Paper sharing

1.Task: POS Tagging in 22 languages

2.Model: Bidirectional LSTM with auxiliary loss

3.Different representations: words,characters and bytes embedding

Details of the model

1. A context bi-LSTM:

$$v_i = \text{bi-LSTM}_{\theta_2}(w_1, w_2, w_3, \ldots, w_n, i) = \text{RNN}_{\theta_2}(w_1, \ldots, w_{i-1}) \circ \text{RNN}_{\theta_2}(w_n, \ldots, w_{i+1})$$

2.Cross-entropy loss:

$$L(\theta_1, \theta_2) + L(\theta_1, \theta_2)$$

1. POS tag 2. log frequency label

Overall results

Additional

Three ways to evaluate:

1.Rare words
2.Data set size
3.Label noise

Rare words

Data set size

Label noise

1. By artificially corrupting training labels.

2. Our initial results show that at low noise rates, bi-LSTMs and TNT are affected similarly: their accuracies drop to a similar degree.

3. Only at higher noise levels (more than 30% corrupted labels), bi-LSTMs are less robust, showing higher drops in accuracy compared to TNT

Paper sharing

1.Task: POS Tagging in 22 languages

2.Model: Bidirectional LSTM with auxiliary loss

3.Different representations: words,characters and bytes embedding

Details of the model

1. A context bi-LSTM:

$$v_i = \text{bi-LSTM}_{\theta_2}(w_1, w_2, w_3, \ldots, w_n, i) = \text{RNN}_{\theta_2}(w_1, \ldots, w_{i-1}) \circ \text{RNN}_{\theta_2}(w_n, \ldots, w_{i+1})$$

2.Cross-entropy loss:

$$L(\theta_1, \theta_2) + L(\theta_1, \theta_2)$$

1. POS tag 2. log frequency label

Overall results

Additional

Three ways to evaluate:

1.Rare words
2.Data set size
3.Label noise

Rare words

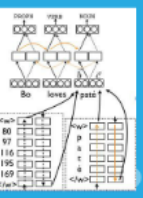
Data set size

Label noise

1. By artificially corrupting training labels.

2. Our initial results show that at low noise rates, bi-LSTMs and TNT are affected similarly: their accuracies drop to a similar degree.

3. Only at higher noise levels (more than 30% corrupted labels), bi-LSTMs are less robust, showing higher drops in accuracy compared to TNT



Paper sharing

1.Task: POS Tagging in 22 languages

2.Model: Bidirectional LSTM with auxiliary loss

3.Different representations: words,characters and bytes embedding

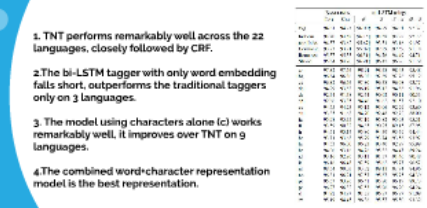
Overall results

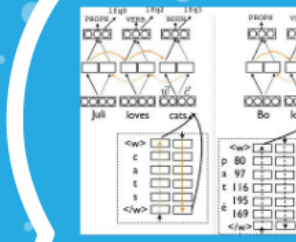
1. TNT performs remarkably well across the 22 languages, closely followed by CRF.

2.The bi-LSTM tagger with only word embedding falls short, outperforms the traditional taggers only on 3 languages.

3. The model using characters alone (c) works remarkably well, it improves over TNT on 9 languages.

4.The combined word+character representation model is the best representation.





Details of the model

1. A context bi-LSTM :

$$v_i = \text{bi-RNN}_{\text{ctx}}(x_{1:n}, i) = \text{RNN}_f(x_{1:i}) \circ \text{RNN}_r(x_{n:i})$$

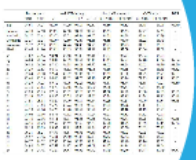
2.Cross-entropy loss:

$$L(\hat{y}_t, y_t) + L(\hat{y}_a, y_a)$$

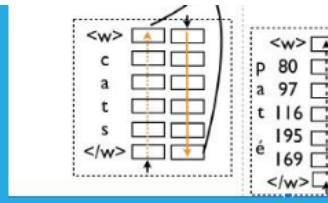
t : POS tag a: log frequency label

Additional

1. Initialize the word embeddings with pre-trained embeddings (POLYGLOT) and combine with the character embedding.
2. Add frequency label (multi-task learning which shares the same parameters)
3. Result: the word embeddings (+POLYGLOT) further improves accuracy and the overall best system is the multi-task bi-LSTM FREQBIN.
4. It is successful in predicting POS for OOV tokens, especially for languages like Ara-bic, Farsi, Hebrew, Finnish.



