The Triplex Approach for Recognizing Semantic Relations from Noun Phrases, Appositions, and Adjectives

Seyed Iman Mirrezaei[1]([⊠]), Bruno Martins[2],
and Isabel F. Cruz[1]

[1] ADVIS Lab, Department of Computer Science,
University of Illinois at Chicago, Chicago, USA
`smirre2@uic.edu, ifc@cs.uic.edu`
[2] Instituto Superior Técnico and INESC-ID,
Universidade de Lisboa, Lisbon, Portugal
`bruno.g.martins@tecnico.ulisboa.pt`

Abstract. Discovering knowledge from textual sources and subsequently expanding the coverage of knowledge bases like DBpedia or Freebase currently requires either extensive manual work or carefully designed information extractors. Information extractors capture triples from textual sentences. Each triple consists of a subject, a predicate/property, and an object. Triples can be mediated via verbs, nouns, adjectives, and appositions. We propose TRIPLEX, an information extractor that complements previous efforts, concentrating on noun-mediated triples related to nouns, adjectives, and appositions. TRIPLEX automatically constructs templates expressing noun-mediated triples from a bootstrapping set. The bootstrapping set is constructed without manual intervention by creating templates that include syntactic, semantic, and lexical constraints. We report on an automatic evaluation method to examine the output of information extractors both with and without the TRIPLEX approach. Our experimental study indicates that TRIPLEX is a promising approach for extracting noun-mediated triples.

Keywords: Open domain information extraction · Relation extraction · Noun-mediated relation triples · Compound nouns · Appositions

1 Introduction

Deriving useful knowledge from unstructured text is a challenging task. Nowadays, knowledge needs to be extracted almost instantaneously and automatically from continuous streams of information such as those generated by news agencies or published by individuals on the social web to enrich properties related to people, places, and organizations in existing large-scale knowledge bases, such as Freebase [3], DBpedia [1], or Google's Knowledge Graph [11]. Subsequently these values can be used by search engines to provide answers for user queries

(e.g., the resignation date of a given politician, or the ownership after a company acquisition). For many natural language processing (NLP) applications, including question answering, information retrieval, machine translation, and information extraction, it is important to extract facts from text. For example, a question answering system may need to find the location of the *Microsoft Visitor Center* in the sentence *The video features the Microsoft Visitor Center, located in Redmond.*

Open Information Extractors (OIE) aim to extract triples from text, with each triple consisting of a subject, a predicate/property, and an object. These triples can be expressed via verbs, nouns, adjectives, and appositions. Most OIE systems described in the literature, such as TextRunner [2], WOE [12], or ReVerb [7], focus on the extraction of verb-mediated triples. Other OIE systems, such as OLLIE [9], ClauseIE [6], Xavier and Lima's system [14], or ReNoun [15], may also, or only, extract noun-mediated triples from text. OLLIE was the first approach for simultaneously extracting verb-mediated and noun-mediated triples, although it can only capture noun-meditated triples that are expressed in verb-mediated formats. For example, OLLIE can extract the triple `<Bill Gates; be co-founder of; Microsoft>` from the sentence *Microsoft co-founder Bill Gates spoke at a conference* but it cannot extract a triple `<Microsoft; headquarter; Redmond>` from the sentence *Microsoft is an American corporation headquartered in Redmond.* ClauseIE extracts noun-mediated triples from appositions and possessives based upon a predefined set of rules. ReNoun uses seeds (i.e., examples gathered through manually crafted rules) and an ontology to learn patterns for extracting noun-mediated triples.

The OIE system that we have built is named Triplex. It is designed specifically to extract triples from noun phrases, adjectives, and appositions. Systems like OLLIE, which can only extract triples corresponding to relations expressed through verb phrases, can be assisted by Triplex, which extracts triples from grammatical dependency relations involving noun phrases and modifiers that correspond to adjectives and appositions. Triplex recognizes templates that express noun-mediated triples during its automatic bootstrapping process. The bootstrapping process finds sentences that express noun-mediated triples using Wikipedia pages. Then, it constructs templates from sentences in the bootstrapping set. The templates express how noun-mediated triples occur in sentences and they allow for information to be extracted relating to different levels of text analysis, from lexical (i.e., word tokens) and shallow syntactic features (i.e., part-of-speech tags), to features resulting from a deeper syntactic analysis (i.e., features derived from dependency parsing). In addition, semantic constraints may be included in some templates to obtain more precise extractions. Templates are then generalized to broaden their coverage (i.e., those with similar constraints are merged together). Finally, the templates can be used to extract triples from previously unseen text. We evaluated Triplex according to the automated framework of Bronzi et al. [4], extending it to assess noun-mediated triples.

