Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 613–622
Marseille, 20-25 June 2022

Integrating a Phrase Structure Corpus Grammar and a Lexical-Semantic Network: the HOLINET Knowledge Graph

Jean-Philippe Prost
Laboratoire Parole et Langage (LPL), Aix-Marseille Université, France
Jean-Philippe.Prost@univ-amu.fr

Abstract

In this paper we address the question of how to integrate grammar and lexical-semantic knowledge within a single and homogeneous knowledge graph. We introduce a graph modelling of grammar knowledge which enables its merging with a lexical-semantic network. Such an integrated representation is expected, for instance, to provide new material for language-related graph embeddings in order to model interactions between Syntax and Semantics. Our base model relies on a phrase structure grammar. The phrase structure is accounted for by both a Proof-Theoretical representation, through a Context-Free Grammar, and a Model-Theoretical one, through a constraint-based grammar. The constraint types colour the grammar layer with syntactic relationships such as Immediate Dominance, Linear Precedence, and more. We detail a creation process which infers the grammar layer from a corpus annotated in constituency and integrates it with a lexical-semantic network through a shared POS tagset. We implement the process, and experiment with the French Treebank and the JeuxDeMots lexical-semantic network. The outcome is the HOLINET knowledge graph.

Keywords: Knowledge Graph, Lexical-Semantic Network, Grammar, Phrase Structure, Constituency, JeuxDeMots

1. Introduction

While the pipeline software architecture, which steps from one linguistic dimension to the next, has been typical for decades for most Natural Language Processing (NLP) applications, it often prevents many potential interactions across dimensions from actually occurring. A variety of sentence-level ambiguities, for instance, require the full sentence to be parsed morphologically, then syntactically, then semantically, prior to being disambiguated through a pipeline. We hypothesize that a holistic modelling of the linguistic knowledge, which also approaches language as a whole rather than solely as a sum of its parts on various dimensions, would ease its processing and improve its performances in many respects.

Knowledge graphs provide a convenient means for heterogeneous knowledge to interact rather seamlessly. They open perspectives with regard to holistic approaches to the modelling of knowledge, and linguistic knowledge in particular.

Yet, the modelling of knowledge, in ways that make it suitable for Knowledge Graphs, is not always trivial and often constitutes a challenge as such.

Concerning the representation of natural language syntax, the landscape of knowledge graphs has few resources, if any. Syntactic knowledge remains nearly exclusively represented through treebanks and other kinds of annotated corpora, and to a lesser extent through knowledge-based grammars, such as the HPSG grammars from the DELPH-IN consortium[1], or meta-grammars such as FRMG (de La Clergerie, 2005).

Ultimately, our aim is to design and build a knowledge graph which overcomes representational discrepancies among knowledge areas that are relevant to natural language processing and make natural language processable in a holistic manner. Assuming such a holistic knowledge graph, we may wonder whether the presentation of grammar knowledge as a knowledge graph would benefit applications down the road. For instance, would graph embeddings benefit from including syntactic relationships? Would knowledge graph completion enable the inference of syntactic relationships which were not observed in corpora? Etc.

In this paper we focus on the graph modelling of phrase structure grammar knowledge in a way that makes it compatible with a lexical-semantic network. We introduce a base model, detail its acquisition based on corpus grammars, and describe the creation process from the grammar acquisition to the merging of the lexical-semantic and grammar layers into a single structure. We implement the process and experiment with the French Treebank (FTB) and the JeuxDeMots (JDM) lexical-semantic network. The outcome is the HOLINET Knowledge Graph which we introduce here.

Section 2 reviews the literature. Section 3 describes the design of the graph model for the grammar layer, bearing in mind the requirements for compatibility with a lexical-semantic network. Section 4 details the creation process and its implementation with the FTB and JDM. Section 5 gives elements of evaluation. Section 6 discusses perspectives, while Section 7 concludes.

2. Literature Review

In their survey Ji et al. (2021) propose to define a knowledge graph as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$,$\mathcal{R}$ and $\mathcal{F}$ are sets of entities, relations and facts, respectively.

[1] `http://www.delph-in.net`

Language-related data and Knowledge Bases The knowledge bases and resources for human language are rarely holistic. They are essentially found in the different linguistic dimensions. The Linguistic Linked Open Data (LLOD) initiative[2] references resources which are linked with each other. It aims to address problems such as representation formats, federation of multiple data sources, or interoperability. The alignment of knowledge among resources is not the prime concern of the initiative, and often remains a challenge to be addressed. The Multiple Knowledge DB (Faralli et al., 2020), for instance, integrates 5 of these resources: ConceptNet (Speer et al., 2017), DBpedia, WebIsAGraph (Faralli et al., 2019), WordNet and the Wikipedia[3] category hierarchy.

Although many linguistic dimensions are still absent from the picture, such an integration goes in the direction of a holistic representation and processing of language knowledge.

Around syntax LLOD references annotated corpora for syntax, but as far as we know no knowledge base or graph is referenced which would account for grammar knowledge of human language.

