Open Information Extraction via Contextual Sentence Decomposition[1]

Hannah Bast, Elmar Haussmann
Department of Computer Science
University of Freiburg
79110 Freiburg, Germany
{*bast,haussmann*}@*informatik.uni-freiburg.de*

***Abstract*—We show how contextual sentence decomposition (CSD), a technique originally developed for high-precision semantic search, can be used for open information extraction (OIE). Intuitively, CSD decomposes a sentence into the parts that semantically "belong together". By identifying the (implicit or explicit) verb in each such part, we obtain facts like in OIE. We compare our system, called CSD-IE, to three state-of-the-art OIE systems: ReVerb, OLLIE, and ClausIE. We consider the following aspects: accuracy (does the extracted triple express a meaningful fact, which is also expressed in the original sentence), minimality (can the extracted triple be further decomposed into smaller meaningful triples), coverage (percentage of text contained in at least one extracted triple), and number of facts extracted. We show how CSD-IE clearly outperforms ReVerb and OLLIE in terms of coverage and recall, but at comparable accuracy and minimality, and how CSD-IE achieves precision and recall comparable to ClausIE, but at significantly better minimality.[1]**

***Keywords*-open information extraction; contextual sentence decomposition; semantic search;**

I. Introduction

Information extraction (IE) is the task of automatically extracting relational tuples from natural language text. Such relational tuples typically take the form *subject predicate object (SPO)*, for example: *(Ruth Gabriel) (was born) (in San Fernando)*. In early IE systems, the desired relations (predicates) were part of the input, for example *born in*. Such a system was then typically given, for each such relation, a set of correct triples from which it could learn. In recent years, the trend has been towards open information extraction (OIE), where identifying the predicate and hence the relation is part of the problem [1]. Many systems for OIE have been developed in recent years; we describe the most recent ones in Section II.

A classical use case for information extraction is to obtain fact triples for a formal ontology. This use case requires that the S, P, and O parts are disambiguated, that is, different formulations referring to the same entity are mapped to the same identifier. For example, the S part of the triple above should be mapped to the actress Ruth Gabriel, regardless of whether in that part she is referred to as *Ruth Gabriel* (like above), *R. Gabriel*, *Gabriel*, or *she* (assuming, of course, that all these references actually mean her). In OIE, this disambiguation is typically not considered part of the problem. More than that, many facts extracted by traditional OIE systems are not easily disambiguated, because the S and O part often contain references to more than one entity. We come back to this important aspect below, when we discuss the aspect of *minimality* of a triple.

The motivation for our approach comes from an application called semantic full-text search (SFTS) [2]. SFTS combines formal ontology search with classical full-text search. A typical query would be *class:person word:writer*, searching for co-occurrences of a reference to a *person* with the word *writer*. The intention of the query is to find people who are writers. For results of good quality, it is crucial that the two occurrences (or more for a longer query), actually "belong together" semantically. For example, consider the following sentence, which will be our running example throughout the paper:

(S): Ruth Gabriel, daughter of the actress and writer Ana Maria Bueno, was born in San Fernando.

The sentence contains two references to a person, *Ruth Gabriel* and *Ana Maria Bueno*, as well as the word *writer*. However, only the fact that Ana Maria Bueno is a writer is supported by the sentence. Returning Ruth Gabriel as a hit for the query above would be a mistake. In [2], we therefore proposed *contextual sentence decomposition* (CSD). The goal of CSD is to compute, for a given sentence, all sub-sequences of words in that sentence that semantically "belong together". The sub-sequences are then called the *contexts* of the sentence. A correct decomposition of the sentence above would yield the following four contexts (in any order):

#1: *Ruth Gabriel was born in San Fernando*
#2: *Ruth Gabriel, daughter of Ana Maria Bueno*
#3: *actress Ana Maria Bueno*
#4: *writer Ana Maria Bueno*

Note that not splitting *actress and writer Ana Maria Bueno* into #3 and #4 would be considered a mistake in CSD, because the words *actress* and *writer* do not "belong together" semantically. Rather, in this sentence, they are two unrelated attributes related to the same person.

