

# Improving Named Entity Recognition in Tweets via Detecting Non-Standard Words

Chen Li and Yang Liu  
ACL 2015


@最先端NLP

読み手：東北大学 D1 佐々木 彬

# マイクロブログテキスト 解析の難しさ

- Twitter等のマイクロブログテキストには他ドメイン（ニュース記事など）と大きく異なる記述が含まれる

*don't **knw whts** going to happen **nw**...*



NLPにおける従来の解析を行うためには正規化が必要不可欠

以降、こういった正規化が必要な単語を、NSW (Non-Standard-Words) と呼称する

*don't know what's going to happen now...*

# NSWを識別することの意義

- NSWを識別できなければ、正規化の必要がない単語も正規化する恐れがある

*payday* **2mr** 😊

**tomorrow**に正規化  
する必要がある

*want to buy* **ps4**

正規化してはならない

# 辞書でどうにかなるのでは？

- 人名、製品名、企業名等を辞書で網羅することは困難
- 加えて、新語が日々増えていく
- **辞書で全てをカバーすることはできない**

# IV / correct-OOV / ill-OOV

- 本研究では各単語を以下のいずれかに分類

## IV (in-vocabulary)

- 辞書に含まれる単語
- *going, happen, buy*

## correct-OOV (correct out-of-vocabulary)

- 正規化の必要がないOOV
- *PS4, iPhone6, NLP*

## ill-OOV (ill out-of-vocabulary) (= NSW)

- 正規化の必要があるOOV
- *knw, whts, 2mr*

# 本研究の貢献

- NSWを高い精度で識別
  - ※ 正規化は行っていない
  - 後続する解析の精度向上に寄与
- 加えて、NSWとNERを組み合わせた実験
  - マイクロブログテキストに対する、**既存のstate-of-the-artのNERシステムを上回る性能**
  - NSWを識別することの有用性を確認

# マイクロブログテキスト解析の 既存研究

- マイクロブログテキストの正規化に関する研究は多数  
(Liu+, 2012), (Hassan and Menezes, 2013),  
(Yang and Eisenstein, 2013), (Li and Liu, 2014), ...
- ただし、これらはill-OOVの**正規化**のみに着目し、ill-OOVの**識別**は行っていない
  - “ps4”などを正規化してしまう恐れがある

# NSW Detection

単語単位でNSWか否かを判定

## two-step method

- はじめに、辞書に基づいてIVとOOVに分ける
- そのうえで、OOVをcorrect-OOVとill-OOVに分類
- maximum entropy classifier

## 3-way classification

- IV、correct-OOV、ill-OOVの3クラス分類
- CRFs

各々に用いる素性は次ページを参照



# NSW Detectionの主な素性

※ 全素性は末尾付録に記載

## Dictionary Feature

- GNU spell dictionaryに含まれているか否か
- **3-way classification**のみに用いられる

## Lexical Features

- 単語長、母音数、子音数など
- 加えて、上記辞書から学習した文字単位の言語モデル

## Normalization Features

- (Li and Liu, 2014)の単語正規化システムに単語を入れた結果に基づく素性
- 単語に対する正規化候補数などを利用

# NER

- いずれも学習はCRFs
- ※本研究ではNEのClassificationは扱わないあくまでSegmentationのみを対象

## pipeline method

- 先述のNSW Detectionを行ったうえで、その結果を素性に入れてNER ( $isNSW = True/False$ )
- NSW Detectionに失敗すると、後続するNERの性能にも悪影響が出るという問題あり

