Named Entity Analysis and Extraction with non-Common Words

Anonymous EMNLP submission

Abstract

Most previous research treats named entity extraction and classification as an end-to-end task. We argue that the two sub-tasks should be addressed separately. Entity extraction is related to syntax while entity classification to semantics. According to Noam Chomsky's "Syntactic Structures," pp. 93-94 (1957), syntax is not necessarily related to semantics. We analyze two benchmark datasets and find that *non-common words* can distinguish named entities from common text; where non-common words are the words that appear in named entities and hardly in common text, and they are mainly proper nouns, a kind of syntactic features. Experiments on two benchmark datasets validate that under conditional random fields, lexical and syntactic features achieve state-of-the-art performance on entity extraction and that semantic features do not further improve the extraction performance.

1 Introduction

Named entity recognition (NER) is formally defined by Grishman and Sundheim (1996); Chinchor (1997); Sang and Meulder (2003), aiming to extract named entities from free text and classify the extracted named entities into certain categories. NER includes two sub-tasks: entity extraction and entity classification.[1] While most previous research treats the two sub-tasks as an end-to-end task (i.e., NER), the definitions of the two sub-tasks actually point to different central questions. The central question that entity extraction points to is what distinguishes named entities from common words in free text; while the central question of entity classification is what distinguishes different types of named entities from each other.

Besides different central questions, entity extraction and classification lie at different levels of linguistic analysis. Entity extraction is related to syntax while entity classification to semantics.[2] According to Noam Chomsky's "Syntactic Structures," pp. 93-94 (1957), syntax is not necessarily related to semantics and semantics does not affect syntax. In this paper we focus on the entity extraction, specifically, on the question: *what distinguishes named entities from common text?*

To answer the question we analyze two benchmark datasets (i.e., CoNLL03 (Sang and Meulder, 2003) and OntoNotes*, a derived version of OntoNotes5 (Pradhan et al., 2013)) for the characteristics of named entities and have three findings. First, more than 92.2% of named entities contain *non-common words*, which hardly appear in common text. Second, named entities are mainly made up of proper nouns; in the whole text, more than 84.8% of proper nouns appear in named entities, and within named entities, more than 80.1% of words are proper nouns. Third, entity length follows a family of scale-free power law distributions, with means of less than 2 words.

The characteristics motivate us to design a conditional random fields (CRFs) (Lafferty et al., 2001) based learning method named POM to extract named entities from free text.[3] Specifically, POM defines a constituent-based tagging scheme named POM scheme[4] that consists of

[1] Term clarification: 'entity extraction' means the task of extracting named entities from free text; 'entity classification' means the task of classifying named entities into certain categories; and 'named entity recognition' means the task of treating entity extraction and classification as an end-to-end task.

[2] Although entity extraction does not require to explicitly outline the entities' syntactic structure, to certain extent we still need to know the structure to determine their boundaries. Our analysis (Section 3) shows that non-common words/proper nouns can distinguishes named entities from common text and they are lexical/syntactic features; and our experiments (Section 5) validate that lexical and syntactic features achieve state-of-the-art performance on entity extraction and that semantic features do not further improve the extraction performance. This demonstrates that entity extraction is related to syntax, and not necessarily related to semantics. On the other hand, classifying named entities into different categories to certain extent requires to know the entities' meanings. So entity classification is related to semantics.

[3] If the paper were accepted, we will release the code and data.

[4] We use 'POM' to denote our method and use 'POM scheme' to denote the tagging scheme that POM defines.

three tags: P, O, and M. P encodes the non-common words as Predictors, such as 'African' and 'Chicago.' M encodes the words that Modify predictors in named entity; for example, 'North' modifies 'African' in 'North African' and 'University' modifies 'Chicago' in 'Chicago University.' O indicates the words Outside named entity. In modeling, POM assigns one word with one POM tag under a CRFs framework, mainly with lexical features and syntactic features.

POM is inspired by TOMN (which defines a constituent-based tagging scheme to model time expressions) (Zhong and Cambria, 2018) and like TOMN, POM overcomes the problem of inconsistent tag assignment that is caused by the position-based tagging schemes (e.g., IOBES scheme); POM therefore can fully leverage the information of the non-common words and the information that depends on words, such as part-of-speech (POS). The difference between POM and TOMN lies in the differences between general named entities and time expressions. First, time expressions consist of only a small group of time-related words and those words can be collected wholly (e.g., only 350 distinct words in time expressions across four datasets) (Zhong et al., 2017; Zhong and Cambria, 2018). General named entities instead contain diverse and countless words that it is difficult to collect all of them (e.g., 23,698 distinct words in named entities across CoNLL03 and OntoNotes* datasets). Second, POS tags cannot distinguish time expressions from common text (Zhong et al., 2017; Zhong and Cambria, 2018) and TOMN does not use the POS tags nor other syntactic features. However, POS tags are important features in POM; the non-common words, which can distinguish named entities from common text, are mainly the proper nouns, a kind of POS tags. In practice, POM derives non-common words from training data based on the distinction between named entities and common text. Such strategy addresses the difficulty of collecting the whole entity-related words.

