Triple Extraction Using Lexical Pattern-based Syntax Model

Ai Loan Huynh, Hong Son Nguyen and Trong Hai Duong

Abstract This work proposed a new approach to extract relations and their arguments from natural language text without knowledge base. Using the grammar of English language, it allows detecting sentence based on verb types and phrasal verb in terms of extraction. In addition, this approach is able to extract the properties of objects/entities mentioned in text corpus, which previous works have not yet explored. Experimental result is performed by using various real-world datasets which were used by ClausIE and Ollie, and other text were found in the Internet. The result shows that our method is significant in comparison with ClausIE and Ollie.

Keywords Tripe extraction · Semantics · Open information extraction · Relations extraction

1 Introduction

Today, the fast growth of Web as well as the variety of data formats and language enables challenges for both Information Retrieval and Natural Language Processing in finding relevant documents which satisfy the needs of users. This fact leads us to the necessity of discover structured information from unstructured or semi-structured sources to retrieve information. Indeed, the growth demand of searching systems

A.L. Huynh · T.H. Duong (✉)
School of Computer Science and Engineering, International University,
VNU-HCMC, Ho Chi Minh City, Vietnam
e-mail: haiduongtrong@gmail.com

A.L. Huynh
e-mail: ailoan2712@gmail.com

H.S. Nguyen
Faculty of Information Technology, Le Quy Don University, Hanoi, Vietnam
e-mail: son_nguyenhong2002@yahoo.com

© Springer International Publishing Switzerland 2016

T.B. Nguyen et al. (eds.), *Advanced Computational Methods for Knowledge Engineering*, Advances in Intelligent Systems and Computing 453, DOI 10.1007/978-3-319-38884-7_19

induces the development of Information Extraction (IE) systems that analyze text written in plain natural language and to find facts or events in the text.

In IE system, its methods can be used to build knowledge representation models that report relations between words like ontology, semantic network, etc. The traditional IE systems learnt an extractor for each target relation as an input from labeled training examples [7, 10, 11]. Moreover, this approach did not scale to corpora where the number of target relations is very large, or where the target relations cannot be specified in advance [5].

Aiming to overcome the above problem, the Open Information Extraction (Open IE) where relation phrases are identified was introduced with Text Runner system [1]. Open IE systems extracted a large number of triples (*arg1, rel, arg2)* from text based on verb-based relations, where *arg1* and *arg2* are the arguments of the relation and *rel* is a relation phrase. Unlike other relation extraction methods which focused on a pre-defined a set of target relations, Open IE systems based on unsupervised extraction methods. They did not require any background knowledge or manually labeled training data.

The two tools that made up the Open IE state-of-the art, Ollie [9] and ClausIE [3], which intended to export the largest number of relations from the same sentence. They generated triples which are near to reproductions of the text. However, they lost the minimality of triple as mentioned in CSD system [2]. Because *Subject* or *Object* may consist of one or more entities in the same relation, Ollie and ClausIE did not identify the fact of each entity with its relation. For instance, a give sentence "*Anna and Jack have a meeting*". Two entities "*Anna*" and "*Jack*" refer to the relation "*have a meeting.*" A correct decomposition of this sentence would yield the triples "*Anna, have, a meeting*" and *"Jack, have, a meeting*"

The recent system LSOE [14] improves the precision of previous Open IE. It performs rule-based extraction of triples using POS-tagged text by applying lexical syntactic patterns based on Pustejovsky's qualia structure and generic patterns for non-specified relationships. But it takes the limitation of qualia-based patterns. In addition, LSOE as well as CSD use POS-tagged as the input.

Another issue is due to Verb phrase (VP) parsing. Most of Open IE systems relied on verb-based relation. The starting point of a VP in a clause is recognized but the VP is not fully parsed. VP words are simply connected together to create relations so that OIE methods create several relations that have the same meaning but different grammar forms, such as "is the author of" and "are the authors of"; "is", "are", "was", "were" and "has been"; "have", "has", "had"… Voice and affirmative of verbs are also not processed. Phrasal verbs often require complex grammar structures.

