Structural Sentence Decomposition via Open Information Extraction

Cheikh Kacfah Emani, Catarina Ferreira da Silva, Bruno Fiès, Parisa Ghodous, Farzad Khosrowshahi

Structural Sentence Decomposition via Open Information Extraction

Cheikh Kacfah Emani, Catarina Ferreira Da Silva, Bruno Fiès, Parisa Ghodous, and Farzad Khosrowshahi

Abstract— The field of construction engineering is governed by an important volume of legal texts. Each of these texts provide a set of requirements by means of sentences written in natural language. These texts support conformity checking process of objects in construction engineering. Automate or at-least semi-automate this conformity checking process is the target of our project. This automation imposes to be able to make legal requirements processable and therefore as a first step, rewrite them in a more unambiguous and simple way. Thus, we envisage to rephrase each natural language sentence into a set of atomic requirements (i.e a triple: <Subject, Predicate, Object>). Further, to preserve the actual meaning of the requirement, we must identify the relations between these extracted facts. For instance, a two-fact sentence can be written as ($fact_1$ AND $fact_2$) or (IF $fact_1$ THEN $fact_2$), etc. For facts extraction, we have used existing Open Information Extraction-systems. Since they face a drop of precision mainly due to multi-word expressions, we provide a method to handle them before OIE itself. Moreover, we also tackle the problem of enumeration items to improve OIE performance. Improvements due to these two aspects have been evaluated on classic OIE and building engineering construction corpora. Using OIE for legal sentences semantics handling, improvement of OIE by means of multi-word expressions and finally computation of relations between facts constitute the originality of this work. Such decomposition is the prerequisite to our target: automatic conformity checking.

Index Terms— Structural Sentence Decomposition, Open Information Extraction, Multi-word Expressions, Business Rules

INTRODUCTION

Many pieces of text govern the field of construction engineering. In addition to the literary freestyle, long sentences, these texts may be full of ambiguities. In our aim to get a formal representation of requirements expressed in these texts, it is essential to fully get their real content. By (real) content, we mean the main piece(s) of information and its (their) context or meaning modifier. Indeed, this work is part of a bigger process described in [9]. This target process aims to perform automatic conformity checking in the field of building construction. In [9], the author shows how important it is to rephrase each natural language requirement into a set of more easy processable facts. Let us take as example the sentence: "When built in hazardous zones or considered as such, buildings must have entrances with door whose width is greater than or equal to 2 metres for ground floor and 1.5 metre otherwise.". We have: [**main (1)**- buildings must have entrances with door whose width is greater than or equal to 2 metres for ground floor], [**main (2)-** buildings must have entrances with door whose width is greater than or equal to 1.5 metre otherwise] and [**Context-** when built in hazardous zones or considered as such[1]]. Let us note that a semantic decoding of the phrases for ground floor and otherwise in "the mains" might help to point out the *if then else* structure of the sentence. As presented above, this rearrangement is very similar to what Galichet [8, 7] calls *the structural decomposition of the sentence*. Although this task can be seen as Open Information Extraction (OIE), the usage of OIE-tools on our corpus (legal French texts relative to building construction) gives poor precision. It is mainly due to many implicit facts (example: "and 1.5 otherwise" instead of "[...] door whose width is 1.5 otherwise"). Moreover, multi-word terms such as "greater than or equal to" and subordinate clauses like "when built in hazardous zones" are responsible of a lot of noise within facts extracted by an OIE-system.

Consequently, we envision making a number of processing on the original sentence before submitting it to an OIE-tool. All of these early steps aim at performing IE on a more easy to explicit and less noisy version of the original sentence. Next, triples provided by this information extraction step will be organised to better highlight the actual meaning of the sentence. This highlighting concerns phrases, which give more precision to a given fact or the real structure of the sentence even if implicit (like in the example above).

The originality of this work lays in the using of OIE to get the structure of legal sentences, their pre-processing before applying a OIE tool, the improvement of these OIE results when using a priori domain knowledge and the highlighting of the relations between facts in a sentence. Using OIE for legal sentences structuring helps to get every assertion in a sentence regardless to the position (within the sentence) of the words, which constitute this assertion. In addition, when identifying contextual clauses and multi-word expressions, we get more correct and informative facts. Our goal is discussed through the following agenda. Initially, a brief state of the art on OIE and legal sentences decomposition (Sect. 2). Next, we detail our contribution (Sect. 4 and 5). Finally, our algorithms are tested on a set of legal sentences (Sect.6).