[1] An extended version of the paper is available via the authors' website: http://ad.informatik.uni-freiburg.de/publications.

In this paper we explore the use of CSD for OIE. In fact, the contexts above already look close to the kind of triples expected from an OIE system. All that is missing is the distinction into the subject, predicate, and object part. Since CSD is computed from a full parse of the sentence, with explicit markup denoting the verb phrases, this is relatively straightforward. Also note that some of the contexts above are missing a verb. In that case the verb is implicit, but can (typically) be easily deduced from the context, e.g. *is* for contexts #3 and #4 (noun phrase with pre-modifying noun phrase). Our system, called CSD-IE is described in detail in Section III.

A. Quality Aspects

In this paper, we evaluate OIE systems with respect to the following three quality aspects.

1. The accuracy of the extracted facts. Two aspects are important here. First, whether the fact actually has the form of a meaningful relational triple. For example, the triple *(Ruth) (Gabriel was) (San Fernando)* would be considered inaccurate for two reasons: (1) the P part contains words which do not belong to the verb (but to the S part in this case), and (2) the P and the O parts do not fit together. Second, what is expressed by the triple should also be expressed by the sentence. For example, *(Ruth Gabriel) (is) (actress)* would be accurate according to the criterion just mentioned, but it's not expressed in our sentence (S) above.

Accuracy is typically assessed by human judges, see Section IV. The percentage of the extracted facts deemed accurate is typically referred to as the *precision*.

2. The number of extracted facts. An average sentence may express a lot of facts. For example, our example sentence from above expresses four facts:

#1: *(Ruth Gabriel) (was born) (in San Fernando)*
#2: *(Ruth Gabriel) (is) (daughter of Ana Maria Bueno)*
#3: *(Ana Maria Bueno) (is) (actress)*
#4: *(Ana Maria Bueno) (is) (writer)*

It is the explicit goal of our system to extract, from a given sentence, as many facts as possible, and lose as little information as possible. In Section IV we measure this by the *coverage*, that is, the percentage of all word occurrences that occur in at least one extracted triple. For the sentence above, the coverage is 100%.

3. The minimality of the extracted facts. An accurate fact may itself contain other (accurate) facts. For example, *(Ruth Gabriel) (is) (daughter of the actress and writer Ana Maria Bueno)* would be considered an accurate fact according to our definition above. However, we already explained above that this fact contains two other facts (namely, that Ana Maria Bueno is an actress and a writer), which are unrelated to the containing fact and that this mixture of unrelated facts is problematic for applications that require "semantic togetherness" of the words in a fact. It is therefore another explicit goal of our system to extract minimal facts. The four facts listed under 2. above are all minimal.

The goal of minimality comes with a challenge that is not apparent in our example sentences, but occurs in sentences of a more complex type. For example, consider the sentence:

The Embassy said that 6,700 Americans were in Pakistan.

This sentence contains two facts: that the Embassy made some statement, and that 6,700 Americans were in Pakistan. However, by simply extracting these two facts, we would lose the information what statement the Embassy made. We solve this by allowing the S and O part of a fact to contain *references* to other extracted facts. In this case, we would extract:

#1: *(The Embassy) (said) (that #2)*
#2: *(6,700 Americans) (were) (in Pakistan.)*

where the numbers are simply unique ids for each extracted triple. That way, no information is lost, and the application may choose to either keep the facts separate (and avoid mixing of facts) or substitute the references with the referred fact (and thus obtain output as in other OIE systems). A similar issue was addressed by OLLIE [3] using what they call additional context information; see Section II.

