The Workshop Programme

2:00pm: **Welcome and Introduction**

2:20pm: **Ferruz-Beltrán, P. J. and Gervás, P.** Universidad Complutense de Madrid, Spain
Converting Frames into OWL: Preparing Mikrokosmos for Linguistic Creativity

2:40pm: **Lönneker, B.** University of Hamburg, Germany
Lexical databases as resources for linguistic creativity: focus on metaphor

3:00pm: **Pérez y Pérez, R.** Universidad Nacional Autónoma de México, Mexico
Emotions and plot generation in MEXICA

3:20pm: **Stock, O. and Strapparava, C.** ITC-irst, Italy
Resources for "Computational On-line Meditative Intelligence for Computers"

3:40pm **Coffee Break**

3:50pm: **Hayes, J. and Veale, T.** University College Dublin, Ireland
Interpreting noun-noun compounds: A truly large-scale model

4:10pm **Choi, K-S., and Kim, Y-B.** Computer Science Division, Korterm, KAIST, Korea
Knowledge-seeking activities for content intelligence

4:30pm **Mendes, M, Pereira, F. C., and Cardoso, A.** Universidade de Coimbra, Portugal
Creativity in natural language: studying lexical relations

4:50pm **Coffee Break**

5:00pm **Seco, N. and Veale, T.** University College Dublin, Ireland
The paradoxical role of similarity in creative reasoning

5:20pm **Peinado, F., Gervás, P. and Díaz-Agudo, B.** Universidad Complutense de Madrid.
A description logic ontology for fairy-tale generation

5:40pm **Pereira, F. C. and Gervás, P.** Universidade de Coimbra, Portugal
An automatic method for lexical semantics transformation

6:00pm **Round Table / General discussion**
Concluding remarks

Workshop Organisers

Tony Veale,
Department of Computer Science,
University College Dublin,
Ireland
Tony.Veale@UCD.ie

Francisco C. Pereira,
Department Engenharia Informáica,
Universidade de Coimbra,
Pólo II,
Portugal
Camara@dei.uc.pt

Amílcar Cardoso,
Department Engenharia Informática,
Universidade de Coimbra,
Pólo II,
Portugal
Amilcar@dei.uc.pt

Pablo Gervás,
Department Sistemas Informáticos y Programación,
Universidad Complutense de Madrid,
Spain
PGervas@sip.ucm.es

Workshop Programme Committee

Tony Veale,
Department of Computer Science, University College Dublin, Ireland

Francisco C. Pereira,
Department Engenharia Informáica, Universidade de Coimbra, Pólo II, Portugal

Amílcar Cardoso,
Department Engenharia Informáica, Universidade de Coimbra, Pólo II, Portugal

Pablo Gervás,
Department Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Spain

Diarmuid O'Donoghue,
Department of Computer Science, National University of Ireland Maynooth, Ireland.

Table of Contents

Author Index

Converting frames into OWL: Preparing Mikrokosmos for Linguistic Creativity

Pablo J. Beltrán-Ferruz, Pedro A. González-Calero, Pablo Gervás

GAIA - Applied Artificial Intelligence Group
Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
C/ Juan del Rosal, 8. 28040 Madrid
pablo@fdi.ucm.es, pedro,pgervas@sip.ucm.es

Abstract

Linguistic creativity requires a complex combination of explicitly declared knowledge and problem-specific inference processes. The COLIBRI CBR shell combines ontologies and description logics to develop CBR solutions to complex problems. The ontologies provide a number of ready-made resources that fulfill the need for explicit knowledge without the need to hand-craft it, whereas the description logics that underlie the shell can deal with complex inferences like instance classification or generalization. Currently there is a shortage of ontologies with sufficient linguistic coverage that can be formulated in the description logic formalism. The present paper describes an attempt to convert a frame-based lingusitic resource (Mikrokosmos) to the description logic formalism in order to make it available for further attempts at developing a language generation systems that can exhibit creative behaviour. This has been achieved for OIL and is currently in process for OWL, which is an evolution of hte standard. A description is provided of the structure of Mikrokosmos and its semantics. Then we provide a brief outline of the target resources (OIL and OWL) and the particular tools that we use to manipulate them (JCopernico, RACER, Protégé). The mapping used is outlined, and the expected applications of this resource to linguistic creativity are discussed.

1. Introduction

Linguistic creativity, of the kind involved for instance in the generation of poetry (Gervás, 2002), requires a complex combination of explicitly declared knowledge and problem-specific inference processes. This was empirically tested in the particular case of Spanish formal poetry in (Gervás, 2001a), where the shortcomings of a case-based reasoning system for this particular purpose could easily be traced back to the limitations of the knowledge representation and the lack of enough semantical information.

This led to a reformulation of the approach in a more knowledge intensive framework. The COLIBRI CBR shell (Díaz-Agudo and González-Calero, 2000), based on ontologies and description logics, was chosen. In this set up, the ontologies provide a number of ready-made resources that fulfill the need for for explicit knowledge without the need to hand-craft it, whereas the description logics that underlie the shell can deal with complex inferences like instance classifcation or generalization. A first attempt at reformulating the problem in this framework (Diaz-Agudo et al., 2002) still showed poor results, because the semantic information available to the system was still insufficient. Although the domain specific issues —a separate Spanish formal poetry ontology— were linked to the CBR shell through a CBR ontology (Díaz-Agudo and González-Calero, 2002), the system still lacked semantic information concerning the particular words that it was using as poetical vocabulary.

The philosophy of the COLIBRI system assumes that existing ontologies can be reused, but there is currently a shortage of ontologies with sufficient linguistic coverage that can be formulated in the description logic formalism. Attempts to convert WordNet (Miller, 1995) to this formalisms resulted in a very skeletal representation of the resource, due to the fact that text descriptions of meanings disappeared during the conversion, and the structural information encoded in the synset graph was insufficient to satisfy the requirements.

The present paper describes an attempt to convert a frame-based linguistic resource (Mikrokosmos) to the description logic formalism in order to make it available for further attempts at developing a language generation systems that can exhibit creative behaviour.

2. Resources Involved

Our interest in Mikrokosmos is to use the ontology for converting it to OWL in order to reason with it. First of all, we need to understand the structure of Mikrokosmos and its semantics. Then we provide a brief outline of the target resources (OIL and OWL) and the particular tools that we use to manipulate them (JCopernico, RACER, Protégé).

2.1. Ontology Languages, Description Logics, and Tools

In the last year ontology languajes have developed quickly and we have seen a lot of standards appear and disappear. One of these standars was OIL (Horrocks, 2000), which generated very high expectations arising from its promise to have some inference abilities. These expectations were not met because finally few of the envisaged functionalities were implemented.

Recently, a new standard has reached a high relevance because it really implements reasoning. This new standard is OWL (Bechhofer et al., 2004). Reasoning is implemented using JENA (McBride, 2000) and DIG interface (Sean Bechhofer and Crowther, 2003) in OWL DF version. There are two inference engines that implements DIG interface: RACER and FaCT [1].

JCopérnico is a tool for editing ontologies developed by two PhD students in the Universidad Complutense (see Figure 1). It was originally created for developing ontologies

[1] `http://dl-web.man.ac.uk/dig/`

in OIL and pass them to RACER, so we had to extend it for using Mikrokosmos. This tool was developed under the assumption that any concept represented in OIL - except concrete data types - has a translation into the description logic SHIQ (Horrocks, 2000) implemented by RACER system (Haarslev and Möller, 2003)

Protégé 2.0 is a new version of the Protégé 2000 system, was developped at Stanford University, that can manage ontologies in OWL language (Gennari et al., 2002). The original beta version has now become a stable release and there is an important of ongoing work devoted to improving it.

2.2. Mikrokosmos ontology

The Mikrokosmos project was originally an interlingual system for Knowledge-Based Machine Translation (KBMT) (Nirenburg, 1987) developed in the Computing Research Laboratory from New Mexico State University. Although KBMT was conceived for translation of domain specific texts, no further restrictions are imposed in the contents of the text. Therefore the creators of Mikrokosmos built a rich ontology that contains a lot of general concepts, more than 4.700 concepts that are connected with an average of other 14 concepts using attributes and relations (de Quesada, 2001).

KBMT is an expensive approach that requires a big effort on knowledge acquisition, and it has been considered impractical by some authors. For that reason, the creators of Mikrokosmos were specially concerned about developing real-size systems that would demonstrate the feasibility of their approach. Generating contents for the ontology was their first concern, while the use of a rigorous formalism for knowledge representation was not considered a priority (Moreno-Ortiz et al., 2002). In fact, we have not been able to find any paper where the exact formalism of Mikrokosmos ontology is described.

In Mikrokosmos, concepts are primitive symbols of a world model which includes objects, events and properties organized in a complex hierarchy of language-independent concepts. (See top hierarchy of Mikrokosmos in figure 2.) The concepts are constructed following super ordinates, or hyponymy relations (IS-A links). In addition to its organization into a taxonomy via IS-A links, the ontology contain numerous other links between concepts, such as links using properties (Lonergan, 2001). For example DECEMBER has a relation with WINTER using the property PART-OF-OBJECT.

Each concept that makes up the ontology is language independent and is represented using frames. For example we can see the frame for concept REPLACEMENT-FOR in Table 1.

This frame is saved in a text file using Spencer notation that is based on XML. There is another notation called Beale notation that is based on Lisp, but we will focus in Spencer notation.

In the XML based format we have the whole ontology represented in a list of RECORD entries. Definition of one CONCEPT requires one or more of these RECORD entries. Each entry contains four fields, that are: CONCEPT, SLOT, FACET, and FILLER.

The CONCEPT field can be filled by any *Name* of a concept of the ontology.

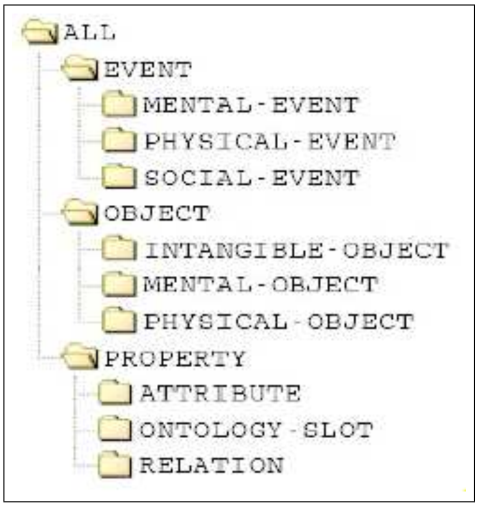

Figure 2: Mikrokosmos top hierarchy.

The second field in each entry is SLOT. This field can be filled with PROPERTY or any of its subclasses using IS-A links. There are two kind of *slot* fillers. One type are descendants of ATTRIBUTE and RELATION that represent links between concepts in the hierarchy. The other type are descendants of ONTOLOGY-SLOT. We will call them *special slots*, and all of them have the sense of determining the structure of the ontology. Possible descendants of ONTOLOGY-SLOT are: DEFINITION, DOMAIN, INSTANCES, INVERSE, IS-A, RANGE, SUBCLASSES and some others that are less important; later we will explain them in detail.

The third field is FACET, and it describes some finer distinctions between the possible fillers of the slot. Possibles FACETs are: VALUE, SEM, DEFAULT, INV, NOT, DEFAULT, DEFAULT-MEASURE and RELAXABLE-TO.

The last field is FILLER, and its value depends on the other fields, but generally it contains a *Name* of a concept of the ontology or an instance.

Initially we can think that there are no restrictions in these representation, but there are some special slots that limit expressiveness. All CONCEPT frames have non-special and special slots. Special slots for all kinds of concepts are

- DEFINITION: Definition in English for the concept.
- IS-A: It is used for asserting parents in the hierarchy.
- SUBCLASSES: It is used for listing concept children.
- INSTANCES, SPANISH1, ENGLISH1: They are only used in the leaves of OBJECT and EVENT. It contains words of the dictionary

Special slots which can only be present in all PROPERTY and only in PROPERTY concept frames are

- DOMAIN: It has fillers usually filled with EVENTs[2] and/or OBJECTs and it determines whether a CONCEPT can have it as a SLOT.