(PLOYGLOT : The data size is more than 10,000 articles for every language on Wikipedia and each language's vocabulary will contain up to 100,000 words)



Details of the model

1. A context bi-LSTM :

$$v_i = \text{bi-RNN}_{\text{ctx}}(x_{1:n}, i) = \text{RNN}_f(x_{1:i}) \circ \text{RNN}_r(x_{n:i})$$

2.Cross-entropy loss:

$$L(\hat{y}_t, y_t) + L(\hat{y}_a, y_a)$$

t : POS tag a: log frequency label

Overall results

1. TNT performs remarkably well across the 22 languages, closely followed by CRF.

2.The bi-LSTM tagger with only word embedding falls short, outperforms the traditional taggers only on 3 languages.

3. The model using characters alone (c) works remarkably well, it improves over TNT on 9 languages.

4.The combined word+character representation model is the best representation.

	BASELINES		BI-LSTM using:			
	TNT	CRF	$\vec{w}$	$\vec{c}$	$\vec{c}+\vec{b}$	$\vec{w}+\vec{c}$
avg	94.61	94.27	96.00†	94.29	94.01	92.37
Indoeur.	94.70	94.58	96.15†	94.58	94.28	92.72
non-Indo.	94.57	93.62	95.67†	93.51	93.16	91.97
Germanic	93.27	93.21	95.09†	92.89	92.59	91.18
Romance	95.37	95.53	96.51†	94.76	94.49	94.71
Slavic	95.64	94.96	96.91†	96.45	96.26	91.79
ar	97.82	97.56	**98.91**	98.68	98.43	95.48
bg	96.84	96.36	98.02	97.89	97.78	95.12
cs	96.82	96.56	97.80	96.38	96.08	93.77
da	94.29	93.83	96.19	95.12	94.88	91.96
de	92.64	91.38	92.64	90.02	90.11	90.33
en	92.66	93.35	94.46	91.62	91.57	92.10
es	94.55	94.23	95.12	93.06	92.29	93.60
eu	93.35	91.63	94.70	92.48	92.72	88.00
fa	95.98	95.65	97.19	95.82	95.03	95.31
fi	93.59	90.32	94.85	90.25	89.15	87.95
fr	94.51	95.14	95.80	94.39	93.69	94.44
he	93.71	93.63	95.79	93.74	93.58	93.97
hi	94.53	96.00	96.23	93.40	92.99	95.99
hr	94.06	93.16	94.76	95.32	94.47	89.24
id	93.16	92.96	93.11	91.37	91.46	90.48
it	96.16	96.43	97.59	95.62	95.77	96.57
nl	88.54	90.03	93.32	89.11	87.74	84.96
no	96.31	96.21	97.57	95.87	95.75	94.39
pl	95.57	93.96	96.41	95.80	96.19	89.73
pt	96.27	96.32	97.53	95.96	96.20	94.24
sl	94.92	94.77	**97.55**	96.87	96.77	91.09
sv	95.19	94.45	96.36	95.57	95.50	93.32

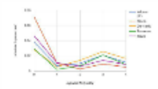
Additional

1. Initialize the word embeddings with pre-trained embeddings (POLYGLOT) and combine with the character embedding.
2. Add frequency label (multi-task learning which shares the same parameters)
3. Result: the word embeddings (+POLYGLOT) further improves accuracy and the overall best system is the multi-task bi-LSTM FREQBIN.
4. It is successful in predicting POS for OOV tokens, especially for languages like Ara-bic, Farsi, Hebrew, Finnish.

