Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Does Semantics Aid Syntax? An Empirical Study on Named Entity Recognition and Classification

Xiaoshi Zhong · Erik Cambria* · Amir Hussain

Received: date / Accepted: date

Abstract Many researchers jointly model multiple linguistic tasks (e.g., joint modeling of named entity recognition and named entity classification and joint modeling of syntactic parsing and semantic parsing) with an implicit assumption that these individual tasks can enhance each other via the joint modeling. Before conducting research on jointly modeling multiple tasks, however, such researchers hardly examine whether such assumption is true or not. In this paper, we empirically examine whether named entity classification improves the performance of named entity recognition as an empirical case of examining whether semantics improves the performance of a syntactic task. To this end, we firstly specify the way to determine whether a linguistic task is a syntactic task or a semantic task according to both syntactic theory and semantic theory. After that, we design and conduct extensive experiments on two well-known benchmark datasets using three representative yet diverse state-of-the-art models. Experimental results demonstrate that named entity recognition does not lie at the semantic level and is not a semantic task, instead, it is a syntactic task, and that the joint modeling of named entity recognition and classification does not improve the performance of named entity recognition. Experimental results also demonstrate that traditional hand-crafted-feature models can achieve state-of-the-art performance in comparison with the auto-learned-feature model on named entity recognition.

Keywords Semantics · Syntax · Syntactic task · Named entity recognition · Named entity classification · Named entity recognition and classification

X. Zhong
School of Computer Science and Engineering, Nanyang Technological University, Singapore

E. Cambria
School of Computer Science and Engineering, Nanyang Technological University, Singapore
*Corresponding author (E-mail: cambria@ntu.edu.sg)

A. Hussain
School of Computing, Edinburgh Napier University, United Kingdom

1 Introduction

In the fields of computational linguistics and natural language processing, researchers usually model multiple tasks simultaneously without realizing they are implicitly assuming that these individual tasks can enhance each other under a joint optimization framework. Sometimes such multiple modeling attempts achieve good results, but sometimes these attempts fail. For example, the joint modeling of syntactic and semantic parsings aims to simultaneously formulate both the syntactic parsing and semantic parsing under a framework, but such joint modeling tasks cannot improve the performance of single task; what is even worse, the joint modeling tasks hurt the single task (**????????**). Another famous joint modeling task goes to the classic named entity recognition and classification (NERC), which aims to jointly model named entity recognition (NER) and named entity classification (NEC) as an end-to-end task (**???**), assuming that NER and NEC can enhance each other under a joint optimization framework. However, there is no existing literature that examines whether such implicit assumption is true or not; perhaps these researchers have not yet realized that they make such implicit assumption in those joint modeling tasks.

In this paper, we aim to examine whether a semantic task can improve the performance of a syntactic task. To this end, we specify the way to determine whether a linguistic task is a syntactic task or a semantic task according to **??**'s syntactic theory and Katz & Fodor's foundation of semantic theory (**????**) (see Section 3 for details). To land down our goal in practice, we conduct our examination on a classical linguistic task, namely NERC, which contains two sub-tasks: NER and NEC.[1]

A line of research on NEC (also known as named entity typing) reports that semantic information is much more effective than syntactic information for NEC (**???**). This indicates, according to our specification of syntactic tasks and semantic tasks described in Section 3, that NEC is a semantic task. In this paper, therefore, we focus on NER, and aim to empirically examine the following two questions: (1) whether can the joint NERC task improve the NER performance? (2) whether NER is a syntactic task?

We conduct extensive experiments on two well-known benchmark datasets, namely CoNLL03 (**?**) and OntoNotes* (a derived version from the OntoNotes5 corpus (**?**)) by using three representative state-of-the-art models, namely StanfordNERC (**?**), LSTM-CRF (**?**), and UGTO (**??**). Experimental results demonstrate that (1) the joint NERC task does not improve the NER performance, (2) NER is not a semantic task but a syntactic task, and (3) semantic information does not further improve the NER performance. This suggests us to separately address the two sub-tasks of NER and NEC, and further suggests us as well that before we conduct research on simultaneously modeling multiple linguistic tasks, we should examine whether these multiple tasks could enhance each other. Experimental results also demonstrate that traditional hand-crafted-feature models can achieve state-of-the-art performance in comparison with the auto-learned-feature model on NER.

[1] Term clarification: in this paper, **named entity recognition (NER)** denotes the task of recognizing named entities from unstructed text; **named entity classification (NEC)** denotes the task of classifying these recognized named entities into certain predefined categories; and **named entity recognition and classification (NERC)** denotes the task of treating NER and NEC as an end-to-end joint task.