SAR-Graphs introduce a kind of resource inclusive of syntactic data with links to different lexical-semantic resources such as WordNet, BabelNet, and YAGO. They are described as *graphs of Semantically-Associated Relations* (Krause et al., 2015a). They address the question of the integration of different aspects of linguistic knowledge within a homogeneous structure referred to as a *language network* (Uszkoreit and Xu, 2013; Krause et al., 2015b). They integrate lexical semantics with semantic relations about facts and events extracted from KBs such as Freebase (Bollacker et al., 2008). However, the grammar knowledge as such is contained in DARE (Xu et al., 2007), a relation extraction system, through the use of the principle-based MINIPAR dependency parser (Lin, 1994).

The gathering of grammar knowledge is a tedious and labour-intensive task, which probably explains why so few resources exist, or are made public. We have already mentioned the DELPH-IN project, which is dedicated to the development of NLP resources and applications, all based on Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994) for syntax, and Minimal Recursion Semantics (MRS) (Copestake et al., 2005) for semantics. A limited number of grammars are made available for a variety of languages (Copestake and Flickinger, 2000). The French Meta-Grammar (FRMG) is another extensive source of computational grammar knowledge. Although not primarily designed as knowledge graphs, resources such as DELPH-IN and FRMG remain very valuable for the structured grammar knowledge they represent.

Complex Syntactic Networks The past decade has seen the emergence of the use of complex networks (Newman, 2010) for studying the syntax of human languages, through Complex Syntactic Networks. However, they have been used exclusively for representing dependency relationships among words, or word forms (Čech et al., 2016): the words (lemmas) are nodes, and the dependency relations are edges between nodes. The term *dependency* must be taken in the sense of the Dependency Grammar formalism (Tesnière, 1959; Hudson, 2006). Although nothing prevents the use of other grammar formalisms (e.g. phrase structure grammars) for syntactic network analysis, the literature shows no evidence of it (Čech et al., 2016).

Lexical-semantic networks WordNet is a lexical-semantic network of 150,000+ words, organised in 170,000+ synsets. Synsets can somehow be seen as concepts. EuroWordnet (Vossen, 1998), a multilingual version of WordNet, and WOLF (Sagot and Fier, 2008), a French version of WordNet, were built automatically through the crossing of WordNet and other lexical resources along with some manual checking.Navigli and Ponzetto (2010) constructed automatically BabelNet, a large multilingual lexical network, from term co-occurrences in the Wikipedia encyclopedia. HowNet (Dong and Dong, 2006) is a hand-crafted lexical network based on concepts linking both English and Chinese. The Réseau Lexical du Français (RLF, *French Lexical Network* (Polguère, 2014) is a resource based on the notion of *lexical function* as defined by Igor Mel'čuk. The resource concerns about 10000 terms and is mainly manually populated with data. FrameNet (Baker et al., 1998) is a lexical-semantic network dedicated to the Frame Semantics theoretical framework (Fillmore, 2008). It links lexical units (i.e., the meaning of terms), semantic frames, and illustrative sentences annotated with predicate-argument structures. The frames are also related to each other through typed semantic relationships.

JDM is a crowdsourced lexical-semantic network for French (Lafourcade, 2007). It is associated with a live ecosystem of semi-automatic processes for validation, evaluation, and knowledge completion. At time of writing, the network comprises[4] 16.5+ million nodes including 5.2+ million terms, 400+ million relations and 150+ relation types. The nodes are terms, concepts and symbolic information. The relations are lexical, morphological, pragmatic, logical, ontological, . . .

3. Graph modelling of a phrase structure grammar

The problem we are concerned with here is to design a graph structure which (i) can represent a phrase structure grammar, and (ii) can be connected to a lexical network. The grammar layer must share at least the POS

[2] `http://linguistic-lod.org/`

[3] `https://www.wikipedia.org/`

[4] As of January 2022, according to the values presented at `http://www.jeuxdemots.org/jdm-about.php`.

nodes and the POS relationships (to relate the terms to their POS nodes) with the lexical layer.

3.1. What kind of grammar knowledge can be represented as a graph structure?

By "grammar knowledge" we do not refer to a collection of syntactic trees or graphs, which would each be specific to an utterance, even though such a reference would be a valid one. Foremost we refer to some form of knowledge that abstracts away from specific utterances, and provides some degree of generalisation.

In this paper, our intention is to model grammar knowledge in a way that makes it compatible with a knowledge graph. This means essentially that we can represent that grammar knowledge as a set of triples, that is, as a set of nodes related to each other by relations. We focus on phrase structure grammars. In other words, we need a way to represent descriptions of phrase structures as a set of objects (nodes), which are related to each other by syntactic relationships. While the POS categories occur quite obviously as good candidates for the nodes, the question of what the syntactic relationships should be is not as obvious.

We propose a model which combines the knowledge from a generative grammar and the knowledge from a constraint-based grammar. The generative grammar provides us with the means to specify tree structures and the objects of a tree domain. The constraint-based grammar provides us with the means to rely on various (constraint) types to further specify relationships among syntactic objects in the tree structure.

3.2. A preliminary model: the Alpha model

Let us start the description of the model with an illustration, and let us consider the example phrase structure in Fig. 1 as annotated in the FTB[5]. The annotated tree structure can be modelled with the three rewrite rules from Fig. 2 (the feature structures have only been stripped out from this example for simplicity's sake). We can model these rules as illustrated in Fig. 3.

In what follows, the node and edge types are given in `true type font`. For reference sake, we call this preliminary model the *Alpha Model*.