## joint decoding

- NSW DetectionとNERを別々に学習しjoint decoding
- 次ページに詳細記載

# NERの主な素性

※ 全素性は末尾付録に記載

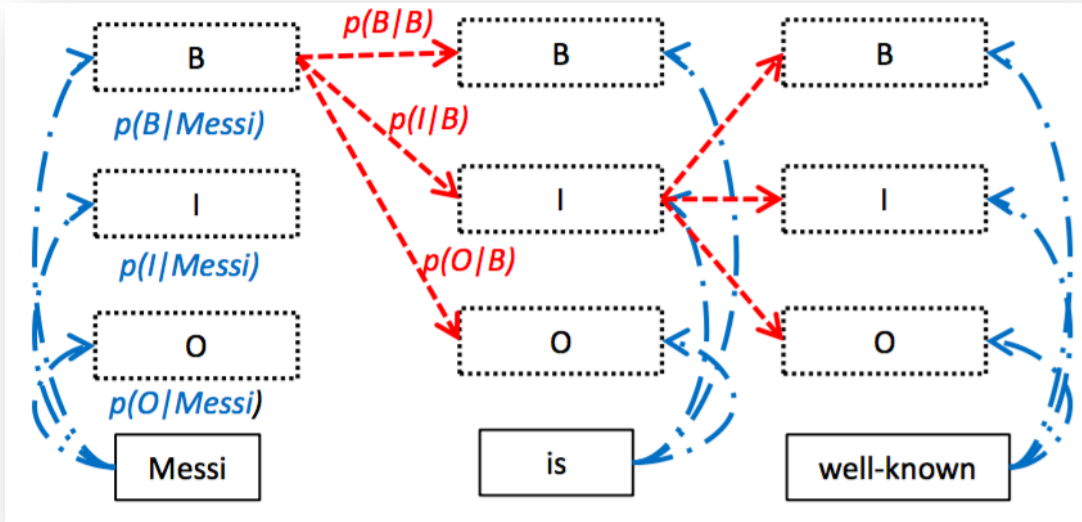
## Basic Features

- word 1,2,3-gram、POS 1,2,3-gram
- 1文字目が大文字か否かの3-gram

## NSW Label Features

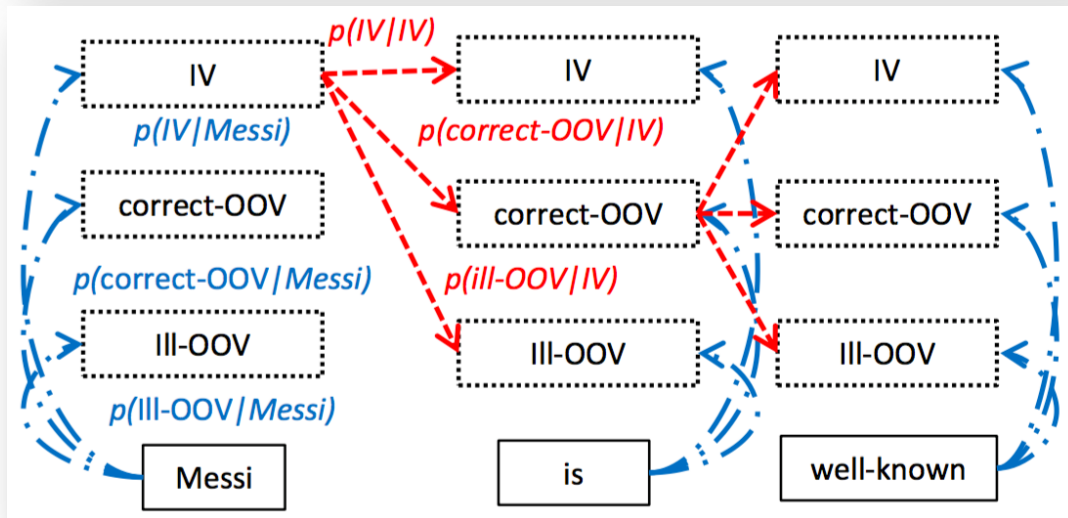
- 辞書中でIVかOOVかの1,2,3-gram
- NSW Detection結果の1,2,3-gram
- 加えて、これらのCompound Feature