1 RELATED WORK

1.1 Open Information Extraction

As mentioned in the introduction, the sentence decomposition we want to achieve is similar to OIE. The difference with OIE is that each triple may be true in a given context and this context must be highlighted as much as possible. During the recent years, many systems were developed to perform OIE. It is the case of ReVerb [6], OLLIE [11], ClausIE [5] and CSD-IE [1]. ReVerb by means of efficient heuristics, focused on incoherent and uninformative triples. Unfortunately, relations extracted by ReVerb were necessarily verb-

- *Cheikh Kacfah Emani is with CSTB, 290 route des Lucioles, BP 209, 06904 Sophia Antipolis, France and Université Lyon 1, LIRIS, CNRS, UMR5205. E-mail: cheikh.kacfah@cstb.fr*
- *Catarina Ferreira Da Silva is with Université Lyon 1, LIRIS, CNRS, UMR5205. E-mail: catarina.ferreira@univ-lyon1.fr*
- *Bruno Fiès is with CSTB, 290 route des Lucioles, BP 209, 06904 Sophia Antipolis, France. E-mail: buno.fies@cstb.fr*
- *Parisa Ghodous is with Université Lyon 1, LIRIS, CNRS, UMR5205. E-mail: parisa.ghodous@univ-lyon1.fr*
- *Farzad Khosrowshahi is with Leeds Metropolitan University. E-mail: F.Khosrowshahi@leedsmet.ac.uk*

[1] We underline this term because in practice, it expresses the context by itself.

based. This is the main reason why OLLIE, developed by the same group of researchers, was provided. In addition to be able to identify non verb-driven facts, OLLIE aims to provide the context/condition, if existing, in which the extracted fact can be considered true. Obviously, this objective of OLLIE is comparable to our, but in many cases, this tool is not always able to identify this context. Moreover, the context, which can be complex (a main fact with a context or multiple coordinated facts), is not processed by OLLIE. If these two previous tools were trained to identify a relation, the recent ones analyse the structure of the sentence and the grammar dependencies to do it. Indeed, ClausIE first identifies the basic constituents of a sentence (clauses) from which it extracts triples. Likewise, CSD-IE first provides contexts (sub-sequences of words of the sentence that are semantically related) and then identifies facts within these contexts. In general, ClausIE and CSD-IE give better results than OIE-systems which use learning. Indeed, learning depends on the training dataset. Unfortunately, a labelled corpus for OIE may not provide a certain type of implicit relations. In addition, results of CSD-IE should respect some quality aspects and these constraints make the results more suitable. These quality requirements are accuracy (a set of heuristics help to identified incoherent triples), coverage (each word in the original sentence should appear in at least one fact) and minimality which considers as inaccurate a triple from which another fact can be extracted. Moreover, going further with informativeness of triples, Bast and Haussmann added some inference rules to improve results of CSD-IE [2]. Finally, CSD-IE is able to link triples (example: $triple_A$ if{$triple_B$, $triple_C$}) in some cases and to extract n-tuples ($n \geq 1$). Indeed, the design of CSD-IE leads to group phrases that semantically "belong together". Consequently, some pieces of text will stand alone (example: the phrase "When built in hazardous zones").

We make some pre-processing of sentences before submitting it to an OIE-tool and then exploit extracted facts to provide the desired result.

1.2 Legal Sentences Decomposition

In the previous paragraph we underlined the pioneering work of Maussam et al. [11] when designing OLLIE. More focusing on legal texts, De Maat and Winkels [3] and Sayah [12] have addressed a similar problem within the (E) Power project. In these two works, few categories have been identified to classify a legal text. For each category, we have a set of Juridical (Natural) Language Construct (JLC). A JLC is a kind of pattern for sentences found in legal texts. For example, the two JLCs **[If]<subject><feature>** and **[Insofar]<subject><feature>** are the patterns for rules which belong to the category *explicit condition*. Since sentences can be very complex, within the sub-sequences of words referenced by **<subject>** or **<feature>,** one or more JLCs could be found. Hypothetically, a sentence can contain many splitting symbols (punctuation marks) like commas, semicolons, colons, etc. Therefore the identification of the exact sequence of words which belongs to an "element" (between the symbols < and >) found in a JLC is not straightforward. Moreover, phrases in a sentence which constitute a fact, are not always contiguous.