In this Alpha model, every rewrite rule is reified as a node, typed `n_g_cfRule`[6] . The left-hand side of each rule is itself reified as a POS node (of type `n_pos`), and every rule is connected to its left-hand side POS node with the `r_g_rewrites` relation. Meanwhile, on the right-hand side (RHS) of each rule, every constituent POS is connected to its rule with the `r_g_constitutes` relation. In order to allow redundancy of POS, like here the AP, every constituent on the RHS is related to its POS node with an `r_g_instantiates` relation.

3.3. The Beta (and final) model

At this stage, we can observe that the Alpha model captures the relationship of Immediate Dominance among the constituents in the rewrite rules, but it fails to represent all the implicit knowledge covered by the rules. For instance, we observe that the following relationships, which are implicit in rule 1, are not accounted for in the Alpha Model (Fig. 3):

- word order (Linear Precedence)
  - the DET precedes the NC
  - the DET precedes the AP ♯1
  - the DET precedes the AP ♯2
- oriented co-occurrence
  - a DET requires an NC

These extra relationships can be modelled as such with *properties* of a Property Grammar (Blache, 2001). To make it short, the PG properties are relational constraints over POS categories. Adding these relationships to the Alpha Model from Fig. 3 gives the Beta Model illustrated in Fig. 4. Next, we detail it further.

3.4. What semantics should define the grammar relationships?

Property Grammar (PG) is a formal framework for specifying constraint-based grammars. It provides us with well-defined semantics for a set of syntactic relationship types called *Properties*. We borrow the following properties from PG for our base model: Constituency (`r_g_constitutes`), Linear Precedence (`r_g_precedes`), and Requirement (`r_g_requires`). Their semantics are re-interpreted for our purpose, following the definitions in (Duchier et al., 2009). Table 1 gives the semantics in use in HOLINET for each type.

For the purpose of our work, a PG property can simply be seen as a triple (Node, relationship, Node).

3.5. What satisfaction value should be assigned to the constraint-based relationships?

A constraint (i.e., a property) from a constraint-based grammar can usually be seen as a statement which may be true (the constraint is satisfied) or false (the constraint is violated). Such a satisfaction value can be assigned to a given property, provided that we know the CFG rule that serves as an evaluation context. So, for instance, in the example above we used Rule 1 as the evaluation context to state that "*the DET precedes the NC*", that is, that property (DET, `r_g_precedes`, NC) is satisfied. Sometimes we may want to be specific and denote the satisfaction value with the type `r_g_precedes+`.

In the end, to build a grammar layer we first need:

- a CFG comprised of a set of rewrite rules,

[5] The functional tags have been removed.

[6] `n_g_cfRule` and `n_g_compound` are synonym.

```
(NP
 (DET##lem=un|cpos=D|g=f|n=s|s=ind## une)
 (NC##lem=bataille|cpos=N|g=f|n=s|s=c## bataille)
 (AP
  (ADJ##lem=politique|cpos=A|n=s|s=qual## politique))
 (AP
  (ADV##lem=extrêmement|cpos=ADV|_## extrêmement)
  (ADJ##lem=ardu|cpos=A|g=f|n=s|s=qual## ardue)))
```

Figure 1: syntactic tree structure for the phrase *une bataille politique extrêmement ardue* (*an extremely arduous political struggle*), from the FTB.

Constituency	(A, `r_g_constitutes`, B)	the `n_g_instance` node A occurs in the rhs of the `n_g_cfRule` node B
Precedence	(A, `r_g_precedes`, B)	the `n_g_instance` node A precedes the `n_g_instance` node B in read direction (implicitely in the context of the `n_g_cfRule` node they are both related to with `r_g_constitutes`)
Requirement	(A, `r_g_requires`, B)	the `n_g_instance` node A requires the `n_g_instance` node B to co-occur (implicitely in the context of the `n_g_cfRule` node they are both related to with `r_g_constitutes`)

Table 1: semantics of the PG property types in use in the HOLINET grammar layer

(1) NP ⟶DET NC AP AP

(2) AP ⟶ADJ

(3) AP ⟶ADV ADJ

Figure 2: CFG for phrase structure in Fig. 1

- a set of consistent properties to complete the CFG.

Given these two resources, we can then move on to assess, for each rule in the CFG, the satisfaction value of each property that is relevant in this context. We call this process the *characterisation* of every rewrite rule, following the PG terminology. The outcome is a set of assessed properties associated with each rule, as illustrated in Fig. 2.

As far as our modelling is concerned, we can see each assessed statement, that is, each evaluated PG property, as a relationship between (instances of) POS categories. Therefore, these assessed properties make as many relationships on our grammar layer.

VanRullen (2005) proposed to model the PG theoretical framework as a graph structure. Representing assessed PG properties is what makes our graph modelling of PG significantly different from VanRullen's one, which represents the properties irrespective of their assessment. Said differently, and to mimic the terminology from Logic or Constraint Programming, we could say that VanRullen's model represents *in intension* the properties that comprise a PG grammar, while we represent them *in extension*.

At this stage, we limit the model to representing only the satisfied properties. Further works will investigate the possible extension of the model with violated properties in order to cover degrees of grammaticality.

In the next section we will detail how to automate the entire creation process of a grammar layer on the basis of a constituency treebank, and how to connect it to JDM.

