Robust Multilingual Part-of-Speech Tagging via Adversarial Training

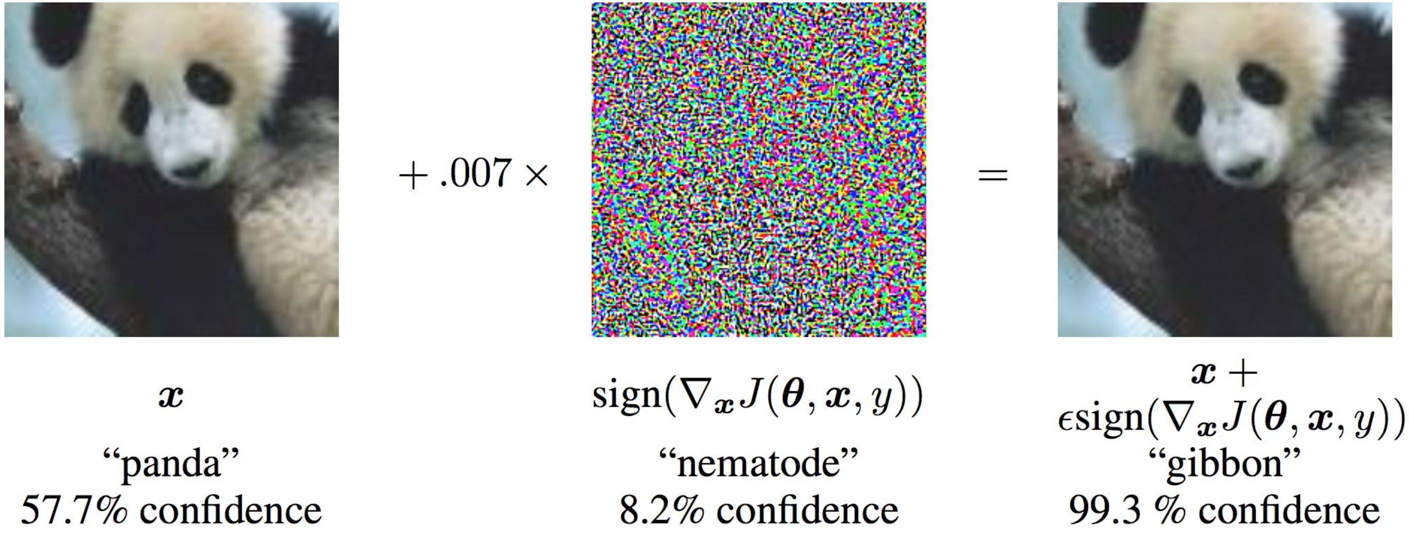
Michihiro Yasunaga, Jungo Kasai, Dragomir Radev

Department of Computer Science, Yale University

Yale - LILY.github.io

Adversarial Examples

Very close to the original input (so should yield the same label) but are likely to be misclassified by the current model


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Adversarial Training (AT)

AT is a regularization technique for neural networks.

1. Generate adversarial examples by adding worst-case perturbations
2. Train on both original examples and adversarial examples

   => improve the model’s robustness to input perturbations

AT has been studied primarily in image classification: e.g.,

- Goodfellow et al. (2015)
- Shaham et al. (2015)

reported success & provided explanation of AT’s regularization effects

Adversarial Training (AT) in … NLP?

Recently, Miyato et al. (2017) applied AT to text classification

=> achieved state-of-the-art accuracy

BUT, the specific effects of AT are still unclear in the context of NLP:

- How can we interpret “robustness” or “perturbation” in natural language inputs?
- Are the effects of AT related to linguistic factors?

Plus, to motivate the use of AT in NLP, we still need to confirm if

- AT is generally effective across different languages / tasks?

