Deep Semantics in an NLP Knowledge Base

Carlos Periñán-Pascual[1] and Francisco Arcas-Túnez[1]

[1] Universidad Católica San Antonio, Campus de los Jerónimos s/n, 30107 Guadalupe–Murcia, Spain
jcperinan@pdi.ucam.edu, farcas@pdi.ucam.edu

Abstract. In most natural language processing systems there is no representation of the semantic knowledge of lexical units, but just subcategorization frames, selection restrictions and links to other paradigmatically-related lexical units. Some NLP systems, e.g. machine translation or dialogue-based systems, attempt to "understand" the input text by translating it into some kind of formal language-independent representation; this approach requires a knowledge base with conceptual representations which reflect the structure of human beings' cognitive system. Even those systems in which surface semantics could be sufficient (e.g. automatic indexing or information extraction), the construction of a robust knowledge base guarantees its use in most natural language processing tasks, consolidating thus the concept of resource reuse. The objective of this paper is to highlight the advantages of storing conceptual meaning representations, and more particularly those in FunGramKB, instead of describing lexical meaning via semantic relations between lexical units.

Keywords. Natural language processing, knowledge base, meaning representation, deep semantics.

1 FunGramKB and Cognitive Knowledge

FunGramKB is a complex knowledge base which comprises two comprehensive information levels, where several independent modules are interrelated:

Lexical level (i.e. linguistic knowledge):

- The lexicon stores morphosyntactic, pragmatic and collocational information of words**.**
- The morphicon helps our system to handle cases of inflectional morphology.

Cognitive level (i.e. non-linguistic knowledge):

- The ontology is presented as a hierarchical structure of all the concepts that a person has in mind when talking about everyday situations.
- The cognicon stores procedural knowledge by means of cognitive macrostructures, i.e. script-like schemata in which a sequence of stereotypical actions is organised on the basis of temporal continuity, and more particularly on Allen's temporal model [1, 2, 3].
- The onomasticon stores information about instances of entities, such as people, cities, products, etc.

This two-level design involves that every lexical module is language-specific, while every cognitive module is shared by all languages. In other words, computational lexicographers must develop one lexicon and one morphicon for English, one lexicon and one morphicon for Spanish and so on, but knowledge engineers build just one ontology, one cognicon and one onomasticon to process any language input cognitively. FunGramKB is multipurpose in the sense that it is both multifunctional and multilingual. In other words, FunGramKB has been designed to be reused in various NLP tasks (e.g. information retrieval and extraction, machine translation, dialogue-based systems, etc) and with several natural languages.[1]

2 Semantic Knowledge Representation in FunGramKB

In cognitive psychology, commonsense knowledge is usually divided into three different types [11]:

- Semantic knowledge - it stores cognitive information about words; it is a kind of mental dictionary.
- Procedural knowledge - it stores information about how events are performed in ordinary situations: e.g. how to ride a bicycle, how to fry an egg...; it is a kind of manual for everyday actions.
- Episodic knowledge - it stores information about specific biographic events or situations: e.g. our wedding-day; it is a kind of personal scrapbook.

Therefore, if there are three types of knowledge involved in human reasoning, there must be three different kinds of knowledge schemata. These schemata are successfully mapped in an integrated way into the cognitive level of FunGramKB:

- Semantic knowledge is represented in the form of meaning postulates in the ontology.
- Procedural knowledge is represented in the form of cognitive macrostructures in the cognicon.
- Episodic knowledge can be stored as a case base.[2]

A key factor for successful reasoning with FunGramKB is that all these knowledge schemata (i.e. meaning postulates, cognitive macrostructures and cases) are represented through the same formal language, so information sharing takes place more effectively among all cognitive modules. Our formal language of cognitive

[1] English, Spanish, German, French and Italian are supported in the current version of FunGramKB.