The remainder of this paper is organized as follows: In Sect. 2, we briefly summarize related work in the area of open-domain information extraction. The TRIPLEX pipeline is presented in Sect. 3. Section 4 describes our experiments, ending with a discussion of the obtained results. Finally, Sect. 5 concludes the paper, summarizing the main aspects and presenting possible directions for future work.

2   Related Work

OIE systems are used to extract triples from text and they can be classified into two major groups. The first group includes systems that extract verb-mediated triples (i.e., TextRunner [2], WOE [12], ReVerb [7], or OLLIE [9]). The second group includes systems that extract noun-mediated triples (i.e., OLLIE [9], ClauseIE [6], Xavier and Lima's system [14], and ReNoun [15]).

In the first group, the earliest proposed OIE system was TextRunner. This system first detects pairs of noun phrases and it then finds a sequence of words as a potential relation (i.e., predicate) between each pair of noun phrases. In a similar way, WOE uses a dependency parser to find the shortest dependency path between two noun phrases. All of the approaches in the first group assume that the object occurs after the subject.

OLLIE, which is a member of both the first and the second groups, was the first approach to extract both noun-mediated and verb-mediated triples. It uses high confidence triples extracted by ReVerb as a bootstrapping set to learn patterns. These patterns, mostly based on dependency parse trees, indicate different ways of expressing triples in textual sources. It is important to note that OLLIE only extracts noun-mediated triples that can be expressed via verb-mediated formats. Therefore, it only covers a limited group of noun-mediated triples. In comparison, TRIPLEX only extracts triples from compound nouns, adjectives, and appositions. ClauseIE uses knowledge about the English grammar to detect clauses based on the dependency parse trees of sentences [6]. Subsequently, triples are generated depending on the type of those clauses. ClauseIE has predefined rules to extract triples from dependency parse trees and it is able to generate both verb-mediated triples from clauses and noun-mediated triples from possessives and appositions. In contrast, TRIPLEX automatically learns rules that extract triples during its bootstrapping process.

Xavier and Lima use a boosting approach to expand the training set for information extractors so as to cover an increased variety of noun-mediated triples [14]. They find verb interpretations for noun and adjective based phrases. Then, the verb interpretations are transformed into verb-mediated triples to enrich the training set. Still, these verb interpretations can create long and ambiguous sentences. Therefore, filtering unrelated interpretations is essential before adding the inferred verb interpretations to the training set of information extractors. TRIPLEX does not depend on verb patterns to extract noun-mediated triples, thereby making such filtering unnecessary. Closer to our work is ReNoun, a system that uses an ontology of noun attributes and a manually crafted set of extraction rules, to extract seeds [15]. The seeds are then used to

learn dependency parse patterns for extracting triples. In contrast, TRIPLEX uses data from Wikipedia during its bootstrapping process without requiring manual intervention.

3   Triplex

OIE systems extract triples from an input sentence according to the format `<subject; relation;object>`. In these triples, a relation phrase (i.e., a predicate or property) expresses a semantic relation between the subject and the object. The subject and the object are noun phrases and the relation phrase is a textual fragment that indicates a semantic relation between two noun phrases. The semantic relation can be verb-mediated or noun-mediated. For example, an extractor may find the triples `<Kevin Systrom;profession;cofounder>` and `<KevinSystrom;appears on; NBC News>` in the sentence *Instagram cofounder Kevin Systrom appears on NBC News.* The first triple is noun-mediated and the second one is verb-mediated.