Our Motivation

Comprehensive analysis of AT in the context of NLP

- Spotlight a core NLP problem: POS tagging
- Apply AT to POS tagging model
    - sequence labeling, rather than text classification

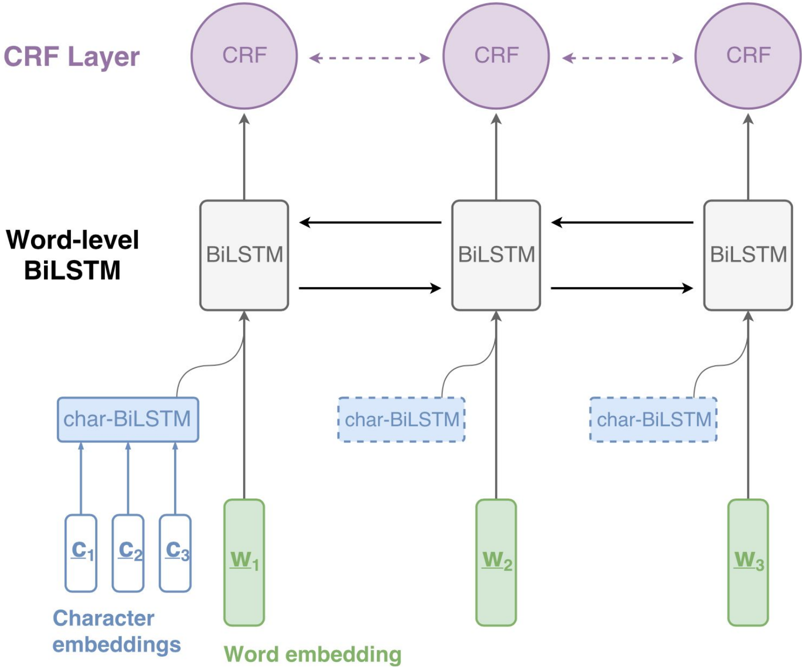
- Analyze the effects of AT:
    - Different target languages
    - Relation with vocabulary statistics (rare/unseen words?)
    - Influence on downstream tasks
    - Word representation learning
    - Applicability to other sequence tasks

Models

Baseline: BiLSTM-CRF
(current state-of-the-art, e.g., Ma and Hovy, 2016)

- Character-level BiLSTM
- Word-level BiLSTM
- Conditional random field (CRF) for global inference of tag sequence
- Input: $\boldsymbol{s} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$
- Loss function:

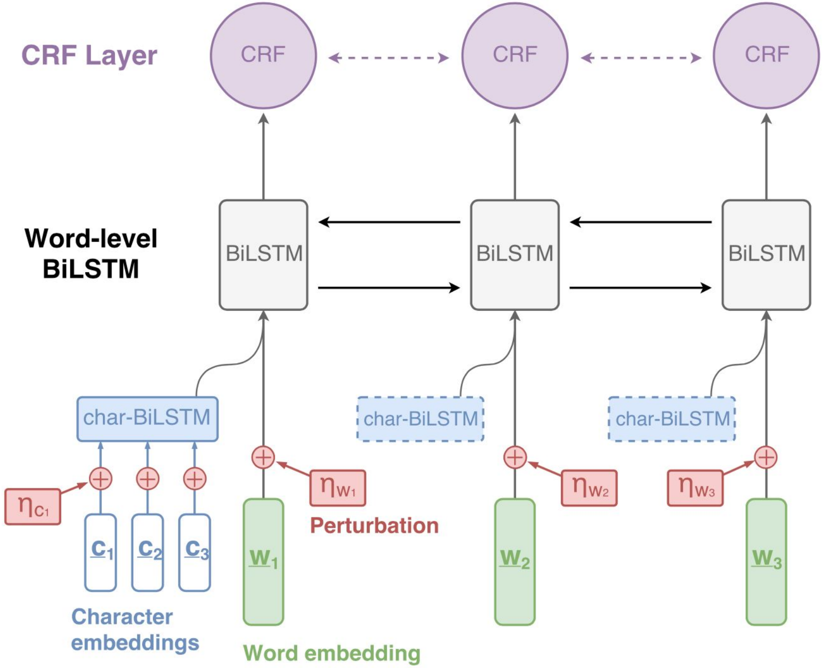
$$L(\boldsymbol{\theta}; \boldsymbol{s}, \boldsymbol{y}) = -\log p(\boldsymbol{y} \mid \boldsymbol{s}; \boldsymbol{\theta})$$