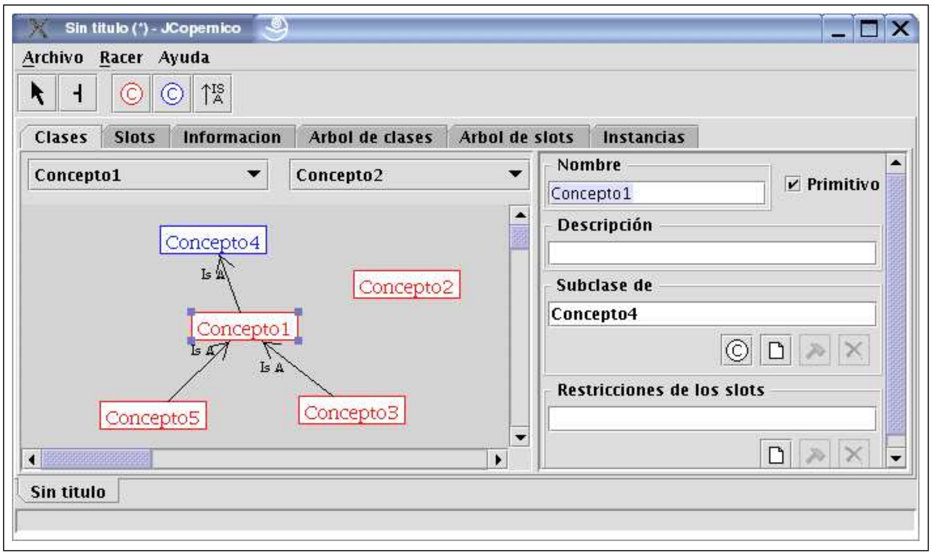

Figure 1: JCopérnico ontology editing tool.

Concept	***Slot***	***Facet***	***Filler(s)***
REPLACEMENT-FOR	DEFINITION	VALUE	”when x is a replacement for y”
	IS-A	VALUE	PHYSICAL-OBJECT-RELATION, EVENT-RELATION
	INVERSE	VALUE	REPLACED-BY
	DOMAIN	SEM	EVENT, OBJECT
	RANGE	SEM	EVENT, OBJECT

Table 1: Example frame: REPLACEMENT-FOR

- RANGE: It is used in RELATIONs and ATTRIBUTEs. In RELATIONs the RANGE slot has only the SEM facet. The fillers of the SEM facet are the names of concepts that are in the range of this RELATION. In ATTRIBUTEs the RANGE slot has only a VALUE facet. The VALUE facet is filled by all the possible literal or numerical values permissible for that ATTRIBUTE. The filler can also be a numerical range specified using appropriate mathematical comparison operators (such as $>$, $<$, ...).

- INVERSE: It is defined only for RELATIONs. It is mandatory for all RELATION frames. The INVERSE slot has only the Value facet which is filled by the name of the RELATION which is the Inverse of the given RELATION.

- MEASURED-IN: It is defined only for the descendants of the SCALAR-ATTRIBUTE concept frame. The MEASURED-IN slot is used to add a measuring unit for the number or scalar range that fills facets of the RANGE slot in SCALAR-ATTRIBUTE concept frames. The facet fillers of the MEASURED-IN slot are the daughters of the MEASURING-UNIT concept. The MEASURED-IN slot is used only in those SCALAR-ATTRIBUTE frames where MEASURING-UNIT has physical sense (e.g. for SIZE, AGE, etc.)

2.3. Description logics language: $\mathcal{SHIQ}$

Description logics (DLs) are a family of logical formalisms that originated in the field of artificial intelligence as a tool for representation of conceptual knowledge. Since then, DLs have been successfully used in a wide range of application areas such as knowledge representation, reasoning about class-based formalisms (e.g. conceptual database models and UML diagrams), and ontology engineering in the context of the semantic web. The basic syntactic entities of description logics are *concepts*, which are constructed from concept names (unary predicates) and role names (binary relations) using the set of concept and role constructors provided by a particular DL (Lutz, 2003).

Our interest in Mikrokosmos ontology is to map its contents to a description logics language. We have chosen $\mathcal{ALCQHI}_{\mathcal{R}^+}$ also known as $\mathcal{SHIQ}$ (Horrocks et al., 2000).

$\mathcal{ALC}$ comprises concepts —denoting sets— as well as roles —denoting binary relations. Unlike roles, concepts can be compound. Compound concepts are constructed by the following operators: intersection $\sqcap$, union $\sqcup$, complementation $\neg$ —taking concepts as arguments—, and the value restrictions $\forall$, and $\exists$ —taking a role and a concept as their arguments. Formally, $\mathcal{ALC}$ is given by the following formation rules, where c denotes a concept symbol and r a role symbol (Schild, 1991):

$$C, D \longrightarrow c \mid \top \mid C \sqcap D \mid \neg C \mid \forall R.C$$

[2] In this paper when we say a concept name in plural we are refering to this concept and his children, using links IS-A defined in the ontology.

$$R \longrightarrow r$$

$\mathcal{SHIQ}$ is the basic logic $\mathcal{ALC}$ augmented with qualifying number restrictions, role restrictions, role hierarchies, inverse roles, and transitive roles.

DL $\mathcal{SHIQ}$ is implemented in the RACER System (Haarslev and Möller, 2003). This makes it a desirable target representation for our ontology. For describing our ontology in $\mathcal{SHIQ}$ we will use the notation explained in Table 2, that contains denotational semantics for our language translation.

3. The Process of Conversion

The Mikrokosmos ontology is a rich and extensive knowledge resource, but use of a rigorous formalism for knowledge representation was not considered a priority during its development. It was developed with a proprietary formalism, and it is necessary to represent this knowledge into a more widely available formalism in order to use it for linguistic creativity. Our main task is to map all knowledge in Mikrokosmos ontology to a DL language.

First we map Mikrokosmos knowledge to $\mathcal{SHIQ}$. This provides us a representation in Description Logics of the ontology independent from a particular implementation.

Then, we implement a converter for this translation. We initially chose OIL as target language because it was a *de facto* standard in the field. This language is used by JCopérnico —a tool developed in our group. So we easily extended it to load the Mikrokosmos ontology.

Recently —in February— OWL has become a W3C recommendation. This decided us to use this language representation for our ontology, so our next step was to make a converter for this language. This converter was developed as a plugin for Protégé.

3.1. Mikrokosmos mapping to $\mathcal{SHIQ}$

Once, we have identified description logics language we want to use —$\mathcal{SHIQ}$— and we have described the Mikrokosmos ontology, we can proceed to map the latter into the former.

The first step is to determine whether a concept is a class or a slot. Although in the Mikrokosmos ontology everything is a concept we need to distinguish between Mikrokosmos concepts that correspond to unary predicates —which map to DL classes— and Mikrokosmos concepts that correspond to binary predicates —which map to DL relations. EVENT, OBJECT and all of their subclasses will be unary predicates so they will be classes. Meanwhile PROPERTY and all its hierarchy except ONTOLOGY-SLOTs (see figure 2) will be binary predicates so they will be slots. There are a few exceptions: concept ALL is **top** in description logics, and ONTOLOGY-SLOT and all of their subclasses are not mapped to DL language because they have the sense of structuring the ontology. ONTOLOGY-SLOT and all of their subclasses encode the structure of the Mikrokosmos ontology. They are not mapped as DL classes or slots. Instead they are incorporated into the DL definition of the Mikrokosmos concepts that they refer to.

Mikrokosmos has some information that can not be mapped to a DL language. We will face up to this problem in two ways. First we will make some annotations to class and slots that are not supported by DL language, but which could be provided by RDFS based languages. Second, extra information about slots that is not supported by DL language will be stored in special concepts created from the corresponding slots.

3.1.1. Building DL classes

Now we will discuss how we extract information stored in the XML based file to build classes in DL language.

The information that has to be extracted is:

class-def (**primitive** | **defined**) `CN`
  subclass-of $C_1 \ldots C_n$
  slot-constraint$_1$
  ⋮
  slot-constraint$_m$

Having identified the set of DL classes we need to identify their superclasses and *slot-constraints*. Information about superclasses is encoded in XML records of the form shown in Figure 3. Additional sources of information about superclasses —such as RECORDs where `CN` appears as FILLER and SUBCLASSES appears as SLOT— actually encode redundant information and are therefore discarded.

```
<RECORD>
    <CONCEPT> CN </CONCEPT>
    <SLOT>IS-A</SLOT>
    <FACET>VALUE</FACET>
    <FILLER> Ci </FILLER>
</RECORD>
```

Figure 3: XML encoding of superclass information

Information about *slot-constraints* is encoded in records having PROPERTYs as a slot. But there are also some ONTOLOGY-SLOT used in class definition and we will assign them a representation.

We collect information about *slot-constraints* from XML records of the form shown in Figure 4:

```
<RECORD>
    <CONCEPT> CN </CONCEPT>
    <SLOT> SN </SLOT>
    <FACET> FACET </FACET>
    <FILLER> C </FILLER>
</RECORD>
```

Figure 4: XML encoding for *slot-constraints*

We obtain different information depending on the value of *FACET*

[3] $\sigma(C)$ is the interpretation of a concept. Interpretation of a concept is the set of all individuals in the domain that satisfies description of the concept.

class-def (**primitive** | **defined**) `CN`	$\mathtt{CN}(\sqsubseteq \mid \doteq)\top$
subclass-of $C_1 \ldots C_n$	$\sqcap\sigma^3(C_1) \sqcap \ldots \sqcap \sigma(C_n)$
slot-constraint$_1$	$\sqcap\sigma($*slot-constraint*$_1)$
$\vdots$	$\vdots$
slot-constraint$_m$	$\sqcap\sigma($*slot-constraint*$_m)$
top | **thing** | **bottom**	$C \sqcup \neg C \mid C \sqcup \neg C \mid C \sqcap \neg C$
$(C_1$ **and** $\ldots$ **and** $C_n)$	$(\sigma(C_1) \sqcap \ldots \sqcap \sigma(C_n))$
$(C_1$ **or** $\ldots$ **or** $C_n)$	$(\sigma(C_1) \sqcup \ldots \sqcup \sigma(C_n))$
(**not** C)	$(\neg\sigma(C))$
(**one-of** $i_1 \ \ldots \ i_n$)	$(P_{i_1} \sqcup \ldots \sqcup P_{i_n})$
slot-constraint `SN`	$\top$
has-value $C_1 \ldots C_n$	$\sqcap \exists\, \mathtt{SN}.\sigma(C_1) \sqcap \ldots \sqcap \exists \mathtt{SN}.\sigma(C_n)$
value-type $C_1 \ldots C_n$	$\sqcap \forall\, \mathtt{SN}.\sigma(C_1) \sqcap \ldots \sqcap \forall \mathtt{SN}.\sigma(C_n)$
max-cardinality n C	$\sqcap \leq n\ \mathtt{SN}.\sigma(C)$
min-cardinality n C	$\sqcap \geq n\ \mathtt{SN}.\sigma(C)$
cardinality n C	$\sqcap \geq n\ \mathtt{SN}.\sigma(C) \sqcap \leq n\mathtt{SN}.\sigma(C)$
has-filler d	$\sqcap \exists\ \mathtt{SN}.\sigma(\mathrm{d})$
slot-def `SN`	
subslot-of $\mathtt{SN_1} \ldots \mathtt{SN_n}$	$(\mathtt{SN} \sqsubseteq \mathtt{SN_1}) \ldots (\mathtt{SN} \sqsubseteq \mathtt{SN_n})$
domain $C_1 \ldots C_n$	$\exists\ \mathtt{SN}.\top \sqsubseteq \sigma(C_1) \sqcap \ldots \sqcap \sigma(C_n)$
range $C_1 \ldots C_n$	$\top \sqsubseteq \forall\ \mathtt{SN}.\sigma(C_1) \sqcap \ldots \sqcap \sigma(C_n)$
inverse `RN`	$(\mathtt{SN}^- \sqsubseteq \mathtt{RN})(\mathtt{RN}^- \sqsubseteq \mathtt{SN})$
properties transitive	$\mathtt{SN} \in \mathtt{S}_+$
properties symmetric	$(\mathtt{SN} \sqsubseteq \mathtt{SN}^-)(\mathtt{SN}^- \sqsubseteq \mathtt{SN})$
properties functional	$\top \sqsubseteq \leq 1\mathtt{SN}$
disjoint $C_1\ C_2 \ldots C_n$	$(\sigma(C_1) \sqsubseteq \neg\sigma(C_2))$
covered C **by** $C_1 \ldots C_n$	$\sigma(C) \sqsubseteq \sigma(C_1) \sqcup \ldots \sqcup \sigma(C_n)$
disjoint-covered C **by** $C_1 \ldots C_n$	$(\sigma(C_1) \sqsubseteq \neg\sigma(C_2))$
	$(\sigma(C) \sqsubseteq \sigma(C_1) \sqcup \ldots \sqcup \sigma(C_n))$
equivalent $C\ C_1 \ldots C_n$	$(\sigma(C) = \sigma(C_1)) \ldots (\sigma(C_{n-1}) = \sigma(C_n))$
instance-of $i\ C_1 \ldots C_n$	$P_i \sqsubseteq \sigma(C_1) \sqcap \ldots \sqcap \sigma(C_n)$
related `SN` $i\ j$	$P_i \sqsubseteq \exists\ \mathtt{SN}.P_j$

Table 2: Denotational semantics for language definition

- If *FACET* = DEFAULT-MEASURE
  `CN` **slot-constraint** `SN` **value-type** `C` is added to the corresponding class definition.
- If *FACET* = DEFAULT. These information is stored as an annotation
- If *FACET* = INV. These information comes from another slot, that it is inverse to `SN`. There is no need to handle here this information because DL has automatic handling for such type of information.
- If *FACET* = NOT. This entry appears when we restrict inheritance of one SLOT in the hierarchy. Information contained in Mikrokosmos about these is affirmative information and negative information, DL only uses affirmative information to handle it, so we do nothing with this information.
- If *FACET* = RELAXABLE-TO. These information is stored as an annotation
- If *FACET* = SEM
  `CN` **slot-constraint** `SN` **value-type** `C` is added.
- If *FACET* = VALUE
  `CN` **slot-constraint** `SN` **has-value** `C` is added.