2 MOTIVATIONS

In this work, we envisage to provide the structure of legal sentences in the field on building engineering construction. However, we aim to propose a methodology easily adaptable in any field, in an open environment. We envision obtaining this structure by identifying any assertion in the sentence (Information Extraction) and then by decoding the hints given by the syntactic tree (subordinate clauses, coordinating conjunctions, etc.).

When we use existing OIE-tools, in some cases we may have incorrect extractions. These wrong extractions are mainly due to parsing errors (*incorrect grammatical dependencies* for a tool like ClausIE and *incorrect parse tree* for a tool like CSD-IE). An analysis of the wrong extractions, thus of the wrong parsing tasks, points out in some cases the problematic role of multi-word expressions (MWE). This assumption is illustrated by the Table 1 which shows the percentage of errors causes by MWE within wrong extractions (using CSD-IE[2] as extraction tool). To get these numbers, we have selected the 30 first sentences (preliminary statistics) of the New York Times dataset where CSD-IE *had at-least one incorrect extraction*. On this small corpus, we have, similarly, added 15 sentences (preliminary tests) from European Norms[3] in the field of building construction. Then, we have summed up the number of errors caused by a MWE and divided it by the total number of errors. In these preliminary statistics, we label an incorrect extraction as due to a (given) MWE if the responsible MWE had been parsed (by the Stanford parser) wrongly. We consider as a MWE in a given sentence every phrase which the meaning will be modified by the addition or the deletion of any of its word. Thus define, a MWE can be: a noun phrase and its determinant, a phrasal verb, a cardinal number and its unit of measurement, a concept (in a given field), a named entity, an idiomatic expression etc. This definition of a MWE makes us notice that in almost every sentence, we have a MWE. Moreover, MWEs cause parsing errors and parsing is the entry step of any IE process. Consequently, the problematic role played by MWEs *must be handled at the top of OIE*. Such preprocessing is one of a goal of this work. In addition, we also use the "unbreakable property" of MWE after OIE. Indeed, we do not allow a triple to have only a fragment of a MWE.

Moreover, another critical problem to solve by OIE-tools is the handling of enumerations. Indeed, ability to correctly identify enumeration items and to "place" them accurately in different triples is a main concern in OIE. We always make a proposal for enhancing existing solutions for this problem.

Finally, as introduced in [11] and improved in [1, 2], we have a focus on how triples must be harmonized to get the actual meaning of the sentence. This organization is a key step for our automatic conformity checking process.

Table 1. Evaluation of the importance of MWE (errors due to MWE and improvement after handling it) using the New York Times dataset and a dataset build from European Norms in the field on building construction

Datasets	Number of errors "before MWE"	Number of errors due to MWE	Number of errors "after MWE"
New Yok Times	51	22 (43.13%)	33 (35.29% fewer errors)
European Norms	31	19 (61.29%)	15 (51.61% fewer errors)

[2] It is the more efficient OIE-tool to our knowledge.

[3] http://sagaweb.afnor.org/

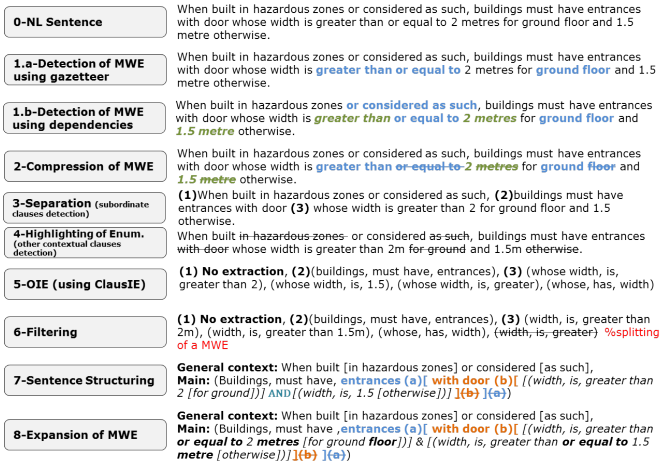

Fig. 1. Overview of an example of structural sentence decomposition. All the steps occur sequentially and the input of the step *i* is the output of the step *i+1*.