The motivation for our approach comes from the grammar structure of a sentence in English as well as verb-phrase patterns. In this paper, according to the approach presented in [8], a triple in a sentence expressed the relation between subject and object, in which the relation is as a verb phrase. The goal of the proposed algorithm is to extract the sets of triple with form {*Subject, Verb-Phrase, Object*} out of syntactically parsed sentences based on Syntax Model (SM) of

English language by determining verb usage pattern (VUP). With SM, not only the relationship between subject and object, but also the relationship between entity and its properties are pointed out in triples.

2 Related Works

In recent years, the process of Open Information Extraction has been improved in some systems such as TextRunner [1], Reverb [5], Ollie [9], ClausIE [3], CSD-IE [2] and LSOE [14]. In the following, we shortly summarize how the old systems (including Reverb, Ollie, ClausIE, CSD-IE and, LSOE) were implemented.

Reverb was a shallow extractor that reduced the incoherent extractions and uninformative extractions in previous open extractors by proposing two concepts on relation phrases including syntactic and lexical constraints. Firstly, Reverb extracted relation phrases that satisfied these two constraints, and then identified arguments as noun phrases that were positioned left and right of the extracted relations. The relation was the longest sequence of words staring from a verb satisfied all constraints.

Ollie improved the Open IE systems by addressing two important limitations of extraction quality: relations not mediated by verbs and relations expressed as a belief, attribution or other conditional context. It used Reverb to make a set of seed tuples and then bootstrapped the training set to learn "open pattern templates" that determined both the arguments and the extract relation phrase as Reverb. Ollie reached from 1.9 to 2.7 times bigger area under precision-yield curve, compared with Reverb and WOE systems.

ClausIE simply detected clauses and clause types via the dependency output from a probabilistic parser. A clause was understood as a part of a sentence, constituted by SVO (subject, verb, object) and some adverbs. Relied on dependency parser, ClausIE recognized one of 12 patterns of 5 verb types (copular verb, intransitive verb, di-transitive verb, mono-transitive verb and complex-transitive verb) to discover the clause types. The verb phrase was not parsed and used as the relation name. The results of this system were done in three datasets and presented as "the number of correct extractions/the total number of extractions": 1706/2975 for 500 sentences from Reverb dataset; 598/1001 for 200 sentences from Wikipedia; and 696/1303 for 200 sentences from New York Times.

CSD-IE decomposed a sentence into sentence constituents by defining a set of rules on the parsed tree structure manually. Then, the identified sentence constituents were combined to form the context, and extract triples from the resulted context. The authors compared their method with Reverb, ClausIE and Ollie with *New York Times* and *Wikipedia* datasets used in [3] and obtained an average of 70 % precision.

Xavier and Lima [15] extracted noun compounds and adjective-noun pairs from noun phrase, interpret extracted information by lexical-syntactic analysis and

exports relations. This method enhanced the extraction of relations within the noun compounds and adjective-noun pairs which is the gap in Open IE.

LSOE was the most recent OIE system which used POS-tagged text as input and applies a pattern-matching technique with using lexical syntactic patterns. It defined two kinds of patterns: generic patterns to identify domain specific non-specified relations and Qualia-based patterns. The weaknesses of LSOE were about the limit number of qualia-based patterns and generic patterns defined manually. Therefore, it missed the extraction of relations expressed by verbs. The authors reported that LSOE extracted less tuples than Reverb, but it achieved 54 % of precision while Reverb obtained 49 %.

3 Triple Extraction Using Lexical Pattern-based Syntax Model

3.1 Observation

The methodology of this approach is proposed via the analysis of English grammar structure. As same as Open IE systems like ClausIE or Ollie, which uses verb-based relation to extract triples, the new approach also depends on the verb type or the phrasal verb in use in order to transfer a sentence into its syntax model.

Considering some English sentences as below:

From three above tables Tables 1, 2 and 3, it is given that a sentence has at least one main clause. Each clause is constituted by subject, verb and objects. A subject can be either noun phrases or clause. An object can be a noun phrase, an adjective phrase, or a clause in Table 4.