Additional information encoded in terms of records with ONTOLOGY-SLOTS —as slots—, must be handled and incorporated into the corresponding class definitions.

The ONTOLOGY-SLOTs to be identified are DEFINITION, SPANISH1 and ENGLISH1.

- If SLOT = DEFINITION. We will make an annotation in class definition.
- If SLOT = SPANISH1 or ENGLISH1. We create two SLOTs called SPANISH1 and ENGLISH1, so we can assert:
  slot-constraint $ENGLISH1$ **has-filler** `d`. [4]

[4] These slots encode cross indexing with lexical information. Another possible mapping would have been to add them as instances, but this would result in loss of this cross indexing information.

3.1.2. Building DL relations

Information required to build DL relations is encoded in XML records with ONTOLOGY-SLOTS in their SLOT field of the form shown in Figure 5

```
<RECORD>
    <CONCEPT> SN </CONCEPT>
    <SLOT>SLOT</SLOT>
    <FACET>FACET</FACET>
    <FILLER> X </FILLER>
</RECORD>
```

Figure 5: XML encoding of slot information

Possible relevant fillers of the ONTOLOGY-SLOTS are:

- DEFINITION, IS-A and SUBCLASSES: This information is handled for DL relations in the same way as for DL classes.
- INVERSE: It can be used with SEM and VALUE *FACET* and represents inverse slots.
  slot-def `SN` **inverses** `X` is added
- DOMAIN: As before when there is a restriction in inheritance Mikrokosmos asserts affirmative and negative information so there is a *FACET* NOT that is rejected, and has no translation to DL language. There are more possibilities for filling the FACET: VALUE, DEFAULT, RELAXABLE-TO and SEM, we make no distinction among them:
  slot-def `SN` **domain disjoint** $\mathtt{X}_1 \ldots \mathtt{X}_n$ is added.
- RANGE: FACET NOT is treated as above. When we have other FACETs there are two possible kinds of FILLERs: CONCEPTS or numeric ranges. For CONCEPTS
  slot-def `SN` **range disjoint** $\mathtt{X}_1 \ldots \mathtt{X}_n$
  is added. For numeric range we create a subclass of Numeric-Range (See Figure 6 and example in Figure 7).
- MEASURED-IN: This information is considered the same as RANGE. It can only have SEM or DEFAULT FACETs.
  slot-def `SN` **range** `X`
  is added.

3.1.3. Building Mikrokosmos PROPERTYs as DL classes

As we have seen in last subsection, not all information about PROPERTYs can be mapped easily to slots. Because of that we have decided to include an extra hierarchy of concepts created from PROPERTYs.

For each slot we will create a class that inherits from CLASS-SLOT called CLASS-<*PROPERTY-NAME*>. These classes contain all information about the PROPERTYs that we could not represent in a DL relations.

For each SLOT applied to a CONCEPT we will create a class that inherits from CLASS-SLOT-CONCEPT called

```
class-def primitive Numeric-Range
    slot-constraint Left-Range-Margin
        max-cardinality 1 int
    slot-constraint Right-Range-Margin
        max-cardinality 1 int

slot-def Numeric-Left-Margin
    range int

slot-def Numeric-Right-Margin
    range int

class-def defined Numeric-Right-Range
    subclass-of Numeric-Range
    slot-constraint Right-Range-Margin
        min-cardinality 1 int

class-def defined Numeric-Left-Range
    subclass-of Numeric-Range
    slot-constraint Left-Range-Margin
        min-cardinality 1 int

class-def defined Numeric-Closed-Range
    subclass-of Numeric-Right-Range
    subclass-of Numeric-Left-Range
```

Figure 6: Range definitions

```
<RECORD>
    <concept>VISCOSITY</concept>
    <slot>RANGE</slot>
    <facet>SEM</facet>
    <filler>(<; >; 0 1)</filler>
    <uid>256</uid>
</RECORD>

class-def VISCOSITY
    subclass-of Numeric-Range
    slot-constraint Left-Range-Margin
        has-filler 0
    slot-constraint Right-Range-Margin
        has-filler 1
```

Figure 7: Example of range restriction

CLASS-<*PROPERTY-NAME*>-<*CONCEPT-NAME*>. These classes have slot-constraints in order to define information not captured in the respective concept.

With this structure of classes we do not lose any information about slots and slot-constraints but almost all information stored in that way is not useful for reasoning in current tools like RACER (Haarslev and Möller, 2001).

3.2. Extending JCopérnico for managing Mikrokosmos

JCopérnico was originally created for creating and editing ontologies in language OIL. It also allows user to communicate JCopérnico with RACER. These features are desirable in our effort to use an extensive ontology in our applications —such as generation of poetry.

JCopérnico was developed using object oriented design patterns to ensure it would be easy to to add more functionalities to it in future. We will profit this and program a new functionality that enables JCopérnico to load XML based file containing Mikrokosmos ontology.

We have made a translation of Mikrokosmos ontology to a DL language —$\mathcal{SHIQ}$—, but JCopérnico works with definitions made in OIL. Our DL language $\mathcal{SHIQ}$ has a simple translation to OIL using (Horrocks, 2000). So now, we can implement a new functionality for JCopérnico that enables it to load Mikrokosmos ontology from an XML based file.

Once we have added this new functionality to JCopérnico we can profit its features. We can store Mikrokosmos ontology using instance OIL. And we can export it to RACER and reason with it.

3.3. MikroOWL: A plugin for Protégé 2.0

There have been a lot of standards of languages for ontologies but now there is a increasing interest in OWL because it implements reasoning using JENA and DIG interface in OWL DF version. Recently —in February— OWL has become a W3C recommendation.

There are several programs in Internet that convert OIL into OWL but we want to develop a plugin for Protégé because it would enable us to use our ontology with other plugins for Protégé and to profit newer features of OWL.

Our plugin in Protégé has been developed as an *import plugin* [5]. This kind of plugins provide us an extensible mechanism for importing Mikrokosmos ontology.

Protégé allows us to store Mikrokosmos ontology in OWL language, and also to export it to RACER.

4. Linguistic Creativity Applications

The work described in this paper was undertaken as a result of the conclusions obtained from previous work on the automatic generation of Spanish formal poetry (Gervás, 2001a). In that work, a CBR approach was applied to build new poems by reusing the structure of existing ones, while adapting the set of words actually filling that structure to fit a given user query. At each stage, the required word was selected solely based on its syntactic category and its relative position in either user proposal, case description or case solution. This works reasonably well for words originating from these sources, but not so well if additional vocabulary is employed. The obvious step to solve this problem was to provide the system with semantical information that could be taken into account when trying to adapt retrieved cases to user queries.

Having access to the sort of relations embodied in an ontology provides the means for finding the most suitable new words to employ during adaptation. Such words will have to fulfil certain requirements in terms of semantical relations both to the words they are replacing in the original poem, to the words provided by the user in his query, and to the words surrounding the position under consideration both in the retrieved case and in the draft that is being built.

The work so far on this line of research has concentrated on the construction of the actual resource, so no examples are available of the type of creativity to be expected. However, the applicability of such a resource in the process of generating poetry can be exemplified over a real example of adaptation of an existing song to fit new circumstances. In 1937 the American volunteers fighting in the Spanish Civil War adapted a well known folk song to fit their circumstances, precisely by reusing the structure of the original and modifying the words. For instance, the original verses...

> ... But remember the Red River Valley
> And the cowboy who loved you so true

where changed into:

> ... So remember the battle for Jarama
> And the people who set that valley free

Although the result of such a process of poem generation is not considered particularly creative, it does present interesting features in as much as the author builds not only on the interplay between the actual words he has chosen, but also on the interplay between them and those in the original lyrics —which are brought to the mind of the listener by the new song being set to the same original tune.

This sort of transformation of a given poem into a different one involves the establishment of a complex network of mappings between concepts in the two versions. A metaphorical association between love (the cowboy's love affair) and war (the battle for the Jarama Valley) is the basis of the transformation. Helping a CBR system to identify this type of relationship during case retrieval would be a major task in which a resource such as the one described here would be involved —though heuristic approximations based on word co-occurrence have proved to be acceptable in the past (Gervás, 2001b). As mentioned above, the main application of the resource and the various operations that its DL representation makes possible would be during adaptation.

Having selected such a case, adaptation requires, for instance, identifying 'the battle for Jarama' as a valid substitution for 'the Red River Valley'. Such a process would involve taking into account factual information about the battle, which took place for control of the Jarama valley, along which ran the last open road into besieged Madrid. While a knowledge base of facts might provide the basic data required, it is clear that to achieve the desired result both a set of semantical relations relating the concepts involved and an inference process capable of operating over them to detect relevant associations are major requirements.

The COLIBRI CBR shell (Díaz-Agudo and González-Calero, 2000) allows the combination of the complex inferences possible in DL —such as instance classification or generalization— with existing ontologies —such as the one described in this paper—, for easy configuration of CBR processes. This configuration is achieved by linking the reused ontology with the operational processes by means of a specific CBR ontology (Díaz-Agudo and González-Calero, 2002). The fact that CBR is used as part of the process instead of alternative algorithmic solutions leaves a certain room for actual creative behaviour on the part of the system.

[5] `http://protege.stanford.edu/doc/pdk/plugins/import_and_export_plugins.html`

5. Conclusions

The process of conversion is currently work in progress, and the material reported here is based on preliminary results. However, a few relevant details are already apparent. For instance, having an ontology that provides wide coverage may address some of the problems of earlier systems, but it also poses new problems in terms of restrictions on available memory. A simplified version of the Mikrokosmos ontology, involving only the hierarchy of concepts with little information about the relations that link them, takes up half a gigabyte of memory. This could signal that attempts to solve the creativity problem simply in terms of increasing coverage may be ill-advised. It is therefore crucial to find ways to supplement broader coverage with adequate inference processes that can bridge problematic cases where the explicitly available information is not enough to produce satisfactory answers.

A CBR approach to the application of ontological resources, such as the one advocated in this paper, would present great advantages over more algorithmic solutions. Although the examples described are focused on very specific problems identified in previous work, once the resource is operative enough to be linked to the proposed system, a number of wider alternatives for its application to linguistic creativity will be open. Some of the options already under consideration include the development of story plots, and its application to the automated direction of user interactions with a pre-authored plot in interactive narrative environments, where conflicts between authors intentions and user freewill would greatly benefit from creative solutions.

6. References

Bechhofer, Sean, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Andrea Stein, 2004. OWL Web Ontology Language Reference. W3C http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

de Quesada, Mercedes García, 2001. Estructura definicional terminografica en el subdominio de la oncologma clmnica. In *Estudios de Linguistica Espanola*, volume 14.

Diaz-Agudo, B., P. Gervás, and P P. Gonzalez-Calero, 2002. Poetry generation in COLIBRI. In S. Craw and A. Preece (eds.), *ECCBR 2002, Advances in Case Based Reasoning*. Springer. Lecture Notes in Artificial Intelligence.

Díaz-Agudo, Belén and Pedro Antonio González-Calero, 2000. An architecture for knowledge intensive CBR systems. In Enrico Blanzieri and Luigi Portinale (eds.), *Advances in Case-Based Reasoning – (EWCBR'00)*. Berlin Heidelberg New York: Springer-Verlag.

Díaz-Agudo, Belén and Pedro Antonio González-Calero, 2002. CBROnto: a task/method ontology for CBR. In *CBR Track (FLAIRS)*.

Gennari, J., M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubizy, H. Eriksson, N. F. Noy, and S. W. Tu, 2002. The evolution of Protégé: An environment for knowledge-based systems development. Technical report, Stanford University.

Gervás, P., 2001a. Automatic generation of poetry using a CBR approach. In *CAEPIA - TTIA 01 Actas Volumen I*. CAEPIA.