3 Preprocessing Of Sentences

We have mentioned earlier that implicit sub-sequences of words and the noise due to multi-word expressions (MWE) were two main drawbacks for an OIE in our corpus. Of course, these remarks can be extended to more general corpus (as shown by Table 1). Moreover, we are strongly interested in the context, if available, of the extracted facts. Consequently, to get better results from the OIE-systems in general, we decide to "help" them by handling MWE (Sect. 4.1), enumerations items (Sect. 4.3) and subordinate clauses (Sect. 4.2). All these tasks aim at providing a sentence more easy to parse and thus to look for information within it. Since we completely have the hand on the inputs of an OIE-process, we finally manage its output to obtain the structure of the original sentence. All these steps are deeply described below and are depicted by Fig. 1.

3.1 Handle Multi-word Expressions

In the introduction, we have highlighted the role played by multi-word expressions (MWE) in the poor precision of OIE-tools. For instance, by processing the phrase "greater than or equal to" as a non-compound term, ClausIE extracts the following (incorrect) triple: *<entrances with door width, is, greater>*. Our solution to improve the quality of OIE-tools when they face this problem is a three-step operation: **(i)** *Detect* MWE, **(ii)** *compress* each MWE and **(iii)** *expand* each MWE at the end of the whole process (it is the latest of all the tasks described in this paper).

3.1.1 Detection of Multi-word Expressions

MWE are usually domain terms or recurrent domain-independent terms. When we mention domain terms we have in mind terms which can be found in a thesaurus or any knowledge base. Domain-independent terms are not related to the field of study but are frequently found in the corpus. This second category of terms has been manually obtained. Indeed, a gazetteer has been built thanks to the experience of experts in construction engineering. A similar idea, master expressions which are not linked to a given domain but which are unavoidable to get a formal representation of sentences written in natural language, can be found in [13, 10]. Therefore, this step is simply a plain text search, where we have list of domain terms and a list of recurrent domain-independent terms. In our running example, this stage enables to detect the two domain-independent MWE *"or considered as such"* and *"greather than or equal to"* and the domain-term *"ground floor"*. In practice, we can extend the list of these predefined MWE by reasoning on some typed dependencies. Indeed NLP tools used for this task can provide dependencies like *mwe*. This relation between two words means that they both act as a single word. Moreover, dependencies like *det* (determiner) or *predet* (predeterminer) and, more important in our context, *num* (numeric modifier[4]) are considered to be synonyms of the *mwe* relation. This step is illustrated by the steps **1.a** and **1.b** on the Fig. 1.

3.1.2 Compression of a Multi-word Expression

The reason why precision of OIE-tools is affected by MWE is that the latter is considered by the former to be non-atomic. Hence, an efficient OIE-system will either try to identify a triple in it or to use parts of the MWE independently. This situation made us decide to replace, within the initial sentence, all the identified MWE by their shortened version. A MWE will be shortened automatically by following this two-steps algorithm:

- if the MWE is hinted by the *num* dependency, the cardinal number is taken as the shortened version of the MWE;
- else, if the MWE contains at-least a noun, the first of thoses nouns is considered to be the shortened version of the MWE;
- else, we take the string provided by the smallest phrase[5] within the syntactic tree.

Let us mention that named entities are not shortened at all.

If we applied this small algorithm by taking as input *"greater than or equal"* and the dependencies graph of Fig. 2, we get as result *"greater than"*. Likewise, the shortened version of *"ground floor"* is "*ground*". Consequently the whole sentence of the example become ***"When built in hazardous zones or considered as such, buildings must have entrances with door whose width is greater than 2 for ground and 1.5 otherwise."*** (Step 2 on Fig. 1)