Models (cont’d)

Adversarial training: BiLSTM-CRF-AT

1. Generate adversarial examples by adding worst case perturbations to input embeddings
2. Train with mixture of clean examples & adversarial examples



1. Generating Adversarial Examples

At the input embeddings (dense).

Given a sentence

$$\boldsymbol{s} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$$

generate small perturbations in the direction that significantly increases the loss (worst-case perturbation):

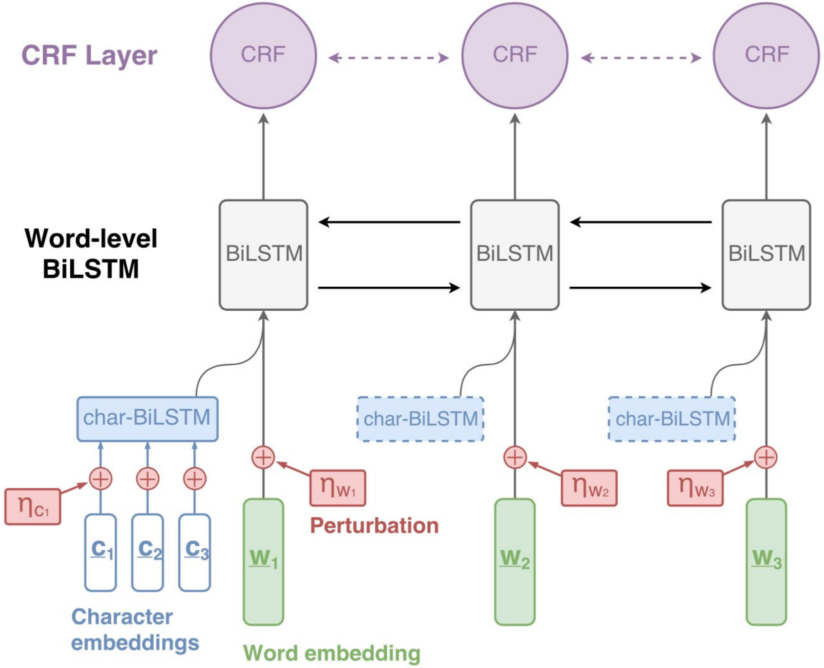
$$\boldsymbol{\eta} = \underset{\boldsymbol{\eta}' : \|\boldsymbol{\eta}'\|_2 \leq \epsilon}{\arg\max} \; L(\hat{\boldsymbol{\theta}}; \boldsymbol{s} + \boldsymbol{\eta}', \boldsymbol{y})$$

approximation:

$$\boldsymbol{\eta} = \epsilon \boldsymbol{g} / \|\boldsymbol{g}\|_2, \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{s}} L(\hat{\boldsymbol{\theta}}; \boldsymbol{s}, \boldsymbol{y})$$

=> Adversarial example:

$$\boldsymbol{s}_{\text{adv}} = \boldsymbol{s} + \boldsymbol{\eta}$$

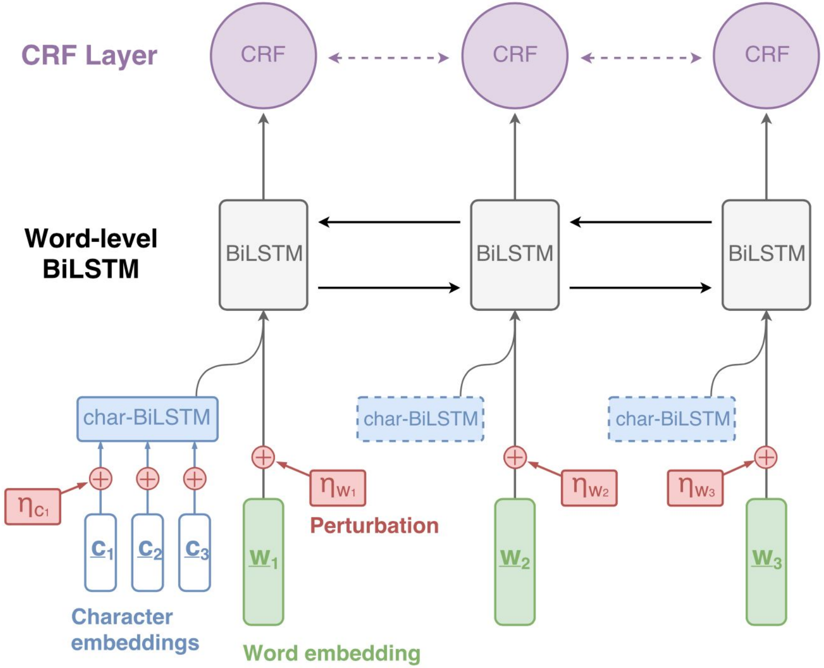


1. Generating Adversarial Examples (cont'd)

Note:

- Normalize embeddings so that every vector has mean 0, std 1, entry-wise.
  - Otherwise, model could just learn embedding of large norm to make the perturbation insignificant

- Set the small perturbation norm ϵ to be $\alpha\sqrt{D}$ (i.e., proportional to $\sqrt{D}$), where D is the dimension of $\boldsymbol{s}$ (so, adaptive).
  - Can generate adversarial examples for sentence of variable length

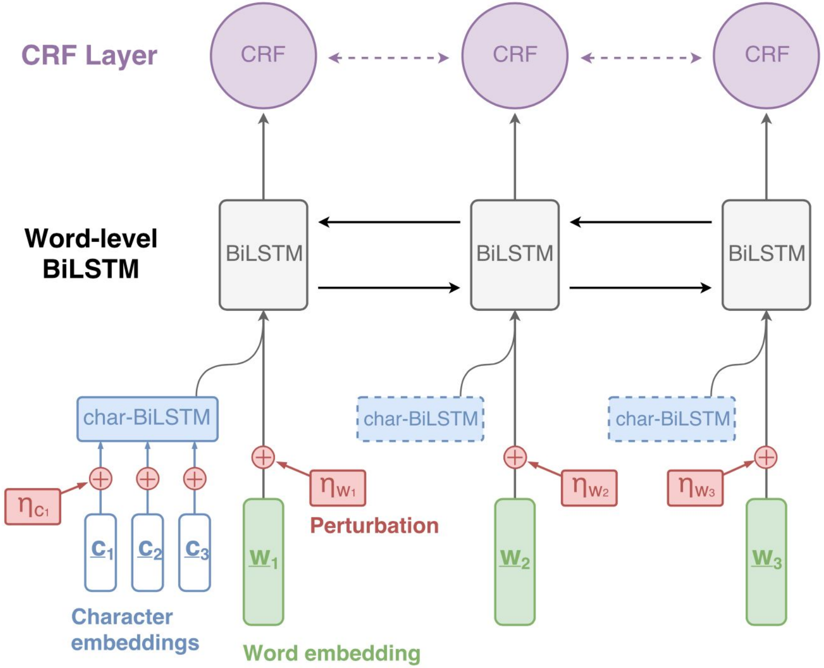


2. Adversarial Training

At every training step (SDG), generate adversarial examples against the current model.