B. Our contribution

We present a new system for open information extraction, called CSD-IE, that is good in all three aspects above. We compare our approach to what we consider the three best previous approaches: ReVerb, OLLIE, and ClausIE. CSD-IE outperforms ReVerb and OLLIE in terms of coverage and recall, but at comparable precision and minimality. It also achieves precision and recall comparable to ClausIE, but at significantly better minimality. For some details about previous systems and how they relate to our approach see the next Section II. The details behind our CSD-IE are described in Section III. Our evaluation is provided in Section IV.

II. Related work

A large variety of OIE systems have been developed in recent years, starting with the original TextRunner [1], over WOE [4], R2A2 [5], ReVerb [6], OLLIE [3], to the very recent ClausIE [7]. A good and up-to-date overview over these and other systems is provided in [7]. In the following we will shortly describe how our strongest competitors (ReVerb, OLLIE, and ClausIE) relate to our approach.

ReVerb explicitly addressed the issues of *incoherent* extractions and *uninformative* extractions. Using shallow NLP and learned extraction patterns ReVerb achieves a significantly better precision than its predecessors. Our approach utilizes a full parse which helps with these problems; see the description of ClausIE below and our description of CSD-IE in Section III.

OLLIE improves over ReVerb by addressing two further issues important for extraction quality. The first issue are facts not mediated by verbs. The second issue is additional information about facts expressed in indirect speech (*He said that* ...) and the like. OLLIE relies on a dependency parse and achieves significantly better recall than ReVerb with basically the same good precision. Our contextual sentence decomposition deals with these issues by allowing triples to contain references to other triples and explicitly considering facts not mediated by verbs. This has the advantage of simultaneously addressing the issue of information loss and of minimality.

ClausIE is the most recent OIE system, and our strongest competitor. In fact, the basic approach of ClausIE is very similar to ours: decompose each sentence into its basic constituents (called "clauses" in ClausIE), and from those constituents derive triples. Our approach has been developed in independent work and the basic ideas behind our contextual sentence decomposition (CSD), as used for our semantic search, have already been published in [8] and [2], long before [7]. Still, there are some important differences between our CSD-IE and ClausIE.

First and foremost, we make minimality a primary goal of our system. This was motivated by our application to semantic full-text search. However, minimality is also important for the more universal task of transforming the OIE triples into disambiguated facts within a formal ontology.

Second, we provide a more principled description of how we obtain our sentence constituents, by first transforming the (fine-grained) parse tree into a (coarse-grained) constituent tree, and then treating that tree like an expression tree (with different operators) to obtain what we call our contexts (from which we then derive our triples). In particular, the second step is tricky when relative clauses are nested in enumerations or vice versa. This aspect is not addressed in the description of [7, Section 4.2].

III. CSD AND CSD-IE

We describe our system CSD-IE. The main idea behind CSD-IE is *contextual sentence decomposition* (CSD). CSD is performed in two steps. First, basic building blocks of our contexts are identified in the *sentence constituent identification* (SCI) phase. A tree expressing the semantics is derived. In the second step, *sentence constituent recombination* (SCR), the tree constituents are combined to form our contexts. To derive triples from resulting contexts we introduce a (relatively trivial) third phase. Throughout this section we use the running sentence (S) from Section I.

The next sub-sections define the conceptual ideas of SCI and SCR. We describe an implementation of SCI based on constituent parse trees in section III-C. The triple generation phase is described in section III-D.

A. Sentence Constituent Identification

The task of SCI is to identify the basic "building blocks" of our contexts in a sentence and arrange them in a tree. It turns out that, for our purposes, mainly *relative clauses* and what we call *enumeration items* are important, because they usually contain separate facts that have no direct relationship to the other parts of the sentence. In our sentence (S) from above, the (reduced) relative clause *daughter of the actress and writer Ana Maria Bueno* refers to *Ruth Gabriel* but has nothing to do with the rest of the sentence. Furthermore, the relative clause contains the nominal (pre-)modifier *the actress and writer* modifying *Ana Maria Bueno*, which we consider to be another type of relative clause. The nominal modifier itself contains an enumeration of two coordinated *enumeration items*: *the actress* and *writer*. These have nothing to do with each other, except that they both refer to Ana Maria Bueno.

