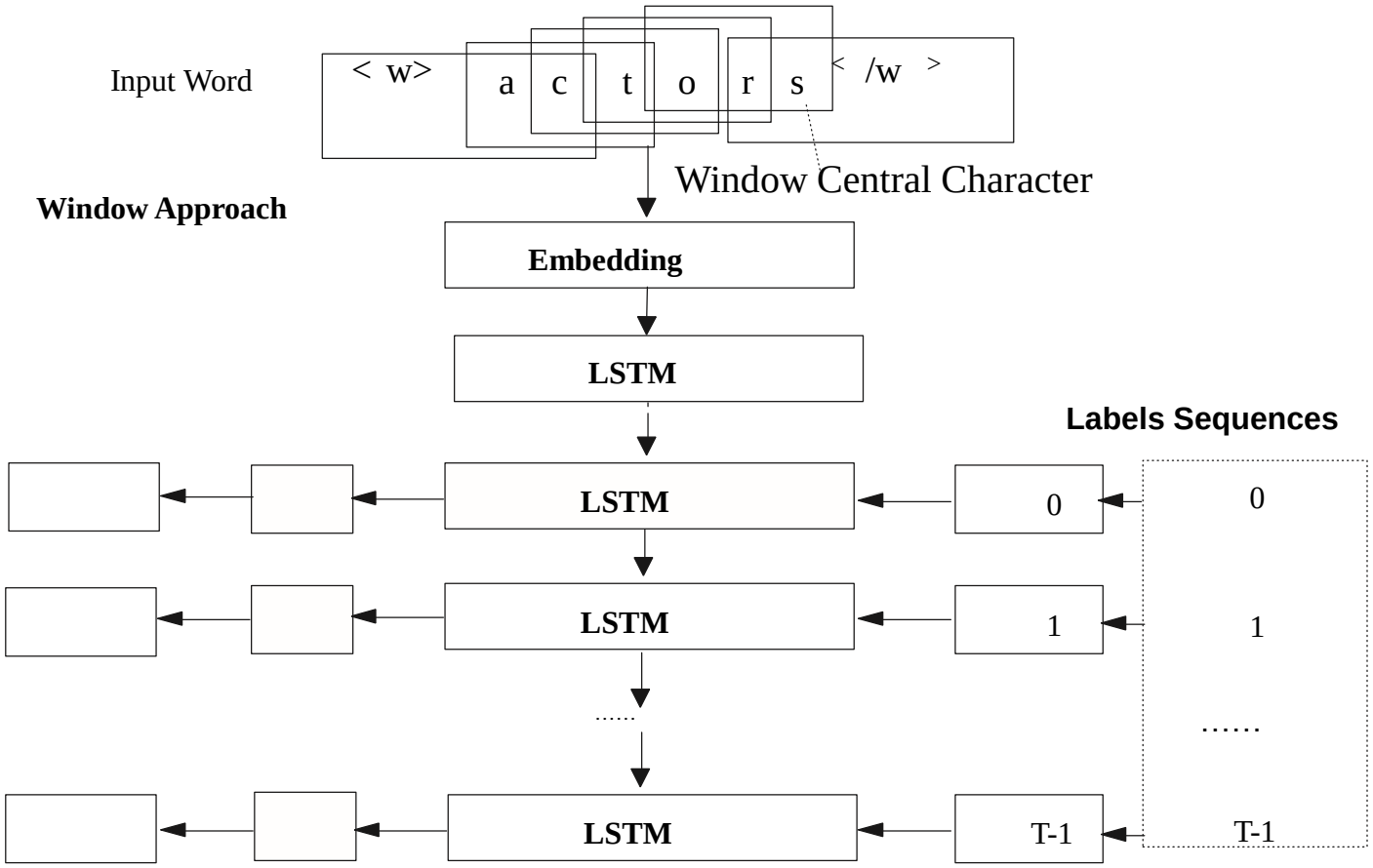
$$T_t - 1 = LT_W(v) \qquad (1)$$

$$T_t = W_e l_0 \qquad (2)$$

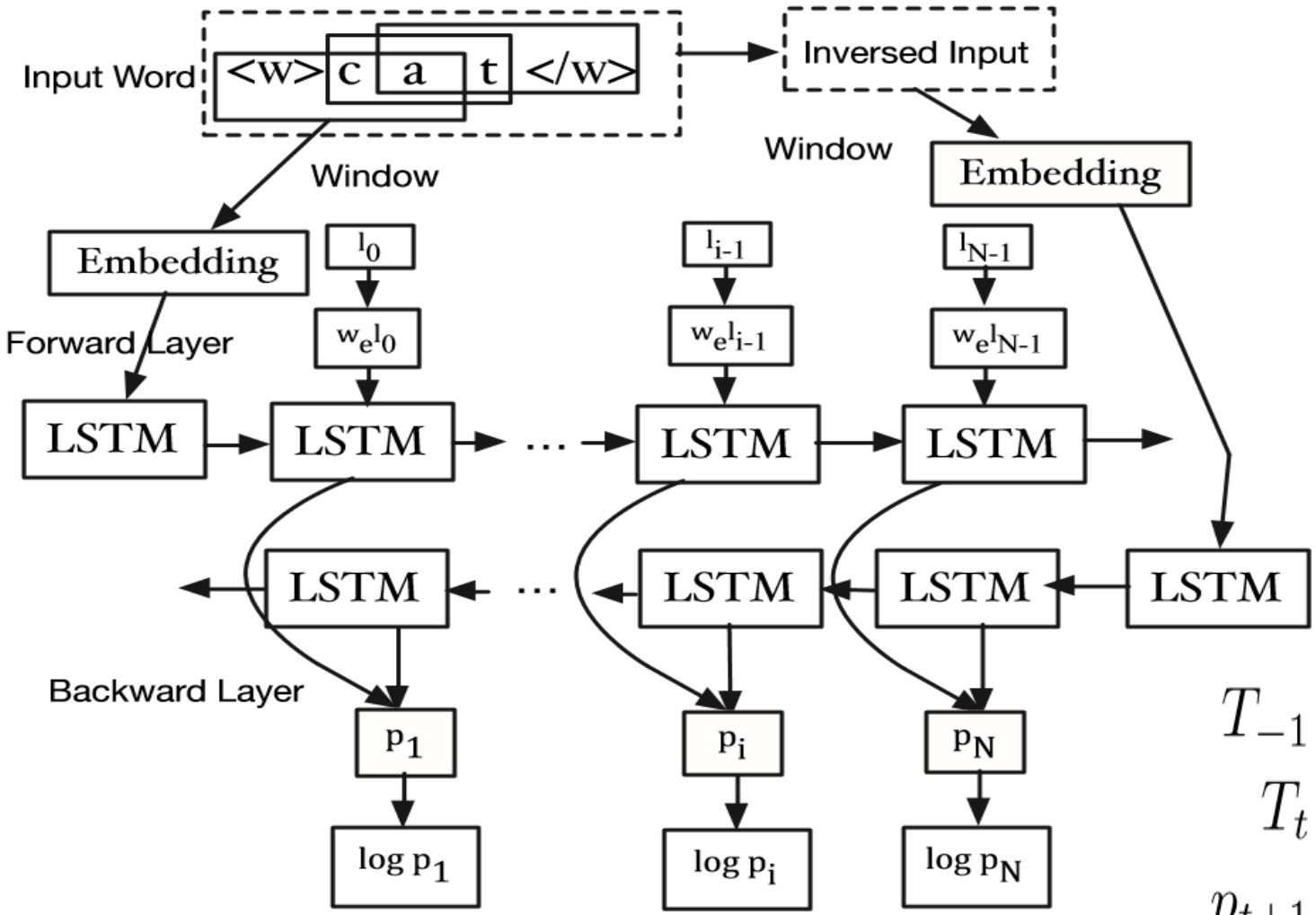
$$p = LSTM(T_t) \qquad (3)$$



Multi Window LSTM



Multi Window BLSTM Network



$$T_{-1} = LT_W(V)$$

$$T_t = W_e l_t \text{ for } t \in \{0, \ldots, T-1\}$$

$$p_{t+1} = LSTM(T_t) \text{ for } t \in \{0, \ldots, T-1\}$$

Results

Arabic	Method	Precision	Recall	F1
25%	CRF	**95.5**	**93.1**	**94.3**
	LSTM	74.6	68.7	71.5
	MW-LSTM	93.3	90.7	92.0
	BMW-LSTM	93.8	90.1	91.9
50%	CRF	**96.5**	**94.6**	**95.5**
	LSTM	72.8	69.9	71.3
	MW-LSTM	95.0	92.9	94.0
	BMW-LSTM	95.0	92.6	93.8
	E. MW-LSTM	95.9	96.1	96.0
75%	CRF	**97.2**	**96.1**	**96.6**
	LSTM	75.4	68.3	71.7
	MW-LSTM	95.6	94.1	94.9
	BMW-LSTM	96.9	93.2	95.0
	E. MW-LSTM	96.5	96.4	96.5
100%	CRF	**98.1**	**97.5**	**97.8**
	LSTM	83.5	65.2	73.3