[2] FunGramKB can be very useful in case-based reasoning, where problems are solved by remembering previous similar cases and reusing general knowledge.

representation is partially founded on Dik's functional model [6, 7], which was initially devised for machine translation [5].[3]

This paper focuses on the semantic knowledge representation, which takes the form of meaning postulates in FunGramKB. A meaning postulate is a set of one or more logically connected predications (e_1, e_2... e_n), which are cognitive constructs carrying the generic features of the concept.[4] Concepts, and not words, are the building blocks for the formal description of meaning postulates, so a meaning postulate becomes a language-independent semantic knowledge representation. To illustrate, predications in the meaning postulate of BIRD are presented in (1):[5]

(1) BIRD
+(e_1: +BE_00 (x_1: +BIRD_00)$_{Theme}$ (x_2: +VERTEBRATE_00)$_{Referent}$)
*(e_2: +COMPRISE_00 (x_1)$_{Theme}$ (x_3: $_m$ +FEATHER_00 & $_2$ +LEG_00 & $_2$ +WING_00)$_{Referent}$)
*(e_3: +FLY_00 (x_1)$_{Agent}$ (x_1)$_{Theme}$ (x_4)$_{Origin}$ (x_5)$_{Goal}$)

These predications have the following natural language equivalents:

(2) Birds are always vertebrates.
A typical bird has many feathers, two legs and two wings.
A typical bird flies.[6]

Dik proposes using words from the own language when describing meaning postulates, since meaning definition is an internal issue of the language [7]. However, this strategy contributes to lexical ambiguity due to the polysemous nature of the defining lexical units. In addition, describing the meaning of words in terms of other words leads to some linguistic dependency [13]. Instead, FunGramKB employs concepts for the formal description of meaning postulates, resulting in an interlanguage representation of meaning.

A parser written in C# takes meaning postulates such as (1) and outputs XML-formatted feature-value structures used as the input for the reasoning engine.[7] An

[3] FunGramKB is not a literal implementation of Dik's lexical database, but we depart from his model in some important aspects with the aim of building a more robust knowledge base.

[4] The formal grammar of well-formed predications for meaning postulates in FunGramKB is described in [9].

[5] For the sake of clarity, the names of concepts—which are presented in upper case—have been oversimplified.

[6] In FunGramKB, each predication taking part in a meaning postulate is preceded by a reasoning operator in order to state if the predication is strict (+) or defeasible (*). Our inference engine handles predications as rules, allowing monotonic reasoning with strict predications, and non-monotonic with defeasible predications.

[7] To illustrate, appendix 1 shows example (1) in XML.

alternative could have been to use second-order predicate logics for the formal representation of lexical meaning. However, the problem lies not only on the little expressive power of predicate logics, but also on the fact that standard logics use monotonic reasoning, which isn't robust enough for the simulation of human beings' commonsense reasoning.

3 Benefits of Conceptual Meaning Representation in FunGramKB

Two strategies are typically used when describing meaning in computational lexicography [12]: the cognitive content in a lexical unit can be described by means of semantic features or primitives (i.e. conceptual meaning), or through associations with other lexical units in the lexicon (i.e. relational meaning). While the analysis of conceptual meaning is related to deep semantics, relational meaning belongs to surface semantics. Strictly speaking, the latter doesn't give a real definition of the lexical unit, but it describes its usage in the language via "meaning relations" with other lexical units. In this section, we demonstrate that surface semantics presents two main types of problems which can be overcome by deep semantics: its expressive power is dramatically restricted, and redundancy is highly spread through the knowledge base. To illustrate this comparative analysis between relational and conceptual meanings, we take semantic relations from EuroWordNet [4, 14] and meaning postulates from FunGramKB.[8]

FunGramKB meaning postulates own greater expressive power than surface semantics. For example, EuroWordNet cannot fully exploit the cognitive content of lexical units, particularly when it is required to use some defining concepts which do not directly describe the ***definiendum*** but qualify neighbouring concepts in the meaning postulate (example 3):

(3) OSTRICH
$((e_3\text{: COMPRISE } (x_1)_{\text{Theme}} (x_4\text{: }_2\text{ LEG \& }_1\text{ NECK})_{\text{Referent}})(e_4\text{: BE } (x_4)_{\text{Theme}} (x_5\text{: LONG})_{\text{Attribute}}))$
$((e_5\text{: COMPRISE } (x_1)_{\text{Theme}} (x_6\text{: }_m\text{ FEATHER})_{\text{Referent}})(e_6\text{: BE } (x_6)_{\text{Theme}} (x_7\text{: LARGE \& SOFT})_{\text{Attribute}}))$
$((e_7\text{: LIVE } (x_1)_{\text{Theme}} (x_8)_{\text{Location}})(e_8\text{: BE } (x_8)_{\text{Theme}} (x_9\text{: HOT})_{\text{Attribute}}))$