# joint decoding (1/2)



NE (BIOタグ)の  
系列ラベリング問題

**pipeline method**  
(p.10)

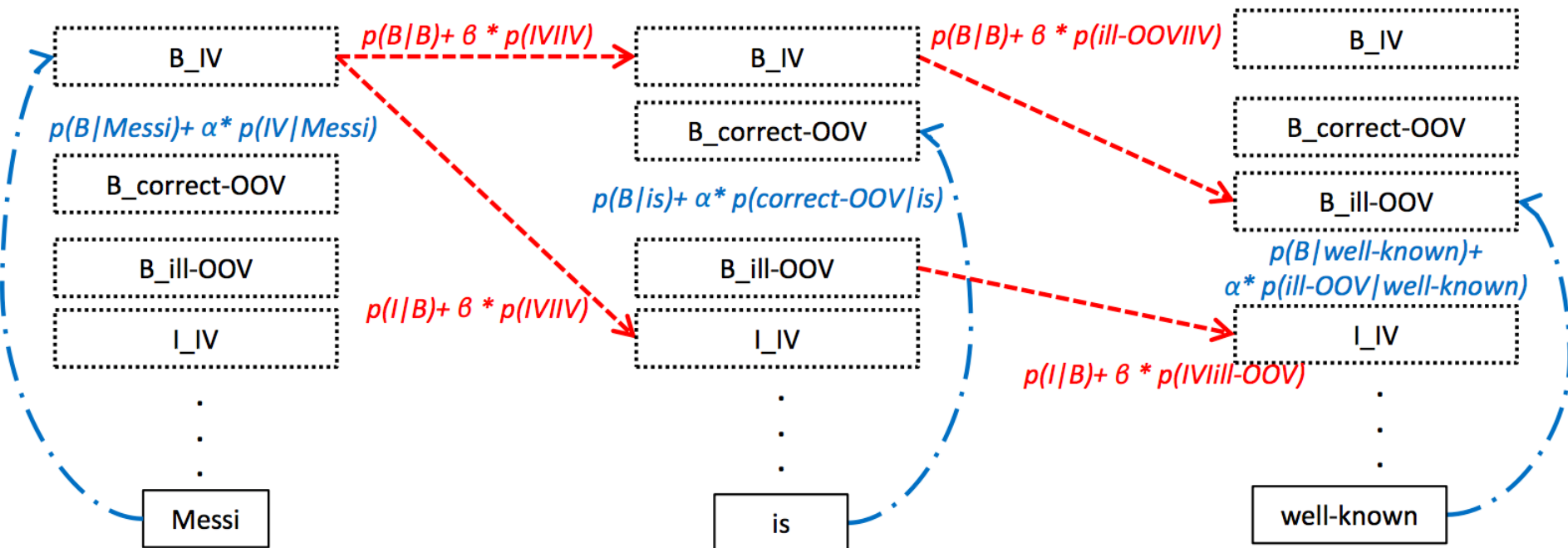


IV / correct-OOV /  
ill-OOVの  
系列ラベリング問題

**3-way classification**  
(p.8)

# joint decoding (2/2)

- 前ページで個別に学習した結果をjoint
- ラベルは(B, I, O)×(IV, correct-OOV, ill-OOV) = 9個



# 訓練・評価用データ

- NSW Detection訓練用データ
  - 2,577ツイート
    - IV: 33,740, correct-OOV: 1,455, ill-OOV: 4,121
- NSW Detection評価用データ
  - Test set 1: 549ツイート (全てill-OOVを含む)
  - Test set 2: 798ツイート ( // )
- NER訓練用・評価用データ
  - 2,396ツイート (NEラベル付き)
  - うち1,012文のみがill-OOVを含む
  - 4分割交差検定

# NSW Detectionの評価結果

System	Test Set 1			Test Set 2		
	R	P	F	R	P	F
Dictionary	88.73	72.35	79.71	67.87	69.59	68.72
Two-step	81.66	88.74	85.05	57.60	90.04	70.26
3-way	87.63	83.49	<b>85.51</b>	73.53	90.42	<b>81.10</b>

Two-step methodでははじめに辞書に基づきOOVを分けるので、3-wayに比べてrecallが悪い

Table 3: NSW detection results.