Although our analysis and experiments demonstrate that neither the NEC task alone nor the joint NERC task can further improve the NER performance, there are still some potential limitations in our work that require to be resolved in the future. One limitation is that our analysis on the NER and NEC tasks is just an empirical case of examining whether semantics or semantic information can improve the performance of a syntactic task. To fully examine the proposition of whether semantics can aid syntax, we still need to examine many other syntactic tasks such as syntactic parsing to see whether semantics or semantic information could improve those syntactic tasks. In the future, we will continue such kinds of examinations to justify the validity or invalidity of this proposition. Another limitation is that although our experiment are designed to learn syntactic information or semantic information from context, we could not guarantee that those models learn only the syntactic information without learning any semantic information, nor that those models learn only the semantic information without learning any syntactic information. What is even worse, it is still not clear whether we could separate the syntactic information from the semantic information. In the future, we will also try to resolve these issues.

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

- We specify the way to empirically examine whether a linguistic task is a syntactic task or a semantic task according to both the syntactic theory and semantic theory.
- We design experiments on NER and NEC as an empirical case to examine whether a semantic task or the joint syntactic and semantic tasks can improve the performance of a syntactic task, or more generally, whether semantics can aid syntax. To the best of our knowledge, this is the first attempt to resolve this problem.
- We conduct extensive experiments and demonstrate that neither the joint NERC task nor the NEC task can improve the NER performance, and our experimental results suggest us to separately address the two sub-tasks of NER and NEC and be carefully examine whether individual tasks could enhance each other before we conduct research on jointly modeling multiple tasks.

The remaining of this paper is organized as follows. In Section 2, we briefly overview the literature that is related to our work, mainly including the joint modelings of syntactic tasks and semantic tasks. After that, we illustrate in Section 3 the way to determine whether a linguistic task is a syntactic task or a semantic task according to both the syntactic theory and semantic theory. In Section 4, we detail our experiment design which aims to examine whether the joint NERC task and semantic information improve the NER performance; and then conduct extensive experiments which demonstrates that the NER task does not lie at the semantic level and is not a semantic task, instead, NER is a syntactic task. Then we describe some potential limitations of our work in Section 5 and finally draw a conclusion and outline some future research in Section 6.

2 Related Works

In this research, we aim to land down the examination of whether semantics aids syntax to examine whether semantic information or semantic tasks can improve the performance of a syntactic task. Those works that are directly related to our research mainly include those research that involves the joint modeling of multiple syntactic and semantic tasks, such as joint syntactic and semantic parsings (**????**) and the end-to-end NERC task (**?????**).

2.1 Syntactic Parsing and Semantic Parsing

There have been considerable efforts trying to jointly model syntactic parsing and semantic parsing under an optimization framework which aims to simultaneously resolve these two parsings in the same time. However, almost all these efforts waste but justify that those attempts that try to jointly model syntactic and semantic parsings will fail in the end. **?** report that their approach for joint syntactic parsing and semantic role labeling gets negative results. In the CoNLL2008 and CoNLL2009 shared tasks on the joint syntactic and semantic parsings, those systems that perform the best are those ones that develop separate syntactic models and semantic models (**??**). Specifically, **?** achieve the best results in the CoNLL2008 shared task by developing separate models; they report that their joint model fails to improve the performance over their separate models. In subsequent research, a series of techniques are employed to develop joint models for syntactic and semantic parsings on the CoNLL2008 and CoNLL2009 datasets, but none of them can further improve the performance in comparison with those best separate models (**????**). **?** jointly model many linguistic tasks, including syntactic tasks and semantic tasks; among their experiments, the joint modeling of multiple syntactic and semantic tasks fails to improve the performance compared with those individual tasks. The most possible reason of these failure is that in theory, syntax and semantics lie at different levels of linguistic analyses, as shown in Figure 1 (**?????**); in practice, the joint modeling of syntactic and semantic tasks requires a trade-off between these two linguistic tasks, and in that trade-off these two linguistic tasks will hurt each other in terms of their performance.

A line of research, which is slightly related to our work, is concerned about the necessity and usefulness of syntactic parsing for semantic analysis (**????**). These empirical results demonstrate that syntactic parsing can significantly improve the performance of semantic analysis, but the premise is that syntactic parsing is finished before semantic parsing starts. These results are consistent with the syntactic and semantic theories (**?????**) as well as the layout of the syntactic-semantic structure, as shown in Figure 1.

2.2 Named Entity Recognition and Classification

NERC research has a long history, with more than two decades' effort devoted to this research topic. Nadeau and Sekine review the development of the early years (**?**) in terms of languages (e.g., English, Chinese, and Persian) (**????**), text genres and domains (e.g., scientific and journalistic) (**??**), statistical learning techniques (e.g., CRFs and maximum entropy models) (**??**), engineering features (e.g., lexical features