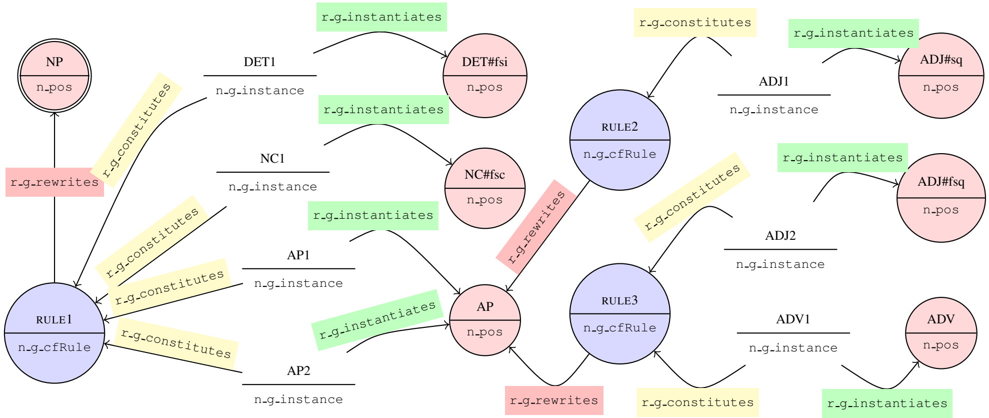
4. The creation process

We assume the existence of the following resources:

- a constituency treebank
- a (coloured) lexical-semantic network of typed nodes and typed relationships, where
  - the lexical entries are terms, represented as nodes of type `n_term`
  - the Part-of-Speech (POS) labels are nodes of type `n_pos`
  - each term is related to one or several POS nodes through a relationship of type `r_pos`
- if necessary, a conversion table between the treebank POS labels and the network POS labels.

The creation process, in short, is made up of the following steps:

1. *extract* the corpus CFG from the treebank
2. *derive* the PG from the CFG
3. *characterise* the CFG according to the PG
4. *convert* the corpus tagset as required
5. *create* the sets of nodes and edges for the grammar layer
6. *merge* the grammar and the lexical layers



with the following node descriptions:

$$\textit{DET\#fsi}\begin{bmatrix} \text{TYPE} & \texttt{n\_pos} \\ \text{CONTENT} & \begin{bmatrix} \text{CAT} & \textit{DET} \\ \text{CPOS} & \textit{D} \\ \text{GEND} & \textit{f} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{ind} \end{bmatrix} \end{bmatrix} \quad \textit{N\#fsc}\begin{bmatrix} \text{TYPE} & \texttt{n\_pos} \\ \text{CONTENT} & \begin{bmatrix} \text{CAT} & \textit{NC} \\ \text{CPOS} & \textit{N} \\ \text{GEND} & \textit{f} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{c} \end{bmatrix} \end{bmatrix}$$

$$\textit{ADJ\#sq}\begin{bmatrix} \text{TYPE} & \texttt{n\_pos} \\ \text{CONTENT} & \begin{bmatrix} \text{CAT} & \textit{ADJ} \\ \text{CPOS} & \textit{A} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{qual} \end{bmatrix} \end{bmatrix} \quad \textit{ADJ\#fsq}\begin{bmatrix} \text{TYPE} & \texttt{n\_pos} \\ \text{CONTENT} & \begin{bmatrix} \text{CAT} & \textit{ADJ} \\ \text{CPOS} & \textit{A} \\ \text{GEND} & \textit{f} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{qual} \end{bmatrix} \end{bmatrix}$$

$$\text{RULE1}\begin{bmatrix} \text{TYPE} & \texttt{n\_g\_cfRule} \\ \text{RULE} & \begin{bmatrix} \text{LHS} & \begin{bmatrix} \text{CAT} & \textit{NP} \end{bmatrix} \\ \text{RHS} & \left\langle \begin{bmatrix} \text{CAT} & \textit{DET} \\ \text{CPOS} & \textit{D} \\ \text{GEND} & \textit{f} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{ind} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \textit{NC} \\ \text{CPOS} & \textit{N} \\ \text{GEND} & \textit{f} \\ \text{NUM} & \textit{s} \\ \text{SUBCAT} & \textit{c} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \textit{AP} \end{bmatrix}, \begin{bmatrix} \text{CAT} & \textit{AP} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \text{(idem for the other rules)}$$

Figure 3: Alpha model: the syntactic relationships from the CFG in Fig. 2

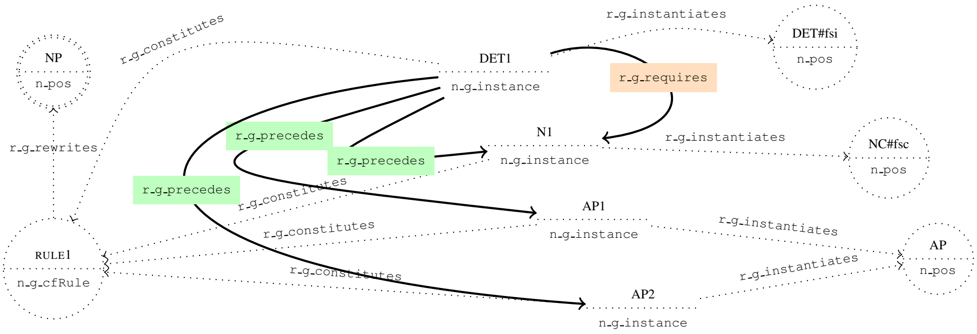


Figure 4: Beta Model: The PG relationships added to the Alpha model for RULE1

4.1. Implementation

In the following we work with the French Treebank (FTB) (Abeillé et al., 2003)[7], and the JeuxDeMots (JDM) (Lafourcade, 2021) lexical-semantic network[8], but the process is fairly basic and should easily apply to other resources with little changes. Given the respective nature and size of the resources involved (a fixed

[7] version 1.0 2016, annotated with the Penn Treebank tagset. The functional tags have been removed.

