

Rationale, Concepts, and Current Outcome of the Unit Graphs Framework

Maxime Lefrançois, Fabien Gandon

► **To cite this version:**

Maxime Lefrançois, Fabien Gandon. Rationale, Concepts, and Current Outcome of the Unit Graphs Framework. Angelova, Galia and Bontcheva, Kalina and Mitkov, Ruslan. RANLP - 9th International Conference Recent Advances in Natural Language Processing, Sep 2013, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA, pp.382-388, 2013, <http://lml.bas.bg/ranlp2013/docs/RANLP_main.pdf>. <hal-00857652>

HAL Id: hal-00857652

<https://hal.inria.fr/hal-00857652>

Submitted on 3 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rationale, Concepts, and Current Outcome of the Unit Graphs Framework

Maxime Lefrançois and Fabien Gandon

Wimmics, Inria, I3S, CNRS, UNSA

2004 rte des Lucioles, BP. 93, 06902 Sophia Antipolis, France

{maxime.lefrancois, fabien.gandon}@inria.fr

Abstract

The Unit Graphs (UGs) framework is a graph-based knowledge representation (KR) formalism that is designed to allow for the representation, manipulation, query, and reasoning over linguistic knowledge of the Explanatory Combinatorial Dictionary of the Meaning-Text Theory (MTT). This paper introduces the UGs framework, and overviews current published outcomes. It first introduces rationale of this new formalism: neither semantic web formalisms nor Conceptual Graphs can represent linguistic predicates. It then overviews the foundational concepts of this framework: the UGs are defined over a UG-support that contains: i) a hierarchy of unit types which is strongly driven by the actantial structure of unit types, ii) a hierarchy of circumstantial symbols, and iii) a set of unit identifiers. On these foundational concepts and on the definition of UGs, this paper finally overviews current outcomes of the UGs framework: the definition of a deep-semantic representation level for the MTT, representation of lexicographic definitions of lexical units in the form of semantic graphs, and two formal semantics: one based on UGs closure and homomorphism, and one based on model semantics.

1 Introduction

The Meaning-Text Theory (MTT) is a theoretical dependency linguistics framework for the construction of models of natural language. As such, its goal is to write systems of explicit rules that express the correspondence between meanings and texts (or sounds) in various languages (Kahane, 2003). From semantic representations to surface

phonologic representations, seven different levels of linguistic representation are supposed for each set of synonymous utterances. Thus, two times six modules containing transformation rules are used to transcribe representations of a level into representations of an adjacent level. The main constituent of the MTT is the dictionary model where lexical units are described, which is called the Explanatory Combinatorial Dictionary (ECD) (Mel'čuk, 2006).

As for any community of interest, linguists and lexicographers of the MTT framework produce knowledge. Knowledge Representation (KR) is an area of artificial intelligence that deals with recurrent needs that emerge with such knowledge production.

The aim of this paper is to introduce the Unit Graphs KR formalism that is designed to allow for the representation, manipulation, query, and reasoning over dependency structures, rules and lexicographic knowledge of the ECD.

The rest of this paper is organized as follows. We will first introduce rationale of this new KR formalism (§2), then the fundamental concepts of the UGs framework (§3), implications for the MTT, lexicographic definitions and application to a specific MTT lexicographic edition project (§4), and finally two approaches to assign UGs with logical semantics, so as to enable reasoning in the UGs framework (§5).

2 Rationale: Representation of Valency-based Predicates

Most past or current projects that aimed at implementing the ECD did so in a lexicographic perspective. One important example is the RELIEF project (Lux-Pogodalla and Polguère, 2011), which aims at representing a lexical system graph named RLF (Polguère, 2009), where lexical units are interlinked by paradigmatic and syntagmatic links of lexical functions (Mel'čuk, 1996). In

the RELIEF project, the description of Lexical Functions is based on a formalization proposed by Kahane and Polguère (2001). Moreover, lexicographic definitions start to be partially formalized in the RELIEF project using the markup type that has been developed in the Definiens project (Barque and Polguère, 2008; Barque et al., 2010).

One exception is the proprietary linguistic processor ETAP-3 that implements a variety of ECD for Natural Language Processing (Apresian et al., 2003; Boguslavsky et al., 2004). Linguistic knowledge are asserted, and transformation rules are directly formalized in first order logic.