	BASELINES		BI-LSTM using:				$\vec{w}+\vec{c}$ +POLYGLOT		OOV ACC		BTS
	TNT	CRF	$\vec{w}$	$\vec{c}$	$\vec{c}+\vec{b}$	$\vec{w}+\vec{c}$	bi-LSTM	FREQBIN	bi-LSTM	FREQBIN	
avg	94.61	94.27	96.00†	94.29	94.01	92.37	**96.50**	**96.52**	83.48	87.98	95.70
Indoeur.	94.70	94.58	96.15†	94.58	94.28	92.72	**96.63**	**96.63**	82.77	87.63	–
non-Indo.	94.57	93.62	95.67†	93.51	93.16	91.97	96.21	**96.28**	87.44	90.39	–
Germanic	93.27	93.21	95.09†	92.89	92.59	91.18	**95.55**	95.49	81.22	85.45	–
Romance	95.37	95.53	96.51†	94.76	94.49	94.71	**96.93**	**96.93**	81.31	86.07	–
Slavic	95.64	94.96	96.91†	96.45	96.26	91.79	97.42	**97.50**	86.66	91.69	–
ar	97.82	97.56	**98.91**	98.68	98.43	95.48	98.87	**98.91**	95.04	96.21	–
bg	96.84	96.36	98.02	97.89	97.78	95.12	**98.23**	97.97	87.40	90.56	97.84
cs	96.82	96.56	97.80	96.38	96.08	93.77	98.02	**98.24**	89.02	91.30	98.50
da	94.29	93.83	96.19	95.12	94.88	91.96	96.16	**96.35**	77.09	86.35	95.52
de	92.64	91.38	92.64	90.02	90.11	90.33	**93.51**	93.38	81.95	86.77	92.87
en	92.66	93.35	94.46	91.62	91.57	92.10	**95.17**	95.16	71.23	80.11	93.87
es	94.55	94.23	95.12	93.06	92.29	93.60	95.67	**95.74**	71.38	79.27	95.80
eu	93.35	91.63	94.70	92.48	92.72	88.00	95.38	**95.51**	79.87	84.30	–
fa	95.98	95.65	97.19	95.82	95.03	95.31	**97.60**	97.49	80.00	89.05	96.82
fi	93.59	90.32	94.85	90.25	89.15	87.95	95.74	**95.85**	86.34	88.85	95.48
fr	94.51	95.14	95.80	94.39	93.69	94.44	**96.20**	96.11	78.09	83.54	95.75
he	93.71	93.63	95.79	93.74	93.58	93.97	96.92	**96.96**	80.11	88.83	–
hi	94.53	96.00	96.23	93.40	92.99	95.99	96.97	**97.10**	81.19	85.27	–
hr	94.06	93.16	94.76	95.32	94.47	89.24	96.27	**96.82**	84.62	92.71	–
id	93.16	92.96	93.11	91.37	91.46	90.48	93.32	**93.41**	88.25	87.67	92.85
it	96.16	96.43	97.59	95.62	95.77	96.57	97.90	**97.95**	83.59	89.15	97.56
nl	88.54	90.03	93.32	89.11	87.74	84.96	**93.82**	93.30	76.62	75.95	–
no	96.31	96.21	97.57	95.87	95.75	94.39	**98.06**	98.03	92.05	93.72	–
pl	95.57	93.96	96.41	95.80	96.19	89.73	**97.63**	97.62	91.77	94.94	–
pt	96.27	96.32	97.53	95.96	96.20	94.24	**97.94**	97.90	92.16	92.33	–
sl	94.92	94.77	**97.55**	96.87	96.77	91.09	**96.97**	96.84	80.48	88.94	–
sv	95.19	94.45	96.36	95.57	95.50	93.32	96.60	**96.69**	88.37	89.80	95.57

(PLOYGLOT : The data size is more than 10,000 articles for every language on Wikipedia and each language's vocabulary will contain up to 100,000 words)

Rare words



1. Especially for Slavic and non-Indoeuropean languages, having high morphologic complexity, most of the improvement is obtained in the Zipfian tail.