The TRIPLEX approach focuses on noun-mediated triples from noun phrases, adjectives, and appositions. First, it finds sentences that express noun-mediated triples. These sentences are detected by using a dependency parser to find grammatical relations between nouns, adjectives, and appositions. Second, it automatically extracts templates from the sentences. Finally, these templates are used to extract noun-mediated triples from previously unseen text.

The TRIPLEX pipeline uses the Stanford NLP toolkit[1] to parse sentences, extract dependencies, label tokens with named entity (NE) and with part-of-speech (POS) information, and perform coreference resolution. The coreference resolution module is used to replace in all the sentences pronouns and other coreferential mentions with the corresponding entity spans prior to subsequent processing. The dependency parser discovers the syntactic structure of input sentences. A dependency parse of a sentence is a directed graph whose vertices are words and whose edges are syntactic relations between the words. Each dependency corresponds to a binary grammatical relation between a governor and a dependent [5]. For example, the dependency relations `nsubj<went,Obama>` and `prep-to<went,Denver>` can be found in the sentence *Obama went to Denver.* In the dependency relation `prep-to<went,Denver>`, the word `went` is the governor and the word `Denver` is the dependent. The part-of-speech tagger assigns a morpho-syntactic class to each word, such as noun, verb, or adjective. The Named Entity Recognition (NER) model[2] labels sequences of words according to pre-defined entity categories: *Person*, *Organization*, *Location*, and *Date.*

The other components of the pipeline are a noun phrase chunker, which complements the POS, NER, and dependency parsing modules from the Stanford NLP toolkit, WordNet synsets, and Wikipedia synsets. The noun phrase chunker extracts noun phrases from sentences. WordNet is a lexical database that categorizes English words into sets of synonyms called synsets. WordNet synsets

---

[1] http://nlp.stanford.edu/software/corenlp.shtml.
[2] http://nlp.stanford.edu/software/CRF-NER.shtml.

are used to recognize entities within each sentence according to the pre-defined categories, complementing the Stanford NER system. Several synsets are also built for each Wikipedia page. There are different mentions for a Wikipedia page (e.g., redirects and alternative names) and also in the hypertext anchors that point to a Wikipedia page. For example, in the Wikipedia page for the *University of Illinois at Chicago*, the word *UIC* is extensively used to refer to the university. Synsets of Wikipedia pages are constructed automatically by using redirection page links, backward links, and hypertext anchors. These links are retrieved using the Java-based Wikipedia Library[3].

TRIPLEX uses infobox properties and infobox values of Wikipedia during its bootstrapping process. We use a Wikipedia English dump[4] to extract all Wikipedia pages and we query Freebase and DBpedia according to the Wikipedia page ID to determine the type of the page. Wikipedia pages are categorized under the following types: *Person*, *Organization*, or *Location*. Additionally, we perform coreference resolution on the extracted Wikipedia pages to identify words that refer to the same Wikipedia page subject. We then use these words to enrich synsets of the respective Wikipedia page. We now describe the TRIPLEX approach for extracting templates, starting with the generation of the bootstrapping set of sentences.

3.1 Bootstrapping Set Creation

Following ideas from the OLLIE system, which leverages bootstrapping [9], our first goal is to construct automatically a bootstrapping set that expresses in multiple ways how the information in noun phrases, adjectives, and appositions is encapsulated. The bootstrapping set is created by processing the extracted Wikipedia pages and their corresponding infoboxes.

Wikipedia pages without infobox templates are ignored during sentence extraction, while the other pages are converted into sets of sentences. Finally, we perform preprocessing on the sentences from the extracted Wikipedia pages and we use custom templates (i.e., regular expressions) to identify infobox values from the text. We also convert dates to strings. For instance, the infobox with value 1961|8|4 is translated to August 4, 1961. We begin template extraction by processing 3,061,956 sentences from the extracted Wikipedia pages that are matched with infobox values.

The sentence extractor automatically constructs a bootstrapping set by matching infobox values of the extracted Wikipedia pages with phrases from the text of the corresponding Wikipedia pages. If in a sentence there exists a dependency path between the current infobox value and the synset of the page name, and if this dependency path only contains nouns, adjectives, and appositions, then the sentence is extracted. For instance, given the page for *Barack Obama*, the extractor matches the infobox value *August 4, 1961* with the sentence *Barack Hussein Obama II (born August 4, 1961)*. This process is repeated for all infobox values of a Wikipedia page.