Adding to these formalization works, our goal is to propose a formalization from a knowledge engineering perspective, compatible with standard KR formalisms. The term *formalization* here means not only *make non-ambiguous*, but also *make operational*, i.e., *such that it supports logical operations* (e.g., knowledge manipulation, query, reasoning). We thus adopt a knowledge engineering approach applied to the domain of the MTT.

At first sight, two existing KR formalisms seem interesting for this job: semantic web formalisms (e.g., RDF¹, RDFS², OWL³, SPARQL⁴), and Conceptual Graphs (CGs) (Sowa, 1984; Chein and Mugnier, 2008). Both of them are based on directed labelled graph structures, and some research has been done towards using them to represent dependency structures and knowledge of the ECD (OWL in (Lefrançois and Gandon, 2011; Boguslavsky, 2011), CGs at the conceptual level in (Bohnet and Wanner, 2010)). Yet Lefrançois (2013) showed that neither of these KR formalisms can represent valency-based predicates, therefore lexicographic definitions. One crucial issue is the following: in RDFS, OWL and the CGs, there is a strong distinction between concept types and relations. Yet, a linguistic predicate may be considered both as a concept type as it is instantiated in dependency structures, and as a relation as its instances may link other instances. The simple semantic representation illustrated on figure 1 thus cannot be represented with these formalisms unless we use reification of *n*-ary rela-

tions. But then these formalisms lack logical semantics to reason with such relations.

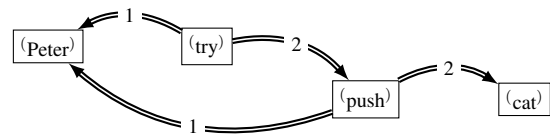


Figure 1: Semantic representation of sentence *Peter tries to push the cat*.

As the CGs formalism is the closest to the semantic networks, the following choice has been made to overcome these issues: *Modify the CGs formalism basis, and define transformations to syntaxes of Semantic Web formalisms for sharing and querying knowledge*. As we are to represent linguistic units of different nature (e.g., semantic units, lexical units, grammatical units, words), term *unit* has been chosen to be used in a generic manner, and the result of this adaptation is thus *the Unit Graphs (UGs) framework*.

3 Fundamental Concepts of the UGs Framework

First, for a specific Lexical Unit L, Mel'čuk (2004, p.5) distinguishes considering L in language (i.e., in the lexicon), or in speech (i.e., in an utterance). KR formalisms and the UGs formalism also do this distinction using types. In this paper and in the UGs formalism, there is thus a clear distinction between *units* (e.g., semantic unit, lexical unit), which will be represented in the UGs, and their *types* (e.g., semantic unit type, lexical unit type), which are roughly classes of units that share specific features. It is those types that will specify through their so-called actancial structure (Mel'čuk, 2004) how their instances (i.e., units) are to be linked to other units in a UG.

3.1 Hierarchy of Unit Types

The core of the UGs framework is a structure called *hierarchy of unit types* and noted \mathcal{T} , where unit types and their actancial structure are described. This structure is thoroughly described in (Lefrançois, 2013; Lefrançois and Gandon, 2013b) and studied in (Lefrançois and Gandon, 2013d).

Whether they are semantic, lexical or grammatical, unit types are assigned a set of *Actant Slots (ASlots)*, and every ASlot has a so-called *Actant*

¹RDF - Resource Description Framework, c.f., <http://www.w3.org/RDF/>

²RDFS - RDF Schema, c.f., <http://www.w3.org/TR/rdf-schema/>

³OWL - Web Ontology Language, c.f., <http://www.w3.org/TR/owl2-overview/>

⁴SPARQL, c.f., <http://www.w3.org/TR/sparql11-overview/>

Symbol (ASymbol) which is chosen in a set denoted $S_{\mathcal{T}}$. $S_{\mathcal{T}}$ contains numbers for the semantic unit types, and other "classical" symbols for the other levels under consideration (e.g, roman numerals **I** to **VI** for the Deep Syntactic actants). The set of ASlots of a unit type t is represented by the set $\alpha(t)$ of ASymbols these ASlots have. Moreover,

- some ASlots are obligatory, they form the set $\alpha_1(t)$ of *Obligatory Actant Slots* (OblASlots);
- other are prohibited, they form the set $\alpha_0(t)$ of *Prohibited Actant Slots* (ProASlots);
- the ASlots that are neither obligatory nor prohibited are said to be optional, they form the set $\alpha_2(t)$ of *Optional Actant Slots* (OptASlots).