In this example, HOT does not describe an attribute of the entity referenced by OSTRICH but of the typical places where instances of this entity live in. Similar cases of "cognitive subordination" are found in the other two predications of this example (e.g. "a typical ostrich has many feathers, which are large and soft"). It is also very hard for surface semantics to represent phenomena such as quantification,

[8] EuroWordNet is one of the best examples of multilingual "relational" database, which provides elaborate lexical networks by means of semantic relations between *synsets* (or cluster of synonymous words) within every language-dependent wordnet. In the examples of this paper, EuroWordNet relational specifications are presented more meaningfully by stating the most representative synset members involved instead of synsets' unique identifiers.

aspectuality, temporality or modality. In example (3), operators *2*, *1* and *m* specify absolute quantification (e.g. “two legs and one neck”) and relative quantification (e.g. “many feathers”) for selection preferences in arguments (x_4) and (x_6) respectively. In example (4), operators *egr* (egressive) and *past* place some meaning components within the dimensions of aspectuality and temporality respectively:

(4) FORGIVE
(e_1: $_{egr}$ FEEL (x_1: HUMAN)$_{Theme}$ (x_2: ANGRY)$_{Attribute}$ (f_1: HUMAN)$_{Goal}$)
(e_2: $_{past}$ BLAME (x_1)$_{Theme}$ (x_3)$_{Referent}$ (x_4: f_1)$_{Goal}$)

In addition, EuroWordNet has introduced ten semantic relations (e.g. ROLE_AGENT, ROLE_INSTRUMENT, etc), and their reverse counterparts (e.g. INVOLVED_AGENT, INVOLVED_INSTRUMENT, etc), to encode data about arguments and adjuncts strongly involved in the meaning of verbs. By integrating frame semantics into surface semantics, meaning postulate (5) could have a near-equivalent in (6), except for the lack of distinction between inclusive and exclusive disjunctions:

(5) SWIM
(e_1: MOVE (x_1: HUMAN ^ ANIMAL)$_{Theme}$ (f_1: WATER)$_{Means}$ (f_2: ARM | LEG)$_{Instrument}$)

(6)

swim	HAS_HYPERONYM	move	
swim	INVOLVED_AGENT	person	
swim	INVOLVED _AGENT	animal	*dis*
swim	INVOLVED _LOCATION	water	
swim	INVOLVED_INSTRUMENT	arm	
swim	INVOLVED_INSTRUMENT	leg	*dis*

However, when meaning postulates become more complex cognitively, there is no way to state co-reference between internal conceptual units just via semantic relations. For example, co-indexation of arguments and satellites in example (4) allows the system to “understand” that the person who is forgiven did something wrong to the forgiver.

Taking advantage of the descriptive power of FunGramKB semantic knowledge formalism, we also use it as interlingua in the analysis and generation of texts, what favours the integration of lexical meaning in text semantics. Moreover, meaning postulates in the ontology and cognitive macrostructures in the cognicon are represented through the same formal language; thus knowledge can be shared more effectively between FunGramKB cognitive modules, particularly when reasoning mechanisms are triggered. To illustrate, example (7) presents some predications of the cognitive macrostructure EATING_AT_A_RESTAURANT:

(7) EATING_AT_A_RESTAURANT
(e_1: ENTER (x_1: CUSTOMER)$_{Theme}$ (x_2: RESTAURANT)$_{Target}$ (f_1: (e_2: BE (x_1)$_{Theme}$ (x_3: HUNGRY)$_{Attrribute}$))$_{Reason}$)
(e_3: ACCOMPANY (x_4: WAITER)$_{Theme}$ (x_1)$_{Referent}$ (f_2: TABLE)$_{Target}$)

(e_4: SIT (x_1)$_{Theme}$ (x_5: f_1)$_{Location}$)
(e_5: BRING (x_4)$_{Theme}$ (x_6: MENU ^ WINE_LIST)$_{Referent}$ (f_4: x_1)$_{Target}$)
(e_6: REQUEST (x_1)$_{Theme}$ (x_7: FOOD | BEVERAGE)$_{Referent}$ (x_4)$_{Target}$)
(e_7: TELL (x_4)$_{Theme}$ (x_2: (e_8: COOK (x_{10}: COOK)$_{Theme}$ (x_8: FOOD)$_{Referent}$)$_{Referent}$ (x_{10})$_{Target}$)
(e_9: BRING (x_4)$_{Theme}$ (x_9: BEVERAGE)$_{Referent}$ (f_3: BAR)$_{Source}$)