A noun phrase starts with a head noun. It may contain pre-modifier or post-modifier of this head noun. Pre-modifier can be a noun, an adjective phrase or a participle. Post-modifier can be a subordinate clause or prepositional phrase (shown in Table 5).

Table 1 The grammar structure of a simple sentence

A simple sentence		
He	*Is*	*Handsome*
Subject	**Verb**	**Object**

Table 2 The grammar structure of a sentence with multiple clauses

A sentence with multiple clauses						
I	*Have*	*a shirt*	;	*It*	*Is*	*Beautiful*
Subject	Verb	Object		Subject	Verb	Object
Clause 1				**Clause 2**		

Table 3 The grammar structure of a complex sentence

A complex sentence is constituted by subordinate clause and main clause					
Since	*we*	*can't go*	*You*	*can have*	*the tickets.*
Subordinator	Subject	Verb			
Subordinate Clause			**Main Clause**		
Adverbial			**Subject**	**Verb**	**Object**

Table 4 The grammar structure of an object as a clause

A subordinate clause functioning as object		
I	*Know*	*that you lied.*
		Subordinate Clause
Subject	**Verb**	**Direct Object**

Table 5 The grammar structure of a noun phrase

Some Examples of the Noun Phrase				
Function	*Determiner*	*Pre-modifier*	*Head*	*Post-modifier*
1			Lions	
2	The	Information	Age	
3	Several	new mystery	Books	which we recently enjoyed
4	A	Marvelous	data bank	filled with information
FORMS	***Pronoun***	***Participle***	***Noun***	***Prepositional Phrase***
	Article	***Noun***	***Noun***	***Relative Clause***
	Quantifier	***Adjective Phrase***	***Pronoun***	***Nonfinite Clause***
				Complementation

Table 6 The grammar structure of prepositional phrase

Some Examples of the Prepositional Phrase		
Function	*Preposition*	*Complement*
1	With	Her
2	On	The table
3	From	what I can see
FORMS	**Preposition**	***Adverb***
		Noun Phrase
		-ing Clause or Relative Clause

A prepositional phrase is subdivided into a preposition and a complement. In which, complement may be an adverb, a noun phrase or a subordinate clause (shown in Table 6).

Table 7 The grammar structure of adjective phrase

Some Examples of the Adjective Phrase			
Function	*Pre-modifier*	*Head*	*Post-modifier*
1		Happy	
2		Young	in spirit
3	Too	Good	to be true
4		Excited	Indeed
FORMS	***Adverb***	***Adjective***	***Prepositional Phrase***
			Infinitive Clause
			Adverb

An adjective phrase begins with an adjective as a head word. It also has pre-modifier and post-modifier to help the adjective word to be more clearly as given examples in Table 7.

In general, the analysis method on the grammar structure of a sentence is based on the observations in Fig. 1. These observations are concluded by examining the sentence writing of human being as in previous examples.

The simple sentence only has one clause which contains subject and predicate (*for exp: I have a book)*. But the complex sentence includes many clauses which are linked together via subordinators, coordinating conjunctions or semi-colon. And the format of sentence depends on the human writing style. For instance, some writers prefer to use coordinating conjunction "and" replaced to semi-colon in order to connect multiple independent clauses of a sentence.

There are several verb types, e.g. transitive, intransitive, linking, etc. For each verb type, a grammar structure is required in usage. For example, an intransitive verb does not have a direct object but a transitive verb requires a noun or noun phrase as a direct object. However, a di-transitive verb requires two nouns or noun phrases as direct and indirect objects. A verb can have several verb types, for

a) A sentence can be one of the below forms:
   a1. Sentence = Subject + Predicate
   a2. Sentence = Independent Clause + coordinating conjunction + Independent Clause
   a3. Sentence = Adverbial Clause + Main Clause
b) Clause (C) = Subject (Subj) + Predicate (Pre)
c) Subject can be a noun phrase (NP) or a clause
d) Predicate = Verb + Direct Object (Dobj) + Indirect Object (Iobj) or Complement (Comp)
e) DO can be the list of NP or Clause or Pronoun
f) IO can be the list of NP or Clause or Pronoun
g) Comp can be the list of NP, list of Adjective Phrase (AdjP), an Adverb Phrase (AdvP) , a Prepositional Phrase (PrepP) or a Clause