We evaluate POM on two benchmark datasets (i.e., CoNLL03 (Sang and Meulder, 2003) and OntoNotes* (Pradhan et al., 2013)) against two representative state-of-the-art methods (i.e., StanfordNER (Finkel et al., 2005) and LSTM-CRF (Lample et al., 2016)). Experiment results demonstrate the effectiveness and efficiency of POM compared with the state-of-the-art methods. Moreover, the experiments indicate that under CRFs, the lexical and syntactic features (mainly the non-common words and proper nouns) achieve state-of-the-art performance on entity extraction and that the semantic features do not further improve the extraction performance. The performance of lexical/syntactic features and semantic features supports Chomsky's view (1957) that syntax is not necessarily related to semantics and suggests us to address the entity extraction and entity classification separately.

To summarize, we make in this paper the following contributions.

- We recognize from two benchmark datasets the capability of the non-common words to distinguish named entities from common text and discover the beauty of scale-free power law in entity length.
- We design a CRFs-based learning method with a constituent-based tagging scheme for named entity extraction. Our method fully leverages the information of non-common words and addresses the difficulty of collecting the whole entity-related words.
- We conduct experiments on two benchmark datasets, and the results demonstrate the effectiveness and efficiency of our method. And our method together with Chomsky's syntax theory (1957) suggests to address the entity extraction and classification separately.

2 Related Work

We analyze named entities for their characteristics, and based on the characteristics we design a method for named entity extraction. One of the entity characteristics (i.e., Finding 3) is related to power law in language. The named entity extraction is related to named entity recognition.

2.1 Power Law in Language

Zipf's law reveals that words' rank-frequency relation in corpora follows a family of power law distributions (Zipf, 1936, 1949; Mandelbrot, 1961). Researchers observe that the word length and sentence length can be roughly fitted by variants of Poisson distributions (Wimmer et al., 1994; Best, 1996) and gamma distributions (Sigurd et al., 2004) but not a power law distribution. We now discover that the length of named entities follows a family of scale-free power law distributions, with well-defined means and finite variances.

2.2 Named Entity Recognition

Named entity recognition has a long history. Nadeau and Sekine (2007) review the development of early years (from 1991 to 2006) in terms of languages (e.g., English, German, and Chinese) (Wang et al., 1992; Chen and Lee, 1996; Grishman and Sundheim, 1996; Chinchor, 1997; Sang and Meulder, 2003), text genres (e.g., scientific and journalistic) and domains (e.g., sports and business) (Maynard et al., 2001; Poibeau and Kosseim, 2001; Minkov et al., 2005), statistical learning techniques (e.g., hidden Markov models, maximum entropy models, and conditional random fields) (Bikel et al., 1997; Borthwick et al., 1998; Asahara and Matsumoto, 2003; McCallum and Li, 2003), engineering features (e.g., word-level features and dictionary features) (Bikel et al., 1997; Ravin and Wacholder, 1997; Yu et al., 1998; Collins and Singer, 1999; Collins, 2002; Silva et al., 2004), and shared task evaluations (e.g., ACE, MUC, and CoNLL) (Grishman and Sundheim, 1996; Chinchor, 1997; Sang and Meulder, 2003; Doddington et al., 2004).

Before deep learning era, there are also works that concern several aspects of NER, like leveraging unlabeled data for NER (Liang, 2005), leveraging external knowledge for NER (Kazama and Torisawa, 2007; Ratinov and Roth, 2009; Chiu and Nichols, 2016), nested NER (Alex et al., 2007; Finkel and Manning, 2009), and NER in informal text (Liu et al., 2011; Ritter et al., 2011).

In deep learning era, researchers use neural networks and word embeddings to develop models on CoNLL03 dataset (Collobert et al., 2011; Passos et al., 2014; Huang et al., 2015; Ling et al., 2015; Santos and Guimaraes, 2015; Luo et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Xu et al., 2017; Liu et al., 2018).

POM benefits some features from the traditional methods, and refines the significant features and filters out the insignificant features according to a deeper analysis for the characteristics of named entities. Unlike the neural network based methods that mainly compute the semantic similarities among words, POM focuses on the distinction between named entities and common text. And unlike most NER methods that treat the entity extraction and classification as an end-to-end task, POM together with Chomsky's syntax theory (1957) suggests to address the entity extraction and entity classification separately.