	W-LSTM	95.6	91.6	93.6
	MW-LSTM	***97.1***	***96.1***	***96.6***
	BMW-LSTM	***96.0***	***95.0***	***95.5***

Results

Hebrew	Method	Precision	Recall	F1
25%	CRF	90.5	90.6	90.6
	LSTM	58.2	69.9	63.5
	MW-LSTM	91.7	85.0	88.2
	BMW-LSTM	95.8	90.7	93.2
50%	CRF	94.0	91.5	92.7
	LSTM	52.5	76.2	62.2
	MW-LSTM	92.9	88.8	90.8
	BMW-LSTM	92.9	88.8	90.8
75%	CRF	94.0	92.7	93.4
	LSTM	53.9	73.3	62.2
	MW-LSTM	91.1	92.4	91.8
	BMW-LSTM	92.4	90.4	91.4
	E. BMW-LSTM	93.1	93.2	93.1
100%	CRF	94.9	94.0	94.5
	LSTM	60.3	65.7	62.9
	W-LSTM	91.0	92.8	91.9
	MW-LSTM	93.2	90.1	92.0
	BMW-LSTM	92.7	91.8	92.2
	E. BMW-LSTM	95.2	95.2	95.2

Attention based BLSTM

- The BLSTM as encoder can increase the amount of information available to the network.
- Captures past and future information effectively.
- The attention mechanism in decoder focuses on certain contexts of current character.
- Provide comparable performance to existing best models without feature engineering.
- It treats the segmentation as sequence to sequence problem and pays attention to the parts of input text.

Conditional probability

$$p(y|x) = \prod_i p(y_i|y_0, y_1, \ldots, y_{i-1}, C)$$

Context presentation

$$C_f = \left(\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_{T_x}\right)$$

$$\overrightarrow{h}_i = LSTM_{forward}\left(\overrightarrow{h}_{i-1}, x_i\right)$$

Context information

$$C_b = \left(\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_{T_x}\right)$$

$$\overleftarrow{h}_j = LSTM_{backward}\left(\overleftarrow{h}_{j+1}, x_j\right)$$

$$\hat{z}_i = \tanh(z_i, c_i)$$

$$\hat{z}_i = LSTM(z_{i-1}, y_{i-1}, \hat{z}_{i-1})$$

$$C = (h_1, h_2, \ldots, h_{T_x}).\ g(y_i, \hat{z}_i, c_i) = softmax(y_i \cdot \hat{z}_i).$$