---

[3] https://code.google.com/p/jwpl/.
[4] http://dumps.wikimedia.org/backup-index.html.

In order to match complete names with abbreviations such as *UIC*, the extractor uses a set of heuristics that was originally proposed in WOE [12], named *full match*, *synset match*, and *partial match*. The full match heuristic is used when the page name is found within a sentence of the page. The synset match heuristic is used when one member of the synset for the page name is discovered within a sentence. The partial match heuristic is used when a prefix or suffix of a member of the synset is used in a sentence. Finally, a template is created by marking an infobox value and a synset member in the dependency path of a selected sentence. We apply a constraint on the length of the dependency path between a synset member and an infobox value to reduce bootstrapping errors. This constraint sets the maximum length of the dependency path to 6, a value which was determined experimentally by checking the quality of our bootstrapping set. Specifically, we randomly selected 100 sentences for manual examination; of these, 90 % satisfied the dependency path length constraint.

After creating the bootstrapping set, the next step is to automatically create templates from dependency paths that express noun-mediated triples. Templates describe how noun-mediated triples can occur in textual sentences. Each template results from a dependency path between a synset member (a subject) and an infobox value (an object). We annotate these paths with POS tags, named entities, and WordNet synsets. In the template, to each infobox value we add the name of the infobox. In addition, a template includes a template type, based on the types of the Wikipedia page where the sentence occurred. The types of dependencies between synset members and infobox values are also attached to the template. If there is a copular verb or a verbal modifier in the dependency path, we will add them as a lexical constraint to the template. For example, *headquartered* is a verbal modifier added as a lexical constraint to the corresponding template for the sentence: *Microsoft is an American corporation headquartered in Redmond* (see Fig. 1). *Born* is another lexical constraint for templates related to nationality, as in the sentence *The Italian-born Antonio Verrio was frequently commissioned*. We merge templates if the only differences among them relate to lexical constraints. We keep one template and a list of lexical constraints for the merged templates. Finally, we process all templates and remove redundant ones.

Infobox values may occur before or after synset members of the page name in sentences. If there exists a dependency path between these values independently of their position, the related template is extracted. For example, the infobox value occurs before the synset member in the sentence *Instagram co-founder Kevin Systrom announced a hiring spree.* In this example, *co-founder* is the infobox value and *Steve Hafner* is the synset member of the Wikipedia page. The infobox value may also occur after the synset member, as shown in the sentence *Microsoft is an American corporation headquartered in Redmond*. In this case, *corporation* is the synset member and *Redmond* is the infobox value (see Fig. 1).

The noun phrase chunker is finally used to search dependency paths and merge words that are part of the same noun phrase chunk. In addition, we do
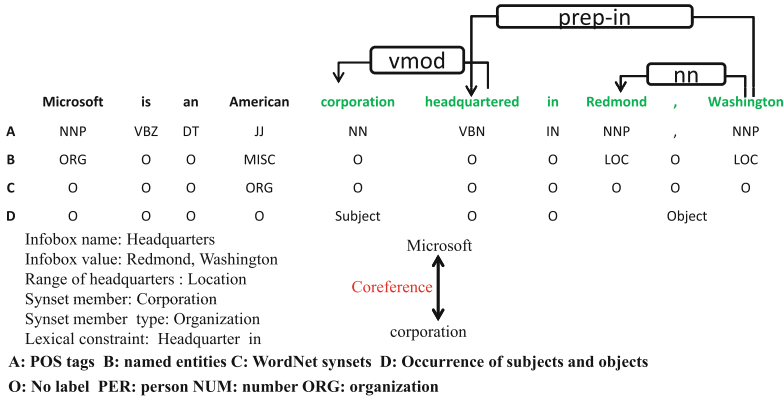
Fig. 1. An example sentence annotated with the corresponding dependency relations, the POS tags for the word tokens, named entities, WordNet synsets, and occurrences of the synset member of the Wikipedia page (subjects) and the infobox values (objects).

not apply the noun phrase chunker if a synset member and the infobox value occur in the same chunk.