Table 1: Dataset statistics. 'Entire' indicates the entire dataset. '#T' denotes the number of entity types.

Dataset		#Docs	#Words	#Entities	#T
CoNLL03	Train	946	203,621	23,499	4
	Dev.	216	51,362	5,942	
	Test	231	46,435	5,648	
	Entire	**1,393**	**301,418**	**35,089**	
OntoNotes*	Train	2,729	1,578,195	81,222	11
	Dev.	406	246,009	12,721	
	Test	235	155,330	7,537	
	Entire	**3,370**	**1,979,534**	**101,480**	

3 Named Entity Analysis

3.1 Datasets

We use two benchmark datasets for named entity analysis: CoNLL03 and OntoNotes*.[5]

CoNLL03 is a benchmark dataset derived from Reuters RCV1 corpus, with 1,393 news articles between August 1996 and August 1997; it contains 4 entity types: PER, LOC, ORG, and MISC (Sang and Meulder, 2003).

OntoNotes* is a dataset derived from OntoNotes5 dataset (Pradhan et al., 2013). OntoNotes5 is a portion of OnteNotes 5.0 corpus for named entity analysis and consists of 3,370 articles collected from different sources (e.g., newswire and web data) over a long period of time; it contains 18 entity types.[6] Although OntoNotes5 is a benchmark dataset, we find that its annotation is far from perfect. For example, "OntoNotes Named Entity Guidelines (Version 14.0)" states that the ORDINAL includes all the ordinal numbers and the CARDINAL includes the whole numbers, fractions, and decimals, but we find in common text 3,588 numeral words, 7.10% of the total numeral words. Besides, some sequences are annotated inconsistently; for the 'the Cold War,' for example, in some cases the whole sequence is annotated as an entity (i.e., '<ENAMEX>the Cold War</ENAMEX>'; where 'ENAMEX' is the annotation mark) while in some cases only the 'Cold War' is an entity (i.e., 'the <ENAMEX>Cold War</ENAMEX>').

To get a high quality dataset for named entity analysis, we derive a dataset named OntoNotes* from OntoNotes5 by removing the entity types[7] whose entities are mainly composed of numbers

[5] The original CoNLL03 and OntoNotes5 datasets include data in English and other languages, here we focus on the English data.

[6] OntoNotes5's 18 entity types are CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, and WORK_OF_ART.

[7] The removed entity types include CARDINAL, DATE, MONEY, ORDINAL, PERCENT, QUANTITY, and TIME.

Table 2: Percentage of named entities each has at least one word that hardly appears in common text

	Entire	Train	Dev.	Test
CoNLL03	97.77	98.77	99.19	98.62
OntoNotes*	92.91	92.20	95.22	95.61

and ordinals and moving all the 'the' at the beginning of entities and all the ''s' at the end of entities outside their entities (e.g., all the '<ENAMEX>the Cold War 's</ENAMEX>' are changed to 'the <ENAMEX>Cold War</ENAMEX> 's').

In setting the training, development, and test sets, we follow the setting by (Sang and Meulder, 2003) for CoNLL03 and follow the setting[8] by OntoNotes5's author for OntoNotes*. Table 1 summarizes the statistics of the two datasets.

3.2 Findings

Although the two datasets vary in source, corpus size, text genre, generated time, and concern different entity types, we find that their named entities demonstrate some similar characteristics.

Finding 1 *Named entity contains non-common word(s); more than 92.2% of named entities each has at least one word that hardly appears in the common text.*

Table 2 reports the percentage of named entities that have words hardly appearing in common text (case sensitive); 'common text' here means the whole text with named entities excluded. The percentage is computed within a set that contains named entities and common text; and the set can be an entire dataset (e.g., CoNLL03 dataset) or only a splitting set (e.g., CoNLL03's training set). Within a set, for a word w, the rate of its occurrences in named entities over the ones in the whole text is defined by Equation (1).

$$r(w) = \frac{f_{entity}(w)}{f_{entity}(w) + f_{common}(w)} \quad (1)$$

where $f_{entity}(w)$ denotes w's occurrences in named entities while $f_{common}(w)$ denotes its occurrences in common text. If $r(w)$ reaches a threshold t, then the word w is treated as hardly appearing in common text. For CoNLL03 and its splitting sets, t is set by 1; which means the word does not appear in common text. For OntoNotes* and its splitting sets, t is set by 0.95; because its annotation is imperfect: its common text contains some words that should be treated as named entities, such as 'Google' and 'American.' We call such kind of words that hardly appear in common text *non-common words*.