Normalization

$$p(y|S) = softmax\left(\mathrm{W}^{(S)}\mathrm{h}' + \mathrm{b}^{(S)}\right)$$

$$\mathrm{y}' = \mathrm{argmax}\ p'(\mathrm{y}|\mathrm{S})$$

Evaluation Metric

$$P = \frac{\#(correctly\_tagged\_characters)}{\#(characters\_tagged)}$$

$$R = \frac{\#(correctly\_tagged\_characters)}{\#(characers\_should\_be\_tagged)}$$

$$F1 = \frac{2 * P * R}{P + R}$$

Hyper parameters

Model=BLSTM

Optimizer=SGD

Initial state = 0.0

State size =300

Drop out =0.5

Datasets

Finish

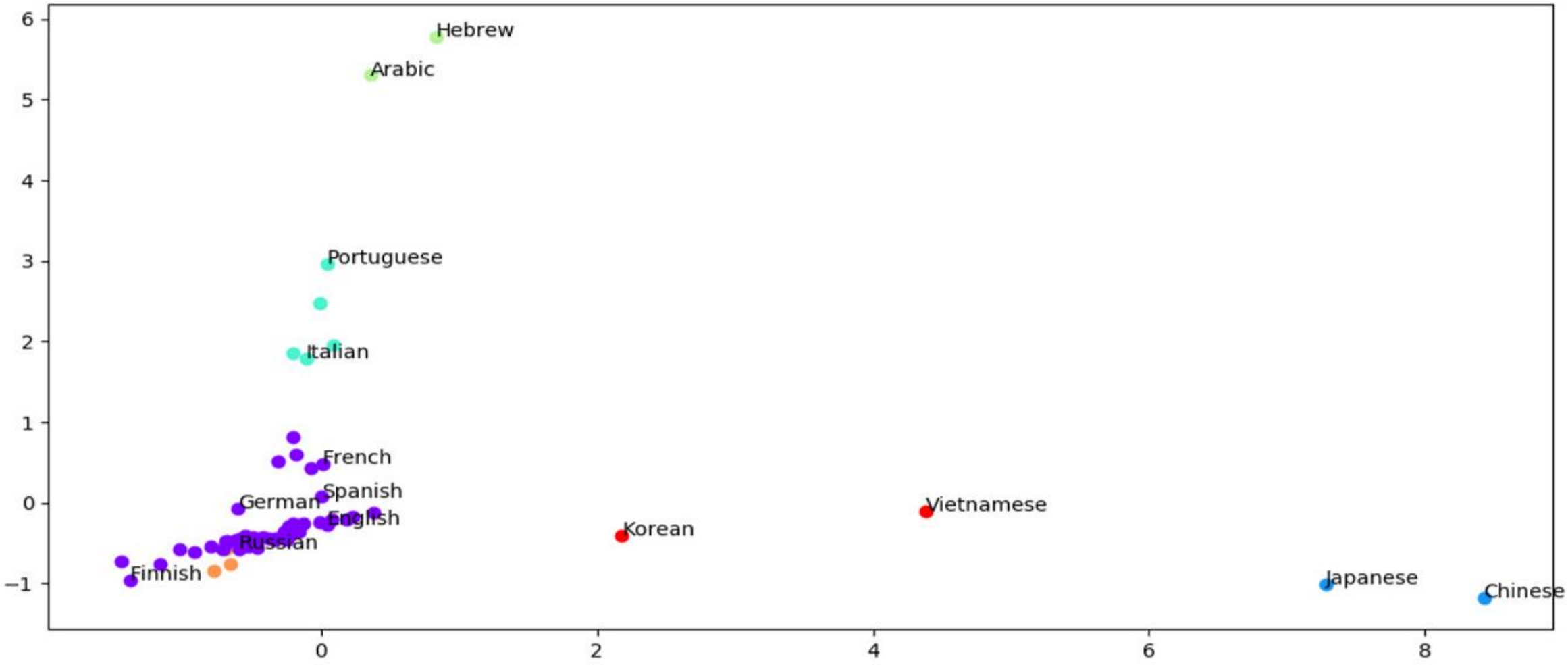
Turkish

English

Results (BMES tag tagging scheme)