Minimize the loss for the mixture of clean examples and adversarial examples:

$$\tilde{L} = \gamma L(\boldsymbol{\theta}; \boldsymbol{s}, \boldsymbol{y}) + (1-\gamma) L(\boldsymbol{\theta}; \boldsymbol{s}_{\mathrm{adv}}, \boldsymbol{y})$$



Experiments

Datasets:

- Penn Treebank WSJ (PTB-WSJ): English
- Universal Dependencies (UD): 27 languages

for POS tagging

Initial embeddings:

- English: GloVe (Pennington et al., 2014)
- Other languages: Polyglot (Al-Rfou et al., 2013)

Optimization:

Minibatch stochastic gradient descent (SDG)

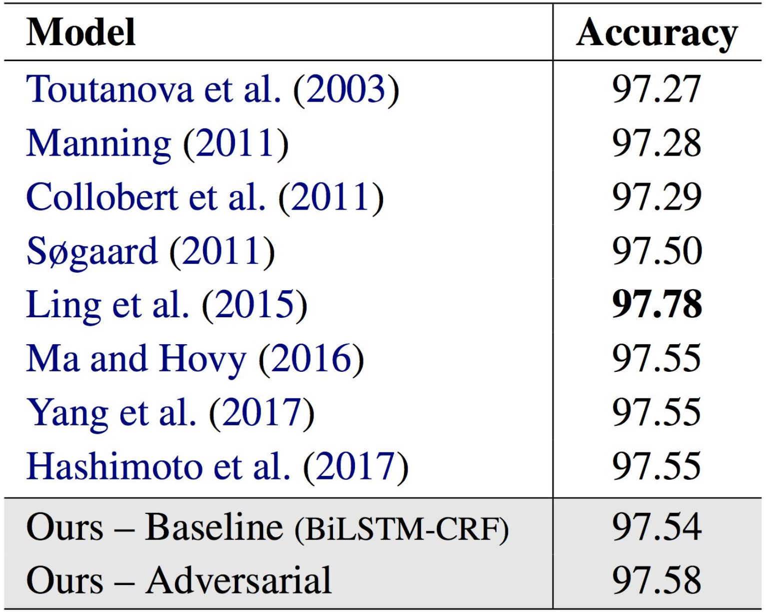
Results

PTB-WSJ (see table):

Tagging accuracy:

97.54 (baseline) → 97.58 (AT)

outperforming most existing works.



Model	Accuracy
Toutanova et al. (2003)	97.27
Manning (2011)	97.28
Collobert et al. (2011)	97.29
Søgaard (2011)	97.50
Ling et al. (2015)	**97.78**
Ma and Hovy (2016)	97.55
Yang et al. (2017)	97.55
Hashimoto et al. (2017)	97.55
Ours – Baseline (BiLSTM-CRF)	97.54
Ours – Adversarial	97.58

UD (27 languages):

Improvements on all the languages

- Statistically significant
- 0.25% up on average

=> AT's regularization is generally effective across different languages.

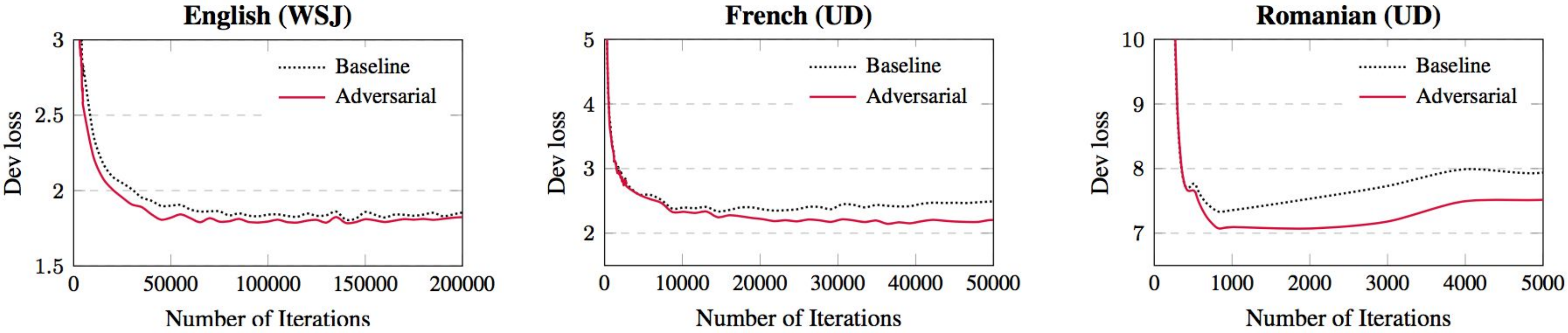
Results (cont'd)