[8] https://github.com/ontonotes/conll-formatted-ontonotes-5.0

Table 3: Top 4 POS tags in named entities and their percentage over the whole tags within named entities (P_{entity}) and over the corresponding tags in the whole text (P_{text})

CoNLL03			OntoNotes*		
POS	P_{entity}	P_{text}	POS	P_{entity}	P_{text}
NNP	83.81	84.82	NNP	77.67	85.88
JJ	5.82	17.57	JJ	4.60	6.77
NN	4.89	6.46	NN	4.57	2.91
NNPS	1.55	94.12	NNPS	2.50	93.04

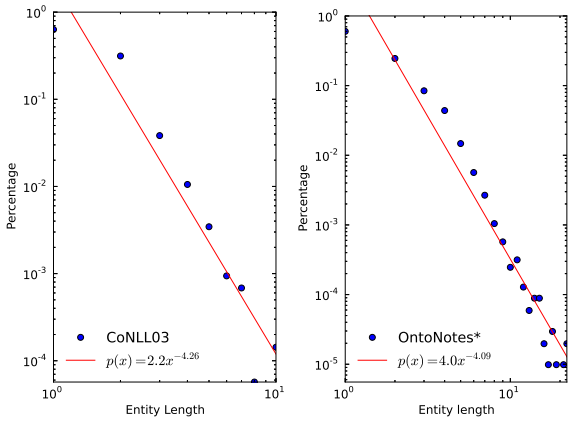
We can see that for a set, more than 92.2% of its named entities contain at least one non-common word; and the phenomenon of non-common words widely exists in CoNLL03 and OntoNotes* datasets as well as their training, development, and test sets. A corollary of this phenomenon is that for a dataset, the non-common words of its development and test sets also hardly appear in the common text of its training set. This suggests that the words of its development set and test set that hardly appear in the common text of its training set are likely to predict named entities.

Finding 2 *Named entities are mainly made up of proper nouns. In the whole text, more than 84.8% of proper nouns appear in named entities; and within named entities, more than 80.1% of the words are proper nouns.*

We find that named entities are mainly made up of proper nouns.[9] Table 3 lists the top 4 POS tags in named entities and their percentage over the whole tags in named entities (P_{entity}) and over the corresponding tags in the whole text (P_{text}). We can see that the top 4 POS tags in both of CoNLL03 and OntoNotes* are the same and they are NNP, JJ, NN, and NNPS. The P_{entity} of proper nouns (including NNP and NNPS) reaches more than 80.1%, and this indicates that named entities are mainly made up of proper nouns. The P_{text} of proper nouns reaches more than 84.8%, and this indicates that in the whole text, the proper nouns mainly appear in named entities.[10] Within named entities, the JJ words are mainly the words that indicate nationalities (e.g., 'American').

[9] If consider the whole OntoNotes5 dataset, then named entities are mainly made up of proper nouns and cardinal numbers.

[10] P_{text} of proper nouns does not reach 100% mainly because individual dataset concerns certain types of named entities and partly because some NNP* words (e.g., 'SURPRISE/NNP DEFEAT/NNP') are POS tagging error.



(a) CoNLL03 (b) OntoNotes*

Figure 1: Length distribution of named entities. Entity length denotes the number of words in named entities.

Table 4: Percentage of named entities in length

Length	CoNLL03	OntoNotes*
1	63.19	60.08
2	31.40	24.53
3	3.83	8.31
≥4	1.58	6.96

Finding 3 *Entity length follows a family of scale-free power law distributions, with well-defined means and finite variances.*

Figure 1 plots the length distribution of named entities in a log-log scale. We can see that the entity length can be fitted by a family of power law distributions $p(x) = Cx^{-\alpha}$, in which the constant C is unimportant and the exponent α is of interest. The power law fits CoNLL03's entity length distribution with $\alpha = 4.26$ (see Figure 1(a)) and fits OntoNotes*'s with $\alpha = 4.09$ (see Figure 1(b)).[11] The two α are all greater than 3, indicating that the two power law distributions have well-defined means and finite variances (Newman, 2005). The power law distributions also possess the scale-free property (Barabási and Albert, 1999).

Table 4 shows the distribution of entity length in percentage. The percentage of one-word entities in CoNLL03 is 63.19% and the one in OntoNotes* is 60.08%. On average, the named entities in CoNLL03 contain 1.45 words and the ones in OntoNotes* contain 1.67 words.

To summarize, an average named entity contains less than two words, with at least one non-common word that is mainly a proper noun. To extract the named entity, it is essential to model its non-common word.

[11] The power law fits original OntoNotes5's entity length with $\alpha = 4.34$.