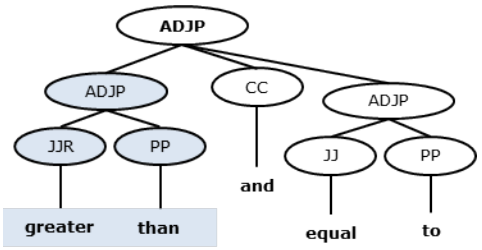

Fig. 2. Syntactic tree of the multi-word expression "greater than or equal to" (provided by the Stanford parser)

4.1.1 Expansion of a multi-word expression

This final step aims at reconciling facts which will be extracted from the sentence and the original sentence. Therefore it occurs as the final step of the whole process of our structural sentence decomposition (see the last step of the pipeline depicted by Fig. 1). Expansion therefore consists in replacing shortened representation of each MWE by its original one as presented in the initial sentence. This task is depicted by the step 8 (Fig. 1)

4.2 Separation of Contextual Clauses from Main Clause

We have identified contextual clauses as a kind of noise which spoils the precision of OIE-tools results. This remark has been done when designing pioneering OIE-systems OLLIE [11] and CSD-IE [1]. These works point out the importance not to give a wrong role to such clauses. In our work, putting these contextual clauses to their right place is crucial. Maussam and colleagues [11] who call these

[4] Example: 'metres' is the *numeric modifier* of the cardinal number 2 in our illustrative sentence.

[5] An exhaustive list of labels for phrases is available in the Penn Treebank.

clauses ClausalModifiers, aim to identify the clause, which is actually modified by these ClausalModifiers. Although Bast and Haussmann [1] circumscribe subordinate and relative clauses, they are not interested in the deep semantic role of these clauses.

In our current work, we want both to detect all contextual clauses and to assign to each of them their actual semantic role. Similarly to the work described in [1], we use a syntactic parser to detect subordinate clauses (signaled by the label SBAR). In addition, we also identify prepositional and adverbial phrases (respectively labeled PP and ADVP). Moreover, each phrase is a child of a bigger tree. We thus attach each clause we are interested in as the context of its parent. Consequently, if one of these contextual clauses is a direct child of the root of the syntactic tree, it means that it is part of the whole context. The role played by contextual elements is better more highlighted in Sect. 5.2.

In addition, a contextual clause may be complex. Thus it could contain many atomic assertions. In practice, we perform separately OIE on each clause: contextual ones and the clause formed by remaining sequence of words after pruning contextual phrases from the input sentence. On the Fig. 1, steps 3 and 4 illustrate this task.

4.3 Better Highlighting of Enumeration Items

Within our corpus, made up of French law texts, it is common to find pieces of text which, if stood by themselves, are deeply ambiguous. In our sample sentence, it is the case of the fragment *"and 1.5 metre"*. A human reader, when looking at this sentence will understand that *1.5 metre* is the value of the width of the entrance doors. This kind of ambiguity is mainly encountered in enumerations.

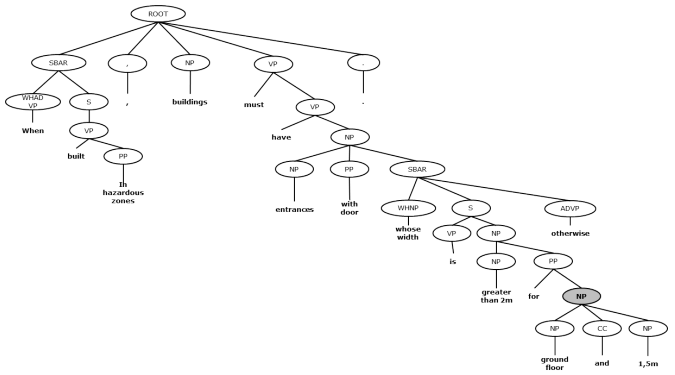

Fig. 3. The problematic role played by a prepositional phrase for the identification of enumeration

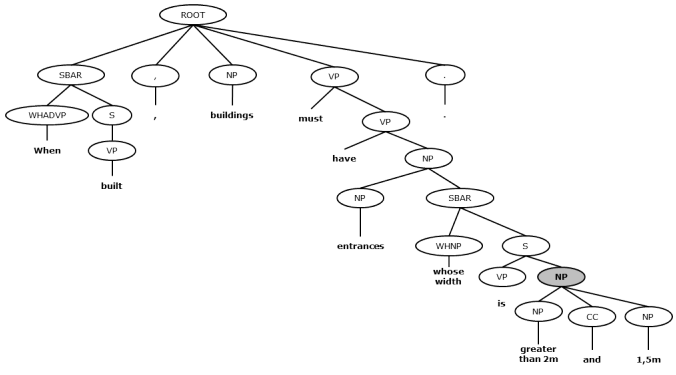

Fig. 4. The removal of prepositional and adverb phrases provides a meaningful enumeration