3.2 Template Matching

This section describes how we use the dependency paths of a sentence together with the extracted templates to detect noun-mediated triples. First, named entities and WordNet synsets are used to recognize the candidate subjects of a sentence together with their types. Then, dependency paths between candidate subjects and all potential objects are identified and annotated by the NLP pipeline. Finally, candidate infobox names (which are properties in DBpedia) are assigned to a candidate subject and a candidate object, derived from matching templates with subject types, dependency types, WordNet synsets, POS tags, and named entity annotations. If there are lexical constraints in a template, the words in the dependency path between a subject and an object must be matched with one of the phrases in the lexical constraint list. We also consider the semantic similarity between the words and the member of the lexical constraint list, using Jiang and Conrath's approach to calculate the semantic similarity between words [8].

When there is a specific range (*Person*, *Organization*, *Location*, or *Date*) for an infobox name (property) of a triple, and when the object type of a triple is unknown, a previously trained confidence function is used to adjust the confidence score of the triple. A logistic regression classifier is used in this confidence function after it is trained using 500 triples extracted from Wikipedia pages. Our confidence function is an extension of the confidence function proposed for OLLIE [9] and for ReVerb [7]. A set of features (i.e., frequency of the extraction template, existence of particular lexical features in templates, range of properties, and semantic object type) are computed for each extracted triple. The confidence score is assigned the probability computed by the classifier.

Finally, each candidate triple has an infobox name that is mapped to a DBpedia property and the object type of a candidate triple should be matched with the range of that property. When the range of a property is a literal, all possible values of the property are retrieved from DBpedia and compared with the candidate object. If their values are not matched, the candidate triple is discarded.

4   Evaluation

We conducted a comprehensive set of experiments to compare the outputs of TRIPLEX, OLLIE, and ReVerb based upon the approach by Bronzi et al. [4]. These authors introduced an approach to evaluate verb-mediated information extractors automatically. We improve on their approach by expanding it to the evaluation of noun-mediated triples. Additionally, we compare TRIPLEX, OLLIE, and ReVerb using a manually constructed gold standard. Finally, we compare the various information extractors according to the quality of their extracted triples.

We first created a dataset by taking 1000 random sentences from Wikipedia that have not been used during the bootstrapping process. Each sentence in the test dataset has a corresponding Wikipedia page ID. All extracted facts gathered by information extractors from these sentences needed to be verified. We recall that a fact is a triple `<subject;predicate;object>` that expresses a relation between a subject and an object. A fact is correct if its corresponding triple has been found in the Freebase or DBpedia knowledge bases or if there is a significant association between the entities (subjects and objects) and its predicate [4] according to Eq. 2. In order to estimate the precision of an information extractor, we use the following formula:

$$Precision = \frac{|a| + |b|}{|S|} \tag{1}$$

In Eq. 1, $|b|$ is the number of extracted facts from Freebase and DBpedia, $|S|$ is the total number of facts extracted by the system, and $|a|$ is the number of correct facts returned by the information extractor, which have been validated by using the pointwise mutual information (PMI), as defined in Eq. 2. Since values of properties in Freebase and DBpedia are not completely filled, Bronzi et al. [4] compute the PMI to verify a fact. The PMI of a fact measures the likelihood of observing the fact given that we observed its subject ($subj$) and object (obj), independently of the predicate ($pred$):

$$\mathrm{PMI}(subj, pred, obj) = \frac{\mathrm{Count}(subj \; \wedge \; pred \; \wedge \; obj)}{\mathrm{Count}(subj \; \wedge \; obj)} \tag{2}$$

When verifying the extracted facts, we use the corresponding Wikipedia ID of each sentence to retrieve all possible properties and their values from Freebase or DBpedia. These values are then used to verify extracted facts from sentences. The semantic similarity between the properties of those knowledge bases and the

predicate of a fact are calculated [8]. The semantic similarity measure uses Word-Net together with corpus statistics to calculate the semantic similarity between phrases. If the semantic similarity is above a predetermined threshold and if the entities corresponding to the subject and object also match the knowledge base properties, then the fact is deemed correct [4].