More specifically now, SCI computes a tree with the following types of nodes:

ENUM: an enumeration. Each child corresponds to an enumeration item and belongs to a different context.

CONC: a group of child-nodes (constituents) that belong to the same context.

REL: a relative clause with a link to its head (the noun phrase or clause it closer describes).

SUB: a clause or sub-sequence corresponding to a self-sufficient context, e.g., a prepositional phrase describing a complex circumstance representing some fact on its own.

LEAF: a leaf that contains words (terminals) of the sentence.

Nodes can be nested in an arbitrary fashion and arbitrarily deep. Figure 1 depicts an SCI tree of our example sentence.

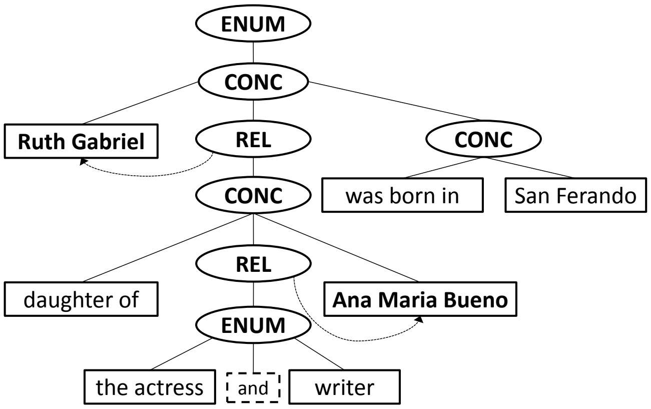

Figure 1. The SCI tree for our example sentence (S). The head of each relative clause is printed in bold, filler words in striped rectangles.

B. Sentence Constituent Recombination

The SCR phase recombines the constituents identified by the SCI phase to form the final *contexts*. SCR recursively computes contexts from an SCI tree or subtree as follows:

(SCR 0) Take out each subtree labeled REL or SUB and change the root of this new tree to CONC. For REL, add the head as the leftmost child (but leave it in the SCI tree, too). For SUB, leave a reference to the newly created tree in the original tree. This is the only place references to other contexts need to be considered. Then process each such subtree and the remaining part of the original SCI tree (each of which then only has ENUM and CONC nodes left) separately as follows:

(SCR 1) For a leaf, there is exactly one context: the part of the sentence stored in that leaf.

(SCR 2a) For an inner node, first recursively compute the set of contexts for each of its children.

(SCR 2b) If the node is marked ENUM, the set of contexts for this node is computed as the *union* of the sets of contexts of the children.

(SCR 2c) If the node is marked CONC, the set of contexts for this node is computed as the *cross-product* of the sets of contexts of the children.

Applying these rules to the SCI tree in Figure 1 yields the desired contexts shown in section I:
In a final step we generate triples from the extracted contexts, see Section III-D. We note that, given the SCI tree and the definition above, SCR is straightforward and fully defined. Therefore, the challenging part of CSD is computing the SCI tree.

C. Sentence Constituent Identification Based on Deep Parse Trees

We present an approach to SCI that is based on the output of a state-of-the-art constituent parser. Figure 2 depicts the parse tree for our example sentence (S).

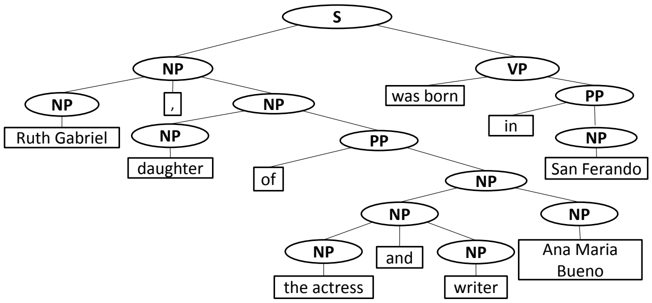

Figure 2. Constituent parse tree for our example sentence. For the sake of readability, the parse tree has been simplified.