Finally, every unit type t has a signature function ς_t that assigns to every ASlot of t a unit type, which characterises units that fill such a slot.

The set of unit types is then pre-ordered⁵ by a specialization relation \lesssim , and for mathematical reasons as one goes down the hierarchy of unit types the actantial structure of unit types may only become more and more specific: (i) some ASlot may appear, be optional a moment, and at some points become obligatory or prohibited; (ii) the signatures may only become more specific.

3.2 Hierarchy of Circumstantial Symbols

UGs include actantial relations, which are considered of type predicate-argument and are described in the hierarchy of unit types. Now UGs also include circumstantial relations which are considered of type instance-instance. Example of such relations are the deep syntactic representation relations **ATTR**, **COORD**, **APPEND** of the MTT, but we may also use such relations to represent the link between a lexical unit and its associated surface semantic unit for instance. Circumstantial relations are labelled by symbols chosen in a set of so-called *Circumstantial Symbols* (CSymbols), denoted $S_{\mathcal{C}}$, and their categories and usage are described in a hierarchy denoted \mathcal{C} , that has been formally defined in (Lefrançois and Gandon, 2013a).

3.3 Unit Graphs

UGs are defined over a so-called support, $\mathcal{S} \stackrel{\text{def}}{=} (\mathcal{T}, \mathcal{C}, \mathbf{M})$ where \mathcal{T} is a hierarchy of unit types, \mathcal{C}

⁵A pre-order is a reflexive and transitive binary relation.

is a hierarchy of CSymbols, and \mathbf{M} is a set of *unit identifiers*.

A UG G defined over a support \mathcal{S} is a tuple denoted $G \stackrel{\text{def}}{=} (U, \mathbf{I}, A, C, Eq)$, where U is the set of unit nodes, \mathbf{I} is a labelling mapping over U that associate every unit node with a unit type and one or more unit identifiers, A and C are respectively actantial and circumstantial triples, and Eq is a set of asserted unit node equivalences. Unit nodes are illustrated by rectangles with their label written inside, actantial triples are illustrated by double arrows, circumstantial triples are illustrated by simple arrows, and asserted unit node equivalences are illustrated by dashed arrows.

For instance, figure 1 is a semantic representation of sentence *Peter tries to push the cat.* in which units are typed by singletons and ASymbols are numbers, in accordance with the MTT. Figure 2 is a simplified deep syntactic representation of *Peter is gently pushing the cat.* In this figure unit nodes u_2 and u_4 are typed by singletons, and only unit node u_2 is not generic and has a marker: $\{Peter\}$. P is composed of (u_1, \mathbf{I}, u_2) and (u_1, \mathbf{II}, u_3) , where **I** and **II** are ASymbols. C is composed of $(u_1, \mathbf{ATTR}, u_4)$ where **ATTR** is a CSymbol. In the relation Eq there is (u_1, u_1) , (u_2, u_2) , and (u_3, u_3) .

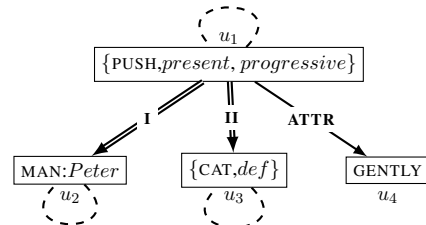


Figure 2: Deep syntactic representation of sentence *Peter is gently pushing the cat.*

UGs so defined are the core dependency structures of the UGs mathematical framework.

4 Unit Graphs and the Meaning-Text Theory

4.1 A Deep-Semantic Representation Level

As the unit types hierarchy \mathcal{T} is driven by the actantial structure of unit types, and as semantic ASymbols are numbers, the pre-order over unit types at the semantic level represents a specialization of the actantial structure, and not of meanings. For instance, the french lexical unit INSTRUMENT

(en: instrument) has a Semantic ASlot 1 that corresponds to the activity for which the instrument is designed. Now PEIGNE (en: comb) has a stricter meaning than INSTRUMENT, and also two Semantic ASlots: 1 correspond to the person that uses the comb, and 2 is a split variable⁶ that corresponds either to the hair or to the person that is to be combed. Then semantic unit type (peigne) cannot be more specific than (instrument) in the hierarchy of unit types because the signature of its ASlot 1 is not more specific than the signature of the ASlot 1 of (instrument), i.e., $\varsigma_{(peigne)}(1) = (person) \not\prec (activity) = \varsigma_{(instrument)}(1)$. In fact, the meaning of ASlot 1 is not the same for (instrument) and (peigne).