Fig. 1 The observations

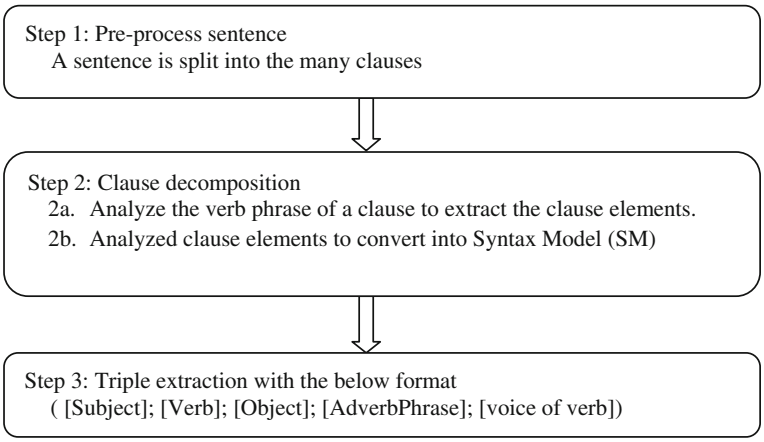

Fig. 2 Steps process a sentence

examples, "remember" can be transitive as well as intransitive verb. Therefore, as in observation (d), the predicate should follow the predefined grammar structure of that verb and its verb type.

3.2 New Triple Extraction Algorithm

Following the observation in Fig. 1, it is given that a sentence has at least a clause. In addition, depending on the verb type or phrasal verb, a verb requires a particular grammar structure called Verb Usage Pattern (VUP). Whether a verb is used in a sentence, the grammar structure of the sentence has to be satisfied the current used VUP of this verb. For instance, word "*remember*" is both transitive and intransitive verb. A sentence made by this verb has one of two VUPs: *Subject + Verb* or *Subject +Verb + Object*. For another example, the separable phrasal verb "*switch on*" has two patterns in grammar structure of the sentence: *John switches on the radio (Subject + Verb + particle + Object)* or *John switches the radio on (Subject + Verb + Object + particle)*.The construction of VUP(s) is based on Internet resources from Oxford Online Dictionary,[1] the Free Dictionary[2] or collected from linguistic resources in [6, 12, 13].

In this paper, triple extraction system performs in three steps as Fig. 2. Firstly, the sentence is divided into many clauses. In the second step, each clause is processed to decompose into clause elements and convert into Syntax Model (SM) by

[1] www.oxforddictionaries.com.

[2] www.thefreedictionary.com.