We manually created a small set of rules with which we can derive an SCI tree from a parse tree. In the following description when we speak of, for example, an *NP* (noun phrase) we refer to nodes in the parse tree with that tag.

(SCI 1) Mark as ENUM each node, for which the children are all of the same type (e.g. all *VP*), but interleaved by punctuation or conjunctive constructions. (This avoids splitting objects of di- and complex-transitive verbs).

(SCI 2a) If the sequence consist of only two *NPs* split by a comma (and not some conjunction) this is an apposition. Mark the second *NP* as REL and the first *NP* as its head.

(SCI 2b) Mark as REL each *SBAR* and *PP*, if it starts with a phrase in *WHPP*, *WHADV*, *WHNP* or with a word from a positive-list (e.g., *such as* or *who*) but not with one from a negative-list indicating temporal relations (e.g., *before*, *when*).

(SCI 2c) Mark as REL each *PRN* contained in round brackets *"(", ")"*.

(SCI 2d) Mark as REL each *S* that is preceded by an *NP* followed by a comma.

(SCI 2e) Mark as REL each *VP* that is preceded by a comma, if it has not previously been marked as part of an enumeration and starts with a word with part-of-speech tag of *VBN*, *VBG* or *VBD*. The verb can optionally be preceded by an adverbial phrase *ADVP*.

(SCI 2f) Mark as REL each *VP* below a *WHNP* or *NP*, starting with a word with a part-of-speech tag of *VBN*, *VBG*, or *VBD*. These are participial clauses acting as relative clauses.

(SCI 2g) For all REL from above define the closest left sibling *NP* as the *head*. If there is no left sibling *NP* move down towards the leaves and use the closest first *NP* to the left as head (low/local attachment).

(SCI 3) Mark as REL each *NP*, which has an *NP* as parent and which has exactly one right sibling *NP*. This *NP* is premodifying the right sibling *NP*. Mark the right sibling *NP* as head.

(SCI 4) Mark as REL each *NP*, which has as a first (or last) word with part-of-speech tag *PRP$*. This indicates a possessive relation. Mark the last (or first) part of the *NP* as head. If the *NP* also has a right sibling mark it as additional head.

(SCI 5a) Mark as SUB each *PP* starting with a preposition from a positive-list of temporal indicators (e.g., *before* or *while*), each *PP* enclosed in commas and all *PP*s at the beginning of a sentence. These often describe circumstances of events and should be considered separately.

(SCI 5b) Mark as SUB each *SBAR* if it is not already marked REL.

(SCI 5c) Mark as SUB each *S* if it is below an *SBAR* and contains *NP* as well as *VP*, but only if it is not directly below a node of type *PP* or *PRN* (these were treated separately).

(SCI 6) Mark as CONC all remaining nodes. Contract away each CONC with only text nodes in its sub-tree (by merging the respective text) and merge CONC nodes that only have CONC nodes as children.

Applying these rules to the parse tree in Figure 2 produces the SCI-tree displayed in Figure 1. As our quality evaluation in Section IV shows, in general, this small set of rules already works reasonably well.

D. Triple Generation

The triple generation from extracted contexts is (relatively) straight-forward. For each context, we identify the first explicit verb phrase and surrounding adverbs to be the predicate. Everything before that belongs to the subject, and everything after that to the object. For the contexts from our example sentence (S) this gives us the triple #1 from the desired triples shown under 2 in Section I-A.

For relations resulting from SCI 2a-g and SCI 3 rules we add the verb *is* between the head and its attachment, but only if the attachment does not begin with a verb phrase (we then use this verb phrase as predicate). This gives us the remaining triples #2, #3 and #4 for our example sentence from above.

To deal with possessive relations identified by the SCI 4 rule (see section III-C above) we use the predicate *has* between head and attachment.

In some cases the object consists of a list of noun or prepositional phrases (that are not enumeration items), for example in the sentence:

Soubry graduated in law from the University of
Birmingham in 1979.