[8] As built from the dump dated 01 November 2021

treebank and an ever-changing network), we choose to adapt the treebank tagset to the network one.

The distribution of all the software involved in this section is detailed in Appendix 7.

Step 1. Extract CFG from FTB is straightforward.

Step 2. Derive PG from CFG Given an observed corpus CFG, it is possible to automate the derivation of the relationships we are interested in (Prost et al., 2016). In order to do so, each type of relationship, according to its semantics, corresponds to a derivation rule. We recall those derivation rules below.

Let $lhs(x)$ be the function that maps the non-terminal category x to the set of all the rewrite rules in the CFG that take x as its left-hand side.

Constituency The POS category B is in a Constituency relationship with the non-terminal category A iff at least one rewrite rule in $lhs(A)$ can be found where B occurs in the right-hand side.

Linear Precedence The two POS categories B and C are in a Linear Precedence relationship in the context of the non-terminal A (denoted $A : B \prec C$ for short), iff at least one rewrite rule in $lhs(A)$ can be found where $B \prec C$, and no rule can be found where $C \prec B$.

Requirement The two POS categories B and C are in a Requirement relationship in the context of the non-terminal category A iff no rewrite rule can be found in $lhs(A)$ where B occurs without C.

Step 3. Characterise the CFG The characterisation process can somehow be seen as checking the satisfaction of a constraint system against an assignment, where the constraint system is the grammar made up of a set of constraints/properties/relationships, and the assignment is the constituents of the tree structure represented here by one or more rewrite rules. In our case, since the relationships that make the PG grammar were observed in corpus, and since they were derived from the very tree structures described in the CFG we want to characterise, they are all deemed satisfied. Therefore, the characterisation process comes down to simply listing, for each rewrite rule, which relationships from the PG are relevant in this context. By "relevant" we mean that the relationship applies in this context and is not trivially satisfied, as would be, for instance, the relationship (NP,`r_g_precedes`,NC) in the context of the rewrite rule (AP $\longrightarrow$ADJ), since the NP and AP categories mismatch.

Table 2 reports a sample of the characterised corpus grammar that results from this step. The first two columns are for the rewrite rules, one per line. The right-hand side of rule is a comma-separated list of labels. Each of the next three columns is for a relationship type: Constituency is a comma-separated list of constituent POS; Linear Precedence and Requirement are comma-separated lists of semi-colon-separated pairs. The full grammar is available as part of the current release (see Appendix 7 for details).

Step 4. Convert the FTB tagset to JDM The mismatches encountered are mainly caused by the following reasons:

- A single FTB tag might correspond to a combination of several distinct JDM tags. For instance, the FTB tag `P+PRO##cpos=P+PRO|g=f|n=p|p=3|s=rel##` should, in theory, correspond in JDM to `Pre+Pro:Fem+PL+Rel`. The full tag does not exist, that is, there exists no POS node labelled with this tag. Instead, there exist four distinct nodes: `Pre:`, `Pro:Fem+PL`, `Pro:Rel`, and `Pre:`. Hence, all the required information is present in JDM, but spread across several nodes, while a single one would be required by rewrite rules in the FTB.
- Since JDM adopted as a general strategy not to overload the network with data that has never been encountered, some features are absent from JDM. E.g, the definite/indefinite subcategorisation values for determiners.
- Some base categories in the FTB do not exist in JDM and had to be created. E.g., `VPinf`, `VPpart`.

Table 3 reports a small sample of the conversion table. The full table is available as part of the current release (see Appendix 7 for details)[9] .

Step 5. Create the sets of nodes and edges for the grammar layer Algorithm 1 describes the procedure for creating a grammar layer. The algorithm must be adapted to the base lexical-semantic network in order to enforce consistency between the two layers. For instance, the pre-existence of POS nodes in the network may require that the existing nodes be listed prior to running the algorithm.

Step 6. Merge JDM and the grammar layer In short, the merging involves the creation of the JDM graph from a released dump, and the consistent merging of the grammar layer. By 'consistent' we mean, for instance, making sure that no duplicate nodes and triples are created. Detailing this step further goes beyond the scope of this paper, but the software is publicly available.

Our implementation of the entire process is available as a combination of open source projects, which are all made available in various public repositories (see Appendix 7).

Fig. 5 illustrates the outcome of our implentation of the process with a sample of the grammar layer of HOLINET and part of the utterance *<u>une bataille</u> politique extrêmement ardue*.

[9] Since the JDM crew was notified about these issues, the live version of the network might partially fix them by the time this paper is published.