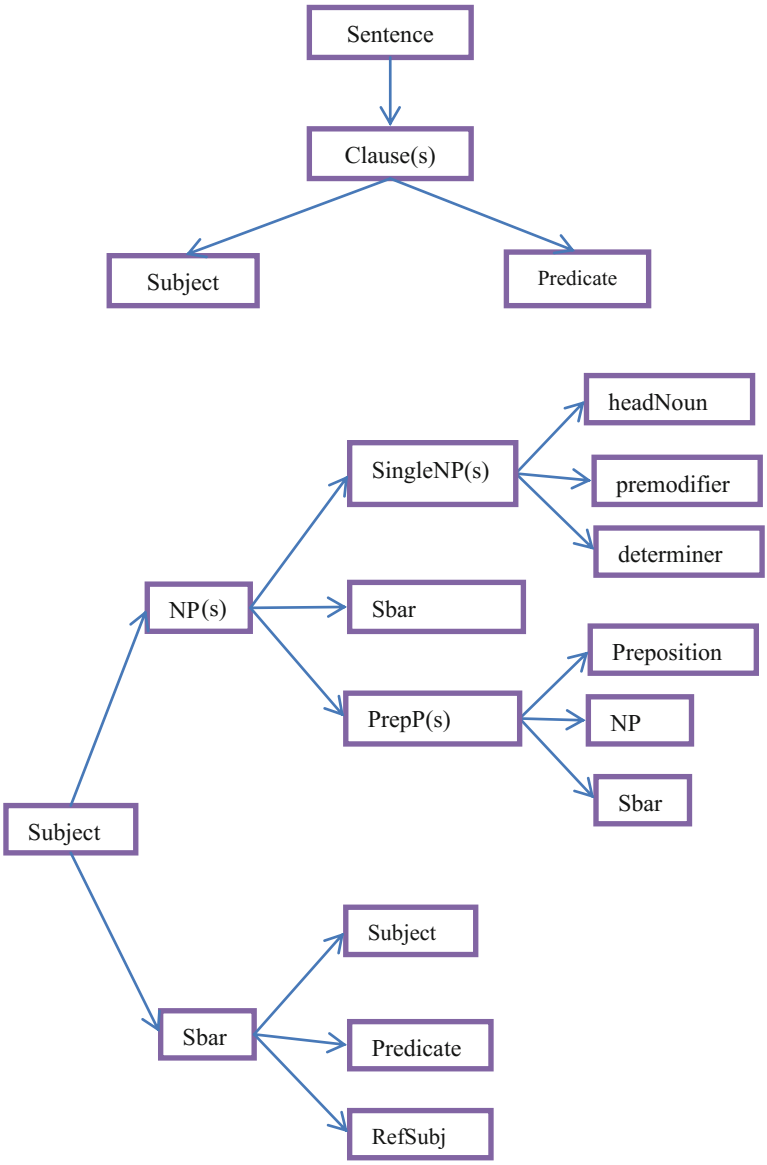

Fig. 3 Syntax model (part 1)

applying VUP. Finally, the tree constituents by clause elements are combined to extract triples.

Step 1: Sentence does pre-processing to divide into multiple clauses. As mentioned in part 3.1, the format of sentence relies on the human writing. Therefore, the sentence firstly detects whether it has multiple independent clauses via semi-colon.

Next, each independent clause plays the role as a simple sentence; and this sentence continues to check whether it has format type a3 in observation Fig. 1 through subordinators. Finally, the list of clauses is gotten to do the analysis in next steps.

Step 2: Clause Decomposition consists of two main sub-tasks: (2a) parsing a verb phrase (VP) and (2b) converting into SM

- **Sub-task 2a**: a probabilistic parser is called to produce the parsed structure, which can be a phrase structure trees or a typed dependencies. The method can work on both forms of parsed structure, but in this paper, only the Stanford typed dependencies are used to illustrate. This task starts with searching main