Here, the object consists of three parts: *(in law)*, *(from the University of Birmingham)* and *(in 1979)*. We identify these noun and prepositional phrases in the object when creating our contexts in the SCR-phase, utilizing information from the parse tree. From *n* identified phrases in the object we derive triples, by in turn appending zero or one of the *n - 1* last phrases to the first phrase:

#1: *(Soubry) (graduated) (in law)*
#2: *(Soubry) (graduated) (in law from*
the University of Birmingham)
#3: *(Soubry) (graduated) (in law in 1979)*

For *n* identified phrases in the object, this results in *n* triples.

IV. Evaluation

We compare CSD-IE against the three OIE systems ReVerb, OLLIE, and ClausIE. CSD-IE was implemented as described in Section III, using the Stanford Constituent Parser [9]. ClausIE was run in its default mode to extract triples.

We evaluated the systems on two datasets: 200 random sentences from the English *Wikipedia*, and 200 random sentences from the *New York Times*. For comparability, we used the exact same datasets as in [7], which the authors made available on their web site[2]. The labels of their evaluation are also available, however, we chose to label all extracted triples (by all four systems) again, for the following reasons. First, for reasons of consistency (there is always room for interpretation when it comes to assessing accuracy and minimality). Second, to avoid a bias in favor or against a certain system (all extractions were labeled together). Third, because we also wanted to explicitly assess minimality for each extraction.

	ReVerb	OLLIE	ClausIE	CSD-IE
#facts	249	408	610	677
#facts correct	188	230	421	474
prec-a	75.5%	56.4%	69.0%	70.0 %
prec-m	87.2%	80.4%	57.0%	76.8 %
coverage	47.2%	62.7%	95.4%	97.5 %
triple length	7.3	9.7	11.0	8.4

	ReVerb	OLLIE	ClausIE	CSD-IE
#facts	271	358	644	743
#facts correct	177	200	412	531
prec-a	65.3%	55.9%	64.0%	71.5 %
prec-m	93.8%	79.5%	71.4%	90.4 %
coverage	43.8%	61.0%	89.0%	91.8 %
triple length	7.0	10.4	11.6	7.8

Table I
Results of our quality evaluation for all four systems for the two datasets Wikipedia (above) and New York Times (below).

The evaluation of [7] also comprises the ReVerb dataset from [6], which consists of 500 random sentences from the Web. However, the relative quality differences between evaluated systems on that dataset were essentially the same. Preliminary experiments on the ReVerb dataset have confirmed this impression, and we therefore excluded it from our (already cumbersome) evaluation.

As explained in Section I, we labeled each extraction (from each of the four systems) with two labels: one for accuracy (yes or no) and one for minimality (yes or no). From these labels, we computed the following accumulated measures for each system:

precision wrt accuracy (prec-a) = the percentage of triples labeled as accurate; see Section I
precision wrt minimality (prec-m) = the percentage of correct triples labeled as minimal; see Section I
coverage = the percentage of word occurrences that occur in at least one extracted triple for that system
average triple length in words = average length of extracted triples for that system in words (ignoring special characters)

Table I shows the results for the two datasets. We note that the numbers for both datasets closely agree with those reported in [7], confirming the reasonability of our labels. We first discuss the results for the Wikipedia dataset (Table I, upper part). From all systems CSD-IE extracts the largest number of facts overall (*#facts*), as well as the largest number of correct facts (*#facts correct*). It also provides the highest coverage of all systems (*coverage*). The precision wrt accuracy (*prec-a*) and coverage of CSD-IE is comparable

[2] http://www.mpi-inf.mpg.de/departments/d5/software/clausie.