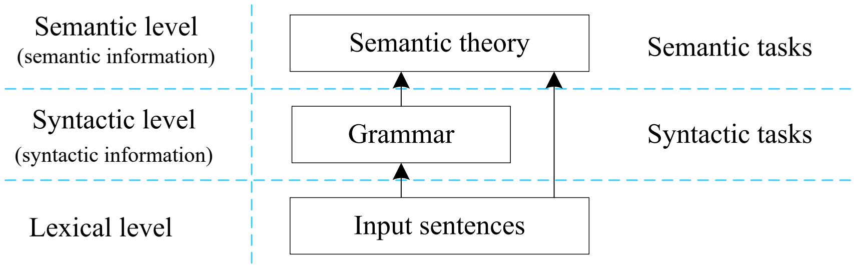

Fig. 1 Relations between syntactic theory and semantic theory (the middle part) and between syntactic tasks and semantic tasks (the right-hand side), with referring to the syntactic theory by (**??**) and the one of semantic theory by (**?**). The lexical level lies at the bottom and includes specific tokens, phrases, and sentences; in the middle is the syntactic level that stores syntactic information for grammar construction and other syntactic tasks; the semantic level lies above the syntactic level where semantic information is for semantic theory development and other semantic tasks.

and dictionary features) (**??**), and shared task evaluations (e.g., MUC, CoNLL, and ACE) (**????**).

Before the deep learning era, there were also works that concern several aspects of NERC, like leveraging unlabeled data for NERC (**?**), leveraging external knowledge for NERC (**??**), nested NERC (**??**), and NERC in informal text (**??**). In the deep learning era, many researchers use neural networks and word embeddings to develop variants of models on the CoNLL03 dataset (**??????????????**). **?** conduct a brief survey on the recent advances in NERC from these deep learning models.

Almost all these research treat the NER and NEC as an end-to-end task, with an implicit assumption that these two sub-tasks (i.e., NER and NER) can enhance each other under an optimization framework for joint modeling which tries to simultaneously resolve the two sub-tasks. However, we could not find any existing works in the literature that examine whether such implicit assumption is true or not before they conduct research to jointly model these two sub-tasks. In this paper, we examine whether such implicit assumption is true or not.

3 Syntactic Task and Semantic Task

In this section, we describe how to empirically determine whether a linguistic task is a syntactic task or a semantic task, according to Chomsky's syntactic theory (**??**) and Katz & Fodor's foundation of semantic theory (**????**).

On the one hand, Chomsky's syntactic theory suggests that syntax does not appeal to semantics; in other words, semantics does not affect the study of syntax (**??**). On the other hand, semantic theory treats syntactic structures (i.e., grammar) as a part of it, but it requires the syntactic analysis to be completed before starting semantic analysis (**????**). According to both the syntactic theory and semantic theory, we outline the relationships between syntax and semantics and between syntactic tasks and semantic tasks in Figure 1, with referring to the Figure 6 and Figure 7 in **?**. It contains three levels in the layout of syntactic-semantic structure: lexical level, syntactic level, and semantic level. In the lexical level, there are specific tokens, phrases, and sentences, which are general units or components we see in languages. The syntactic level lies at the middle and stores syntactic information that is employed for syntactic structures (i.e., grammar) construction and other syntactic tasks. Above the syntactic level is the semantic level where semantic information is stored and is employed for semantic theory development and other semantic tasks.

According to the layout of syntactic-semantic structure shown in Figure 1, we specify the way to empirically determine whether a linguistic task is a syntactic task or a semantic task.

Semantic Task: To verify that a linguistic task is a semantic task, we need only to verify that semantic information is more effective than syntactic information for this task; or empirically speaking, using semantic information achieves higher performance than using syntactic information on this task. Because the semantic level lies above the syntactic level, if the semantic information is more effective for a linguistic task than the syntactic information, then the linguistic task must be a semantic task.

Syntactic Task: To verify that a linguistic task is a syntactic task, we need to conduct experiments that satisfy the following two conditions: (1) using lexical and syntactic information can achieve state-of-the art performance for this linguistic task and (2) adding or using semantic information can not improve the performance on this task. The first condition indicates that this task is at least a syntactic task, and the second condition indicates that this task is not a semantic task. If a linguistic task is at least a syntactic task but not a semantic task, then it must be a syntactic task.

We would like to emphasize that the relationships between syntax and semantics and between syntactic tasks and semantic tasks belong to linguistic phenomena, and they are independent of those statistical models that we employ to process the language text. That means these phenomena will appear in most state-of-the-art models. Therefore, if a linguistic phenomenon is empirically demonstrated to appear in a state-of-the-art model, then theoretically speaking, this linguistic phenomenon will also appear in other state-of-the-art models.

4 Experiments

As mentioned in Section 1, our ultimately goal is to empirically examine the proposition of whether semantics aids syntax; and to this end, we land down the goal by examining whether semantic information can improve the performance of a syntactic task. More specifically, we analyze the classic task of NERC as an empirical case study. In the remaining of this section, we firstly detail the experiment designs as well as their purposes for our examination, and then demonstrate the experimental results and our analysis.

4.1 Experimental Design

We design the following four experiments with two goals. The first goal is to examine whether the joint NERC task can improve the NER performance, and this goal is denoted by *G1*. The second goal is to examine whether semantic information can improve the NER performance, and this goal is denoted by *G2*.