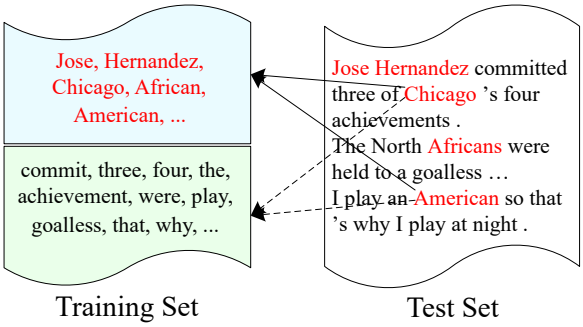


Figure 2: Main idea: words (red font) of test set that appear in training set's named entities (top-left) and hardly in training set's common text (bottom-left) tend to predict named entities. Solid arrow denotes appearing while dashed arrow denotes hardly appearing.

4 POM: Named Entity Extraction with non-Common Words

Finding 1 and 2 suggest that for a dataset, words of its development and test sets that hardly appear in the common text of training set tend to predict named entities; and they are mainly the proper nouns and nationality words. This is our main idea for named entity extraction. Figure 2 visualizes the idea with a simple example: in the test set, words like 'Chicago' and 'American' that appear in the named entities of training set and hardly appear in the common text of training set are likely to predict named entities. Following we illustrate how we develop the idea in POM.

POM models named entities under a CRFs framework and follows the CRFs procedure. POM includes four components: (1) non-common word induction, (2) a set of entity-related word lexicon, (3) a constituent-based POM scheme, and (4) named entity modeling, with the help of non-common words, word lexicon, and POM scheme.

4.1 Non-Common Word Induction

For each dataset, we induce a list of non-common words from its training set.[12] At the beginning, there is an empty list L. For each word w in named entities, we compute its rate ($r(w)$) of hardly appearing in common text by Equation (1). If $r(w)$ reaches a threshold t, then we add w to L. t is set by 1 for CoNLL03 and 0.95 for OntoNotes*. Finding 1 and 3 indicate that the one-word entities contain only the non-common words; so we add all the words of the one-word entities to L.

[12] In analysis (Section 3), we analyze the phenomenon of non-common words using the entire dataset; but in experiments (Section 4 and 5), we induce the non-common words using only the training set.

Table 5: # of entity tokens and trigger words

Category	#Entity Tokens	#Trigger Words
PER	8,424	27
LOC	835	6
ORG	-	78
MISC	207	68

4.2 Word Lexicon

Word lexicon includes two kinds of entity-related words: entity token and modifier. Entity tokens are collected from external sources. We collect from the entity list provided by CoNLL03 shared task (Sang and Meulder, 2003) some PER and LOC entity tokens and from Wikipedia[13] some LOC and MISC entity tokens. Modifiers are collected from training set according to the dataset's annotation guideline and include two kinds: generic modifier and trigger word. Generic modifiers can modify several types of entity tokens, such as 'of' and 'and'; while trigger words modify a specific type of entity tokens, such as 'Inc' modifies ORG entity tokens. Unlike previous works (Kazama and Torisawa, 2007; Ratinov and Roth, 2009) that use lexicon in word sequences, we use lexicon in words. For example, we do not use 'North African' but use 'North' and 'African.' Table 5 summarizes the number of entity tokens and trigger words; the generic modifiers include 18 words. We collect word lexicon for only CoNLL03's entity types and use them for CoNLL03 and OntoNotes* datasets. Note that the word lexicon are collected with only a little effort; and the size of nationality words (i.e., MISC entity tokens) and modifiers is far smaller than the one of distinct words in named entities (404 vs. 23,698).

4.3 POM Scheme

POM scheme consists of three tags: P, O, and M; they indicate the constituents of named entity: Predictor, Modifier, and the words Outside named entity. P encodes the non-common words and entity tokens as predictors. M encodes the generic modifiers and trigger words as modifiers. Predictor and modifier are defined based on the constituent role that they play in named entities. Take 'North African' as an example; 'African' is a predictor because itself alone indicates a named entity while 'North' is a modifier because itself alone does not indicate a named entity.

[13] https://en.wikipedia.org/wiki/Lists_of_cities_by_country for LOC entity tokens; we mainly collect from the European and North American countries. https://en.wikipedia.org/wiki/Lists_of_people_by_nationality for MISC entity tokens.

4.4 Named Entity Modeling

Named entity modeling includes two parts: feature extraction and model learning and tagging.

4.4.1 Feature Extraction

We extract three kinds of features: POM pre-tag features, word cluster features, and lexical & POS features. The ith word in text is denoted by w_i.