Little effort has been made to build large-scale databases of procedural-knowledge schemata by means of semantic relations. To this respect, ThoughtTreasure [8] is an exceptional case of knowledge base for commonsense reasoning, containing about one hundred scripts. Thus, the first predication in cognitive macrostructure (8) would be closely mapped to the following relations in ThoughtTreasure:

(8) SCRIPT eat-in-restaurant
[r1 eat-in-restaurant human]
…
[r3 eat-in-restaurant restaurant]
[role01-of eat-in-restaurant customer]
…
[goal-of eat-in-restaurant [s-hunger customer]]
…
[event02-of eat-in-restaurant [arrive customer na restaurant]]

However, this type of scripts presents the inherent deficiencies of relational notation, i.e. less descriptive power and more redundancy.

As far as redundancy in lexical meaning representations is concerned, duplication of knowledge is particularly remarkable when reverse relations or one-to-many relations are stored (examples 9 and 10):

(9)	bird	HAS_MERO_PART	feather	
	feather	HAS_HOLO_PART	bird	*rev*
(10)	bird	HAS_MERO_PART	feather	
	bird	HAS_MERO_PART	leg	*con*
	bird	HAS_MERO_PART	wing	*con*

The problem is that most NLP ontologies work with asymmetric binary semantic relations. On the one hand, the relation between a source concept and a target one is not usually the same as that between the target concept and the source one. The most direct consequence is that the number of semantic relations outgrows. For example, EuroWordNet provides an inventory of sixty-five relations, but just nine of them are really symmetric—mainly those related to the language phenomena of synonymy and antonymy; consequently, twenty-seven relations could have been ignored if the relations themselves hadn't displayed an intrinsic conceptual unidirectionality (example 9). Moreover, label *rev* is used to emphasize the asymmetric condition of the relation. On the other hand, whenever a source concept is linked to multiple target

concepts, the database must store the whole relational specification for every target concept (example 10).

Furthermore, multilingual databases contribute to redundancy if relations are not assigned to concepts in a language-independent ontology, as this is the case of EuroWordNet (example 11):

(11)

bird	HAS_MERO_PART	feather
Vogel	HAS_MERO_PART	Feder
pájaro	HAS_MERO_PART	pluma
uccello	HAS_MERO_PART	piuma
oiseau	HAS_MERO_PART	plume

The efficiency of knowledge management in FunGramKB is clearly shown in example (12), whose predication can infer all semantic relations in examples (9-11):[9]

(12) BIRD
*(e_2: COMPRISE (x_1: BIRD)$_{Theme}$ (x_3: $_m$ FEATHER & $_2$ LEG & $_2$ WING)$_{Referent}$)

Redundancy originated by multilingualism does not occur in FunGramKB, since meaning postulates are cognitive representations of concepts, to which lexical units from different lexica are assigned.

Finally, FunGramKB reasoning engine also contributes to minimize redundancy as well as maximizing informativeness in our semantic knowledge repository. A meaning postulate in FunGramKB is like an iceberg - only a small amount is visible from the surface, so a lexical unit is associated to much more semantic information which is really shown in the meaning postulate of the concept to which that lexical unit is linked. In FunGramKB, all this underlying cognitive information is revealed through a process called MicroKnowing (Microconceptual-Knowledge Spreading), which takes place in the ontology of our system. This multi-level process is performed by means of two types of reasoning mechanisms: inheritance and inference. Our inheritance mechanism strictly involves the transfer of one or several predications from a superordinate concept to a subordinate one in the ontology. On the other hand, our inference mechanism is based on the structures shared between predications linked to conceptual units which do not take part in the same subsumption relation within the ontology. Both inheritance and inference can be successfully applied providing that the "stepwise conceptual decomposition" process is also triggered, i.e. conceptual units in a predication are substituted by their respective meaning postulates until a meaning representation composed of root basic concepts is reached. It is has been demonstrated elsewhere [10] that a meaning

[9] Moreover, semantic knowledge representation in (13) provides a more accurate and realistic account of the world model, since FunGramKB does not assert that "birds have feathers" but that "a typical bird has many feathers", what can allow non-monotonic reasoning when dealing with concepts such as PENGUIN. Non-monotonicity is a key issue in both human and machine reasoning, because it permits the withdrawal of conclusions which are true just for the typical members of a particular class.

postulate consisting of just four predications can be easily spread to a set of twenty-four predications.