Gervás, P., 2002. Linguistic creativity at different levels of decision in sentence production. In G.A. Wiggins (ed.), *Proceedings of the AISB 02 Symposium on AI and Creativity in Arts and Science, 3rd-5th April 2002, Imperial College*. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Gervás, Pablo, 2001b. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems*, 14(3–4):181–188.

Haarslev, Volker and Ralf Möller, 2001. Description of the RACER system and its applications. In *Proceedings International Workshop on Description Logics (DL-2001)*. Stanford, USA.

Haarslev, Volker and Ralf Möller, 2003. *RACER User s Guide and Reference Manual Version 1.7.7*. Concordia University and Univ. of Appl. Sciences in Wedel.

Horrocks, I., 2000. A denotational semantics for standard OIL and instance OIL. Technical report, Department of Computer Science, University of Manchester, UK.

Horrocks, I., U. Sattler, and S. Tobies, 2000. Reasoning with individuals for the description logic SHIQ. In David MacAllester (ed.), *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, number 1831. Germany: Springer Verlag.

Lonergan, E., 2001. Lexical knowledge engineering: Mikrokosmos revisited. In *PACLING2001 - Pacific Association for Computational Linguistics 2001*. Kitakyushu, Japan.

Lutz, C., 2003. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*. King's College Publications.

McBride, Brian, 2000. Jena: Implementing the RDF model and syntax specification. Technical report, Hewlett Packard Laboratories, Bristol, UK.

Miller, G. A., 1995. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11).

Moreno-Ortiz, Antonio, Victor Raskin, and Sergei Nirenburg, 2002. New developments in ontological semantics. In *Proceedings of LREC-02, Spain, June.*.

Nirenburg, S., 1987. Knowledge-based machine translation, the CMU approach. In *Machine Translation: theoretical and methodological issues, Studies in Natural Language Processing*. Cambridge: Cambridge University Press.

Schild, Klaus, 1991. A correspondence theory for terminological logics: preliminary report. In *Proceedings of IJCAI-91, 12th International Joint Conference on Artificial Intelligence*. Sidney, AU.

Sean Bechhofer, Ralf Möller and Peter Crowther, 2003. The DIG description logic interface. In *Description Logics 2003*, CEUR Workshop Proceedings.

Lexical databases as resources for linguistic creativity: Focus on metaphor

Birte Lönneker

Institute for Romance Languages
Von-Melle-Park 6, 20146 Hamburg, Germany
birte.loenneker@uni-hamburg.de

Abstract

This paper discusses the shortcomings of current general-domain lexical databases, as well as their potential with respect to metaphor representation. By metaphor representation, we mean here a minimal set of relations inside the source domain of the metaphorical mapping and a relation between the source and target domain. A case study based on material from the Hamburg Metaphor Database, which combines data from corpora, EuroWordNet and the Berkeley Master Metaphor List, exemplifies the claims made in the paper.

1. Introduction

Current general-domain lexicons are of very restricted usefulness for creative systems that aim at understanding or creating metaphorical expressions. However, lexical databases like WordNet have the potential to become useful basic resources for metaphor representation, as will be shown in this paper. After recalling some basic notions on metaphor (Section 2.) and the general needs of systems for metaphor handling (Section 3.), the Hamburg Metaphor Database (HMD) will be presented in Section 4. It combines data from corpora, EuroWordNet and a freely available metaphor list. General results of the work on HMD as well as a case study based on HMD data (Section 5.) support the claims made at the beginning. Section 6. is the conclusion.

2. Metaphor as a form of linguistic creativity

Metaphor is probably one of the most widespread forms of linguistic creativity. At the same time, it is a phenomenon that occurs itself under many forms. One of the scales on which metaphor can be characterized is that of conventionality. At one end of the continuum, there are novel poetical and spontaneous metaphors, which are by definition of a very low frequency (example: *My horse with a mane made of short rainbows*, Navaho song cited by (Lakoff, 1993, 230)). At the other end, there are conventionalized metaphors that can be very common, and sometimes even difficult to replace by a non-metaphoric expression (example: *He* defended *his belief that the letters were genuine*). Before discussing the role lexical databases could play as resources for treating metaphorical creativity, it is therefore necessary to outline some basic theoretic assumptions and terminological distinctions. The subsections of this section will briefly

1. recall the cognitive basis of metaphor;
2. explain the notion of lexical metaphor; and
3. mention some types of novel metaphor.

2.1. Cognitive basis

One of the first researchers to notice the abondance of metaphor in common language use was Michael Reddy. (Reddy, 1979) shows that speakers of English use a large number of conventionalized metaphorical expressions when talking about communication: *to pack thoughts into words*, *the sentence was filled with emotion*, *hollow words*, *find good ideas in the essay*, *seal up meaning in sentences*, to mention but a few examples. The “story” told by these common expressions suggests that signal-entities like words or sentences are containers which directly hold “reified” mental and emotional content. Conventionalized metaphors like those discussed by Reddy are understood and produced by children already fairly early in life, as brought forward by (Feldman and Narayanan, forthcoming) who discuss similar examples like *to grasp an idea*.

According to (Feldman and Narayanan, forthcoming), metaphorical utterances are understood and reasoned on in terms of underlying “embodied” meaning: For example, *grasping an idea* is a simulation of a situation involving the body, like *grasping the salt container*. This is in line with the theory of Cognitive Metaphor introduced by (Lakoff and Johnson, 1980) and since then further elaborated by many scholars. According to Cognitive Metaphor theory, the primary basis of metaphor as a phenomenon is not language, but thought. The vast majority of metaphorical utterances like *to grasp an idea* rely on mental generalizations, which relate a conceptual source domain and a conceptual target domain. The target domain is understood and acted on in terms of the source domain. For example, the expression *to grasp an idea* makes use of the conceptual metaphor IDEAS ARE OBJECTS, in which OBJECTS are the source domain and IDEAS the target domain. In general, we can interpret those conceptual domains as follows:

- The **source domain** is a concept that is closer to basic concepts accessible by bodily experience, in a continuum of concepts. *Example:* OBJECT. Physical objects can be perceived visually, touched, and manipulated.
- The **target domain** is a concept that is closer to abstract concepts which cannot be immediately experienced, in the same continuum of concepts. *Example:* IDEA. An idea is an “abstract” object which cannot be immediately perceived by the senses.

2.2. Lexical metaphors as instantiations of conceptual metaphors

Individual metaphors are lexical instantiations of conceptual metaphors. For example, the figurative uses of the verbs *to pack* (as in Reddy's example *to pack thoughts into words*) and *to grasp* (as in *to grasp an idea*) are lexical instantiations of the conceptual metaphor IDEAS ARE OBJECTS. Usually, a single conceptual metaphor accounts for the metaphorical meanings of a number of different words belonging to the source domain: "[The] unified way of *conceptualizing* [a domain] metaphorically is realized in many different linguistic expressions." (Lakoff, 1993, 209)

Lexical metaphors can be encountered in everyday life conversations, in ordinary newspaper texts, and in many other text types including academic writing. According to (Martin, 1994), the frequency of lexical metaphors in a newspaper text can be estimated to about 4 to 5 words per 100. Not only all humans, but also most systems dealing with natural language will thus encounter metaphor.

2.3. Main types of novel metaphor

(Lakoff, 1993) distinguishes several types of novel lexical metaphors. Three main types will be briefly presented in what follows.

Lexical extension of conventional conceptual metaphors. Lexical metaphors of a higher degree of creativity, and having at the same time a high potential of "success" in terms of comprehensibility, are those that extend the set of conventionally mapped lexical items inside the source domain. A process aiming at producing a narrative of any kind could start out using some conventionally mapped lexical items of a selected source domain and continue using lexemes from the same source domain that are usually not encountered in a metaphorical sense. In fact, humans do creatively produce such metaphors. For example, the conceptual metaphor THEORIES ARE CONSTRUCTED OBJECTS shows conventionalized lexical mappings in the sentence *He is trying to buttress his argument with a lot of irrelevant facts, but it is still so shaky that it will easily fall apart under criticism*. A creative, but comprehensible lexical extension of this conceptual metaphor is exemplified in the sentence *Your theory is constructed out of cheap stucco*.

Image metaphors. Image metaphors are singular metaphors that most often map only one image onto another image, for example the image of an hour-glass onto the image of a woman. They do not refer to conventional conceptual metaphors, in which many elements of the source domain can be mapped onto many corresponding concepts in the target domain, nor do they establish new conventional metaphorical mappings. While in most image-metaphors aspects of a part-whole structure are mapped onto aspects of another part-whole structure, also other aspects like attributes can be mapped. In the example *My horse with a mane made of short rainbows* discussed above, the colorfulness and beauty of the object in the source domain (rainbow) are mapped onto the object in the target domain (mane).

Analogies. Analogies like the famous example *my job is a jail* usually make use of several well-established conceptual mappings. Understanding the expression *my job is a jail* involves the processing of the independently existing metaphors GENERIC IS SPECIFIC, PSYCHOLOGICAL FORCE IS PHYSICAL FORCE, and ACTIONS ARE SELF-PROPELLED MOVEMENTS (cf. (Lakoff, 1993)).

(Lakoff, 1993, 231) suggests as a generalization that a certain structure is preserved in the metaphorical mapping of all metaphors, whether conventional, image metaphor or analogy. He calls this structure *image-schema structure*, where parts are mapped onto parts and wholes onto wholes (as in the hourglass-woman example), containers onto containers as in the IDEAS ARE OBJECTS example, and so on. As (Green, 2002) further elaborates, it seems to be the most important to map the system of *relationships* present in the source domain. It is interesting to note that while Lakoff argues that "symbol manipulation systems cannot handle image-schemas" (Lakoff, 1993, 249), most NLP or AI systems aiming at the processing of metaphors do exactly that (among other tasks like inferencing): They try, in one way or the other, to *represent the structure* of the source and target domains.

3. Metaphor in NLP and resource building

As outlined above, conventionalized metaphor is an everyday issue. Most systems dealing with NLP have to face it sooner or later. A successful handling of conventional metaphor is also the first step towards the processing of novel metaphor. However, only very few systems have been designed with special attention to metaphor handling. Some examples of these, and the implications behind them, are mentioned in Subsection 3.1. Subsection 3.2. then addresses general aspects of resource building that are related to metaphor.

3.1. Processing of metaphor

Obvious problems for NLP systems caused by lexical metaphors consist in the incompatibility of metaphorically used nouns as arguments of verbs. In systems which constrain the type of arguments for every verb by semantic features like `human`, `living`, `concrete` or `abstract`, metaphors can cause inconsistencies that will have to be solved. For example, if the grammatical subject of the English verb *go* was restricted to entities classified as `living` in a given system, the following sentence (1.) taken from (Hobbs, 1992) could not be parsed.

(1) The variable N goes from 1 to 100.

Obviously, there is an open-ended number of such sentences. In fact, there have been many attempts to increase the ability of systems to deal with incompatibilities of this kind, caused by instantiations of conceptual metaphors. In most cases, a representation of at least a part of the conventionalized mapping is encoded in the system. Those systems can be called *knowledge-based* systems; they "leverage knowledge of systematic language conventions in an attempt to avoid resorting to more computationally expensive methods" (Martin, 1994). Those systems generally reason in the source domain and transfer the results back to the target domain using the provided mapping; this procedure is applied, for example, in KARMA's networks (Feldman and Narayanan, forthcoming) or in the rules of TACITUS (Hobbs, 1992) and ATT-META (Barnden and Lee, 2001).

As (Martin, 1994) points out, one of the problems for knowledge-based systems with integrated metaphor handling is the acquisition of sufficient and suitable knowledge. Still nowadays, systems like KARMA and ATT-meta have to be provided with knowledge by the users. It would therefore be useful to provide more knowledge about metaphor in lexical resources, which could be either directly used in NLP systems, or used as a basis for building rules and networks in systems designed especially for metaphor handling. If well-studied linguistic knowledge supported by attestations in corpora was encoded in lexical resources, they could also be regarded as a common starting point for different systems, and the results of the systems would become more directly comparable.

3.2. Resource building

One of the reasons why general lexical resources have not been used as input for metaphor processing systems are their fine-grained and sometimes arbitrary sense distinctions (Martin, 1994). In order to overcome the lack of knowledge in his MIDAS system, (Martin, 1994) therefore built the MetaBank database, which is independent from any other lexicon or knowledge base. MetaBank contains mappings of single lexical metaphors like *enter* or *kill*. The grammatical objects of these lexical items can refer to containers (source) or processes (target), among others: *enter a computer program*, *kill a process*. For *both* senses of the words (in the source and target domain), representations of the meaning are built (including for example the superconcept as well as type restrictions for result, victim, and actor), which are then mapped by so-called metaphor maps (Martin, 1992). For building the knowledge base used in MIDAS, (Martin, 1994) and his colleagues analysed three sources:

1. the Berkeley Metaphor List, the latest version of which is (Lakoff et al., 1991), to be discussed in more detail in Subsection 4.2.,

2. a specialized corpus from the computer domain containing questions and answers about UNIX;

3. (probes of) a newspaper corpus (three years of the Wall Street Journal).