- **Experiment 1** *Do not incorporate entity types into labeling tags in the whole process, including modeling, tagging, and evaluation.*
- **Experiment 2** *Incorporate entity types into labeling tags during modeling and tagging (i.e., training and testing), but not the evaluation.*
- **Experiment 3** *Add word embeddings as features to the model in Experiment 1.*
- **Experiment 4** *Add word embeddings as features to the model in Experiment 2.*

To easily explain the goals of these designed experiments, we let $<\mathbf{X},Y>$ be the representations for words, where $\mathbf{X}$ denotes the feature vectors and Y denotes the labeling tags.

Exp. 1 is a basic experiment for NER, in which a model optimizes for NER with an aim to learn syntactic information from context while without an aim to learn semantic information.[2]

Designing Exp. 2 is to achieve both the goals *G1* and *G2*. On the one hand, incorporating entity types into labeling tags Y during modeling indicates this experiment is a joint NERC task, therefore, Exp. 2 can achieve *G1*, namely to examine whether the joint NERC task can improve the NER performance. On the other hand, entity types are semantic types and carry semantic information, therefore, in Exp. 2, a model optimizes for the joint NERC task with an aim to learn both syntactic information and semantic information from context; and such experiment can achieve *G2*, namely to examine whether semantic information can improve the NER performance.

Designing Exp. 3 is to achieve the goal *G2* by incorporate semantic information into features $\mathbf{X}$. Word embeddings are proposed to capture both the syntactic and semantic information from large corpus (**???**). The Figure 3 in (**?**) suggests that word embeddings capture much more semantic information than syntactic information from context. Therefore, adding word embeddings into a model can incorporate semantic information into features $\mathbf{X}$ for modeling, and thus can examine whether semantic information can improve the NER performance.

Designing Exp. 4 is to achieve both the goals *G1* and *G2* by conducting Exp. 2 and Exp. 3 simultaneously, namely merging Exp. 2 and Exp. 3 into an experiment.

In all the above four experiments, we report only the NER performance by all the models that are used in our experiments. Specifically, after recognizing named entities from unstructured text, we convert tagged text to the CoNLL-format with the BIO scheme for evaluation; the BIO scheme indicates the **B**eginning word, the **I**nside word in a named entity, and those words appearing **O**utside named entities. For Exp. 2 and 4, we remove entity types during evaluation; more specifically, we incorporate entity types into labeling tags during modeling and tagging, but remove entity types during evaluation, so that we can evaluate the impact of the joint NERC task on the NER performance. We do the same conversion and evaluation for all the three used models that are described in Section 4.2.2.

[2] Language context contains both the syntactic and semantic information, and statistical models (e.g., word embeddings (**???**)) can learn both the information from context. A model that is optimized for NER does not aim to learn the semantic information but aims to learn the syntactic from context, while a model that is optimized for NEC aims to learn the semantic information from context. In this paper, we are mainly concerns with the impact of the semantic information that is learned from context for the NER performance.

Table 1 Statistics of datasets

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

4.2 Experimental Setup

In this subsection, we describe the main experimental setup for our experiments, including the experimental datasets, state-of-the-art models, and evaluation metrics.

4.2.1 Datasets

We use the following two well-known benchmark datasets in our experiments: CoNLL03 (**?**) and OntoNotes* (**?**).

CoNLL03 is a benchmark dataset collected from the Reuters RCV1 corpus, and it contains 1,393 news articles written between August 1996 to August 1997 (**?**). This dataset has 4 entity types: PER, LOC, ORG, and MISC; and it is a golden dataset used for the NERC analysis.

OntoNotes* is a clean version of dataset that we derive from the OntoNotes5 dataset (**?**). OntoNotes5 consists of 3,370 articles collected from various sources (e.g., newswire, weblogs, and web data) over a long period of time; and it contains 18 entity types.[3] Although OntoNotes5 is a benchmark dataset, we find that its annotation is far from perfect. For example, the "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 3,588 numeral words in common text, and that is about 7.1% of the total numeral words. In addition, some sequences are annotated inconsistently; for example, for the "the Cold War," in some cases the whole sequence is annotated as a named entity (i.e., "<ENAMEX>the Cold War</ENAMEX>"; where "ENAMEX" is the annotation mark) while in some other cases only the "Cold War" is annotated as a named entity (i.e., "the <ENAMEX>Cold War</ENAMEX>").

To get a high-quality dataset for our analysis, we derive "OntoNotes*" from the OntoNotes5 dataset by removing those entity types[4] whose named entities are mainly composed of numbers and ordinals and moving all the "the" at the beginning of named entities and all the "'s" at the end of named entities to the outside of their

[3] The 18 entity types of the OntoNotes5 dataset are CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PNERCENT, PNERSON, PRODUCT, QUANTITY, TIME, and WORK_OF_ART.

[4] The entity types we remove from the OntoNotes5 dataset to derive the OntoNotes* dataset include CARDINAL, DATE, MONEY, ORDINAL, PNERCENT, QUANTITY, and TIME.