4 Conclusion

Currently most NLP systems adopt a relational approach to represent lexical meanings, since it is easier to state associations among lexical units in the way of meaning relations than describing the cognitive content of lexical units formally. Although large-scale development of deep-semantic resources requires a lot of time and effort, the expressive power of conceptual meanings is much more robust and the management and maintenance of their knowledge becomes more efficient. In addition, even when surface semantics can be sufficient in some NLP systems (e.g. information retrieval or data mining), the construction of a knowledge base such as FunGramKB guarantees its use for any NLP task, consolidating thus the concept of resource reuse.

References

1. Allen, J.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM 26-11 (1983) 832-843
2. Allen, J.: Time and Time Again: The Many Ways to Represent Time. International Journal of Intelligent Systems 6-4 (1991) 341-355
3. Allen, J., Ferguson, G.: Actions and Events in Interval Temporal Logic. Journal of Logic and Computation 4-5 (1994) 531-579
4. Alonge, A., Calzolari, N., Vossen, P., Bloksma, L., Castellón, I., Martí, M.A., Peters, W.: The Linguistic Design of the EuroWordNet Database. Computers and the Humanities 32-2/3 (1998) 91-115
5. Connolly, J.H., Dik, S.C. (eds.): Functional Grammar and the Computer. Foris, Dordrecht (1989)
6. Dik, S.C.: Functional Grammar. Foris: Dordrecht (1978)
7. Dik, S.C.: The Theory of Functional Grammar. Mouton de Gruyter, Berlin New York (1997)
8. Mueller, E.T.: A Database and Lexicon of Scripts for ThoughtTreasure (1999) [http://cogprints.ecs.soton.ac.uk/archive/00000555/]
9. Periñán-Pascual, C., Arcas-Túnez, F.: Meaning Postulates in a Lexico-Conceptual Knowledge Base. 15th International Workshop on Database and Expert Systems Applications. Zaragoza (2004) 38-42
10. Periñán-Pascual, C., Arcas-Túnez, F.: Microconceptual-Knowledge Spreading in FunGramKB. 9th IASTED International Conference on Artificial Intelligence and Soft Computing. ACTA Press, Anaheim Calgary Zurich (2005) 239- 244
11. Tulving, E.: How Many Memory Systems Are There?. American Psychologist 40 (1985) 385-398
12. Velardi, P., Pazienza, M.T., Fasolo, M.: How to Encode Semantic Knowledge: A Method for Meaning Representation and Computer-aided Acquisition. Computational Linguistics 17-2 (1991) 153-170
13. Vossen, P.: The End of the Chain: Where Does Decomposition of Lexical Knowledge Lead us Eventually?. In: Engberg-Pedersen, E., Jakobsen, L., Schack Rasmussen, L.

(eds.): Function and Expression in Functional Grammar. Mouton de Gruyter, Berlin New York (1994) 11-39
14. Vossen, P.: Introduction to EuroWordNet. Computers and the Humanities 32-2/3 (1998) 73-89

Appendix 1: FunGramKB Meaning Postulate for BIRD in XML

```
<MP>
    <e N="1" OPr="+">
        <lbcv>
            <cv>+BE_00</cv>
        </lbcv>
        <lx>
            <x N="1" SFx="Theme">
                <lbceq>
                    <ceq>+BIRD_00</ceq>
                </lbceq>
            </x>
            <x N="2" SFx="Referent">
                <lbceq>
                    <ceq>+VERTEBRATE_00</ceq>
                </lbceq>
            </x>
        </lx>
    </e>
    <e N="2" OPr="*">
        <lbcv>
            <cv>+COMPRISE_00</cv>
        </lbcv>
        <lx>
            <x N="1" SFx="Theme"/>
            <x N="3" SFx="Referent">
                <lbceq>
                    <and>
                        <ceq OPxf="m">+FEATHER_00</ceq>
                        <ceq OPxf="2">+LEG_00</ceq>
                        <ceq OPxf="2">+WING_00</ceq>
                    </and>
                </lbceq>
            </x>
        </lx>
    </e>
    <e N="3" OPr="*">
        <lbcv>
            <cv>+FLY_00</cv>
```

```
        </lbcv>
        <lx>
            <x N="1" SFx="Agent"/>
            <x N="1" SFx="Theme"/>
            <x N="4" SFx="Origin"/>
            <x N="5" SFx="Goal"/>
        </lx>
    </e>
</MP>
```