The Berkeley list helps to perform directed searches (for example, on container metaphors), as (Martin, 1994) exemplifies using the computer corpus. He points out that an exhaustive metaphor analysis of general corpora would be most fruitful for building knowledge resources, but is not feasible with a large collection. Therefore, (Martin, 1994) analyses six newspaper samples of each about 100 sentences. One of his empiric insights is that a relatively small number of general conventional metaphors accounts for a high number of lexical metaphors, according to frequency counts. The consequence of these findings is that it is worthwhile to undertake a thourough study of metaphor domains that are lexicalized with a high frequence in general corpora, because a better representation of those areas could help many systems.

4. The Hamburg Metaphor Database

With respect to resources for linguistic creativity focusing on metaphor, we can summarize the discussion of the preceding section as follows:

1. Knowledge acquisition is a preprequisite for metaphor handling programs.

2. General lexicons could provide knowledge and at the same time be a point of comparison for various systems, but they are currently built with too little attention to metaphor.

In order to both show the potential and tackle the shortcomings of current lexical and conceptual resources for the processing of metaphor, the Hamburg Metaphor Database (HMD) was created. Our basic task is to annotate French and German attested example sentences and phrases containing lexical metaphors, using EuroWordNet as a lexical resource and the Berkeley Master Metaphor List as a conceptual resource. In the following subsections, we briefly present those two resources (Subsections 4.1. and 4.2.) as well as our corpora (Subsection 4.3.) and the annotation methodology (Subsection 4.4.). Subsection 4.5. indicates the current status of the database. For a more detailed description of HMD, see (Lönneker and Eilts, 2004).

4.1. Wordnets

For the lexical data used in annotation, we refer to the French and German EuroWordNet lexical databases. EuroWordNet (Vossen, 1999) was a European project aiming to build a multilingual database along the lines of the Princeton WordNet (Miller, 1990). The data of the English wordnet situated in Princeton can be freely queried and obtained via the WordNet website[1], while EuroWordNet has to be acquired from the ELRA/ELDA agency[2] against a fee.

The basic notions of WordNet are those of synset and relation. A *synset* is a set of synonyms or near-synonyms referring to the same concept. For example, in WordNet 2.0 the verbs *to tumble* and *to topple* both refer to the concept of ‘falling down, as if collapsing’ and are therefore grouped in the synset {tumble, topple}. The synonyms inside a synset are called *variants* or *literals* of that synset. A *relation* characterizes the way in which two synsets are connected. Most relations in wordnets pertain to semantic or conceptual-semantic relations like subsumption and part-whole-relation (usually called *meronymy* in lexical resources) and thus hold between synsets as a whole. There are also some relations that exist only between variants.

Especially now that WordNet data and common ontologies are being matched onto each other (Pease and Fellbaum, 2003; Kiryakov and I..Simov, 2000), WordNet data will get more and more accessible to knowledge-based systems. In order to illustrate why wordnets might help directly or indirectly in metaphor processing, let us consider the relevant synsets for the words in italics in the following example sentences taken over from (Martin, 1994).

[1]URL: http://www.cogsci.princeton.edu/˜wn [12 April 2004]
[2]URL: http://www.elda.fr [14 April 2004]

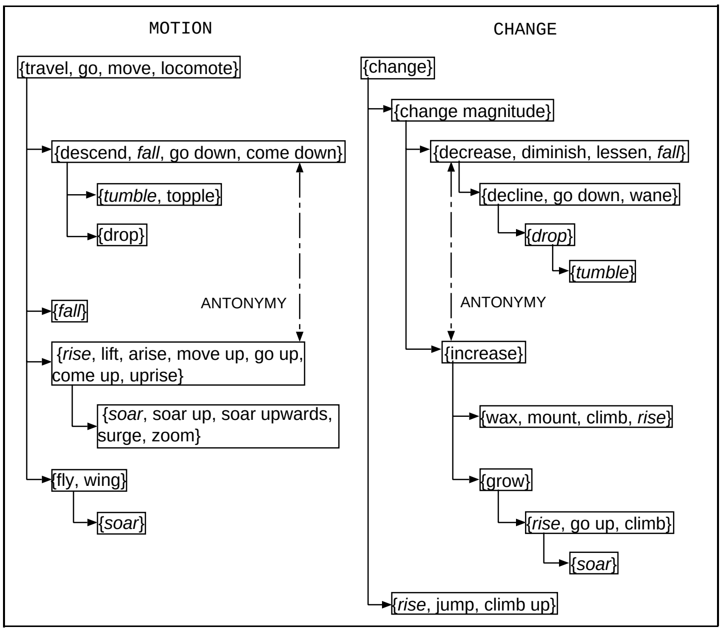

Figure 1: A sample of polysemic motion verbs in WordNet 2.0 hyponym hierarchies.

(2.) The Financial Times 30-share index *tumbled* 34.9 to 1822.9.

(3.) While net income doubled to an estimated $9.2 million during the nine months ended July 31, profit margins *fell* to 3.9% from 5.8% a year earlier, as general expenses *soared* nearly sevenfold.

(4.) Over the course of the latest recession of 1981-82, service-industry employment *rose* about 200,000, against a 2.7 million *drop* in goods-producing jobs; even so, overall unemployment *soared* to nearly 11% from about 7%.

(Martin, 1994) considers these sentences as illustrating the conceptual metaphor VALUE-CHANGE IS MOVEMENT, which we can interpret as being a sub-metaphor of CHANGE IS MOTION. Figure 1 shows excerpts from WordNet 2.0 hyponym hierarchies illustrating the polysemic motion verbs appearing in example sentences (2.) to (4.); these verbs are indicated in italics. It can be seen from the figure that the domains of MOTION and CHANGE both contain two main opposing components (concepts), rendered in WordNet as synsets related via the antonymy relation. The selected areas of the WordNet lexicon are thus *structured in a similar way* in both domains. The Hamburg Metaphor Database therefore refers to synsets of the French and German EuroWordNet when annotating corpus examples.

However, as all lexical resources, WordNet and EuroWordNet have shortcomings. One of those pertaining to metaphor is the absence of relations showing the literal-figurative (or source-target) connection between any of the concepts in the two domains, from which a system could infer the meaning of further related or subconcepts. Section 5. will discuss more shortcomings and possibilities to overcome them.

4.2. List of conceptual metaphors

For describing metaphors at the conceptual level, we use the Berkeley Master Metaphor List (Lakoff et al., 1991) as a reference. The list can be queried online and is freely available at the Conceptual Metaphor homepage[3]. It describes mappings between conceptual domains that underly lexical metaphors, illustrated in English example sentences. An example of such a mapping is the IDEAS ARE OBJECTS metaphor, for which the list holds the following examples, among others:

(5.) Sally gave the idea to Sam.

(6.) Sally took the idea from Sam.

(7.) Sally put the idea aside.

Domain names in the Hamburg Metaphor Database try to reflect the titles of the conceptual metaphors as closely as possible. For example, IDEAS ARE OBJECTS would be entered as follows: IDEA is the name of the target domain, OBJECT is the name of the source domain.

4.3. Corpora

The examples that we annotate stem from written text or transcribed speech corpora. The corpora focus on certain domains and contents (for instance, on a political event like the creation of the Euro currency) and are collected from the mass media as material for master theses dealing with metaphorical language. The theses have been written since the 1990s at the Institute for Romance Languages under the supervision of Wolfgang Settekorn. Most of them cover comparable French and German text corpora dealing with the same event. They do usually not provide the entire texts, but selected and classified examples. The evaluation and annotation of this material in the Hamburg Metaphor

[3]URL: http://cogsci.berkeley.edu/ [13 April 2004]

Database is completely independent from the production and evaluation of the theses.

4.4. Methodology

For each metaphorically used lexeme in the example sentences, we try to find entries in EuroWordNet. Synsets containing the lexeme are entered as either metaphorical or literal, according to the meaning in which it is understood. If we find both literal and metaphorical synsets, we enter both of them in the corresponding fields of the database. This work is not as straighforward as it sounds, due to several problems like scarse or unclear glosses, unclear hierarchies and literal-figurative inconsistencies (cf. Section 5.1.). We also identify the conceptual mapping underlying each lexical metaphor. If it is already listed in the Master Metaphor List, the names of source and target domain are taken over. As not all domain mappings are actually present in the Master Metaphor List, HMD also uses a parallel German naming system for conceptual domains. Most of them are translations of the English domain names from the Metaphor List, some of them are specializations of these, and still others are mappings that are not covered in the Master Metaphor List.

4.5. Current status

At the time of this writing, the Hamburg Metaphor Database contains more than 400 entries covering 308 metaphorically used lexemes. The examples stem from corpora collected in ten different master theses. The metaphor database is accessible via the HMD project webpage[4] and can be freely queried according to French and German EuroWordNet synsets, domain names in German and English, or titles of the master theses.

5. Results

This section is subdivided into two parts. Subsection 5.1. contains a general evaluation of EuroWordNet and WordNet with respect to metaphor repreentation. Subsection 5.2. presents the results and consequences of a case study conducted on HMD data.

5.1. EuroWordNet evaluation

When building or querying the Hamburg Metaphor Database, it turns out very fast that the French and German lexical networks included in EWN have a rather low coverage even of some conventionalized metaphors. In part, this fact can be explained by the fairly low general coverage. For example, the German part of EuroWordNet contains 15,132 synsets, while the further developed GermaNet[5] contains 41,777 synsets. The data in HMD can therefore be used directly in order to fill gaps on the synset level in the existing networks. We provide a list of missing EWN data and some comments on apparently erroneous EWN data on the Hamburg Metaphor Database webpage.

At this point, it might be appropriate to come back to Martin's criticism of lexical resources, especially as far as their fine-grainedness and arbitrariness are concerned. As a result of our research, I discovered a phenomenon that I called *literal-figurative inconsistency* (Lönneker, 2003), which might be one of the "disturbances" causing the perceived arbitrarity of lexicon entries. A literal-figurative inconsistency is caused by the subsumption of source-domain concepts (referred to by lexical items) under concepts belonging to the target domain, or vice versa. It can also consist in other semantic relations (for example, part-whole relations) between concepts of two distinct domains, as discussed in (Lönneker, 2003). In a weaker form, also the attribution of "wrong" example sentences is a literal-figurative inconsistency, and it might lead to a low performance of systems for word sense disambiguation that make use of these sentences.

To give an example in English, let us consider the definitions in examples (8.) to (10.) from WordNet 2.0, describing a small hyponym hierarchy of motion concepts. Example (8.) defines the most general concept in this hierarchy and (10.) the most specific one.

(8.) {travel, go, move, locomote} – (change location; move, travel, or proceed; "How fast does your new car go?"; "We travelled from Rome to Naples by bus"; "The policemen went from door to door looking for the suspect"; [. . .])

(9.) {descend, fall, go down, come down} – (move downward and lower, but not necessarily all the way; **"The temperature is going down"; "The barometer is falling"**; "The curtain fell on the diva"; "Her hand went up and then fell again")

(10.) {tumble, topple} – (fall down, as if collapsing; "The tower of the World Trade Center tumbled after the plane hit it")

While the concepts are clearly taken from the domain of physical motion, the two example sentences that are in bold face illustrate concepts from a target domain of MOTION, which can be most generally named CHANGE (here in the form of a change of numerical values). Similar literal-figurative inconsistencies exist not only in EuroWordNet and WordNet, but also in other general lexical resources.

5.2. Case study

In this section, results of a case study conducted on data from the Hamburg Metaphor Database will be presented. The following aspects will be treated separately: Domain-internal relations (Subsection 5.2.1.), world-knowledge relations (Subsection 5.2.2.), and inter-domain relations (Subsection 5.2.3.).