More pragmatically, it brings researchers to be able to assign the right role to a comma or a coordinating conjunction. This issue is clearly prioritized in the designing of CSD-IE [1]. Bast and Haussmann [1] use the syntactic tree of the input sentence to identify items of a possible enumeration. In that tree, they consider that a node denotes an enumeration if its children are "all of the same type (e.g. all VP), but interleaved by punctuation or conjunctive constructions" [1, page 4].

Surely, this definition may be true in many cases and is a key reason of the performance of CSD-IE. Unfortunately, there are some cases where items of an enumeration do not fit this definition. It occurs when items are separated by sequence of words which are neither punctuation marks nor conjunctions. Such noisy sequence of words act as fact modifiers (contextual phrases): they give more information about other words. Syntactically, they could be prepositional phrases, adverb phrases, adjective phrases, etc. But, in the previous step, we have pruned contextual clauses from the original sentence. This step of contextual clauses identification hence has a broader scope. Indeed, it also has a part in better highlighting of enumeration items.

These problematic roles of contextual clauses in the detection of items of an enumeration are depicted by the Fig. 3 and 4. On the former figure, applying the definition given in [1] will make us identify *"for ground floor"* and *"1.5 metre"* as enumeration items. Thus, the following two triples may be extracted <*width, is, greater than 2 metres for ground floor*> and <*width, is, greater than 2 metres for 1.5 metre*>. Surely, the second triple is incorrect. Now, if we try to perform facts extraction with the tree provided by Fig. 4. we get the following results. <*Width, is, greater than 2 metres*> and <*width, is, 1.5 metre*>. As mentioned in Sect. 4.2 the prepositional phrase *"for ground floor"* will be attached to the first triple as its context.

When we look more carefully to the sub-tree denoted by the node labelled PP and which structures the sentence fragment *"for ground floor and 1.5 metre"*, we notice that this prepositional phrase is incorrect. Thus, pruning it without any preprocessing will still result in incorrect extractions. Consequently, we perform a further checking when extracting contextual phrases and that they contain enumeration items according to the definition of Bast and Haussmann [1]. We verify that words which are coordinated are of the same type. A syntactic tree highlights the relation between chunks (constituents) of a phrase. On the contrary, an analysis of typed dependencies will point out the two words which support the relation. In this case the *conj* dependency links the noun *floor* and the cardinal number *1.5*. Hence, the subsequence *"and 1.5m"* has been removed from the prepositional clause *"for ground floor and 1.5m"*. This heuristic may encounter coarse cases but helps to recover from some parsing errors.

All the processing steps described in this section are highlighted step by step through the Fig. 1.

5 Improving Open Information Extraction Results

Earlier in this work, we have pointed out a set of things which degrades precision of OIE-tools. We have thus decided to perform a set of operations on the sentence to ease the work of OIE-systems. Obviously, none of these tasks use triples provided by an OIE-tool. In the current section, we have now going to exploit facts provided by an OIE-tool. We will first filter the set of facts and finally, organize the remaining triples to get the structure of our sentence.

5.1 Filtering

At this stage, we browse the set of triples resulting from each clause of the sentence. A triple is accepted after a filtering process. As mentioned in our state-of-the-art, we take advantage of the filters (accuracy and coverage) described in [1]. Moreover, we add the following requirements:

- If a triple contains only a fragment of a multi-word expression, it is considered as incorrect.
- When a triple is included in another triple (e.g <*whose width, is, greater than 1.5m*> is included in <*whose width,*

is, greater than 1.5m otherwise>), the smaller fragment is rejected. This rule ensures us to take all the words of the initial sentence and therefore to have a good coverage (see Sect. 2).