Table 2 Some main characteristics of used experimental models

Model	Feature Type	Learning Framework	Tagging Scheme
StanfordNERC	hand-crafted	CRFs	BIO scheme (position-based)
LSTM-CRF	auto-learned	LSTM & CRFs	IOBES scheme (position-based)
UGTO	hand-crafted	CRFs	UGTO scheme (constituent-based)

named entities. For example, all the "<ENAMEX>the Cold War 's</ENAMEX>" are changed to "the <ENAMEX>Cold War</ENAMEX> 's".

When setting the training, development, and testing sets, for CoNLL03, we follow the original setting described in (**?**); for OntoNotes*, we follow the setting by one of the authors of the OntoNotes5 dataset, and this setting can be found at `https://github.com/ontonotes/conll-formatted-ontonotes-5.0`. Table 1 summarizes the statistics of these datasets.

4.2.2 Models

We use the following three diverse state-of-the-art models in our experiments for analysis: StanfordNERC (**?**), LSTM-CRF (**?**), and UGTO (**??**).

StanfordNERC: StanfordNERC is a traditional and widely used state-of-the-art model that derives hand-crafted features to model named entities under the framework of conditional random fields (CRFs) (**?**) using the position-based BIO scheme (Beginning-Inside-Outside) as the labeling tags (**?**).

LSTM-CRF: LSTM-CRF derives auto-learned features, which are learned by long short-term memory networks (LSTMs) (**?**), to model named entities under the CRFs framework with the IOBES scheme (**B**eginning-**I**nside-**E**nd-**S**ingle-**O**utside) as the labeling tags (**?**).

UGTO: UGTO derives only lexical features and syntactic features, which belong to hand-crafted features, according to an in-depth analysis on some common characteristics of named entities and uses these features to model named entities under CRFs with a constituent-based tagging scheme called UGTO scheme as the labeling tags; the UGTO scheme indicates the **U**ncommon words, **G**eneral modifiers, and **T**rigger words of named entities, and those words appearing **O**utside named entities (**??**).

We use the StanfordNERC model as the representative of those traditional hand-crafted-feature models under CRFs with traditional position-based tagging schemes; use the LSTM-CRF model as the representative of those auto-learned-feature models under CRFs with traditional position-based tagging schemes; and use the UGTO model as the representative of those hand-crafted-feature methods under CRFs with newly constituent-based tagging schemes. Table 2 summarizes three main characteristics of these used experimental models. It shows that all the three models use the CRFs framework, and the main differences among them lie at whether using hand-crafted features or auto-learned features, whether using position-based tagging schemes or constituent-based tagging schemes.

There are other advanced auto-learned-feature models that can be used for the NER task and the joint NERC task, such as (**???**), but we do not use those advanced models but use only the above three models in our experiments, because of the following reasons: (1) our goal is to not to demonstrate the effectiveness of these models but to empirically examine whether the NEC task or the joint NERC task can improve the NER performance as an empirical case of examining whether a semantic task can improve a syntactic task; (2) the relationships between semantics and syntax and between semantic tasks and syntactic tasks are linguistic phenomena that are independent of specific statistical models; if such linguistic phenomena appear in a state-of-the-art model, then these linguistic phenomena will appear in most state-of-the-art models. Therefore, we do not need to use all the advanced models in our experiments; instead, experiments on those representative models can provide enough evidence to validate or invalidate the propositions of our examination.

For the four experiments described in Section 4.1, we conduct all the experiments (i.e., Exp. 1, 2, 3, and 4) for the UGTO and StanfordNERC models, while conduct only Exp. 1 and 2 for the LSTM-CRF model because the LSTM-CRF model already takes into account embedding features for modeling and tagging.

4.2.3 Evaluation Metrics

We use the evaluation toolkit[5] provided by the CoNLL2003 shared task (**?**) to report the results under three standard evaluation metrics: *Precision* (*Pr*), *Recall* (*Re*), and F_1; they are described in Eq. (1), (2), and (3), respectively.

$$Pr = \frac{TP}{TP+FP} \tag{1}$$

$$Re = \frac{TP}{TP+FN} \tag{2}$$

$$F_1 = \frac{2 \times Pr \times Re}{Pr+Re} \tag{3}$$

where TP is the number of named entities that appear in both the system predictions and the ground-truth, FP is the number of named entities that appear in the system predictions but not appear in the ground-truth, and FN is the number of named entities that appear in the ground-truth but not appear in the system predictions.

4.3 Experimental Results

Table 3 reports the overall NER performance of the three models conducted in the four experiments described above on both the CoNLL03 and OntoNotes* datasets. Note again that we are mainly concerned with the NER performance. For the LSTM-CRF model, as mentioned above, since it already leverages auto-learned features and takes into account embedding features, we do not need to conduct Exp. 3 and 4 for it.

In the following few subsections, we analyze these experimental results and demonstrate the empirical examination of our goals *G1* and *G2* described in Section 4.1.