5.2.1. Domain-internal relations

As discussed above (Subsection 3.1.) AI systems with metaphor handling usually perform most of the reasoning in the source domain (Barnden, 2004), just like (supposedly) humans (Feldman and Narayanan, forthcoming). One of the first goals of lexical databases that include metaphor information should thus be to provide adequate relations

[4]URL: http://rrz.uni-hamburg.de/metaphern/index_en.html [12 April 2004]

[5]http://www.sfs.nphil.uni-tuebingen.de/lsd/ [29 February 2004]

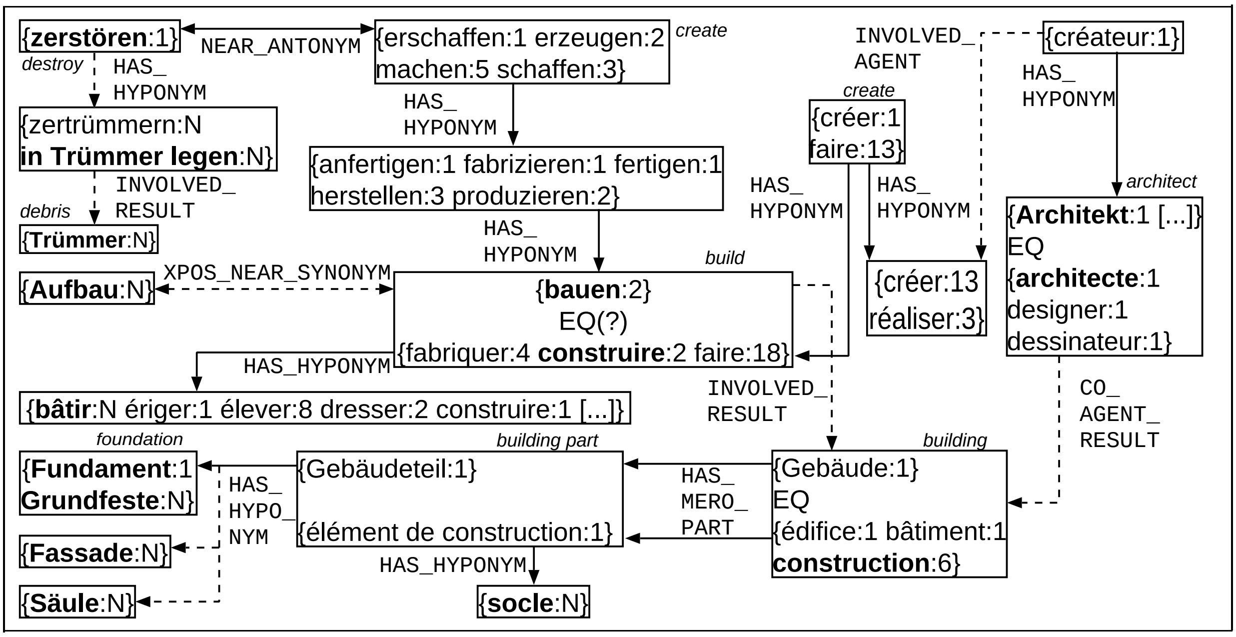

Figure 2: Current EWN relations and potential new relations in the BUILDING source domain.

between concepts *inside* the source domain, as those relations could be used by systems and humans in order to constitute domains and to infer possible metaphorical uses of lexical items (variants) referring to source domain concepts (synsets).

In wordnets, different sense relations can be represented between synsets. In order to illustrate how these sense relations help structuring conceptual domains, the remainder of this subsection will present a case study of HMD data for the source concept BAUWERK (‘BUILDING’). The examples found in the Hamburg Metaphor Database can be classified as illustrating the ORGANIZATIONS ARE STRUCTURED OBJECTS metaphor identified by (Martin, 1994), referring to States as a special kind of organization. In the Berkeley Master Metaphor List (cf. Subsection 4.2. above), this conceptual metaphor is not listed. It can however be subsumed under CREATING IS MAKING, which is also found as annotation in the Hamburg Metaphor Database fields for Berkeley domain names in most of the examples.

Figure 2 shows both the current state of relations in the actual EuroWordNet databases (solid lines) and a potential, more complete representation of the conceptual source domain BUILDING using more instantiations of existing EuroWordNet relations (dashed lines).

The BUILDING event is probably the most central concept of the source domain and lexicalized by a high number of metaphors in HMD.[6] As a first distinction, concepts belonging to the source domain can be subdivided into three groups with the following working definitions:

1. The central event and concepts that are directly linked to it are primary concepts.

2. Non-primary concepts that are represented in lexical metaphors and that are linked to primary concepts by actual or potential sense relations are secondary concepts.

3. Further concepts that are represented in lexical metaphors are tertiary concepts.

As **primary concepts** of the BUILDING domain, we can so far identify three concepts, based on lexical metaphors in the Hamburg Metaphor Database (EWN synsets are indicated in brackets, where applicable):

1. the BUILDING event, probably the most central concept of the source domain, lexicalized in: Ge. *bauen* ‘to build’ ({bauen:2}), *Aufbauphase* containing *Aufbau* ‘construction, assembly’; Fr. *bâtir* ‘to build’, *construire* ‘to construct’ ({fabriquer:4 construire:2 faire:18}), *édification* ‘construction, assembly’

2. the PRODUCT of the BUILDING event, lexicalized in: Fr. *construction* and *édifice* ‘construction, builing’ ({édifice:1 bâtiment:1 construction:6})

3. the contrary of the BUILDING event (i.e. DESTRUCTION), lexicalized in: Fr. *ébranlement* ‘stroke (causing vibrations)’ {dommage:3 mal:8 ébranlement:1}; Ge. *zerstören* ‘destroy’ ({zerstören:1}), *in Trümmer legen* ‘to ruin’

As **secondary concepts** of the BUILDING domain, we can so far identify two concepts, based on lexical metaphors in HMD:

1. PARTS of the PRODUCT of the BUILDING event, lexicalized in: Ge. *Fassade* ‘facade’, *Fundament* and *Grundfeste* ‘foundation’ ({Fundament:1}), *Säule* ‘column, pillar’; Fr. *socle* ‘foundation, plinth’, *pierre*

[6] An event can be represented by a noun or a verb. EWN accounts for this by the top-level category of 2ndOrderEntities, which correspond to events and states.

‘stone’[7]

2. the PRODUCT of the contrary of the BUILDING event, lexicalized in: Ge. *Trümmerfeld* containing *Trümmer* ‘debris’ (see also *in Trümmer legen* above).

The ACTOR of the PRECEDING EVENT (planning, designing) of the BUILDING event, lexicalized as Fr. *architecte* ‘architect’, has to be classified as a tertiary concept according to HMD data, as no metaphorical lexicalizations of the intermediate designing event have been documented.

The minimal internal structure of the BUILDING domain consists of the enumerated concepts and appropriate relations between them. Figure 2 reveals that the current EurWordNet database does not yet reflect that structure. This shortcoming is due to the fact that EuroWordNet focuses on hyponymic links and that it is (at least in the domains and languages studied in HMD) quite poor in other language-internal sense relations. The figure shows that EuroWordNet currently accounts only for the relations between the central event its contrary (following the hyperonym and antonym links), and for the relations between the event product and its parts.

Unfortunately, relations involving events, which are however available in EuroWordNet (for example, INVOLVED relations and CO_ROLE-relations), have been only very scarsely encoded, and are totally missing from the studied BUILDING domain. Basically, the INVOLVED relation links different types of participants to events, and the CO_ROLE relation links different participants of an event to each other, if certain semantic tests are fulfilled. For more information on EuroWordNet relations, cf. (Vossen, 1999). The insertion of INVOLVED and CO_ROLE relations for the BUILDING domain (cf. dashed lines in Figure 2) show that they could indeed help to reflect more of the domain-internal structure.

5.2.2. From lexical relations to world knowledge

The available sense relations in EWN could be used to cover the minimal structuring of the BUILDING source domain, as illustrating in the preceding subsection. However, it seems that there are also relations of a different type, not any more predictable from the meaning of the words or checkable using semantic tests, and that also those relations structure the domain and provide useful knowledge for metaphor creation and interpretation. The relations might be called “typical relations” or “world-knowledge relations”. They could in fact contribute to a higher density of relations inside the domain, which would show more clearly where the domain “ends” (as, of course, also concepts that are at the margins of Figure 2 can still be related to other concepts via sense relations). For example, adequate relations for interlinking the following concepts would be welcome:

- STONE and HOUSE. A house does not necessarily “consist of” stone, and stone is not necessarily used to build houses. If the relation was encoded as a “typical” relation (for example, bearing a new EuroWordNet relation feature like `typical`), it would still be difficult to find the right kind of meronymy subrelation in the EuroWordNet set.
- DEBRIS and BUILDING. There is a quite strong world knowledge connection between these two concepts. Even if debris might not necessarily be caused by destruction of buildings, certain subtypes of building destruction necessarily produce debris.
- ARCHITECT, DESIGN, and BUILDING. Architect and house can be linked by a CO_AGENT_RESULT relation as in Figure 2, because the word *Gebäudearchitekt* referring to the two participants of the “designing event” is lexicalized in German. However, the representation of a concept for the designing of buildings would also be needed, as well as appropriate relations between all three concepts (for example, designing is the typical activity of an architect).

Finally, the source domain element for the lexical metaphor *architecture*, encountered in HMD for German and French, but not discussed so far here, would have to be integrated into the source domain representation. Appropriate world-knowledge relations should link ARCHITECTURE to the concepts of ARCHITECT and HOUSE.

Already now, the set of EWN relations is not limited to strictly lexical-semantic links with the constraint of being always true; for example, relations might be `non-factive` or `reversed`. The integration of a level of “typical” or “world knowledge relations” would turn lexical databases like wordnets more and more into common-sense knowledge bases. Whether this is useful or not depends on the applications that use the database, and on the extent to which they use further, external knowledge resources.

5.2.3. Inter-domain relations

In order to indicate the system how the concepts of the source domain could be understood metaphorically, a link to the target domain is necessary.

A single mapping between the source domain synset {bauen:2} representing the BUILDING event and the an existing or new target domain synset representing the CREATING event could thus be used by a system as a starting point to construct the target domain. For a small number of lexical metaphors, parallel synsets in the source and target domain could be identified in EuroWordNet and are entered as such in the Hamburg Metaphor Database, as explained in Subsection 4.4. above. Instead of a structure-mapping algorithm (Falkenhainer et al., 1989; Veale, 1998), a *structure-completion algorithm* would be needed after the initial mapping is known. The task of the process would be to build the target domain starting from the concepts (synsets) of the source domain for which attested metaphorical usages exist. It would then map also the intermediate concepts, retaining the structure of the source domain.

Source-domain concepts for which no attested lexical metaphors exist are nevertheless candidates for the mapping if they are related to the central starting concept in

[7] Note that *stone* is in a different relation to *house* than, for example, *foundation*.

the same way as concepts with attested metaphorical lexicalisations. They might be used as uncommon or novel metaphors. Let us consider the example of Fr. *socle* 'foundation', which is the only lexical metaphor of our case study sample not documented as such in the huge corpus-based French dictionary *Trésor de la langue fran caise*[8], and therefore maybe less conventionalized than the others.

(11.) l'Allemagne avait voulu, au lendemain de la guerre, construire la République féd'erale sur un socle de probité, de transparence et de respect absolu de la Constitution

If the lexical database included this lexical item either as a synonym (variant) in a known source domain synset or in a synset that was linked to the central synset in the same way as other source domain synsets, a metaphor-handling system could detect the possibility of a metaphorical meaning of the word, and infer what kind of entity the metaphor refers to. An enhanced wordnet-like lexical database could therefore show candidates of novel lexical metaphors both to human writers and to AI systems.

6. Conclusion

The Hamburg Metaphor Database shows that lexical databases like EuroWordNet could and should contain more conventionalized lexical metaphors. An in-depth case study of the source domain BUILDING in HMD indicates also that these databases could convey important information on the internal conceptual structure of the mapping domains. However, while many relations that would contribute to that aim are already available in EuroWordNet, they have been much too scarsely encoded to make it immediately useful as a knowledge resource for metaphor-handling systems. If the proposed relations were indeed encoded, the EuroWordNet database could be used as a basis for understanding conventionalized metaphors as well as novel metaphors extending existing mappings, and eventually also as a basis for the interpretation of analogies and image metaphors.

7. Acknowledgements

I would like to thank Carina Eilts and Jan Dirks for annotating the material of the Hamburg Metaphor Database, as well as for insightful discussions. My thanks also go to Antonietta Alonge and two anonymous reviewers for comments on the paper abstract.

8. References

Barnden, J. A., 2004. Implications of an AI Metaphor Understanding Project. In *Proceedings of the Second International WordNet Conference – GWC 2004*. Brno, Czech Republic.

Barnden, J. A. and M. G. Lee, 2001. Understanding open-ended usages of familiar conceptual metaphors: an approach and artificial intelligence system. CSRP 01-05, School of Computer Science, University of Birmingham.

Falkenhainer, B., K. D. Forbus, and D. Gentner, 1989. The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41:1–68.

Feldman, J. and S. Narayanan, forthcoming. Embodied meaning in a neural theory of language. *Brain and Language*.

Green, R., 2002. Internally-Structured Conceptual Models in Cognitive Semantics. In R. Green, C. Bean, and S. H. Myaeng (eds.), *The Semantics of Relationships*. Dordrecht et al.: Kluwer, pages 73–89.