3-way classification が最も優れた結果に

# NERの評価結果

System	R	P	F
Pipeline w basic features	55.85	74.33	63.76
Pipeline w all features	60.00	77.09	67.40
Joint decoding w all features	73.56	65.02	<b>69.00</b>
(Ritter et al., 2011)	73.00	61.00	67.00

Table 7: NER results from different systems on data from (Ritter et al., 2011).

ツイートに対するNERのState-of-the-art (Ritter et al., 2011) を上回る性能

Ritterらの手法は本研究より多くの外部リソースを利用

- ・ Freebaseのtype list
- ・ brown clusters etc...



# 本研究の貢献

- NSWを高い精度で識別
  - ※ 正規化は行っていない
  - 後続する解析の精度向上に寄与
- 加えて、NSWとNERを組み合わせた実験
  - マイクロブログテキストに対する、既存の **state-of-the-art**のNERシステムを上回る性能
  - joint decodingを採用
  - NSWを識別することの有用性を確認



# 付録

# NSW Detectionの全素性

## Dictionary Feature

1. is token categorized as IV or OOV by the given dictionary (Only used in 3-way classification)

## Lexical Features

2. word identity
3. whether token's first character is capitalized
4. token's length
5. how many vowel character chunks does this token have
6. how many consonant character chunks does this token have
7. the length of longest consecutive vowel character chunk
8. the length of longest consecutive consonant character chunk
9. whether this token contains more than 3 consecutive same character
10. character level probability of this token based on a character level language model

## Normalization Features

11. whether each individual candidate list has any candidates for this token
12. how many candidates each individual candidate list has
13. whether each individual list's top 10 candidates contain this token itself
14. the max number of lists that have the same top one candidate
15. the similarity value between each individual normalization system's first candidate  $w$  and this token  $t$ , calculated by 
$$\frac{\text{longest\_common\_string}(w,t)}{\text{length}(t)}$$
16. the similarity value between each individual normalization system's first candidate  $w$  and this token  $t$ , calculated by 
$$\frac{\text{longest\_common\_sequence}(w,t)}{\text{length}(t)}$$

# NERの全素性

## Basic Features

### 1. Lexical features (word n-gram):

Unigram:  $W_i (i = 0)$

Bigram:  $W_i W_{i+1} (i = -2, -1, 0, 1)$

Trigram:  $W_{i-1} W_i W_{i+1} (i = -2, -1, 0, 1)$

### 2. POS features (POS n-gram):

Unigram:  $P_i (i = 0)$

Bigram:  $P_i P_{i+1} (i = -2, -1, 0, 1)$

Trigram:  $P_{i-1} P_i P_{i+1} (i = -2, -1, 0, 1)$

### 3. Token's capitalization information:

Trigram:  $C_{i-1} C_i C_{i+1} (i = 0)$  ( $C_i = 1$  means this token's first character is capitalized.)

## Additional Features by Incorporating Predicted NSW Label

### 4. Token's dictionary categorization label:

Unigram:  $D_i (i = 0)$

Bigram:  $D_i D_{i+1} (i = -2, -1, 0, 1)$

Trigram:  $D_{i-1} D_i D_{i+1} (i = -2, -1, 0, 1)$

### 5. Token's predicted NSW label:

Unigram:  $L_i (i = 0)$

Bigram:  $L_i L_{i+1} (i = -2, -1, 0, 1)$

Trigram:  $L_{i-1} L_i L_{i+1} (i = -2, -1, 0, 1)$

6. Compound features using lexical and NSW labels:  $W_i D_i, W_i L_i, W_i D_i L_i (i = 0)$

7. Compound features using POS and NSW labels:  $P_i D_i, P_i L_i, P_i D_i L_i (i = 0)$

8. Compound features using word, POS, and NSW labels:

$W_i P_i D_i L_i (i = 0)$