POM Pre-tag Features. POM pre-tag features are designed to encode the information of the non-common words and word lexicon. Specifically, a word is pre-tagged by P if it satisfies two conditions: (1) it appears in the list L induced in Section 4.1 or does not appear in the common text of training set; (2) it has a POS tag of NNP* (i.e., NNP or NNPS) or is matched by the entity tokens or is hyphenized by at least one entity token (e.g., 'U.S.-based'). A word is pre-tagged by M if it is matched by the generic modifiers or the LOC, ORG, MISC trigger words; the word that is matched by the PER trigger words is pre-tagged by a separate tag of MP. Other words are pre-tagged by O.

Word Cluster Features. We follow previous works (Miller et al., 2004; Liang, 2005) to derive the prefix paths of 4, 8, and 12 bits from a hierarchical word clusters as features for a word. In experiments, we use the publicly available word clusters; specifically, we use bllip-clusters[14] for CoNLL03 and use the one[15] trained by OntoNotes 5.0 corpus (Pradhan et al., 2013) for OntoNotes*.

Lexical & POS Features. The basic lexical & POS features of w_i include three kinds: (1) the word w_i itself, its lowercase, and its lemma; (2) whether its first letter is capitalized and whether it is the beginning of a sentence; and (3) its POS tag.

For the POM pre-tag features and lexical & POS features, we extract them for w_i in a 5-word window, namely the features of w_{i-2}, w_{i-1}, w_i, w_{i+1}, and w_{i+2}. For the word cluster features we consider them for only the current w_i.

Labeling Tag Assignment. We use POM scheme as our labeling tags. The words outside named entity are assigned with O. Within named entities, the word that appears in the list L or has a POS tag of NNP* or is matched by entity tokens (including those with hyphens) is assigned with P; otherwise it is assigned with M.

[14] http://people.csail.mit.edu/maestro/papers/bllip-clusters.gz

[15] https://drive.google.com/file/d/0B2ke42d0kYFfN1ZSVExLNlYwX1E/view

Table 7: Entity extraction performance of POM and baselines on CoNLL03 and OntoNotes*

Dataset	Method	Dev.			Test		
		$Pr.$	$Re.$	F_1	$Pr.$	$Re.$	F_1
CoNLL03	StanfordNER	96.03	95.27	95.65	93.57	93.11	93.34
	LSTM-CRF	95.60	94.80	95.20	92.19	91.57	91.88
	POM	**96.04**	**95.61**	**95.83**	**94.20**	**93.41**	**93.80**
OntoNotes*	StanfordNER	93.19	91.57	92.38	**93.58**	91.45	92.50
	LSTM-CRF	92.19	**92.63**	92.41	93.03	92.60	92.81
	POM	**93.25**	91.96	**92.60**	93.47	**92.79**	**93.13**

Table 6: Examples of named entity extraction. For the model with entity types in labeling tags, the words that appear together and are tagged with same entity type form a named entity. For the model without entity types, the consecutive non-O words form a named entity. Color background indicates named entities.

Model with entity types in labeling tags
(1) United/M-ORG Arab/P-ORG Emirates/P-ORG 3/O Kuwait/P-LOC 2/O (/O halftime/O 0-2/O)/O ...
(2) Australian/P-MISC Tom/P-PER Moody/P-PER took/O six/O for/O 82/O ...
Model without entity types in labeling tags
(3) United/M Arab/P Emirates/P 3/O Kuwait/P 2/O (/O halftime/O 0-2/O)/O ...
(4) Australian/P Tom/P Moody/P took/O six/O for/O 82/O ...

4.4.2 Model Learning and Tagging

In experiments, POM uses Stanford Tagger[16] to obtain word lemma and POS tags and uses CRFSuite[17] with its default parameters to learn the model and tag the sequences.

Although POM focuses on named entity extraction, the baselines (see Section 5.1) concern the NER task and use the entity types. For fair comparison, we incorporate the entity types into the labeling tags during model learning and tagging but report only the results of entity extraction. The model without incorporating entity types is left to the factor analysis (see Section 5.2.1).

Named Entity Extraction. After sequence tagging, we extract named entities from the tagged sequences. For the model that incorporates entity types in labeling tags, those words that appear together and are tagged with same entity type form a named entity. See Example (1) and (2) in Table 6. For the model without incorporating entity types, those P and M words that appear together form a named entity. See Example (3) and (4) in Table 6.

[16] http://nlp.stanford.edu/software/tagger.shtml

[17] http://www.chokkan.org/software/crfsuite/

5 Experiments

5.1 Setting

We evaluate POM on CoNLL03 and OntoNotes* datasets (detailed in Section 3.1) against two representative state-of-the-art baselines.