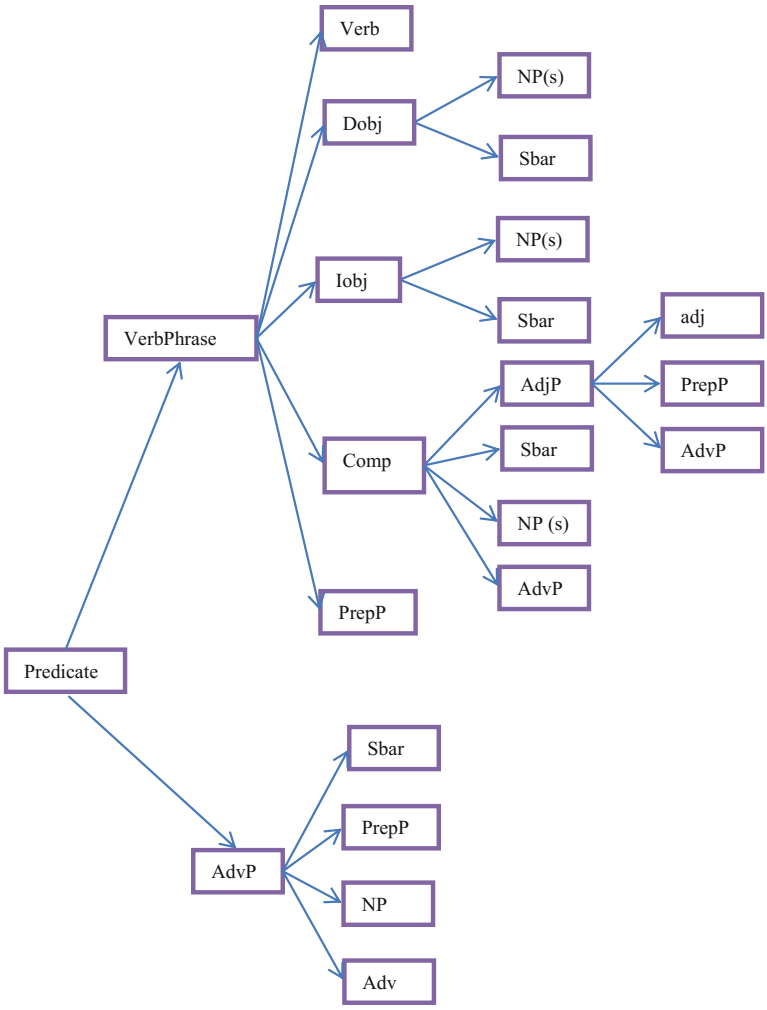

Fig. 4 Syntax model (part 2)

verb of clause based on ROOT and COP typed dependencies using Stanford Parser. Applying the defined VUP before, parts of clause such as Subject, Verb and Object are identified.

- **Sub-task 2b**: Each of extracted elements in previous task is sub-divided and mapped into smaller elements like Noun Phrase, Adjective Phrase, Prepositional Phrase … etc. as same as syntax model (SM) in Figs. 2 and 3.

Step 3: Triples with above format are extracted for each clause as well as for whole sentence (Fig. 4)

With SM, more detail triples which describe the relationship between sub-elements with each part of clause are given. For instance, a noun phrase consisting of prepositional phrase as a modifier like "*a girl with blond hair*" can also have a triple which shows the meaning in relation between these two elements. In this case, triple *{a girl, is (complemented by), with blond hair}* is given. Moreover, SM can identify more specific elements in objects compared with Ollie and ClausIE. It separates Objects into lists of noun phrase, list of preposition phrases supporting for clause or adverb phrase. Therefore, it makes enhancement in the accuracy of extracted triples as well as their minimality [2].

4 Experimental Result Comparison

We compare the proposed system against the two Open IE systems Ollie and ClausIE. This system was implemented as describe in Sect. 3, using the Stanford Constituent Parser [4]. ClausIE was run in default mode to extract triples.

Datasets used for this experience are: 200 random sentences from English Wikipedia and 200 random sentences from New York Time which are the exact same datasets as in [3]. Our results are summarized in Table 8.

Table 8 shows the total number of correct extractions as well as the total number of extractions for each method and each dataset. These result numbers are evaluated by human experience.

This algorithm achieves the better results in giving the detailed triples compared with previous Open IE systems. However it still remains some limitations in the improvement of quality and quantity of triples because of parser failure and the complexity of sentence.

Table 8 Number of correct extractions and total number of extractions

	Ollie	**ClausIE**	**New system**
New York dataset	270/500	662/926	1119/1542
Wiki dataset	357/573	602/794	1008/1323

Table 9 Result of example 1

System	Output
ClausIE	(“the girl with blond hair”, “is”, “beautiful”)
Ollie	(the girl; is; beautiful)
	(beautiful; be the girl with; blond hair)
New algorithm	{girl with blond hair; be; beautiful;; active}
	{girl; be (complemented by); with blond hair;; active}
	{blond; is property of; hair;;}

4.1 Good Points

There are two main points that new algorithm give a better result compared with Ollie and ClausIE. They will be described in some examples as below:

Example 1 “*the girl with blond hair is beautiful*”

Firstly, both Ollie and ClausIE did not give any triples which show the relationship between the modifier and its noun phrase, or between noun phrase and its properties. In example 1, “*blond*” is a property of “*hair*” (Table 9).