UD (more detail): Improvements on all the 27 languages

- 21 resource-rich: 96.45 → 96.65 (0.20% up on average)
- 6 resource-poor[1]: 91.20 → 91.55 (0.35% up on average)

[1] Less than 60k tokens of training data, as in (Plank et al., 2016)

Learning curves:



Results (observations)

- AT’s regularization is generally effective across different languages

- AT prevents overfitting especially well in low-resource languages
  - e.g., Romanian’s learning curve

- AT can be viewed as a augmentation technique:
  - we generate and train with new examples the current model is particularly vulnerable to, at every step

Further Analysis -- overview

More analysis from NLP perspective:

1. Word-level analysis
   a. Tagging performance on rare/unseen words
   b. Influence on neighbor words? (sequence model)
2. Sentence-level & downstream task performance
3. Word representation learning
4. Applicability to other sequence labeling tasks

1. Word-level Analysis

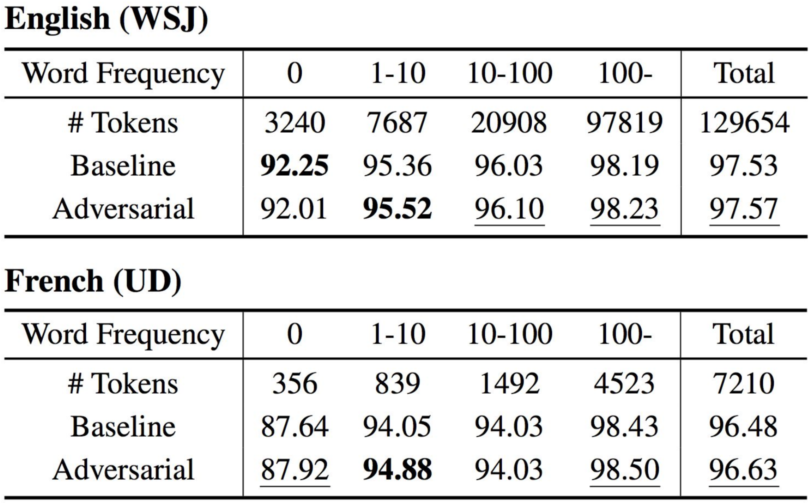
Motivation:

- Poor tagging accuracy on rare/unseen words is a bottleneck in existing POS taggers. Does AT help for this issue?

Analysis:

(a). Tagging accuracy on words categorized by the frequency of occurrence in training.

=> Larger improvements on rare words



English (WSJ)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	3240	7687	20908	97819	129654
Baseline	**92.25**	95.36	96.03	98.19	97.53
Adversarial	92.01	**95.52**	96.10	98.23	97.57

French (UD)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	356	839	1492	4523	7210
Baseline	87.64	94.05	94.03	98.43	96.48
Adversarial	87.92	**94.88**	94.03	98.50	96.63