Lefrançois and Gandon (2013b) therefore introduced a deeper level of representation to describe meanings: the *deep semantic level*, and defined the deep and surface semantic unit types and their actantial structure. The Deep Semantic Unit Type (DSemUT) associated with a Lexical Unit Type (LexUT) L is denoted $/L\backslash$. So that the ASlots of deep semantic unit types convey meanings, the set of ASymbols that is used to symbolize ASlots at this level is a set of lexicalized semantic roles (e.g., *agent*, *combedhair*, *combedperson*). For instance the DSemUT $/instrument\backslash$ associated with the LexUT INSTRUMENT may have an ASlot arbitrarily symbolized *activity*, which would be inherited by the DSemUT $/peigne\backslash$. Then $/peigne\backslash$ also introduces three new ASlots: one arbitrarily symbolized *possessor* that corresponds to the ASlot 1 of (peigne), and two arbitrarily symbolized *combedhair*, and *combedperson* that correspond to the ASlot 2 of (peigne).

Actually, one may need to introduce a new ASymbol every time a Semantic ASlot that conveys a new meaning is introduced. The set of semantic roles thus cannot be bound to a small set of universal semantic roles.

4.2 Lexicographic Definitions

It is at the deep semantic representation level that one may represent the actual meaning of a LexUT L. The lexicographic definition of L corresponds to the definition of its associated DSemUT $/L\backslash$, which is roughly an equivalence between two deep semantic UGs. Unit type definitions have been formally defined in (Lefrançois

⁶For details about split Semantic ASlots, see (Mel'čuk, 2004, p.43)

and Gandon, 2013a), and the definition of $/L\backslash$ is a triple $D_{/L\backslash} \stackrel{\text{def}}{=} (D_{/L\backslash}^-, D_{/L\backslash}^+, \kappa)$, where (roughly):

- $D_{/L\backslash}^-$ represents only a central unit node typed with $/L\backslash$, and some other unit nodes that fill some of the ASlots of $/L\backslash$;
- $D_{/L\backslash}^+$ is a UG called the *expansion* of $/L\backslash$,
- there is no circumstantial triple in these two UGs because circumstantials must not be part of the lexicographic definition of a LexUT.
- κ is a mapping from the unit nodes of $D_{/L\backslash}^-$ to some unit nodes of $D_{/L\backslash}^+$.

Figure 3 is an example of lexicographic definition of PEIGNE: an instrument that a person X uses to untangle the hair Y_1 of a person Y_2 .

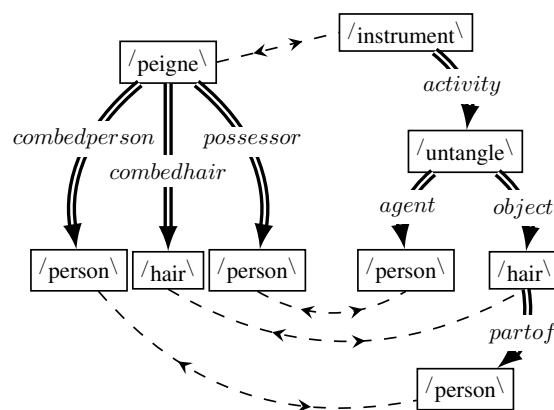


Figure 3: Lexicographic definition of PEIGNE.

Intuitively, a definition corresponds to two reciprocal rules. If there is the defined PUT in a UG then one may infer its definition, and vice versa. A set of unit type definitions \mathcal{D} may thus be added to the unit types hierarchy.

Lefrançois et al. (2013) illustrated how the UGs framework may be used to edit lexicographic definitions in the RELIEF lexicographic edition project (Lux-Pogodalla and Polguère, 2011). Lexical Units are assigned a semantic label that may be considered as a deep semantic unit type and to which one may assign an actantial structure. A lexicographer may then manipulate nodes in a graphical user interface so as to little by little construct a deep semantic UG that represents the decomposition of the DSemUT associated with the defined LexUT. A prototype web application has been developed, and a demonstration is available online: <http://wimmics.inria.fr/doc/video/UnitGraphs/editor1.html>. We currently lead an ergonomic study in partnership

with actors of the RELIEF project in order to enhance the workflow of this prototype.