2. Rare tokens benefit from the sub-token representations.

Three ways to evaluate:

1. Rare words
2. Data set size
3. Label noise

Three ways to evaluate:

1. Rare words
2. Data set size
3. Label noise

Rare words

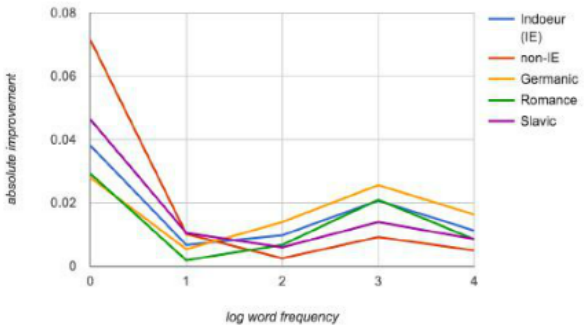


Figure 2: Absolute improvements of bi-LSTM ($\vec{w} + \vec{c}$) over TNT vs mean log frequency.

1. Especially for Slavic and non-Indoeuropean languages, having high morphologic complexity, most of the improvement is obtained in the Zipfian tail.

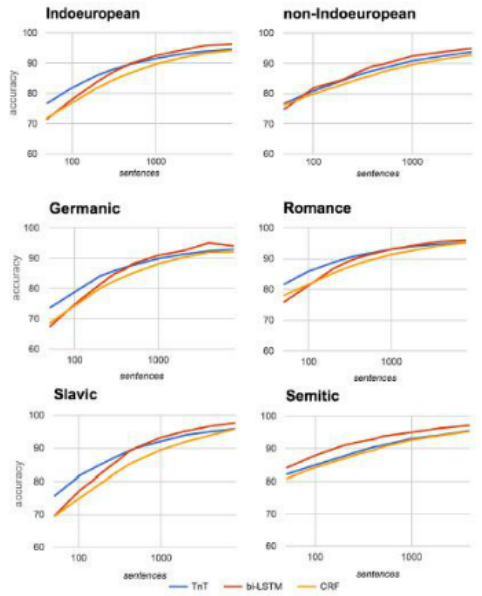
2. Rare tokens benefit from the sub-token representations.

Data set size

1. TNT is better with little data, bi-LSTM is better with more data, and bi-LSTM always wins over CRF.

2. The bi-LSTM model performs already surprisingly well after only 500 training sentences.

3. For non-Indoeuropean languages it is on par and above the other taggers with even less data (100 sen-tences). This shows that the bi-LSTMs often needs more data than the generative markovian model, but this is definitely less than what we expected.



Label noise

1. By artificially corrupting training labels.

2. Our initial results show that at low noise rates, bi-LSTMs and TNT are affected similarly, their accuracies drop to a similar degree.

3. Only at higher noise levels (more than 30% corrupted labels), bi-LSTMs are less robust, showing higher drops in accuracy compared to TNT

Paper sharing

1.Task: POS Tagging in 22 languages

2.Model: Bidirectional LSTM with auxiliary loss

3.Different representations: words,characters and bytes embedding

Details of the model

1. A context bi-LSTM:

$$v_i = \text{bi-LSTM}_{\theta_2}(w_1, w_2, w_3, \ldots, w_n, i) = \text{RNN}_{\theta_2}(w_1, \ldots, w_{i-1}) \circ \text{RNN}_{\theta_2}(w_n, \ldots, w_{i+1})$$

2.Cross-entropy loss:

$$L(\theta_1, \theta_2) + L(\theta_1, \theta_2)$$

1. POS tag 2. log frequency label

Overall results

Additional

Three ways to evaluate:

1.Rare words
2.Data set size
3.Label noise

Rare words

Data set size

Label noise

1. By artificially corrupting training labels.

2. Our initial results show that at low noise rates, bi-LSTMs and TNT are affected similarly: their accuracies drop to a similar degree.

3. Only at higher noise levels (more than 30% corrupted labels), bi-LSTMs are less robust, showing higher drops in accuracy compared to TNT