Hobbs, J. R., 1992. Metaphor and abduction. In A. Ortony, J. Salck, and O. Stock (eds.), *Communication fromm an Artifical Intelligence Perspective: Theoretical and Applied Issues*. Berlin: Springer, pages 35–58.

Kiryakov, A. and K. I..Simov, 2000. Mapping of EuroWordNet top ontology to Upper Cyc ontology. In *Proceedings of the EKAW-2000 workshop on Ontologies and Text*. Juan-les-Pins, France.

Lakoff, G., 1993. The contemporary theory of metaphor. In A. Ortony (ed.), *Metaphor and thought. Second edition*. Cambridge: Cambridge University Press, pages 202–251.

Lakoff, G., J. Espenson, and A. Schwartz, 1991. Master metaphor list. Second draft copy. Technical report, Cognitive Linguistics Group, University of California Berkeley. http://cogsci.berkeley.edu.

Lakoff, G. and M. Johnson, 1980. *Metaphors we live by*. University of Chicago Press.

Lönneker, B. and C. Eilts, 2004. A current resource and future perspectives for enriching WordNets with metaphor information. In *Proceedings of the Second International WordNet Conference – GWC 2004*. Brno, Czech Republic.

Lönneker, Birte, 2003. Is there a way to represent metaphors in WordNets? Insights from the Hamburg Metaphor Database. In *Proceedings of the ACL 2003 Workshop on the Lexicon and Figurative Language*.

Martin, J. H., 1992. Computer understanding of conventional metaphoric language. *Cognitive Science*, 16(2):233–270.

Martin, J. H., 1994. MetaBank: A knowledge-base of metaphoric language conventions. *Computational Intelligence*, 10(2):134–149.

Miller, G. A., 1990. Special issue on WordNet: An on-line lexical database. International Journal of Lexicography. 3(4).

Pease, A. and C. Fellbaum, 2003. Language to logic translation with phrasebank. In *Proceedings of the Second International WordNet Conference—GWC 2004*. Brno, Czech Republic.

Reddy, M. J., 1979. The conduit metaphor: A case of frame conflict in our language about language. In A. Ortony (ed.), *Metaphor and thought*. Cambridge: Cambridge University Press, pages 284–324.

Veale, T., 1998. 'Just in Time' Analogical Mapping. An Iterative-Deepening Approach to Structure-Mapping. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence (ECAI'98)*. Brighton, UK.

Vossen, P., 1999. EuroWordNet general document. Version 3, final. Technical report, University of Amsterdam. Http://www.illc.uva.nl/EuroWordNet/docs.html.

[8]URL: http://atilf.inalf.fr/tlfv3.htm [14 April 2004]

Words, Emotions and Plot-Generation in MEXICA

Rafael Pérez y Pérez
Laboratorio de Interacción Humano-Máquina y Multimedia.
Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Universidad Nacional Autónoma de México
Circuito Exterior s/n-Ciudad Universitaria, México D. F. 04510
Tel: (52-55) 5622-8602x1174.
Fax: (52-55) 5622-8653
E-mail: rpyp@servidor.unam.mx

Abstract

MEXICA is a computer program for plot-generation. As a distinctive characteristic, the system employs emotional links between characters and the dramatic tension of the story in progress as cue to probe memory and retrieve sequences of actions. All valid actions in MEXICA are defined in a text file known as the dictionary of Linguistic Representations of Actions. This dictionary, together with a set of previous stories, constitute the material employed to construct the knowledge structures that drive the generation of frameworks for short-stories. This paper focuses on explaining the relationship between Linguistic Representation of Actions and emotions, and their role during plot generation.

1. Introduction.

MEXICA (Pérez y Pérez, 1999; Pérez y Pérez & Sharples, 2001) is a program that generates frameworks for short stories about the Mexicas (the old inhabitants of what today is México city) based on the engagement-reflection cognitive account of writing (Sharples, 1999). During engagement the system focuses on generating sequences of actions driven by content and rhetorical constraints and avoids the use of explicit goals or predefined story-structures. During reflection MEXICA evaluates the novelty and interestingness of the material produced so far and verifies the coherence of the story. Figure 2 shows an example of a story developed by MEXICA. The design of the system is based on structures known as Linguistic Representations of Actions (LIRAs), which are a set of actions that any character can perform in the story and whose consequences produce some change in the story-world context. These representations (also known as Primitive Actions) are defined as single words (usually verbs) like *A found B*, strings of words like *A followed and found B*, or whole phrases like *A followed the trace through the forest and finally found B swimming in a beautiful waterfall,* where A and B represent characters in the story. MEXICA requires a dictionary of LIRAs to work. In such a dictionary one must specify the word or phrase that identifies the action, the number of characters that participate in it (maximum three actors), and a set of preconditions and postconditions associated with the action (see figure 1).

Preconditions } A *followed and found* B { Postconditions

Figure 1. Elements that constitute a Linguistic Representation of an Action.

In this way, in MEXICA a story is defined as a sequence of LIRAs. There are two types of possible preconditions and postconditions in MEXICA: 1) emotional links between characters and 2) dramatic tensions in the story.

Tlatoani was an inhabitant of the Great Tenochtitlan. Priest was an ambitious person and wanted to be rich and powerful. So, priest kidnapped tlatoani and went to Chapultepec Forest. Priest's plan was to ask for an important amount of cacauatl (cacao beans) and quetzalli (quetzal) feathers to liberate tlatoani. With a hidden knife tlatoani was able to cut all the ropes and escape. Tlatoani was really angry for what had happened and affronted priest. Priest thoroughly observed tlatoani. Then, took a dagger and attacked tlatoani. Suddenly, tlatoani and priest were involved in a violent fight. In a fast movement, priest wounded tlatoani. An intense haemorrhage arouse which weakened tlatoani. Priest felt panic and ran away.
Pince was an inhabitant of the Great Tenochtitlan. Early in the morning prince went to Chapultepec Forest. Suddenly, prince realized that priest wounded tlatoani. Tlatoani always felt a special affection for prince. Even when prince knew about the sympathy that tlatoani felt, prince saw a unique opportunity to become rich and attempted to take advantage of the situation by asking tlatoani for an important amount of cacauatl (cacao beans). Tlatoani was really angry for what had happened and affronted prince. Prince, knowing that tlatoani's life was at risk, resolved not to cure tlatoani. Prince decided to go back to the Great Tenochtitlan City. The injuries that tlatoani received were very serious. However, tlatoani knew that when a Mexica dies fighting, the Gods protect that soul in order it arrives safely to the other world. So, tlatoani died in peace.

Figure 2. *The Kidnapped Tlatoani* (a story develop by MEXICA).

2. Emotions as Preconditions and Postconditions.

Emotional Links. MEXICA allows defining two types of emotional links between characters. For practical reasons all types of emotions are implemented in discrete terms with a value in the range of -3 to +3. Type 1 represents a continuum between love (brotherly love) and hate. Type 2 represents a continuum between being in love with (amorous love) and feeling hatred towards. For example, the action where character A fells in love with character B includes as a postcondition an emotional link from A towards B of type 2 and intensity +3. In the same way, the action A wounds B includes as a precondition the fact that A hates B, i.e. A has an emotional link of type 1 and intensity -3 towards B.

Dramatic Tensions. Tension is a key element in any short story. In MEXICA, it is assumed that a tension in a short story arises when a character is murdered, when the life of a character is at risk, when the health of a character is at risk (e.g. when a character has been wounded) and when a character is made a prisoner. These tensions can be defined as part of LIRA's postconditions and triggered when the action is performed in the story: e.g. the action A wounds B triggers the postcondition *the health of B is at risk*. In the same way, tensions can be deactivated through postconditions: e.g. the action C cures B deactivates the postcondition *the health of B is at risk*. Notice that C cannot cure B at least B is wounded (or ill); so, the tension *the health of B is at risk* is a precondition of the action C cures B. There is a second group of three tensions referred to as inferred tensions: 1) clashing emotions: when a character establishes two opposite emotional links towards other character; 2) love competition: when two different characters are in love with a third one; and 3) potential danger: when a character hates another character and both are located in the same place. These tensions are not defined as part of LIRAs; they are hard-coded and become active only when the emotions that trigger them are present in the story. Thus, each time an action is executed in the story in progress MEXICA verifies if inferred tensions must be triggered or deactivated. Figure 3 shows a representation of a complete definition of a LIRA.

LIRA
A saved the life of B
List of preconditions:
The life of B must be at risk [tension].
List of postconditions:
The life of B is not anymore at risk [deactivation of a tension].
B develops an emotional link of type 1 and intensity +3 towards A.
Alternative Texts
A desperately ran to forest to get some magic plants and saved the life of B

Figure 3. Definition of a Linguistic Representation of an Action.

Notice that MEXICA allows defining alternative texts to represent a LIRA. In this way, when MEXICA generates the final version of a story, it can employ any of the alternative texts to represent the action.

Each tension in MEXICA has associated a value. Thus, each time an action is executed the value of the tension accumulated in the tale is updated; this value is stored in a vector called Tensional Representation. The Tensional Representation records the different values of the tension over time. The Tensional Representation permits representing graphically a story in terms of the tension produced in the story. In MEXICA, a story is considered interesting when it includes increments and decrements of the tension (see figure 4).

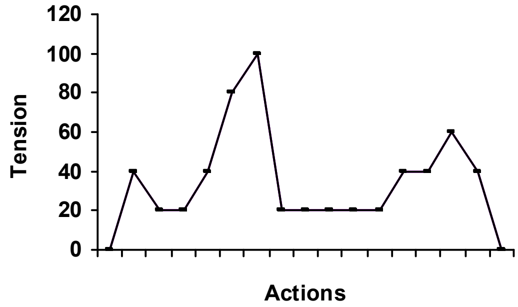

Figure 4. Tensional Representation of *The Kidnapped Tlatoani.*

3. Creating Structures in Memory.

All knowledge structures in MEXICA are built from the dictionary of LIRAs and from a set of Previous Stories. MEXICA is a tool to study the engagement-reflection cycle in plot generation. Thus, the user can define an important number of parameters that control de system. Between these parameters are included the mentioned dictionary and a set of previous stories. MEXICA includes a language to define each entry in the dictionary of LIRAs. Details of such a language can be found in (Pérez y Pérez, 1999 Appendix A). The purpose of the dictionary is to create a collection of actions with their associated preconditions and postconditions. In MEXICA, preconditions and postconditions must be as general as possible. They represent essential requirements and consequences of an action in terms of emotional links and dramatic tensions. For example, a fight between two knights irremediably has as a consequence that their life are at risk (dramatic tension) and, probably, that they develop negative emotional links towards each other. The quality of MEXICA's outputs strongly depends on the dictionary of LIRAs. Each previous story is formed by a sequence of actions. As in the case of LIRAs, MEXICA includes a language to define the previous stories.

MEXICA creates in memory a group of structures known as atoms. Each atom is formed by a collection of emotional links and tensions, and by a set of possible next actions to be employed during the development of a story. The process to create atoms works as follows:

1. MEXICA reads an action from the files of Previous Stories.
2. It updates the story-world context with the action's consequences.
3. All characters in the updated story-world context are substitute by variables; then, the context is employed to build a new atom in memory.
4. MEXICA reads the next action from the files of Previous Stories.
5. The system adds this new action to the set of possible next actions of the atom created in step number 3.
6. The system goes to step 2 until the story ends.

This sequence is repeated for each previous story in the file. If MEXICA generates two identical atoms, only one is kept. The purpose of atoms is to associate groups of emotional links and tensions with a set of possible next actions to be performed. For example, if character A hates character B (an emotional link of type 1 an intensity -3) some members of the set of possible next actions might be: A insults B, A punches B, A kills B, etc. In this way, MEXICA knows that when the story-world context represents a situation where someone hates someone else, any of the elements in the set of possible next actions is a logical action to continue the story. The system can generate very complex groups of emotional links and tensions during the unfolding of a story.

4. Plot Generation.

The process of developing new stories consists of a cycle between the Engaged and Reflective States. During engagement an action is performed producing a story-world context. Such a context is used to match in memory atoms representing similar situations. These structures have associated a set of possible next actions, which are retrieved. Then, one of them is selected as the next action in the story. This action is performed in the story producing a new story-world context and the cycle starts again. As part of the engage state MEXICA employs a set of heuristics to modify the story-world context in order to retrieve novel sequences of actions. If the cycle is interrupted (e.g. by an impasse) the system switches to the reflective state. During reflection all preconditions are verified (notice that preconditions are not checked during engagement) and if necessary actions are inserted to satisfy them, impasses broken, and the material produced is evaluated for originality and interestingness. The system then returns to the engage state or finishes the story. So, plots develop in a non-linear way rather than linearly progressing from the start of the story to its end. The following lines describe how MEXICA produced *The Kidnapped Tlatoani*. For reasons of clarity, the texts employed to describe actions in this example are not exactly the same as those used in figure 2. The user selects the first action (in bold):

*** NEW STORY:
0 The tlatoani liberated himself (0)

The number on the left side (in this case zero) indicates that the action was produced at time 0; the number between parentheses on the right side indicates the value of the tension (at this moment zero). MEXICA switches to engagement but cannot retrieve any action from memory. Thus, an impasse is declared and the system switches back to reflection to try to break the impasse.