Baselines. Our state-of-the-art baselines include StanfordNER (Finkel et al., 2005) and LSTM-CRF (Lample et al., 2016). StanfordNER derives handcrafted features under CRFs with IO scheme (Inside-Outside). LSTM-CRF uses automatic features learned by long short-term memory networks (LSTMs) under CRFs with IOBES scheme (Beginning-Inside-End-Single-Outside). We use StanfordNER as the representative of the traditional methods and use LSTM-CRF as the representative of the neural network based methods. In experiments, we use their source codes to report their performance on the datasets.[18]

Evaluation Metrics. We use the evaluation toolkit[19] of CoNLL03 shared task (Sang and Meulder, 2003) to report the results under the three standard metrics: $Precision$, $Recall$, and F_1.

5.2 Results

Table 7 reports the overall performance of POM and baselines.[20] On CoNLL03, POM outperforms the baselines in all the measures. On OntoNotes*, POM achieves the best F_1. In terms of error reduction, POM reduces 2.89% to 6.90% of errors in F_1. Compared with StanfordNER which mainly treats the training set's named entities as a kind of dictionary, POM explicitly considers the training set's named entities and common text. Such consideration can help extract more unseen named entities.

[18] For StanfordNER, we use its 3.8.0 version with default setting except disuse the features of 'useSequences' and 'usePrevSequences' for saving memory; this setting training on CoNLL03 gets similar results compared with its provided model. For LSTM-CRF, we use its default parameters and its source code can be found at https://github.com/glample/tagger.

[19] http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt

[20] Note that Table 7 and 8 report only the performance of entity extraction.

Consider the LSTM-CRF. According to literature, LSTM-CRF significantly outperforms StanfordNER in NER task on CoNLL03; LSTM-CRF achieves 90.94% of F_1 on CoNLL03's test set (Lample et al., 2016) while StanfordNER achieves only 86.86% (Finkel et al., 2005). However, LSTM-CRF performs comparably with StanfordNER (and worse than POM) in entity extraction. That means the features learned by LSTM-CRF for entity classification do not improve the extraction performance. This coincides with our analysis that lexical and syntactic features can differentiate named entities from common text (see Section 3.2) and supports Chomsky's view (1957) that syntax is not necessarily related to semantics.

5.2.1 Factor Analysis

We conduct controlled experiments to further investigate whether semantic features improve the extraction performance and analyze the impact of the main factors in POM, and report the results on only the test sets (space limited!) in Table 8.

We add to POM the GloVe embeddings, which are trained on Wikipedia 2014 and Gigaword 5 corpora (Pennington et al., 2014). We try all the 50, 100, 200, and 300 dimensional embeddings and the 50 dimensional version achieves the best result and we report that result. We can see that the embedding features do not improve the extraction performance. This confirms that the semantic features do not improve the syntactic extraction.

Table 8 shows that the POM pre-tag features significantly improve the performance, with about absolute 2.0% improvements. This validates the predictive power of the non-common words.

To analyze the effect of addressing the difficulty of collecting the whole entity-related words, we remove the PER and LOC entity tokens from POM but keep the POM pre-tag features and other word lexicon; because the nationality words and modifiers are in a small size and can be collected with little effort (see Section 4.2). We can see that the PER and LOC entity tokens improve the performance but their impact is far less significant than the POM pre-tags'. That means POM addresses that difficulty at the cost of only a little accuracy.

Entity types improve the performance, due to their function of separating some consecutive entities (compare Example (2) and (4) in Table 6). CoNLL03 dataset contains 304 pairs of consecutive entities, 1.73% of the total named entities; OntoNotes* contains 905 pairs, 1.78% of total. A post-processing should be helpful but we do not conduct in POM so as to keep the comparison fair.

Table 8: Impact of factors. '+' indicates adding the kind of factors to POM while '−' indicates removing. 'PER&LOC' denotes the PER and LOC entity tokens.

Dataset	Method	Test		
		$Pr.$	$Re.$	F_1
CoNLL03	POM	**94.20**	**93.41**	**93.80**
	+Embeddings	93.60	92.65	93.12
	−POM Pre-tags	93.26	90.37	91.79
	−PER&LOC	94.16	92.00	93.07
	−Entity Types	93.43	93.15	93.29
	−Word Clusters	94.06	92.58	93.32
OntoNotes*	POM	**93.47**	92.79	**93.13**
	+Embeddings	93.25	92.84	93.04
	−POM Pre-tags	93.09	89.70	91.36
	−PER&LOC	93.41	92.71	93.06
	−Entity Types	92.71	**92.90**	92.80
	−Word Clusters	93.32	92.31	92.81

Table 9: Runtime that POM and baselines cost to complete a whole training and test process

Method	CoNLL03	OntoNotes*
StanfordNER	5 minutes	101 minutes
LSTM-CRF	54 hours	984 hours
POM	4 minutes	53 minutes

Word clusters are helpful in POM (about 0.4% improvement) but the helpfulness is not significant as their impact in some other works (Miller et al., 2004; Liang, 2005; Ratinov and Roth, 2009; Owoputi et al., 2013). Such little helpfulness of word clusters is also reported by Liu et al. (2011). The reason is that the non-common words and POM pre-tag features already play similar role as word clusters in connecting words at the abstraction level and improving the coverage.