Let us note that an OIE process on a chunk of the sentence may have no result. In such case, this emptiness will be filled by the chunk itself in the structuring step (Sect. 5.2). It is the case of the fragment "When built in hazardous zones or considered as such" (steps 5 and 6 on Fig. 1) from which an OIE-system may not get any fact.

5.2 Sentence structuring

At this stage, all the fragments (i.e. the triples issued from the previous step) of the initial sentence have to be put together to get the structure. Let us remember that a triple may come from either a contextual clause or the "main clause(s)". Moreover, we know exactly to which part of the sentence a contextual phrase is attached. When a contextual clause is attached to a verb, it determines all the facts predicated by this verb. Otherwise, if the context is attached to a noun or adjective, we consider that it adds precision to this given noun or adjective. Some SBARs are fully independent (e.g.: "When built in hazardous zones"). We consider such subordinate clauses as the general context. They are the gateways without which the other facts are not meaningful. Let us remember that this work will be used in an automatic conformity checking process as described in [9]. Consequently, if the facts extracted in what we consider as general context are inferred as false, it is not useful to perform further checks. For instance, if our building is not "built in a hazardous zone", the width of the entrance door has no interest. This target goal, be able to automate conformity checking using natural language requirements, leads us to highlight implicit if then rules. Indeed, some sequences of words denote the presence of conditional clauses. In our running example, we see that the adverb *"otherwise"* gives an ***if then else*** view to end of the sentence. Consequently this fragment of the sentence may be rephrased as: ***if the door is located at the ground floor, then the width is greater than 2 metres, else the width is 1.5 metre***.

Relation between facts

One of the key aspects of this task is to highlight how facts in a sentence behave to give its actual meaning. For example, saying that ***sentence = fact$_1$ AND fact2*** is not the same as ***sentence = fact$_1$ XOR fact$_2$***. To perform this task, we decode each coordinating conjunction and find how it links the concerned facts. Indeed, such conjunction coordinates two elements and thus the facts containing them. Further, this coordinating dependency is propagated to all the enumeration items and hence to the fact containing them.

6 Preliminary Evaluation And Discussion

In the current work, we have underlined the similarities between our sentence decomposition goal and deep OIE. Then, focusing on the drop of precision of current OIE-systems on complex legal sentences (more than 63% of errors on our small dataset[6]), we have proposed a set of operations to solve this problem. One of our main goal was to keep the same performance (and doing better if possible) on more simple sentences. Of course not all the sentences of our corpus have such a weird structure. This is why we have taken advantage of existing OIE-system in our pipeline. Early results are promising. Indeed, we have made the following observations:

- Because OIE-systems always separate coordinating terms, the presence of a MWE containing a coordinating conjunction may cause:
  - Incorrect extraction. For instance CSD-IE extracts <whose width, is otherwise, entrances with door greater> and ClausIE <entrances with door width, is, greater otherwise>
  - Semantically (w.r.t to the domain) ambiguous extraction. For example, when a sentence contains a term like *Post and Beam Construction* (which is usually used as a single word in the field of building construction as the name of the country *Trinidad and Tobago*), OIE-tools may extract tuples like <*Post Construction, XXX, XX*> and <*Beam Construction, XXX, XX*>. Surely, such triples would have been more suitable if the term was considered atomic.
- On each sentence, we usually get at-least the same performance on information extraction when using our pipeline than *the OIE-system used at the "OIE-step"*. It makes us confident in the fact that our pre-processing tasks will lead us to better performance than existing OIE-systems. But, in some cases, the compression, which may delete prepositional phrases (e.g "angle of pitch" to "angle") can create inappropriate apposition and thus wrong "is-a" relations.

7 Future Work

In our goal to automatically get the real meaning of a requirement, we must extend the set of hints of relation between facts in a sentence. In the current work, we look for only few coordinating/subordinating conjunctions and conjunctive adverbs (and, or, if, else, then, otherwise, etc.). Decoding more keywords will be relevant. Moreover, we handle sentences separately. This is not in full respect of the manner how legal texts are written. Indeed, the same requirement may be expressed by a set of sentences which have to be semantically related. For instance, the running example could have been expressed like this: ***When built in hazardous zones or considered as such, buildings must have entrances with door whose width is greater than or equal to 2 metres for ground floor. Otherwise, the width is 1.5 metre.*** Hence, we see that the real meaning of the second sentence is fully dependent of the meaning of the first one. In addition, since multi-word expressions play a key role in our work, we have to use an automatic algorithm to detect them within our corpus. Finally, we shall evaluate our algorithm on traditional dataset as done in [6] [5] [1]. More, we need to confirm, on a larger dataset, that on each sentence our algorithm performs at least similarly to the OIE-system taken to do our OIE step. Moreover, we must build a representative dataset from French law texts of regulations and see how our tool performs on it.