to that of ClausIE, but our extracted facts are shorter on average (*triple length*). Furthermore, the precision wrt minimality (*prec-m*) is 20% higher - a drastic improvement. This is because we explicitly consider minimality in our approach and, as a result, our extracted facts are shorter. ReVerb, the only system not utilizing a deep (constituent or dependency) parse, uses patterns that match relatively short facts. These seem to be very accurate, resulting in the best precision and precision wrt minimality scores. In contrast, coverage and the number of extracted facts are not nearly as good as those of CSD-IE or ClausIE. OLLIE improves upon ReVerb by extracting more facts and providing higher coverage. However, overall precision drops drastically. This has also been observed before in [7]. A common problem for deep-parse-based OpenIE systems is the large influence of parser errors. CSD-IE uses the Stanford Constituent Parser [9], OLLIE uses the MaltParser [10] and ClausIE uses the Stanford Dependency Parser [9]. One possible reason for the comparably low precision of OLLIE might be that the MaltParser was trained on data from a different domain and had problems with the sometimes ungrammatical Wikipedia sentences.

The results for the New York Times dataset, (Table I, lower part) are similar to those of the Wikipedia dataset. Sentences are typically longer and deeply nested, causing more broken parses, resulting in slightly worse coverage for all systems. The sentences also contain more facts - almost all systems extract a slightly larger number of facts. Because indirect speech and nested facts are very common in news articles, many triples extracted by CSD-IE contain references to other facts and are therefore more compact and minimal.

A preliminary investigation of errors for CSD-IE on both datasets revealed that most of the inaccurate extractions were caused by mistakes in the parse trees. We consider this an important direction for future research, see the next section.

V. Conclusions and Future Work

We have presented CSD-IE, a new system for open information extraction (OIE). CSD-IE is based on contextual sentence decomposition (CSD), a technique originally developed for semantic full-text search (SFTS). Our evaluation has shown that CSD-IE simultaneously achieves good precision, high recall and very good coverage and minimality. The aspects of coverage and minimality are particularly important for applications such as semantic search, where precise facts and small information loss are desirable. Our work has raised some interesting directions for future research.

A preliminary error analysis shows that most inaccurate extractions are due to a mistake in the (constituent) parse. Since parsing is a complex problem, it would be interesting to make the sentence constituent identification (SCI, see Section III-A) more robust against errors in the parse. Or, alternatively, pre-process the sentence, such that certain common errors are avoided.

A more ambitious plan would be to avoid the "detour" via a deep parse altogether, and try to identify the sentence constituents in a more direct manner. Since our SCI tree III-A is significantly more coarse-grained than the original parse tree, there is hope that this problem can be solved more efficiently and with higher quality.

A common phenomenon in more complex sentences is that of *transitivity*. For example, consider the sentence

The ICRW is a non-profit organization headquartered in Washington.

The following two triples would be considered both accurate and minimal in our approach:

#1: *(The ICRW) (is) (a non-profit organization)*
#2: *(a non-profit organization) (headquartered) (in Washington).*

However, triple #2 would be significantly more informative, if the S part were replaced by *The ICRW*, that is, the concrete entity to which it actually refers. It would be desirable to deal with transitivity of this and more complex kinds as a part of the OIE process.

References

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *IJCAI*, 2007, pp. 2670–2676.

[2] H. Bast, F. Bäurle, B. Buchhold, and E. Haussmann, "Broccoli: Semantic full-text search at your fingertips," *CoRR*, 2012.

[3] Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni, "Open language learning for information extraction," in *EMNLP-CoNLL*, 2012, pp. 523–534.

[4] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *ACL*, 2010, pp. 118–127.

[5] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam, "Open information extraction: The second generation," in *IJCAI*, 2011, pp. 3–10.

[6] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *EMNLP*, 2011, pp. 1535–1545.

[7] L. D. Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *WWW*, 2013, pp. 355–366.

[8] E. Haussmann, "Contextual sentence decomposition with applications to semantic full-text search," Master's thesis, University of Freiburg, July 2011.

[9] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL*, 2003, pp. 423–430.

[10] J. Nivre, J. Hall, and J. Nilsson, "Memory-based dependency parsing," in *Proceedings of CoNLL*, 2004, pp. 49–56.