1. Word-level Analysis (cont'd)

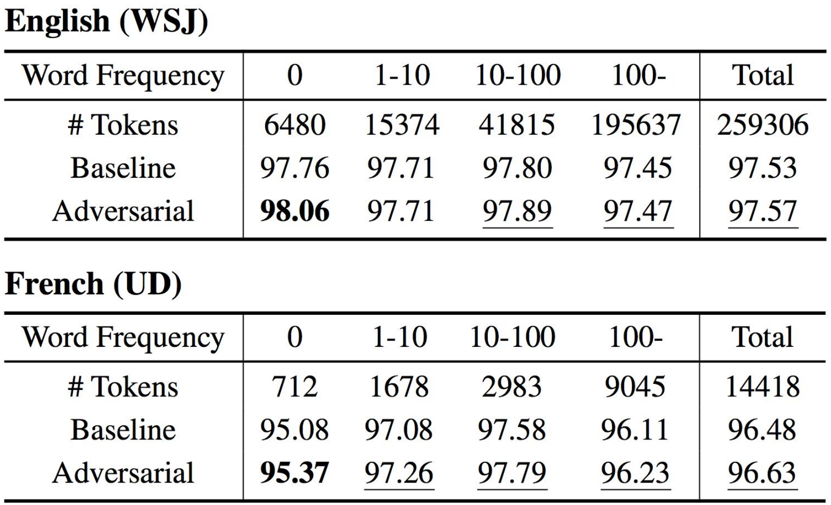
Motivation:

- Poor tagging accuracy on rare/unseen words is a bottleneck in existing POS taggers. Does AT help for this issue?

Analysis:

(b). Tagging accuracy on neighbor words.

=> Larger improvements on neighbors of unseen words



English (WSJ)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	6480	15374	41815	195637	259306
Baseline	97.76	97.71	97.80	97.45	97.53
Adversarial	**98.06**	97.71	97.89	97.47	97.57

French (UD)

Word Frequency	0	1-10	10-100	100-	Total
# Tokens	712	1678	2983	9045	14418
Baseline	95.08	97.08	97.58	96.11	96.48
Adversarial	**95.37**	97.26	97.79	96.23	96.63

2. Sentence-level Analysis

Motivation:

- Sentence-level accuracy is important for downstream tasks, e.g., parsing (Manning, 2014). Is AT POS tagger useful in this regard?

Analysis:

- Sentence-level POS tagging accuracy
- Downstream dependency parsing performance

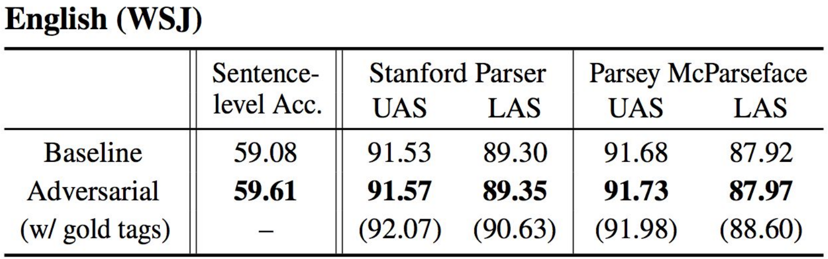
2. Sentence-level Analysis (cont'd)

Analysis:

- Sentence-level POS tagging accuracy
- Downstream dependency parsing performance

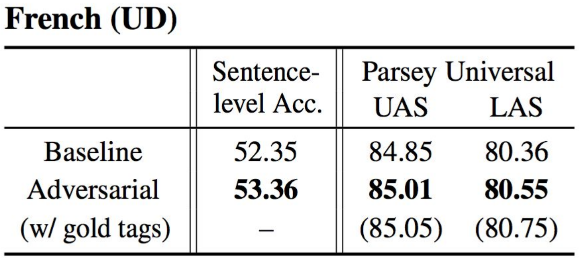
Observations:

- Robustness to rare/unseen words enhances sentence-level accuracy
- POS tags predicted by the AT model also improve downstream dependency parsing



English (WSJ)

	Sentence-level Acc.	Stanford Parser UAS	Stanford Parser LAS	Parsey McParseface UAS	Parsey McParseface LAS
Baseline	59.08	91.53	89.30	91.68	87.92
Adversarial	**59.61**	**91.57**	**89.35**	**91.73**	**87.97**
(w/ gold tags)	–	(92.07)	(90.63)	(91.98)	(88.60)