The function $\text{Count}(q)$ returns the number of results retrieved by the Google search engine for query q, where the elements of the query occur within the maximum distance of 4 words. The range of the PMI function is between 0 and 1. The higher the PMI value, the more likely that the fact is correct. Specifically, a fact is deemed correct if its PMI value is above the threshold of 10^{-3}, which was determined experimentally. We also use the method in Eq. 3 to estimate recall [4]:

$$Recall = \frac{|a| + |b|}{|a| + |b| + |c| + |d|} \tag{3}$$

The parameters $|a|$ and $|b|$ are computed as in Eq. 1. We now describe how $|c|$ and $|d|$ are computed. First, all correct facts within sentences of the dataset are identified. Each fact contains two entities and a relation. All possible entities of a sentence are detected by the Stanford NER and from the WordNet synsets. Furthermore, we use the Stanford CoreNLP toolkit to detect all verbs (predicates) in a sentence. Finally, we expand the set of predicates from sentences by adding DBpedia and Freebase properties.

We use three sets S, P, and O to create all the possible facts, which are respectively the set of recognized subjects, predicates, and objects in the sentences. All possible facts are produced by the Cartesian product of these three sets, $G = (S \times P \times O)$. Assuming that D is the set of all the facts in Freebase and in DBpedia, $|c|$ is computed as follows:

$$|c| = |D \cap G| - |b| \tag{4}$$

Finally, $|d|$ is determined by subtracting $|a|$ from the size of the set of all facts in G that are not in D, which have been validated using PMI (that is, those whose PMI is above the threshold).

We further select 50 sentences from the dataset of 1000 sentences, and a human judge extracts all of the correct facts. Then, we use the method by Bronzi et al. [4] to compute the agreement between the automatic and manual evaluations. The agreement is defined as the ratio between the number of facts where the human and automatic evaluators agree and the total number of facts. This agreement was found to be 0.71. With this information, we are able to determine the precision and recall of our information extractors.

We ran OLLIE, ReVerb, and TRIPLEX individually and then we combine TRIPLEX with OLLIE and with ReVerb. Table 1 shows the results in terms of precision, recall, and F_1 (harmonic mean of precision and recall).

ReVerb only generates verb-mediated triples and OLLIE extracts verb-mediated triples and also noun-mediated triples, if they are expressed in verb-mediated styles. TRIPLEX generates noun-mediated triples and it can complement the results of OLLIE and ReVerb. OLLIE, ReVerb, and TRIPLEX all

assign a confidence score to each extracted triple. In these experiments, the extracted triples are only considered if their confidence scores are above a threshold of 0.2. TRIPLEX shows an improvement in Table 1 when using the manual evaluation instead of the automatic evaluation because extracted facts with very low PMI are considered false in the automatic evaluation. However, these facts are often evaluated as true by a human judge. We analyze the errors made by TRIPLEX in the gold standard dataset that was manually annotated. The errors made by TRIPLEX can be classified into two groups: false positives and false negatives. In the gold standard, 65 % of the triples are related to verb-mediated triples, which are not extracted by TRIPLEX.

Table 1. Automatic and manual evaluation of information extractors. OLLIE * only generates noun-mediated triples. The confidence scores of all extracted triples are above 0.2.

	Automatic evaluation			Manual evaluation		
	Precision	Recall	F1	Precision	Recall	F1
ReVerb	0.61	0.15	0.24	0.55	0.11	0.18
OLLIE	**0.64**	0.30	0.40	**0.65**	0.32	0.42
OLLIE*	0.62	0.10	0.17	0.63	0.11	0.18
TRIPLEX	0.55	0.17	0.25	0.62	0.22	0.32
TRIPLEX + OLLIE	0.57	**0.40**	**0.47**	0.63	**0.44**	**0.51**
TRIPLEX + ReVerb	0.58	0.32	0.41	0.55	0.35	0.42