[5] The official version is written by Perl: `http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt`; an alternative version written by Python can be found at `https://github.com/spyysalo/conlleval.py`

Table 3 NER performance of the three models conducted in the four experiments on the CoNLL03 and OntoNotes* datasets. The subscript "$E1$" represents Exp. 1; "$E2$" represents Exp. 2; "$E3$" represents Exp. 3; and "$E4$" represents Exp. 4. The best result under each metric is highlighted in boldface. Because the LSTM-CRF model already takes into account embeddings as auto-learned features, we do not need to conduct Exp. 3 and 4 so as to add embedding features for it.

Dataset	**Model**	**Development Set**			**Testing Set**		
		Pr.	*Re.*	F_1	*Pr.*	*Re.*	F_1
CoNLL03	StanfordNERC$_{E1}$	95.80	95.93	95.86	93.28	93.59	93.43
	StanfordNERC$_{E2}$	**96.43**	95.36	95.89	93.77	92.49	93.13
	StanfordNERC$_{E3}$	95.97	95.82	95.89	93.34	93.46	93.40
	StanfordNERC$_{E4}$	95.78	95.49	95.63	93.64	93.12	93.38
	LSTM-CRF$_{E1}$	94.96	95.46	95.21	92.02	93.48	92.74
	LSTM-CRF$_{E2}$	95.68	94.36	95.02	92.99	91.55	92.27
	UGTO$_{E1}$	95.49	**95.81**	95.65	93.81	**94.44**	**94.12**
	UGTO$_{E2}$	96.14	95.69	**95.92**	**94.29**	93.77	94.03
	UGTO$_{E3}$	95.74	95.79	95.77	93.04	93.66	93.35
	UGTO$_{E4}$	96.07	95.52	95.80	93.85	92.67	93.26
OntoNotes*	StanfordNERC$_{E1}$	92.38	91.62	92.00	93.11	**91.99**	92.54
	StanfordNERC$_{E2}$	**93.17**	91.17	92.16	**93.69**	90.96	92.31
	StanfordNERC$_{E3}$	92.45	91.48	91.96	92.98	91.92	92.45
	StanfordNERC$_{E4}$	93.09	91.16	92.11	93.21	90.88	92.03
	LSTM-CRF$_{E1}$	91.41	91.86	91.64	92.35	91.91	92.13
	LSTM-CRF$_{E2}$	92.52	90.32	91.41	93.37	90.28	91.80
	UGTO$_{E1}$	92.32	92.08	92.20	93.43	91.67	92.55
	UGTO$_{E2}$	92.58	**92.11**	**92.34**	93.60	91.72	**92.65**
	UGTO$_{E3}$	92.06	91.66	91.86	93.38	91.41	92.38
	UGTO$_{E4}$	92.27	91.35	91.81	93.45	91.22	92.32

4.3.1 Experiment 1

Table 3 shows that UGTO$_{E1}$ achieves either the best results or near the best results among all the three models; the differences between UGTO$_{E1}$ and the best results in all the F_1 are less than 0.27%, which ranges within the scope of experimental errors. Note that in Exp. 1, the model does not aim to learn semantic information from context, and that UGTO$_{E1}$ derives only lexical and syntactic features (**?**). That means, using only lexical and syntactic features achieves state-of-the-art performance on the single NER task, and this indicates that the NER task is at least a syntactic task.

Table 3 also shows that StanfordNERC$_{E1}$ performs comparably with UGTO$_{E1}$ and LSTM-CRF$_{E1}$ performs worse than UGTO$_{E1}$. Note again that UGTO$_{E1}$ derives only the lexical and syntactic features. By contrast, both the StanfordNERC and LSTM-CRF models are originally designed for the joint NERC task, with an aim to learn semantic information from context for the NEC task. The experimental results however demonstrate that those semantic information learned by StanfordNERC$_{E1}$ and LSTM-CRF$_{E1}$ does not improve the NER performance. This indicates that the NER task does not lie at the semantic level and is not a semantic task.

4.3.2 UGTO$_{E2}$, UGTO$_{E3}$, UGTO$_{E4}$ vs. UGTO$_{E1}$

We add three public word embeddings into UGTO, and they are (1) word2vec, which is trained on the Google News dataset (**?**), (2) GloVe, which is trained on the Wikipedia

2014 and Gigaword 5 corpora (**?**), and (3) FastText, which is trained on the Wikipedia 2017, UMBC corpus, statmt.org news, and Common Crawl datasets (**??**). We try all the embeddings of word2vec (300-dimension), GloVe (50-, 100-, 200-, and 300-dimension), and FastText (300-dimension) and the GloVe 50-dimension version achieves the best results with the least runtime, therefore we report the results of using GloVe 50-dimension embeddings to analyze the impact of word embeddings features on the NER task.

From Table 3 we can see that UGTO$_{E2}$, UGTO$_{E3}$, and UGTO$_{E4}$ perform either comparably with or worse than UGTO$_{E1}$ on both datasets. The differences of their performance range from 0.23% to 0.86%, which is within the scope of experimental errors. That means, both the semantic information that are incorporated from entity types into labeling tags and incorporated from word embeddings into features do not further improve the NER performance but simply cost additional runtime.[6] This indicates again that the NER task does not lie at the semantic level and is not a semantic task.