French (UD)

	Sentence-level Acc.	Parsey Universal UAS	Parsey Universal LAS
Baseline	52.35	84.85	80.36
Adversarial	**53.36**	**85.01**	**80.55**
(w/ gold tags)	–	(85.05)	(80.75)

3. Word representation learning

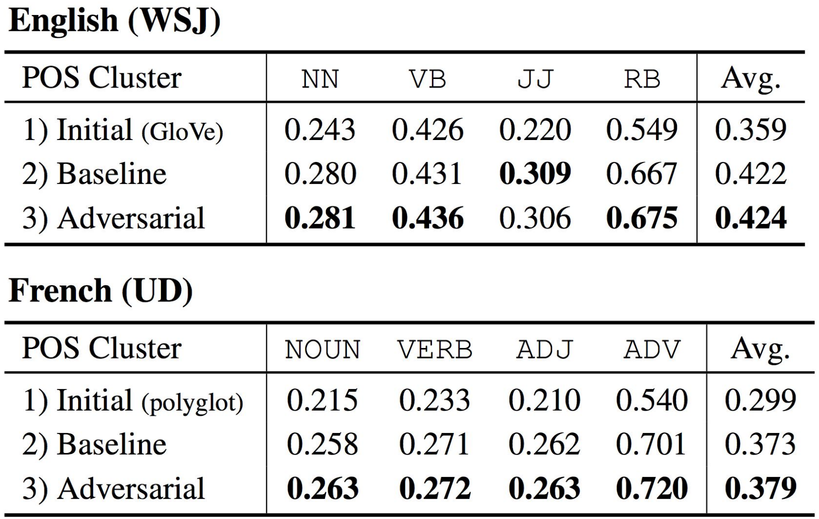
Motivation:

- Does AT help to learn more robust word embeddings?

Analysis:

- Cluster words based on POS tags, and measure the tightness of word vector distribution within each cluster (using cosine similarity metric)
- 3 settings: beginning, after baseline / adversarial training

=> AT learns cleaner embeddings (stronger correlation with POS tags)



English (WSJ)

POS Cluster	NN	VB	JJ	RB	Avg.
1) Initial (GloVe)	0.243	0.426	0.220	0.549	0.359
2) Baseline	0.280	0.431	**0.309**	0.667	0.422
3) Adversarial	**0.281**	**0.436**	0.306	**0.675**	**0.424**

French (UD)

POS Cluster	NOUN	VERB	ADJ	ADV	Avg.
1) Initial (polyglot)	0.215	0.233	0.210	0.540	0.299
2) Baseline	0.258	0.271	0.262	0.701	0.373
3) Adversarial	**0.263**	**0.272**	**0.263**	**0.720**	**0.379**

4. Other Sequence Labeling Tasks

Motivation:

- Does the proposed AT POS tagging model generalize to other sequence labeling tasks?

Experiments:

- **Chunking (PTB-WSJ)**.

  F1 score: 95.18 (baseline) → 95.25 (AT)
- **Named entity recognition (CoNLL-2003)**.

  F1 score: 91.22 (baseline) → 91.56 (AT)

=> The proposed AT model is generally effective across different tasks.

Conclusion

AT not only improves the overall tagging accuracy! Our comprehensive analysis reveals:

1. AT prevents over-fitting well in <u>low resource languages</u>
2. AT boosts tagging accuracy for <u>rare/unseen words</u>
3. POS tagging improvement by AT contributes to downstream task: dependency parsing
4. AT helps the model to learn cleaner word representations

=> AT can be interpreted from the perspective of natural language.

5. AT is generally effective in different languages / different sequence labeling tasks

=> motivating further use of AT in NLP.

Acknowledgment

Thank you to:

Dragomir Radev
Jungo Kasai

Rui Zhang, Jonathan Kummerfeld, Yutaro Yamada

Thank you!

michiyasunaga.github.io/