5 Reasoning in the Unit Graphs Framework

The prime decision problem of the UGs framework is the following: *Considering two UGs G and H defined over the same support \mathcal{S} , does the knowledge of G entail the knowledge of H ?*

5.1 Reasoning with UGs-Homomorphisms

Lefrançois and Gandon (2013a) proposed to use the notion of UGs homomorphism to define this entailment problem. There is a homomorphism from a UG H to a UG G if and only if there is a homomorphism from the underlying oriented labelled graphs of H to that of G .

Now one need to define the notion of knowledge of a UG. In fact, the UGs framework makes the open-world assumption, which means that a UG along with the support on which it is defined represents explicit knowledge, and that additional knowledge may be inferred. Consider the UG $G = (U, I, A, C, Eq)$ defined over the support \mathcal{S} illustrated in figure 4a. Some knowledge in G is implicit:

1. two unit nodes u_1 and u_2 share a common unit marker *Mary*, so one may infer that they represent the same unit. (u_1, u_2) may be added to *Eq*.
2. every unit type is a subtype of the prime universal unit type \top , so one could add \top to all the types of unit nodes in G .
3. there are two unit nodes v_1 and v_2 that fill the same ASlot *activity* of the unit node typed */instrument*. So one may infer that v_1 and v_2 represent the same unit. Said otherwise, (v_1, v_2) may be added to *Eq*.
4. one may recognize the expansion of */peigne* as defined in figure 3, so this type may be made explicit in the unit node typed */instrument*.

Each of the rules behind these cases explicit knowledge in G . More generally, Lefrançois and Gandon (2013a) listed a set of rules which defines the axiomatization of the UGs semantics. The process of applying this set of rules on a UG G until none of them has any effect is called closing G . Figure 4b illustrates the closure of G , where all of the inferable knowledge has been made explicit.

The notion of *entailment* may hence be defined as follows: G entails H , noted $G \models_h H$, if and only if there is a homomorphism from H to the closure of G . Lefrançois and Gandon (2013a) illustrated problematic cases where the closure is infinite for finite UGs. If that occurs it makes the closure undecidable, along with the entailment problem. We are currently working of the definition of restrictions of the unit types hierarchy and the set of definitions in order to ensure that any UG has a finite closure.

5.2 Model Semantics for the UGs framework

Another approach to defining the entailment problem has been presented in (Lefrançois and Gandon, 2013c), using model semantics based on relational algebra. The model of a support $\mathcal{S} = (\mathcal{T}, \mathcal{C}, \mathbf{M})$ is a couple $M = (D, \delta)$, where D is a set called the domain of M , and δ is denoted the *interpretation function*. In order to deal with the problem of prohibited and optional ASlots, D contains a special element denoted \bullet that represents *nothing*, plus at least one other element, and must be such that:

- M is a model of \mathcal{T} ;
- M is a model of \mathcal{C} ;
- for all unit identifier $m \in \mathbf{M}$, the interpretation of m is an object of the domain D except for the special *nothing* element;

Lefrançois and Gandon (2013c) listed the different equations that the interpretation function must satisfy so that a model is a model of a unit types hierarchy and of a CSymbols hierarchy.

A model of a UG G is a model of the support on which it is defined, augmented with an assignment function β , which is a mapping from the set of unit nodes of G to the domain D . Such a model needs to satisfy a list of equations so that it may be said to satisfy the unit graph G .

Then the notion of entailment is defined as classically done with model semantics: Let H and G be two UGs defined over a support \mathcal{S} . G entails H , or H is a *semantic consequence* of G , noted $G \models_m H$, if and only if for any model (D, δ) of \mathcal{S} and for any assignment β_G such that (D, δ, β_G) satisfies G , then there exists an assignment β_H of the unit nodes in H such that (D, δ, β_H) satisfies H .

There are multiple directions of research for the reasoning problem.