The extracted triple in this case has formats:

([adjective], “be property of”, [head Noun])
([single NP], “be complemented by”, [prepPhrase])

Considering two another examples:

Example 2 “*LaBrocca scored his first goal for the club in a 4-1 home victory vs. Chicago Fire.*”

Example 3 “*Jack and Janny went to school*”

Secondly, the problem of ClausIE and Ollie is still missing the details of an extraction. It means that the extracted fact by both systems may be described in more facts. Our system has the same idea in improve quality of extraction in the minimality of the extracted facts. It is proved in example 2 and example 3 obviously (Tables 10 and 11).

4.2 Weak Points

Mausam et al. [9] and Del Corro and Gemulla [3] claimed that most of the extraction errors are due to two problems: parser failures and inability to express relationships in the text into binary relations. And this approach also takes the mistakes related to these issues:

Table 10 Result of example 2

System	Output
ClauseIE	(“LaBrocca”, “scored”, “his first goal for the club”)
	(“LaBrocca”, “scored”, “his first goal in a 4-1 home victory vs. Chicago Fire”)
	(“LaBrocca”, “scored”, “his first goal”)
	(“his”, “has”, “first goal”)
Ollie	(LaBrocca; scored for; the club)
	(LaBrocca; scored; his first goal)
New algorithm	{LaBrocca; score; his first goal;; active}
	{LaBrocca; score; for club;; active}
	{LaBrocca; score; his first goal for club;; active}
	{LaBrocca; score; in 4-1 home victory vs. Chicago Fire;; active}
	{LaBrocca; score; his first goal in 4-1 home victory vs. Chicago Fire;; active}
	{4-1 home victory; be (complemented by); vs. Chicago Fire;; active}
	{first; is property of; goal;;}
	{he; have; first goal;;}

Table 11 Result of example 3

System	Output
ClausIE	(“Jack and Janny”, “went”, “to school”)
Ollie	(Jack and Janny ; went to ; school)
New algorithm	{Jack ; go to ; school ; ; active}
	{Janny ; go to ; school ; ; active}
	{Jack, Janny ; go to ; school ; ; active}

Example 4 “*Judy laughed and Jimmy cried*”

Stanford Parser shows that

```
(ROOT
  (S
    (NP (NNP Judy) (NNP laughed)
      (CC and)
      (NNP Jimmy))
    (VP (VBD cried))))
nn(laughed-1, Judy-0)
nsubj(cried-4, laughed-1)
conj_and(laughed-1, Jimmy-3)
root(ROOT-0, cried-4)
```

In this example, the failure in extraction is due to the error of Stanford dependency in Stanford parser. The word “*laughed*” should be a VP instead of NNP. Therefore, the output of this sentence as below in new algorithm is not correct:

{Judy laughed ; cry ; ; ; active}
{Jimmy ; cry ; ; ; active}
{Judy laughed , Jimmy ; cry ; ; ; active}

Another issue is due to the sentence writing of human beings. This algorithm cannot cover syntax model of complex sentence which is constituted by too many clauses as the below example:

Example 5 "*Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria Bueno (better known as Ana Rosetti), Gabriel was born in San Fernando, Cadiz, but spent her childhood in Madrid*" (Table 12).

Table 12 Result of example 5

System	Output
ClausIE	("Gabriel", "was born", "in San Fernando")
	("Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria Bueno better known as Ana Rosetti", "Cadiz")
	("Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria Bueno better known as Ana Rosetti", "spent", "her childhood in Madrid")
	("Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria Bueno better known as Ana Rosetti", "spent", "her childhood")
	("her", "has", "childhood")
Ollie	(Gabriel; was born in; San Fernando)
	(her childhood; be spent in; Madrid)
	(Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria; was born in; San Fernando)
	(Gabriel; be known as; Ana Rosetti))
	(Gabriel; was born at; San Fernando)
	(Gabriel; was born on; San Fernando)
	(Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria; be known as; Ana Rosetti))
	(San Fernando; was born in; Cadiz)
	(Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria; was born at; San Fernando)
	(Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria; was born on; San Fernando)
New Algorithm	{Daughter of actor Ismael Sanchez Abellan, actress Ana Maria Bueno ; bear ; in San Fernando ; better known as Ana Rosetti ; passive}
	{San Fernando ; be ; Cadiz ; ; active}
	{actor Ismael Sanchez Abellan,actress Ana Maria Bueno ; be ; Gabriel ; ; active}
	{Daughter ; be (complemented by) ; of actor Ismael Sanchez Abellan,actress Ana Maria Bueno ; ; active}
	{Ismael Sanchez Abellan ; be ; actor ; ; active}
	{Ana Maria Bueno ; be ; actress ; ; active}