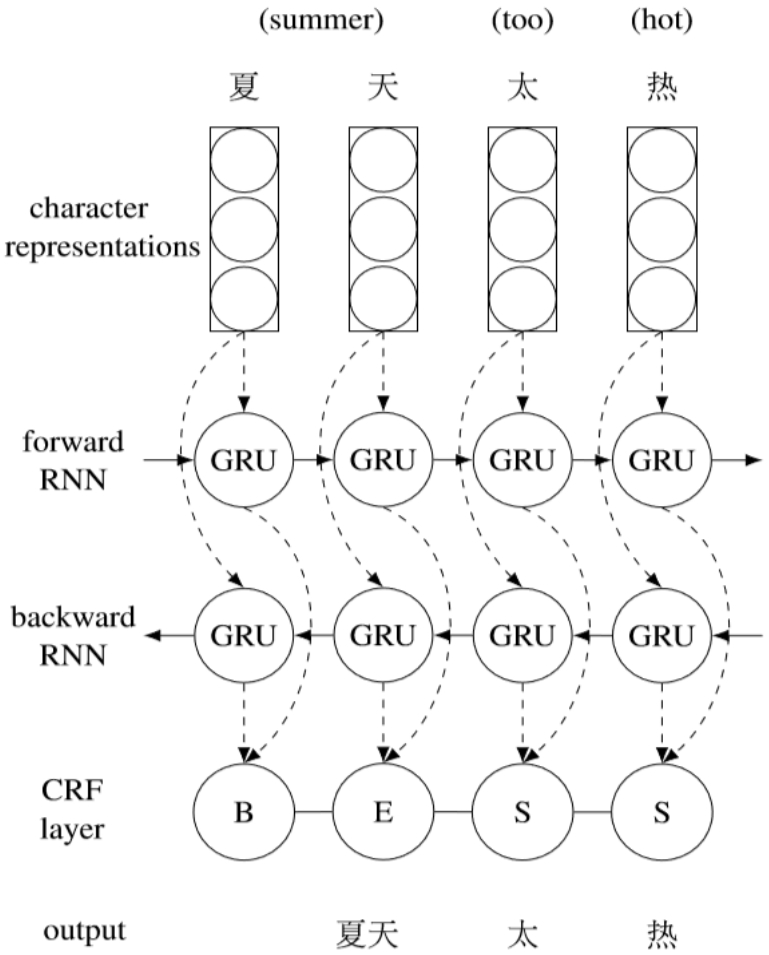
Models	FIN			TURK			ENG		
	P	R	F1	P	R	F1	P	R	F1
CRFs	92.46	91.34	91.90	92.58	92.27	92.42	95.07	93.22	94.39
sL	91.02	90.17	90.59	91.86	90.59	91.22	92.95	92.28	92.61
BiL	91.25	90.36	90.80	91.87	90.78	91.32	93.42	92.64	93.03
BiLa$_{(Attn)}$	92.53	91.62	92.07	92.60	91.30	91.95	95.10	93.24	94.16
CRFs*	92.50	91.38	91.94	92.61	91.32	92.46	95.12	93.26	94.18
sL*	91.05	90.22	90.65	91.88	90.62	91.25	93.00	92.32	92.66
BiL*	91.29	90.38	90.83	91.92	90.80	91.36	93.45	92.67	93.06
BiLa*$_{(Attn)}$	*92.61*	*91.70*	*92.15*	*92.87*	*92.38*	*92.62*	*95.09*	*93.28*	*94.18*

Universal word egmenter



The BiRNN-CRF model for segmentation.

(Huang et al., 2015)



The dashed arrows indicate that dropout is applied.

Language specific setting

1. Languages with word-internal spaces like Vietnamese, first separated punctuation and then use space-delimited syllables for boundary detection.