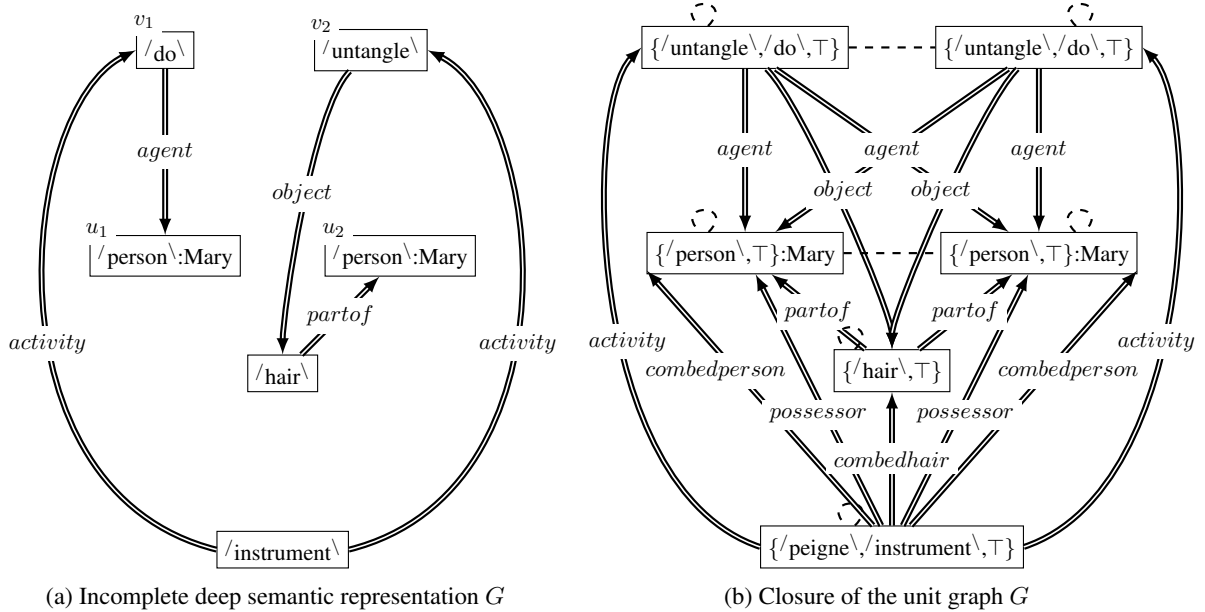


Figure 4: Closure of a UG.

- the definition of the model semantics of the UGs shall be completed so as to take lexicographic definitions into account.
- one need to define algorithms to construct a model that satisfy a UG, and to check the entailment of a UG by another.
- such algorithms may lead to an infinite domain. A condition over the unit types hierarchy and the lexicographic definitions must be found so as to ensure that the model is decidable for a finite UG.
- are the two entailment relations \models_h and \models_m equivalent ?

6 Conclusion

We thus introduced rationale of the new Unit Graphs Knowledge Representation formalism that is designed to formalize, in a knowledge engineering perspective, the dependency structures, the valency-based predicates, and lexicographic definitions in the ECD.

The strong coherence in the unit types hierarchy justifies the introduction of a deep semantic representation level that is deeper than the MTT semantic level, and in which one may represent the lexicographic definitions.

Finally, two different logical semantics have been provided for UGs and the prime entailment decision problem has been defined in two ways. More research is needed to determine if these two decision problems are equivalent, and what their

complexity is.

There are other longer-term directions of research for the Unit Graphs framework:

We are working on a syntax based on semantic web standards for the different objects of the framework. Like WordNet today, the linguistic knowledge written with that syntax could be shared and queried on the web of linked data⁷. This would support their use as a highly structured lexical resource by consumers of the linked data cloud.

Rules have already been defined in the UGs framework. Let G_{DSeM} be a deep semantic UG, we need algorithms to select and apply correspondence rules to transcribe G_{DSeM} to a surface semantic UG G_{SSeM} for instance.

We are working on defining generic rules to formally represent semantic derivations. This is a first step towards representing Lexical Functions that play a very important role in the MTT.

Finally, the design of the Unit Graphs framework is a first step towards Natural Language Processing applications. Future work include (semi-automatically) populating this model with linguistic data, and formulating classical NLP tasks in terms of UGs, such as machine translation, question answering, text summarization, and so on.

⁷The web of data is a W3C initiative, highly active today, <http://linkeddata.org>