In this example, all three systems miss information as below:

(Ismael Sanchez Abellan is actor.
Ana Maria Bueno is actress.
Ana Maria Bueno is a writer.
Ana Maria Bueno is Ana Rosetti.
Gabriel was born in San Fernando, Cadiz.
Gabriel spent her childhood in Madrid.
Daughter of the actor Ismael Sanchez Abellan and actress and writer Ana Maria Bueno is Gabriel.)

Besides, the main idea to convert a sentence into syntax model bases on the defined VUP. We cannot list full types of a verb, and then it cannot give any result in some cases.

5 Conclusion

This paper presented triple extraction based on syntax model of English language. The algorithm is implemented by determining verb usage pattern (VUP) in order to convert sentence into syntax model. The experimental result indicates that new system improved the quality of extraction in the minimality. Besides, its triples can describe the relationship between properties or modifier of a noun phrase. In this experience, that more detailed triples are extracted will enhance the number of relevant triples in sentence.

However, because of the dependence on a probabilistic parser, it leads to some issues which are due to the parser. Additionally, syntax model cannot adapt with all kinds of sentence written by human. Furthermore, the problem concerned about the performance in precision of triples should be considered. All of these issues allow us to do more researches in order to achieve the better result without knowledge-based as well as with using the defined knowledge in future.

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open Information extraction from the web. In: IJCAI, pp. 2670–2676 (2007)
2. Bast, H., Haussmann, E.: Open information extraction via contextual sentence decomposition. In: IEEE Seventh International Conference on Semantic Computing (ICSC), pp. 154–159 (2013)
3. Corro, L.D., Gemulla, R.: ClausIE: clause-based open information extraction. In: WWW, pp. 355–366 (2013)
4. de Marnee, M.-C., Manning, C.D.: Stanford typed dependencies manual
5. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam, M.: Open information extraction: the second generation. In: IJCAI, vol. 11, pp. 3–10 (2011)

6. Hampe, B.: Transitive phrasal verbs in acquisition and use: a view from construction grammar. Lang. Value **4**(1), 1–32 (2012)
7. Kim, J., Moldovan, D.: Acquisition of semantic patterns for information extraction from corpora. In: Proceedings. of Ninth IEEE Conference on Artificial Intelligence for Applications, pp. 171–176 (1993)
8. Leskovec, J., Grobelnik, M., Milic-Frayling, N.: Learning sub-structures of document semantic graphs for document summarization. In: Proceedings of the 7th International Multi-Conference Information Society IS, vol. B, pp. 18–25 (2004)
9. Mausam, Schmitz, M., Soderland, S., Bart, R., Etizioni, O.: Open language learning for information extraction. In: EMNLP-CoNLL, pp. 523–534 (2012)
10. Riloff, E.: Automatically constructing extraction patterns from untagged text. In: Proceedings. of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pp. 1044–1049 (1996)
11. Soderland, S.: Learning Information Extraction Rules for Semi-Structured and Free Text. Mach. Learn. **34**(1–3), 233–272 (1999)
12. Thim, S.: Phrasal verbs: The English Verb-particle Construction and its History, vol. 78. Walter de Gruyter (2012)
13. Trebits, A.: The most frequent phrasal verbs in English language EU documents–A corpus-based analysis and its implications. System **37**(3), 470–481 (2009)
14. Xavier, C.C., De Lima, V.L.S., Souza, M.: Open information extraction based on lexical semantics. J. Braz. Comput. Soc. **21**(1), 1–14 (2015)
15. Xavier, C., Lima, V.S.: Boosting open information extraction with noun-based relations. In: Proceedings of the ninth international conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), pp. 96–100. Reykjavik, Iceland (2014)