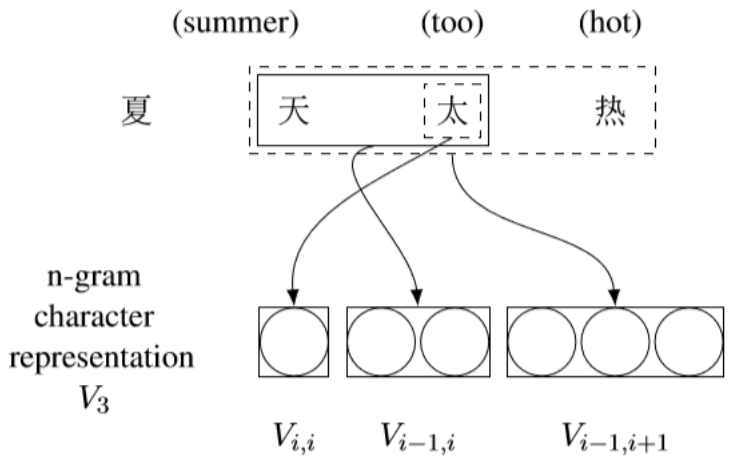
2. Concatenated 3-gram representations. For languages with large character sets and no space delimiters, like Chinese and Japanese,.

3. Encoder-decoder model for transduction with more than 200 unique non segmental multiword tokens, like Arabic and Hebrew.

4. For other languages, the universal model is sufficient without any specific adaptation.

Concatenated 3-gram model

Shao et al. (2017).



Tag Set	Tags	Applied Languages
Baseline Tags	B, I, E, S	Chinese, Japanese, ...
Boundary	X	Russian, Hindi, ...
Transduction	$\overline{\text{B}}$, $\overline{\text{I}}$, $\overline{\text{E}}$, $\overline{\text{S}}$	Spanish, Arabic, ...
Joint Sent. Seg.	T, U	All languages

Tagging scheme

Char.	On considère qu'environ 50 000 Allemands du Wartheland ont péri pendant la période.
Tags	BEXBIIIIIIIEXBIEBIIIIIEXBIIIIEXBIIIIIIIEX$\overline{\text{BE}}$XBIIIIIIIIEXBIEXBIIEXBIIIIIEXBEXBIIIIIES

Implementation

- Grouped sentences with similar lengths into same bucket and padded to the same length.
- Constructed sub-computational graphs for each bucket so that sentences of different lengths are processed more efficiently.
- Used one set of parameters for all the experiments, although fine-tuning the hyperparameters on individual languages might result in additional improvements.
- The encoder-decoder is trained prior to the main network.
- The weights of NNs, including the embeddings, are initialized using the scheme introduced in [Glorot and Bengio (2010)].
- The network is trained using back-propagation.
- All the random embeddings are fine-tuned during training by back-propagating gradients.

Implementation

- Trained Encoder-decoder with unique non-segmental multiword tokens extracted from training set.
- 50 epochs for training and the score of how many outputs exactly match the references is used for selecting weights.
- Word-level F1-score issued to measure the performance of the model after each epoch on the development set.
- The network is trained for 30 epochs and the weight of the best epoch is selected.
- To increase efficiency and reduce memory demand both for training and decoding, we truncate sentences longer than 300 characters.
- At decoding time, the truncated sentences are reassembled at the recorded cut-off points in a post-processing step.