LHS	RHS	CONST+	LIN+	REQ+
VP:Inf	VN:,Ver:PPas,PP:	VN:,PP:,Ver:PPas	(Ver:PPas; PP:),(VN:; Ver:PPas)	(Ver:PPas; VN:),(Ver:PPas; PP:)
NP:	Det:,Nom:,Adj:	Adj:,Nom:,Det:	(Det:; Nom:),(Det:; Adj:)	
NP:	Det:Fem+SG+Def, Nom:Fem+SG+Com	Det:Fem+SG+Def, Nom:Fem+SG+Com	(Det:Fem+SG+Def;Nom:Fem+SG+Com)	(Nom:Fem+SG+Com;Det:Fem+SG+Def)

Table 2: sample characterised corpus grammar derived from the FTB

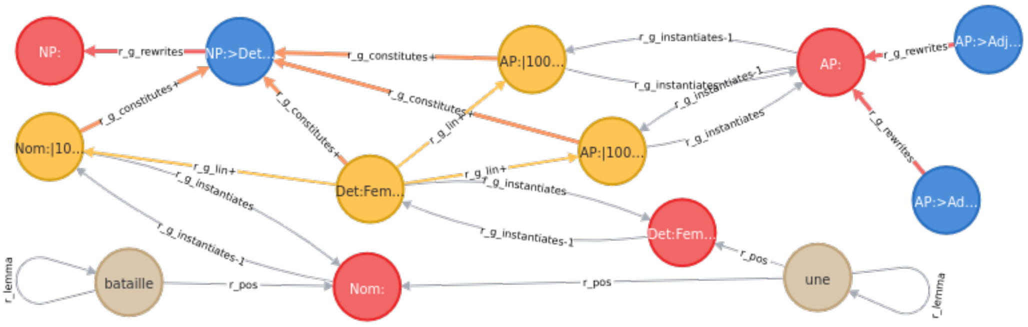


Figure 5: sample HOLINET graph for part of the utterance *une bataille politique extrêmement ardue* (*an* extremely *arduous political struggle*)

FTB	JDM
ADJ##cpos=A|g=f|n=p|s=ord|pred=y##	Adj:Fem+PL+Ord
DET##cpos=D|g=f|n=s|s=def|pred=y##	Det:Fem+SG+Def
P+D##cpos=P+D|s=def|pred=y##	Pre+Det:
VPP+	Ver:PPas

Table 3: sample of the lookup table for converting the FTB tagset to the JDM one.

Algorithm 1 Create the HOLINET Nodes and Edges

```
for all CF rule do
    CREATE an n_pos node for this rule's root if it does not exist yet
    CREATE n_pos nodes for the rule's daughter labels (i.e. RHS) if they do not exist yet
    CREATE the n_g_cfRule node for this rule
    for all categories in this rule's RHS do
        CREATE an n_g_instance node with this category
        CREATE the r_g_instantiates edge, from the current instance node to the r_pos node of its category
        CREATE an r_g_constitutes edge from the current instance node to the current compound
    end for
    CREATE an r_g_rewrites edge from the compound node to the n_pos root node
    CREATE the edges for the PG properties for this rule
end for
```

5. Evaluation

We evaluate the adequacy of our implementation (of our graph model) with respect to the main goal we set initially: integrate a phrase structure grammar with a lexical-semantic network.

Integration we are primarily concerned with evaluating the integration of the grammar layer with the lexical-semantic layer. As emphasized in Section 4.1, many of the POS tags observed in the FTB go missing in JDM. Of course the corresponding missing nodes were created. But as of today, those newly created nodes, although connected to the grammatical layer through the rewrite rules they are involved in, remain disconnected from the lexical-semantic layer. Fixing the problem is quite straightforward at the scale of the FTB. But scaling up the fix to the rest of HOLINET is more challenging, since it would involve the appropriate POS tagging of all the terms in JDM. In order to estimate the extent of the problem, for each POS tag involved in a rewrite rule we check wether the required n_pos node was pre-existing in JDM. Table 4 reports the number of occurrences of each POS in the grammar, whether pre-existing or not. The proportion of connected POS at the scale of the FTB probably gives a rough estimate of the proportion of those at full scale. The 'Null' values are conversion errors.

	Connected	Disconnected	Null	**Total**
Num. nodes	22,742	163,210	9,408	**195,360**
%	11.6%	83.5%	4.8%	**100**

Table 4: measures of integration

Density As suggested in Chen et al. (2019) among a list of possible criteria of evalutation of a knowledge graph, we measure the density of connexions through the number of occurrences of each POS node in rewrite rules. As expected, the number of instance edges by POS node shows a power law distribtuion, as illustrated in Fig. 6.

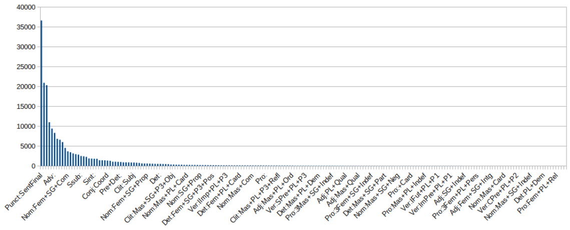


Figure 6: power law distribution of the `r_g_instantiates` edges by POS node

The grammar layer is comprised of 1,048,575 edges, and 180,479 rewrite rules.