5.2.2 Efficiency

Table 9 reports the runtime that POM and baselines cost to complete a whole training and test process on a Mac laptop (8GB Memory×1.4GHz Processor). We can see that POM is more efficient than StanfordNER, especially on the large-scale OntoNotes* dataset. LSTM-CRF is inefficient on both of the CoNLL03 and OntoNotes* datasets.

6 Conclusion

We summarize from two benchmark datasets three characteristics about named entities and design a CRFs-based learning method with a constituent-based tagging scheme for named entity extraction. Experiments show the effectiveness and efficiency of our method and, together with Chomsky's syntax theory (1957), our method suggests to address the entity extraction and classification separately.

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72.

Masayuki Asahara and Yuji Matsumoto. 2003. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 8–15.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science*, 286:509–512.

Karl-Heinz Best. 1996. Word length in old icelandic songs and prose texts. *Journal of Quantitative Linguistics*, 3(2):97–105.

Daniel M. Bikel, Scott L. Miller, Richard M. Schwartz, and Ralph M. Weischedel. 1997. Nymble: A high-performance learning name-finder. In *Proceedings of the fifth Conference on Applied Natural Language Processing*, pages 194–201.

Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. Nyu: Description of the mene named entity system as used in muc-7. In *Proceedings of the 7th Message Understanding Conference*.

Hsin-Hsi Chen and Jen-Chang Lee. 1996. Identification and classification of proper nouns in chinese texts. In *Proceedings of the 16th Conference on Computational Linguistics*, volume 1, pages 222–229.

Nancy A. Chinchor. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference*, volume 29.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton Publishers.

Machael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 489–496.

Ronan Collobert, Jason Weston, Leon Bottou, Machael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(2493-2537).

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the 2004 Conference on Language Resources and Evaluation*, volume 2, pages 1–4.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

Jenny Rose Finkel and Christopher Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. In *https://arxiv.org/abs/1508.01991v1*.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 281–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architecture for named entity recognition. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 260–270.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank F. Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Proceedings of the 32nd AAAI Conference on Artifical Intelligence*.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 359–367.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint named entity recognition and disambiguation. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Benoit Mandelbrot. 1961. On the theory of word frequencies and on related markovian models of discourse. *Structure of Language and its Mathematical Aspects*, 12:190–219.

Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. 2001. Named entity recognition from diverse text types. In *Proceedings of 2001 Recent Advances in Natural Language Processing Conference*, pages 257–274.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Computational Natural Language Learning*.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 443–450.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Mark EJ. Newman. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the 8th Conference on Computational Language Learning*, pages 78–86.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Thierry Poibeau and Leila Kosseim. 2001. Proper name extraction from non-journalistic texts. *Language and Computers*, 37:144–157.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Bjorkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 143–152.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Yael Ravin and Nina Wacholder. 1997. Extracting names from natural-language text. Technical report, IBM Research Division.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 142–147.

Cicero Nogueira Dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the 5th Named Entities Workshop*, pages 25–33.

Bengt Sigurd, Mats Eeg-Olofsson, and Joost van de Weijer. 2004. Word length, sentence length and frequency - zipf revisited. *Studia Linguistica*, 58(1):37–52.

Joaquim Ferreira Da Silva, Zornitsa Kozareva, and Jose Gabriel Pereira Lopes. 2004. Cluster analysis and classification of named entities. In *Proceedings of the 2004 Conference on Language Resources and Evaluation*.

Liang-Jyh Wang, Wei-Chuan Li, and Chao-Huang Chang. 1992. Recognizing unregistered names for mandarin word identification. In *Proceedings of the 14th Conference on Computational Linguistics*, volume 4, pages 1239–1243.

Gejza Wimmer, Reinhard Kohler, Rudiger Grotjahn, and Gabriel Altmann. 1994. Towards a theory of word length distribution. *Journal of Quantitative Linguistics*, 1(1):98–106.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1237–1247.

Shihong Yu, Shuanhu Bai, and Paul S. Wu. 1998. Description of the kent ridge digital labs system used for muc-7. In *Proceedings of the 7th Message Understanding Conference*.

Xiaoshi Zhong and Erik Cambria. 2018. Time expression recognition using a constituent-based tagging scheme. In *Proceedings of the 2018 World Wide Web Conference*, pages 983–992.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 420–429.

George Zipf. 1936. *The Psychobiology of Language*. London: Routledge.

George Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Inc.