Final Results

Dataset	UDPipe	This Paper	Dataset	UDPipe	This Paper	Dataset	UDPipe	This Paper
Ancient Greek	99.98	99.96	Ancient Greek-PROIEL	99.99	100.0	Arabic	93.77	97.16
Arabic-PUD	90.92	95.93	Basque	99.97	100.0	Bulgarian	99.96	99.93
Catalan	99.98	99.80	Chinese	90.47	93.82	Croatian	99.88	99.95
Czech	99.94	99.97	Czech-CAC	99.96	99.93	Czech-CLTT	99.58	99.64
Czech-PUD	99.34	99.62	Danish	99.83	100.0	Dutch	99.84	99.92
Dutch-LassySmall	99.91	99.96	English	99.05	99.13	English-LinES	99.90	99.95
English-PUD	99.69	99.71	English-ParTUT	99.60	99.51	Estonian	99.90	99.88
Finnish	99.57	99.74	Finnish-FTB	99.95	99.99	Finnish-PUD	99.64	99.39
French	98.81	99.39	French-PUD	98.84	97.23	French-ParTUT	98.97	99.32
French-Sequoia	99.11	99.48	Galician	99.94	99.97	Galician-TreeGal	98.66	98.07
German	99.58	99.64	German-PUD	97.94	97.74	Gothic	100.0	100.0
Greek	99.94	99.86	Hebrew	85.16	91.01	Hindi	100.0	100.0
Hindi-PUD	98.26	98.82	Hungarian	99.79	99.93	Indonesian	100.0	100.0
Irish	99.38	99.85	Italian	99.83	99.54	Italian-PUD	99.21	98.78
Japanese	92.03	93.77	Japanese-PUD	93.67	94.17	Kazakh	94.17	94.21
Korean	99.73	99.95	Latin	99.99	100.0	Latin-ITTB	99.94	100.0
Latin-PROIEL	99.90	100.0	Latvian	99.16	99.56	Norwegian-Bokmaal	99.83	99.89
Norwegian-Nynorsk	99.91	99.97	Old Church Slavonic	100.0	100.0	Persian	99.65	99.62
Polish	99.90	99.93	Portuguese	99.59	99.10	Portuguese-BR	99.85	99.52
Portuguese-PUD	99.40	98.98	Romanian	99.68	99.74	Russian	99.66	99.96
Russian-PUD	97.09	97.28	Russian-SynTagRus	99.64	99.65	Slovak	100.0	99.98
Slovenian	99.93	100.0	Slovenian-SST	99.91	100.0	Spanish	99.75	99.85
Spanish-AnCora	99.94	99.93	Spanish-PUD	99.44	99.39	Swedish	99.79	99.97
Swedish-LinES	99.93	99.98	Swedish-PUD	98.36	99.26	Turkish	98.09	97.85
Turkish-PUD	96.99	96.68	Ukrainian	99.81	99.76	Urdu	100.0	100.0
Uyghur	99.85	97.86	Vietnamese	85.53	87.79	**Average**	98.63	**98.90**

Discussion

The proposed Window LSTM learn the structure of input sequences from raw corpus and predict morphological boundaries.

BLSM networks can deal with variable length of sentences, able to capture past and future information effectively.

Window LSTM provided good results on complex morphological languages with limited amount of training data.

Regular LSTM do not learn effectively on small corpora/ do not give good results. .

The regular LSTM network is outperformed by CRF because of feature engineering.

Both models are language independent and provided good results.

Closing Remarks

The attention BLSTM with decoding has the ability to capture semantic relationship between suffixes of current word and other words in the whole sentence.

The attention mechanism only pays attention to the parts of input text not whole sentence.

Entire word is jointly processed in Window and multi window LSTM.

The model effectively learns sequence structure and predicts sequence boundaries in raw words.

The approach is purely language independent and captures more informative features.

State of the art model from basic BiRNN-CRF to attention based seq2seq transducer.

Difficulties are highlighted for cross different languages.

Minimal language specific setting for complex morphological languages.

Required more training data for non space delimited languages.

The accuracy is related to word boundary makers.

Nearly perfect accuracy on space delimited languages.

Best accuracy for Chinese and Japanese, Also best for Arabic and Hebrew

References

Wang, L., Cao, Z., Xia, Y., & de Melo, G. (2016). Morphological Segmentation with Window LSTM Neural Networks. *In AAAI (pp. 2842-2848).*

Zhu, S. (2018). A Neural Attention Based Model for Morphological Segmentation. *Wireless Personal Communications, (pp. 1-8)*

Shao, Y., Hardmeier, C., Nivre, J., (2018). Universal Word Segmentation: Implementation and Interpretation. *In Transactions of the Association for Computational Linguistics, vol.6 (pp. 421-435).*

Future !!!!!!

Self attention models

!!!!only attention will replace LSTM and CNN in future !!!!. ??

Open AI

Google Brain

Stanford NLP

Best Example (Transformer) by Google

Thank You