References

- Juri Apresian, Igor Boguslavsky, Leonid Iomdin, Alexander Lazursky, Vladimir Sannikov, Victor Sizov, and Leonid Tsinman. 2003. ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT. In *First International Conference on Meaning-Text Theory (MTT'2003)*, pages 279–288.
- Lucie Barque and Alain Polguère. 2008. Enrichissement formel des définitions du Trésor de la Langue Française informatisé (TLFi) dans une perspective lexicographique. *Lexique*, 22.
- Lucie Barque, Alexis Nasr, and Alain Polguère. 2010. From the Definitions of the 'Trésor de la Langue Française' To a Semantic Database of the French Language. In Fryske Academy, editor, *Proceedings of the XIV Euralex International Congress*, Fryske Academy, pages 245–252, Leeuwarden, Pays-Bas. Anne Dykstra et Tanneke Schoonheim, dir.
- Igor Boguslavsky, Leonid Iomdin, and Viktor Sizov. 2004. Multilinguality in ETAP-3: reuse of lexical resources. In Gilles Sérasset, editor, *Proc. COLING 2004 Multilingual Linguistic Ressources*, pages 1–8, Geneva, Switzerland. COLING.
- Igor Boguslavsky. 2011. Semantic Analysis Based on Linguistic and Ontological Resources. In Igor Boguslavsky and Leo Wanner, editors, *Proceedings of the 5th International Conference on Meaning-Text Theory (MTT'2011)*, pages 25–36, Barcelona, Spain. INALCO.
- Bernd Bohnet and Leo Wanner. 2010. Open source graph transducer interpreter and grammar development environment. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 19–21, Valtta, Malta. European Language Resources Association (ELRA).
- Michel Chein and Marie-Laure Mugnier. 2008. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer-Verlag New York Incorporated.
- Sylvain Kahane and Alain Polguère. 2001. Formal foundation of lexical functions. In *Proceedings of ACL/EACL 2001 Workshop on Collocation*, pages 8–15.
- Sylvain Kahane. 2003. The Meaning-Text Theory. *Dependency and Valency, An International Handbooks of Contemporary Research*, 25(1):546–569.
- Maxime Lefrançois and Fabien Gandon. 2011. iLexicon: Toward an ECD-Compliant Interlingual Lexical Ontology Described with Semantic Web Formalisms. In Igor Boguslavsky and Leo Wanner, editors, *Proceedings of the 5th International Conference on Meaning-Text Theory (MTT'2011)*, pages 155–164, Barcelona, Spain. INALCO.
- Maxime Lefrançois and Fabien Gandon. 2013a. Reasoning with Dependency Structures and Lexicographic Definitions using Unit Graphs. In *Proc. of the 2nd International Conference on Dependency Linguistics (Depling'2013)*, Prague, Czech Republic. ACL Anthology.
- Maxime Lefrançois and Fabien Gandon. 2013b. The Unit Graphs Framework: A graph-based Knowledge Representation Formalism designed for the Meaning-Text Theory. In *Proceedings of the 6th International Conference on Meaning-Text Theory (MTT'2013)*, Prague, Czech Republic.
- Maxime Lefrançois and Fabien Gandon. 2013c. The Unit Graphs Framework: Foundational Concepts and Semantic Consequence. In *Proceedings of the 9th International Conference on Recent Advances in Natural Language Processing (RANLP'2013)*, Hissar, Bulgaria. ACL Anthology.
- Maxime Lefrançois and Fabien Gandon. 2013d. The Unit Graphs Mathematical Framework. Research Report RR-8212, Inria.
- Maxime Lefrançois, Romain Gugert, Fabien Gandon, and Alain Giboin. 2013. Application of the Unit Graphs Framework to Lexicographic Definitions in the RELIEF project. In *Proceedings of the 6th International Conference on Meaning-Text Theory (MTT'2013)*, Prague, Czech Republic.
- Maxime Lefrançois. 2013. Représentation des connaissances du DEC: Concepts fondamentaux du formalisme des Graphes d'Unités. In *Actes de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'2013)*, pages 164–177, Les Sables d'Olonne, France.
- Veronika Lux-Pogodalla and Alain Polguère. 2011. Construction of a French Lexical Network: Methodological Issues. In *Proceedings of the International Workshop on Lexical Resources*, Ljubljana.
- Igor Mel'čuk. 1996. Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. Benjamins Academic Publishers, Amsterdam/Philadelphia.
- Igor Mel'čuk. 2004. Actants in Semantics and Syntax I: Actants in Semantics. *Linguistics*, 42(1):247–291.
- Igor Mel'čuk. 2006. Explanatory combinatorial dictionary. *Open Problems in Linguistics and Lexicography*, pages 225–355.
- Alain Polguère. 2009. Lexical systems: graph models of natural language lexicons. *Language resources and evaluation*, 43(1):41–55.
- John F Sowa. 1984. *Conceptual structures: information processing in mind and machine*. System programming series. Addison-Wesley Pub., Reading, MA.