4.3.3 StanfordNERC vs. LSTM-CRF

Table 3 shows that the StanfordNERC performs either comparably with or slightly better than the LSTM-CRF in the NER task. According to the literature, however, LSTM-CRF significantly outperforms StanfordNERC in the joint NERC task on the CoNLL03 dataset; specifically, LSTM-CRF achieves the result of F_1 at 90.94% on the test set of the CoNLL03 dataset (**?**) while StanfordNERC achieves the result of F_1 at only 86.86% (**?**). That means those features that are learned by the LSTM-CRF model for the NEC task do not improve the NER performance. Since the NEC task is a semantic task, those features learned by LSTM-CRF for the NEC task mainly includes semantic information; however, those learned semantic information is not effective for the NER task. This therefore indicates again that the NER task does not lie at the semantic level and is not a semantic task.

4.3.4 Experiment 2 vs. Experiment 1

For each model, we compare its performance in Exp. 2 with its performance in Exp. 1 so as to analyze the impact of entity types on the NER performance. Table 3 shows that on both datasets, UGTO$_{E2}$ and UGTO$_{E1}$ achieve similar performance on the NER task; StanfordNERC$_{E2}$ and StanfordNERC$_{E1}$ achieve similar performance on the NER task; and LSTM-CRF$_{E2}$ and LSTM-CRF$_{E1}$ also achieve similar performance on the NER task. That means the semantic information that is incorporated into labeling tags from entity types does not improve the NER performance. This further indicates that the NER task does not lie at the semantic level and is not a semantic task.

[6] The syntactic information from word embeddings does not improve the NER performance, because UGTO$_{E1}$ already leverages sufficient lexical and syntactic information (which includes those syntactic information learned from context) that covers the syntactic information from word embeddings.

Table 4 Controlled experiments using UGTO on the CoNLL03 and OntoNotes* datasets for analyzing the impact of syntactic information on the NER performance. Subscript “$E5$” represents Exp. 5; “$E6$” represents Exp. 6; and “$E7$” represents Exp. 7.

Dataset	**Method**	**Dev. Set**			**Test Set**		
		Pr.	*Re.*	F_1	*Pr.*	*Re.*	F_1
CoNLL03	UGTO$_{E1}$	**95.49**	**95.81**	**95.65**	**93.81**	**94.44**	**94.12**
	UGTO$_{E5}$	94.62	95.29	94.95	91.87	93.41	92.63
	UGTO$_{E6}$	82.63	71.88	76.88	73.70	60.48	66.44
	UGTO$_{E7}$	94.64	95.05	94.84	91.66	92.60	92.13
OntoNotes*	UGTO$_{E1}$	**92.32**	**92.08**	**92.20**	**93.43**	**91.67**	**92.55**
	UGTO$_{E5}$	91.60	91.44	91.52	92.91	91.54	92.22
	UGTO$_{E6}$	80.75	66.50	72.93	82.52	67.81	74.44
	UGTO$_{E7}$	91.41	91.06	91.23	92.80	91.34	92.06

To conclude, the above extensive experimental results demonstrate that the NER task is a syntactic task but not a semantic one, and that the joint NERC task does not improve the NER performance.

4.4 Syntactic Information for the NER Task

Besides the four experiments described in Section 4.1, we also conduct three more controlled experiments using the UGTO model to analyze the impact of syntactic information on the NER task. These three controlled experiments are designed as Exp. 5, 6, and 7, and they demonstrate that (1) syntactic information is effective for the NER task, (2) word embeddings contain some syntactic information that is useful for the NER task, and (3) those syntactic information from word embeddings does not further improve the NER performance.

– **Experiment 5** *Remove the syntactic features from UGTO in Experiment 1, which mainly include the part-of-speech (POS) tags especially the NNP/NNPS tags.*
– **Experiment 6** *Use only the GloVe 50-dimension word embeddings for the NER task in Experiment 1. That is, use only the word embeddings as features under the CRFs framework with the UGTO scheme.*
– **Experiment 7** *Add word embeddings as features to UGTO in Experiment 5.*

The results of these three experiments are reported in Table 4, in which the subscript “$E5$” represents Exp. 5, “$E6$” represents Exp. 6, and “$E7$” represents Exp. 7. Note that in these three experiments, we do not incorporate entity types into labeling tags. For convenient comparison and discussion, Table 4 also reports the results of UGTO$_{E1}$ that is directly copied from Table 3.