In addition, we could envisage solving the problem of shortening of MWE by a semantic resource. Indeed, since the goal is to get the smallest possible synonym of the MWE, we can try to look for a single word equivalent in Linked Open Data or in lexical databases like Wordnet, etc.

8 Conclusion

The goal of this work is to identify any piece of information which constitutes the meaning of legal sentences. It is a prerequisite to the final target of our whole project, which is to perform automatic conformity checking (w.r.t. law) of building engineering products. To fully handle the semantics of our sentences, we have chosen to extract all pieces of information within a sentence and then to rephrase our sentence by semantically relating these chunks of information. Noticing a drop of precision, due to noisy phrases, of current OIE-systems on our legal sentences we have decided to make some pre-processing before an OIE-step. This preprocessing leads us

[6] We remind that this dataset is made up of sentences where CSD-IE get at-least one error! Hence in a completely random selection of sentences, CSD-IE obtains better results.

to handle efficiently multi- word expressions (domain terms and domain independent terms: operators, idiomatic expressions, etc.). Moreover, during this upstream processing of our sentence we have highlighted contextual clauses. Such clauses represent the conditions in which a fact is considered to be true. Finally, the set of facts which compose a sentence are related, using Boolean operators. The computation of these relations is based on the coordinating conjunctions, and thus the coordinated elements within the original sentence. The use of OIE on the road to automatic legal sentence understanding, the handling of multi-word expressions (mainly their shortening), and the highlighting of contextual clauses and logical relations between assertions found in a sentence constitute the original points of this work.

REFERENCES

[1] H. Bast and E. Haussmann. Open information extraction via contextual sentence decomposition. In Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on, pages 154–159. IEEE Computer Society, 2013.

[2] H. Bast and E. Haussmann. More informative open information extraction via simple inference. In Advances in Information Retrieval, volume 8416 of Lecture Notes in Computer Science, pages 585–590. Springer International Publishing, 2014.

[3] E. De Maat and R. Winkels. Categorization of norms. In Legal Knowledge and Information Systems: JURIX 2007: the Twentieth Annual Conference, pages 79–88. IOS Press, 2007.

[4] M.-C. De Marneffe and C. D. Manning. Stanford typed dependencies manual, 2008.

[5] L. Del Corro and R. Gemulla. ClausIE: clause-based open information extraction. In Proceedings of the 22nd international conference on World Wide Web, WWW '13, pages 355–366, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[6] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[7] G. Galichet, La décomposition structurale de la phrase (suite). L'Information Grammaticale, 6(1):32–35, 1980.

[8] G. Galichet. Pour une décomposition structurale de la phrase complexe.

[9] L'Information Grammaticale, 4(1):32–36, 1980.

[10] C. Kacfah Emani. Automatic Detection and Semantic Formalisation of Business Rules. In Extended Semantic Web Conference - PhD Symposium, pages 834–844. Springer, 2014.

[11] J. Lehmann, T. Furche, G. Grasso, A.-C. Ngonga Ngomo, C. Schallhart, A. Sellers, C. Unger, L. Bühmann, D. Gerber, D. Höffner, Kand Liu, and S. Auer. Deqa: Deep web extraction for question answering. In Proceedings of the 11th International Conference on The Semantic Web - Volume Part II, ISWC'12, pages 131–147, Berlin, Heidelberg, 2012. Springer-Verlag.

[12] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open language learning for information extraction. In EMNLP-CoNLL, pages 523–534. Association for Computational Linguistics, 2012.

[13] K. Sayah. Automated norm extraction from legal texts. Master's thesis, Utrecht University, aug 2004.

[14] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. In Proceedings of the 21st International Conference on World Wide Web, WWW '12, pages 639–648, New York, NY, USA, 2012. ACM.