Table 2 shows the results associated with the triples in the gold standard that are not extracted by TRIPLEX. Of those, 10 % obtain low confidence scores (false negatives) because the NER module and WordNet could not find the semantic type for the objects. We penalize the confidence score of a candidate triple if its predicate has one particular property type and if no type is detected for the triple's object. For example, the range of the *nationality* property in DBpedia is a *Location* constraint but neither the NER module nor WordNet can recognize a type in the phrase *Swedish writer* or *Polish-American scientist*. Also, 12 % of the errors are related to the dependency parser, specifically when the parser could not detect a correct grammatical relation between the words in a sentence. Another 7 % of the errors occur when the coreferencing module did not properly resolve coreferential expressions during template extraction. This problem is alleviated by assigning low confidence scores to this group of templates. Finally, 6 % of the errors are caused by over-generalized templates. During template generalization, POS tags are substituted by universal POS tags [10]. Since some templates only extract triples for proper nouns, nouns, or personal pronouns, generalizing and merging these templates together did not produce correct triples.

Approximately 20 % of the false positives are in fact correct triples. This stems from the fact that there are few Google search results for queries that

Table 2. Percentage of the gold standard triples missed by TRIPLEX.

	Missed extractions
10 %	No semantic types
12 %	Dependency parser problems
7 %	Coreferencing errors
6 %	Over generalized templates
65 %	Verb-mediated triples (outside the of scope for TRIPLEX)

contain the subjects of these triples, thus impacting the computation of the PMI scores. Applying the same PMI threshold as used for prominent subjects proved to be ineffective. For example, triples extracted by TRIPLEX are judged incorrect in the sentence, *Alexey Arkhipovich Leonov (born 30 May 1934 in Listvyanka, Kemerovo Oblast, Soviet Union) is a retired Soviet/Russian cosmonaut.* These triples include information about birth date, birth place, origin, and profession, but are not available in the gold standard. Other false positives are due to problems resulting from dependency parsing, named entity recognition, chunking, and over generalized templates.

OIE systems such as ReVerb and OLLIE usually fail to extract triples from compound nouns, adjectives, conjunctions, reduced clauses, parenthetical phrases, and appositions. TRIPLEX only covers noun-mediated triples in sentences.

Table 3. Distribution of correctly extracted triples for TRIPLEX + OLLIE based on their categories. The confidence score of extracted triples by TRIPLEX and OLLIE is above 0.2.

	Distribution	Triple category
Noun-mediated	12 %	Conjunctions, adjectives and noun phrases
	9 %	Appositions and parenthetical phrases
	6 %	Titles or professions
	8 %	Templates with lexicon
Verb-mediated	65 %	Verb-mediated triples

We also examine the output of TRIPLEX with respect to the gold standard, as shown in Table 3. The table shows that 12 % of the noun-mediated triples are related to conjunctions, adjectives, and noun phrases, meaning that TRIPLEX is also able to extract noun-mediated triples from noun conjunctions. For example, TRIPLEX extracts triples about Rye Barcott's professions from the sentences *Rye Barcott is author of It Happened on the Way to War and he is a former U.S. Marine and cofounder of Carolina for Kibera.* Moreover, TRIPLEX is able to extract triples from appositions and parenthetical phrases, and 9 % of the

extracted triples are contained within the triple category of appositions and parenthetical phrases. For example, extracted triples from *Michelle LaVaughn Robinson Obama (born January 17, 1964), an American lawyer and writer, is the wife of the current president of the United States* contain Michelle Obama's two professions, birth date, and nationality. We saw that 6 % of the triples are related to titles or professions, such as *Sir Herbert Lethington Maitland*, *Film director Ingmar Bergman*, and *Microsoft co-founder Bill Gates*. OLLIE is similarly able to capture this kind of triples because they are expressed in a verb-mediated style. However, TRIPLEX does so without using a verb-mediated format. The final fraction of 8 % is for noun-mediated triples that rely on the lexicon of noun-mediated templates. For example, the *headquarters* of Microsoft is extracted from the sentence, *Microsoft is an American multinational corporation headquartered in Redmond, Washington*. Finally, 65 % of the extracted triples are verb-mediated triples. Both ReVerb and OLLIE generate verb-mediated triples from sentences. The majority of errors produced by OLLIE and ReVerb are due to incorrectly identifying subjects or objects. ReVerb first locates verbs in a sentence and then looks for noun phrases to the left and right of the verbs. ReVerb's heuristics sometimes fail to find correct subjects and objects because of compound nouns, appositions, reduced clauses, or conjunctions. OLLIE relies on extracted triples from ReVerb for its bootstrapping process and learning patterns. Although OLLIE produces noun-meditated triples if they can be expressed with verb-mediated formats, it does not cover all formats of noun-mediated triples.