Redundancy Still suggested in Chen et al. (2019), we measure the node redundancy. We found two POS nodes that were created on the grammar layer, whereas they already existed in JDM. This can easily be fixed. Note that depending on the underlying data management system, future such redundancy can easily be avoided with integrity constraints being set on the POS nodes.

6. Perspectives

HOLINET opens the door to a large variety of future works, along several dimensions.

6.1. Extending the graph model

Various extensions of the base model can be considered in order to foster the holistic nature of HOLINET.

Integration of dependency grammar The integration of dependency grammar knowledge should be eased by various works around theoretical frameworks for its constraint-based modelling (Maruyama, 1990; Debusmann et al., 2004). The constraint-based nature of existing frameworks makes quite likely the possibility of integrating a new 'dependency' relationship type with a well-defined semantics.

Integration of syntactic constructions A quite natural extension of a constraint-based phrase structure grammar is to evolve towards a construction Grammar (CxG) (Goldberg, 1995). Recent works (Müller, 2021) show, for instance, that Head-Driven Phrase Structure Grammar (HPSG) can be seen as a CxG. In order to do so, the question of the graph modelling of a form-meaning pair will have to be addressed.

Weighting scheme The question of the assignment of weight to both nodes and edges in the grammar layer is a critical question for a variety of applications. Certain graph algorithms, for instance, such as the classical shortest path, rely on the edges to be associated with weights to perform well. Though we have left aside the question so far, annotated corpora such as the FTB provide with the basic statistical material to turn the underlying CFG into a Probabilistic CFG, hence assign initial weights to the grammar edges.

6.2. Implementation of the grammar layer

Our implementation leaves room for improvement. Different problems that have been stressed in Section 5 should be quite easily fixed (e.g. the 'null' values, the POS-tagging of all the terms in JDM). More relationship types can be extracted from corpus, such as Uniqueness, or the property for category to head a phrase. Other grammar resources, such as FRMG, could also be integrated. The question of maintaining consistency among grammar resources will likely be triggered.

6.3. Graph embeddings

The integration of syntactic knowledge along with lexical and semantic knowledge within the same embeddings is modern ground for investigation (Limisiewicz and Mareček, 2020; Al-Ghezi and Kurimo, 2020). With the introduction in HOLINET of syntactic relationships within the same knowledge graph as lexical and semantic ones, one may wonder whether these relationships could be represented in graph embeddings, and, subsequently, whether such embeddings would provide the NLP applications down the road with a better modelling of the interactions between syntax and semantics.

7. Conclusion

In this paper we addressed the question of the integration of grammar knowledge and lexical-semantic knowledge within a homogeneous knowledge graph. We introduced a graph model of a phrase structure grammar and showed how to integrate it with a lexical-semantic network through a shared POS tagset. This model relies on several types of syntactic relationships which are inspired from Property Grammar, a constraint-based theoretical framework for Model-Theoretic Syntax. We proposed a creation process which derives these relationships from a CFG, and integrates the grammar with a lexical-semantic network. We implemented the creation process with a corpus grammar extracted from the French Treebank and the lexical-semantic network JeuxDeMots. The outcome is the HOLINET knowledge graph (Prost, 2022).

The evaluation we performed of the quality of the data that we produced shows a rather weak integration of the grammar with JDM, with only a small 11% of the POS nodes that are connected to JDM. While it obviously leaves room for improvement implementation-wise, the integration model as such is not to be blamed. Future works should most notably investigate the computation of syntactic-lexical-semantic graph embeddings and measure their impact on NLP applications.

Appendix: data and software release notes

JeuxDeMots is licensed under a Creative Commons 0 1.0 Universal (CC0 1.0) Public Domain Dedication License.

HOLINET version 1.0 is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License. The distribution contains:

- a pre-processed version of JeuxDeMots, version 11012021 (CSV)
- the grammar layer (nodes and edges) (CSV)
- the full lookup table for converting the FTB tagset to the JDM one (CSV)
- a human-readable version of the characterised grammar (CSV)

software The implementation described in Section 4.1 relies on the following pieces of software, all made available in various public repositories:

- rdb4jdm: a fork of the rdb4jdm project by Kevin Cousot, slightly modified for pre-processing the JDM dump. At sourceforge.net, version tag "LREC_2022-final" (`https://sourceforge.net/p/rdb4jdm/code/ci/LREC_2022-final/tree/`)
- treebankGrammatizer: for the steps 1, 2 and 3: at sourceforge.net, version tag "LREC_2022-submission" (`https://sourceforge.net/p/treebankgrammatizer/code/ci/LREC_2022-submission/tree/`)
- holinet: for the steps 4, 5 and 6, and the overall management. At sourceforge.net, version tag "LREC_2022-submission" (`https://sourceforge.net/p/holinet/code/ci/LREC_2022-submission/tree/`)

8. Bibliographical References

Abeillé, A., Clément, L., and Toussenel, F., (2003). *Building a Treebank for French*, chapter Ch. 10, page 165. Springer Netherlands, Treebanks, Kluwer, Dordrecht.

Al-Ghezi, R. and Kurimo, M. (2020). Graph-based syntactic word embeddings. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 72–78.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 1, pages 86–90.