4.4.1 UGTO$_{E5}$ vs. UGTO$_{E1}$

Table 4 shows that the performance of UGTO$_{E5}$ decreases in certain extent in comparison with the one of UGTO$_{E1}$. This means that after syntactic features are removed from the UGTO model, its performance is hurt, and this indicates that the syntactic features are effective for the NER task. But we can see that such decrease is not very significant, with only absolute 0.33% to 1.49% in the F_1. The reason is that statistical models like CRFs can learn syntactic information from context; and those syntactic information that is learned from context for the POS tagging can also be learned from context for the NER task. That means the syntactic information either learned directly from context or carried by the POS tags is effective for the NER task.[7]

Our explanation is also supported by those empirical observations which are reported in other works. On the one hand, the Stanford NLP group reports that the StanfordNERC tagger derives similar features as StanfordPOS tagger does and the performance of their tagger benefits little from the POS tags (see the description under Question 16 at `https://nlp.stanford.edu/software/crf-faq.html`). That means those information that is learned for the POS tagging is similar to the those that is learned for the joint NERC task, and it is effective for both the POS tagging and the joint NERC task. Note that POS tags are syntactic types and POS tagging is a syntactic task, therefore those information that is learned for the POS tagging is syntactic information. On the other hand, entity types are semantic types and the research working on the NEC task reports that semantic information is much more effective than syntactic information for the NEC task (**???**), which demonstrates that the NEC task is a semantic task. And since those information that is effective for POS tagging is syntactic information and it is effective for the NERC task but not effective for the NEC task, those syntactic information must be effective for the NER task.

4.4.2 UGTO$_{E6}$ vs. UGTO$_{E1}$

Table 4 suggests that although word embeddings carry certain amount of syntactic information that is effective for the NER task, such quantity is far less than the one that is learned from context by UGTO$_{E1}$. This performance is consistent with the observation that is reported by (**?**): word embeddings capture only a few syntactic information which is far less than those semantic information that is captured by the same word embeddings model (see Figure 3 in their paper).

4.4.3 UGTO$_{E7}$ vs. UGTO$_{E5}$

Table 4 shows that UGTO$_{E7}$ does not perform better than UGTO$_{E5}$. That means the syntactic information carried by word embeddings does not further improve the performance of a state-of-the-art model on the NER task. This is consistent with the observation reported in Section 4.3. The reason, as illustrated above, is that a state-of-the-art model can learn much more syntactic information from context than those syntactic information that is carried by word embeddings. Of cause, as demonstrated before, the semantic information that is carried by word embeddings is effective for the NEC task but not effective for the NER task.)

To conclude for this subsection, our extensive experiments and thorough analysis demonstrate that syntactic information significantly influences the NER performance. This indicates that the NER task is a syntactic task. Together all the results of all the

[7] In fact, the syntactic information that is carried by the POS tags is also learned from context.

seven experiments described in both Section 4.1 and Section 4.4, we demonstrate clearly that the NER task is a syntactic task, and is not a semantic task.

5 Limitations

Although our analysis and experiments demonstrate that neither the NEC task alone nor the joint NERC task nor the semantic information can improve the NER performance, there are still some potential limitations in our work that require to be resolved in the future. In this section, we discuss two such potential limitations.

One limitation is that our analysis on the NER and NEC tasks is just an empirical case of examining whether semantics or semantic information can improve the performance of a syntactic task. To fully examine whether the proposition of whether semantics can aid syntax, we still need to examine many other syntactic tasks (e.g., syntactic structure construction, syntactic parsing (**????**) and time expression recognition (**???????**)) to see whether semantics or semantic information could improve those syntactic tasks. In the future, we will continue such examinations to justify the validity or invalidity of this proposition.

Another limitation is that although our experiment designs (see Section 4.1) try to learn syntactic information or semantic information from context, we could not guarantee that those models learn only the syntactic information without learning any semantic information, nor that those models learn only the semantic information without learning any syntactic information. What is even worse, it is still not very clear whether we could separate the syntactic information from the semantic information. In the future, we will also try to address these issues.

6 Conclusion and Future Work

In this paper, we aim to examine whether the NEC task or the joint NERC task improves the performance of the NER task as an empirical case study of examining whether semantic information can improve the performance of a syntactic task. To this end, we firstly introduce the way to determine whether a linguistic task is a syntactic task or a semantic task. After that, we design seven experiments on the NER and NERC task to analyze the impacts of semantic information and syntactic information on the NER performance, and conduct these experiments using three representative yet diverse state-of-the-art models on two well-known benchmark datasets. Experimental results demonstrate that neither the joint NERC task nor the semantic information can improve the NER performance. This indicates that the NER task is not a semantic task but a syntactic task. Our analysis suggests us to separately address the two subtasks of NER and NEC; moreover, our analysis also suggests us to examine whether individual tasks can enhance each other under an optimization framework before we jointly model these multiple individual tasks in a framework. In the future work, we plan to investigate these issues we discuss in the previous section; namely, we plan to conduct experiments on other syntactic tasks to empirically examine whether semantics aids syntax and plan to separate semantic information from syntactic information and then investigate their impacts on linguistic tasks.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback which helps us improve the quality of our manuscript. This research/project is supported by A*STAR under its Industry Alignment Fund (LOA Award I1901E0046).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.