Finally, we analyze some sentences to figure out why different information extractors are not able to produce all of the triples in the gold standard. The first reason is that there may not be sufficient information in a sentence to extract triples. For example, TRIPLEX can find the triple `<Antonio;nationality;Italian>` but it cannot find the triple `<Antonio;nationality;England>` in the sentence *The Italian-born Antonio Verrio was responsible for introducing Baroque mural painting into England*. Second, OLLIE and ReVerb cannot successfully extract verb-mediated triples from sentences that contain compound nouns, appositions, parentheses, conjunctions, or reduced clauses. When OLLIE and ReVerb cannot yield verb-mediated triples, recall will be affected because verb-mediated triples are outside of the scope of TRIPLEX. Improvements to OLLIE and ReVerb could substantially lead to better results in TRIPLEX. Also, improvements to the different NLP components can lead to better precision and recall for information extractors that rely heavily on them.

5   Conclusions and Future Work

This paper presented TRIPLEX, an information extractor to generate triples from noun phrases, adjectives, and appositions. First, a bootstrapping set is automatically constructed from infoboxes in Wikipedia pages. Then, templates with semantic, syntactic, and lexical constraints are constructed automatically to capture triples. Our experiments found that TRIPLEX complements the output of

verb-mediated information extractors by capturing noun-mediated triples. The extracted triples can for instance be used to populate Wikipedia pages with missing infobox attribute values or to assist authors in the task of annotating Wikipedia pages. We also extended an automated evaluation method to include noun-mediated triples.

In future work, we plan to improve upon the generation of extraction templates by considering numerical values and their units (e.g., meter, square meter). We would also like to enrich the bootstrapping set generation process by using a probabilistic knowledge base (e.g., Probase [13]), as it may broaden the coverage of the bootstrapping set and support the construction of more templates.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction for the web. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 2670–2676 (2007)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
4. Bronzi, M., Guo, Z., Mesquita, F., Barbosa, D., Merialdo, P.: Automatic evaluation of relation extraction systems on large-scale. In: NAACL-HLT Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pp. 19–24 (2012)
5. De Marneffe, M.C., Manning, C.D.: Stanford typed dependencies manual (2008). http://nlp.stanford.edu/software/dependencies_manual.pdf
6. Del Corro, L., Gemulla, R.: ClausIE: Clause-based open information extraction. In: International World Wide Web Conference (WWW), pp. 355–366 (2013)
7. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Conference on Empirical Methods on Natural Language Processing (EMNLP), pp. 1535–1545 (2011)
8. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: International Conference Research on Computational Linguistics (ROCLING), pp. 19–33 (1997)
9. Mausam, S.M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 523–534 (2012)

10. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: International Conference on Language Resources and Evaluation (LREC), pp. 2089–2096 (2011)
11. Singhal, A.: Introducing the Knowledge Graph: Things. Not Strings, Official Google Blog, May 2012
12. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Annual Meeting of the Association for Computational Linguistics (ACL), pp. 118–127 (2010)
13. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: ACM SIGMOD International Conference on Management of Data, pp. 481–492 (2012)
14. Xavier, C., Lima, V.: Boosting open information extraction with noun-based relations. In: International Conference on Language Resources and Evaluation (LREC), pp. 96–100 (2014)
15. Yahya, M., Whang, S.E., Gupta, R., Halevy, A.: Renoun: fact extraction for nominal attributes. In: Conference on Empirical Methods on Natural Language Processing (EMNLP), pp. 325–335 (2014)