Blache, P. (2001). *Les Grammaires de Propriétés: des contraintes pour le traitement automatique des langues naturelles*. Hermès Sciences.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Čech, R., Mačutek, J., and Liu, H., (2016). *Towards a Theoretical Framework for Analyzing Complex Linguistic Networks*, chapter Syntactic Complex Networks and their Applications, pages 167–186. Understanding Complex Systems. Springer, Berlin, Heidelberg.

Chen, H., Cao, G., Chen, J., and Ding, J. (2019). A practical framework for evaluating the quality of knowledge graph. In *China conference on knowledge graph and semantic computing*, pages 111–122. Springer.

Copestake, A. and Flickinger, D. (2000). An open source grammar development environment and broad-coverage English grammar using HPSG.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332.

de La Clergerie, É. (2005). From Metagrammars to Factorized TAG/TIG Parsers. In *Proceedings of IWPT'05 (poster)*, Vancouver, Canada.

Debusmann, R., Duchier, D., and Niehren, J. (2004). The XDG grammar development kit. In *International Conference on Multiparadigm Programming in Mozart/OZ*, pages 188–199. Springer.

Dong, Z. and Dong, Q. (2006). *HowNet and the Computation of Meaning*. WorldScientific, London.

Duchier, D., Prost, J.-P., and Dao, T.-B.-H. (2009). A Model-Theoretic Framework for Grammaticality Judgements. In *Proceedings of Formal Grammar (FG'09)*, volume 5591 of *LNCS*. FOLLI, Springer.

Faralli, S., Finocchi, I., Ponzetto, S. P., and Velardi, P. (2019). Webisagraph: A very large hypernymy graph from a web corpus. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*, pages 13–15, Bari, Italy.

Faralli, S., Velardi, P., and Yusifli, F. (2020). Multiple knowledge graphdb (mkgdb). In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2325–2331.

Fillmore, C. J., (2008). *Cognitive Linguistics: Basic Readings*, chapter Frame semantics, pages 373–400. De Gruyter Mouton.

Goldberg, A. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

Hudson, R. (2006). *Language Networks: The New Word Grammar*. Oxford University Press.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2021). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans-*

actions on Neural Networks and Learning Systems, pages 1–21.

Krause, S., Hennig, L., Gabryszak, A., Xu, F., and Uszkoreit, H. (2015a). Sar-graphs: A Linked Linguistic Knowledge Resource Connecting Facts with Language. *ACL-IJCNLP 2015*.

Krause, S., Hennig, L., Gabryszak, A., Xu, F., and Uszkoreit, H. (2015b). Sar-graphs: A linked linguistic knowledge resource connecting facts with language. *ACL-IJCNLP 2015*, page 30.

Lafourcade, M. (2007). Making people play for Lexical Acquisition. In *Proc. SNLP 2007*, pages 13–15, Pattaya Thailande, December. 8 p. 7th Symposium on Natural Language Processing.

Limisiewicz, T. and Mareček, D. (2020). Syntax Representation in Word Embeddings and Neural Networks–A Survey. arXiv preprint arXiv:2010.01063.

Lin, D. (1994). Principar: an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 482–488. Association for Computational Linguistics.

Maruyama, H. (1990). Structural Disambiguation with Constraint Propagation. In *Proceedings 28th Annual Meeting of the ACL*, pages 31–38, Pittburgh, PA.

Müller, S., (2021). *Head-Driven Phrase Structure Grammar: The handbook*, chapter HPSG and Construction Grammar, pages 1497–1553. Number 9 in Empirically Oriented Theoretical Morphology and Syntax. Language Science Press, Berlin.

Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden.

Newman, M. (2010). *Networks: An Introduction*. OUP Oxford.

Polguère, A. (2014). From Writing Dictionaries to Weaving Lexical Networks. *International Journal of Lexicography*, 27(4):396–418.

Pollard, C. and Sag, I. (1994). *Head-driven Phrase Structure Grammar*. University of Chicago Press.

Prost, J.-P., Coletta, R., and Lecoutre, C. (2016). Compilation de grammaire de propriétés pour l'analyse syntaxique par optimisation de contraintes. In *Actes de TALN 2016, 23ème conférence sur le Traitement Automatique des Langues Naturelles*, pages 396–402, Paris, France, Juillet. Association pour le Traitement Automatique des Langues.

Sagot, B. and Fier, D. (2008). Construction d'un WordNet libre du français à partir de ressources multilingues. In *Proceedings of TALN 2008*, Avignon, France.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Tesnière, L. (1959). Éléments de syntaxe structurale.

Uszkoreit, H. and Xu, F. (2013). From strings to things sar-graphs: A new type of resource for connecting knowledge and language. In *Proceedings of the 2013th International Conference on NLP & DBpedia-Volume 1064*, pages 109–117. CEUR-WS. org.

VanRullen, T. (2005). *Vers une analyse syntaxique à granularité variable*. Ph.D. thesis, Université de Provence.

Piek Vossen, editor. (1998). *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA.

Xu, F., Uszkoreit, H., and Li, H. (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *ACL*, volume 7, pages 584–591.

9. Language Resource References

Mathieu Lafourcade. (2021). *JeuxDeMots*. Distributed by Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier (LIRMM), Version 11012021.

Jean-Philippe Prost. (2022). *HOLINET*. Distributed by ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr, Version 1.0, ISLRN (upcoming).