*** NEW STORY:
1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)

During reflection MEXICA checks preconditions. So, the system inserts actions at time 1 and 2 to justify why the tlatoani liberated himself (all actions generated during reflection are printed in italics). Notice that the action generated at time 0 (the action given by the user) is the last event in the story produced so far. MEXICA switches to engagement and generates three new actions (a parameter definable by the user specifies the number of actions that can be generated during engagement; in this example, this number is three).

*** NEW STORY:
1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)
3 The priest attacked the tlatoani (40)
4 The tlatoani and the priest fought (80)
5 The priest wounded the tlatoani (100)

Notice that the action generated at time 5 reaches the highest value of the tension in the story (see figure 4). Now MEXICA switches back to reflection and verifies preconditions.

*** NEW STORY:
1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)
6 The tlatoani affronted the priest (20)
3 The priest attacked the tlatoani (40)
4 The tlatoani and the priest fought (80)
5 The priest wounded the tlatoani (100)

In this case the system needs to justify why the priest attacked the tlatoani; so, it inserts the affronted action at time 6. All preconditions are satisfied and MEXICA goes back to engagement.

*** NEW STORY:
1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)
6 The tlatoani affronted the priest (20)
3 The priest attacked the tlatoani (40)
4 The tlatoani and the priest fought (80)
5 The priest wounded the tlatoani (100)
7 The priest ran away (20)
8 The prince decided not to cure the tlatoani (60)
9 The prince went back to Tenochtitlan City (40)

MEXICA generates three actions at times 7, 8 and 9 and switches to reflection. Notice a peculiar moment at time 8 where the system introduces a new character in the story, the prince, which decides not to help the wounded tlatoani. MEXICA needs to explain why this situation occurs.

*** NEW STORY:

1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)
6 The tlatoani affronted the priest (20)
3 The priest attacked the tlatoani (40)
4 The tlatoani and the priest fought (80)
5 The priest wounded the tlatoani (100)
7 The priest ran away (20)
10 The prince lived in Tenochtitlan (20)
11 The prince decided to go to the forest (20)
12 The prince realised that the priest wounded the tlatoani (20)
15 The tlatoani was fond of the prince (20)
14 The prince tried to abuse of the tlatoani (40)
13 The tlatoani affronted the prince (40)
8 The prince decided not to cure the tlatoani (60)
9 The prince went back to Tenochtitlan City (40)

The first step is to introduce the prince in the story at time 10, situate the prince with the tlatoani at the forest at time 11 and make the prince aware that the tlatoani is wounded at time 12. Next, MEXICA inserts the affronted action at time 13 to justify why the prince does not want to help the tlatoani. However, now the system needs to explain why the tlatoani affronted the prince. So, it inserts the abuse or take-advantage action at time 14. Finally, to satisfy the preconditions of action 14 (the goal of this precondition is to increase the tension producing clashing emotions) the system inserts the was-fond action at time 15. MEXICA switches to engagement.

*** NEW STORY:

1 The tlatoani lived in Tenochtitlan (0)
2 The priest kidnapped tlatoani (40)
0 The tlatoani liberated himself (20)
6 The tlatoani affronted the priest 20
3 The priest attacked the tlatoani 40
4 The tlatoani and the priest fought 80
5 The priest wounded the tlatoani 100
7 The priest ran away 20
10 The prince lived in Tenochtitlan
11 The prince decided to go to the forest
12 The prince realised that the priest wounded the tlatoani
15 The tlatoani was fond of the prince
14 The prince tried to abuse of the tlatoani 40
13 The tlatoani affronted the prince 40
8 The prince decided not to cure the tlatoani 60
9 The prince went back to Tenochtitlan City 40
16 The tlatoani died due to his injuries 0

MEXICA cannot retrieve actions from memory and an impasse is declared. So, it switches to reflection and inserts the action where the tlatoani dies in order to break the impasse. The system switches back to engagement and a new impasse is declared. This time MEXICA cannot break it and the story is ended. A detailed description of how this story is generated can be found in (Pérez y Pérez, 1999).

5. Conclusions.

In MEXICA, LIRAs are the basic plot components whereas emotional links and tensions work as the joining units (c.f. Lehnert, 1983). During engagement preconditions are ignored; so, the production of material relies completely on the knowledge recorded in atoms. During reflection actions might be inserted in any part of the story to satisfy preconditions. Thus, the story does not unfold in a linear way and its structure arises as the plot develops. These characteristics allows MEXICA to generate material without the use of explicit goals or pre-defines story structures (c.f. Meehan, 1981; Pemberton, 1989; Turner, 1994). This attribute is relevant since story-predictability, i.e. "the degree to which the output of a computerized storyteller can be predicted when the content of the system's knowledge-structures are known" (Pérez y Pérez & Sharples, 2004), is closely linked to predefined structures. Gelernter (1994) claims that creativity can be reduced to the discovery of new analogies when one thought triggers another one that is related to it by shared emotions. MEXICA suggests that Gelernter ideas can be useful to create more flexible computer programs for plot-generation.

Acknowledgements.

The MEXICA project was sponsored by the National Council of Science and Technology (CONACYT) in México.

References.

Gelernter, D. (1994). The Muse in the Machine. London: Fourth Estate.

Lehnert, W. (1983). Narrative Complexity Based on Summarization Algorithms. In Proceedings of the Eight International Joint Conference in Artificial Intelligence (pp. 713-716). Karlsruhe, Germany.

Meehan, J. (1981) TALE-SPIN. In R.C. Shank & C.K. Riesbeck (Eds.), Inside Computer Understanding: Five Programs plus Miniatures (pp. 197-226). Hillsdale, NJ: Lawrence Erlbaum Associates.

Pemberton, L. (1989) A modular approach to story generation. In: 4th European ACL (pp. 217-224). Manchester, UK.

Pérez y Pérez, R. (1999) MEXICA: a Computer Model of Creativity in Writing. PhD dissertation. University of Sussex.

Pérez y Pérez, R. & Sharples, M. (2001) MEXICA: A computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence, 13 (2), 119-139.

Pérez y Pérez, R. and Sharples, M. (2004) Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. Knowledge Based Systems Journal, 17 (1), 15-29.

Sharples, M. (1999) How we write. London: Routledge.

Turner, S. R (1994) The Creative Process: A Computer Model of Storytelling. Hillsdale NJ: Lawrence Erlbaum Associates.

Resources for “Computational On-line Meditative Intelligence for Computers”

Oliviero Stock and Carlo Strapparava

ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Trento, ITALY
{stock|strappa} itc.it

Abstract

HAHAcronym has been the first project concerned with computational humor sponsored by the European Commission. The project was meant to convince about the potential of computational humor, through the demonstration of a working prototype and an assessment of the state of the art and of scenarios where humor can add something to existing information technologies. The main goal of HAHAcronym was the realization of an acronym *ironic* re-analyzer and generator as a proof of concept in a focalized but non-restricted context. In order to implement it some general tools have been developed or adapted for the humorous context. For all tools, particular attention has been put on reusability. A fundamental tool is an incongruity detector/generator to detect/generate semantic mismatches between the known/expected “sentence” meaning and other interpretations, along some specific dimension.

1. Introduction

To analyze or generate verbal humour as part of a text or of a dialogue requires to include the results of humour research into traditional natural language processing resources such as lexicons, part-of-speech taggers, parsers, annotation tools, knowledge representation formalisms.

So far only very limited effort has been put on building computational humor prototypes. Indeed very few working prototypes that process humorous text and/or simulate humor mechanisms exist. Mostly they are concerned with rather simple tasks.

There has been a considerable amount of research on linguistics of humor and on theories of semantics and pragmatics of humor (Attardo 1994, Attardo and Raskin 1991, Giora and Fein 1999); however, most of the work has not been formal enough to be used directly for computational humor modeling.

Within the artificial intelligence community, most writing on humor has been speculative (Minsky 1980, Hofstadter et al. 1989). Minsky made some preliminary remarks about how humor could be viewed from the artificial intelligence/cognitive science perspective, refining Freud's notion that humor is a way of bypassing our mental "censors" which control inappropriate thoughts and feelings. Utsumi (1996) outlines a logical analysis of irony, but this work has not been implemented. Among other works: Katz (1993) attempted to develop a neural model of humor. Ephratt (1990) has constructed a program that parses a limited range of ambiguous sentences and detects alternative humorous readings.

Probably the most important attempt to create a computational humor prototype is the work of Binsted and Ritchie (1994).

They have devised a formal model of the semantic and syntactic regularities underlying some of the simplest types of punning riddles. A punning riddle is a question-answer riddle that uses phonological ambiguity. The three main strategies used to create phonological ambiguity are syllable substitution, word substitution and metathesis.

Almost all the approaches try to deal with the incongruity theory at various level of refinement (Attardo 1994). The incongruity theory focuses on the element of surprise. It states that humor is created out of a conflict between what is expected and what actually occurs in the joke. This accounts for the most obvious features of a large part of humor phenomena: ambiguity or double meaning.

Specific workshops concerned with Computational Humor have taken place in recent years and have drawn together most of the community active in the field. The proceedings of the most comprehensive events are (Holstijn and Nijholt 1996) and (Stock, Strapparava and Nijholt 2002). Ritchie (2001) has published a survey of the state of the art in the field.

2. Resources and Implementation

The realization of an acronym re-analyzer and generator was proposed to the European Commission as a project that we would be able to develop in a short period of time (less than a year), that would be meaningful, well demonstrable, that could be evaluated along some pre-decided criteria, and that was conducive to a subsequent development in a direction of potential industrial interest. So for us it was essential that a) the work could have many components of a larger system, simplified for the current setting; b) we could reuse and adapt existing relevant linguistic resources; c) some simple strategies for humor effects could be experimented.

3. Resources

One of the purposes of the project was to show that using “standard” resources (with some extensions and modifications) and suitable linguistic theories of humor (i.e. developing specific algorithms that implement or elaborate theories), it is possible to implement a working prototype.

For that, we have taken advantage of specialized thesauri and repositories and in particular of WORDNET DOMAINS, an extension developed at ITC-irst of the well-known English WORDNET (Fellbaum 1998). In WORDNET DOMAINS, synsets are annotated with subject field codes (e.g. MEDICINE, ARCHITECTURE, LITERATURE…) providing cross-categorial information (Magnini et al.

2002). WORDNET DOMAINS is organized for multilinguality and an Italian extension is already available. Other important computational tools (Stock and Strapparava, 2003) we have used are: a rule database of semantic field oppositions with humorous potential; a parser for analyzing input syntactically and a syntactic generator of acronyms; general lexical resources, e.g. acronym grammars, morphological analyzers, rhyming dictionaries, proper nouns databases.

3.1. WordNet Domains

WORDNET is a thesaurus for the English language inspired by psycholinguistics principles and developed at the Princeton University by George Miller (Miller, 1990). It has been conceived as a computational resource, therefore improving some of the drawbacks of traditional dictionaries, such as circularity of definitions and ambiguity of sense references. Lemmata (about 130,000 for version 1.6) are organized in synonym classes (about 100,000 *synsets*). WORDNET can be described as a "lexical matrix" with two dimensions: a dimension for *lexical relations*, that is relations holding among words, and therefore language specific, and a dimension for *conceptual relations*, which hold among senses (the synsets) and that, at least in part, we consider independent from a particular language. A synset contains all the words by means of which it is possible to express the synset meaning: for example the Italian synset `{calcium, calcio, Ca}` describes the sense of "calcio" as a chemical substance, while the synset `{calcio, pedata}` describes the sense of "calcio" as a leg movement. A list of the main relations present in WORDNET follows.

3.2. Augmenting WORDNET with Domain information

Domains have been used both in linguistics (i.e. Semantic Fields) and in lexicography (i.e. Subject Field Codes) to mark technical usages of words. Although this is useful information for sense discrimination, in dictionaries it is typically used for a small portion of the lexicon. WORDNET DOMAINS is an attempt to extend the coverage of domain labels within an already existing lexical database, WORDNET (version 1.6). The synsets have been annotated with at least one domain label, selected from a set of about two hundred labels hierarchically organized.

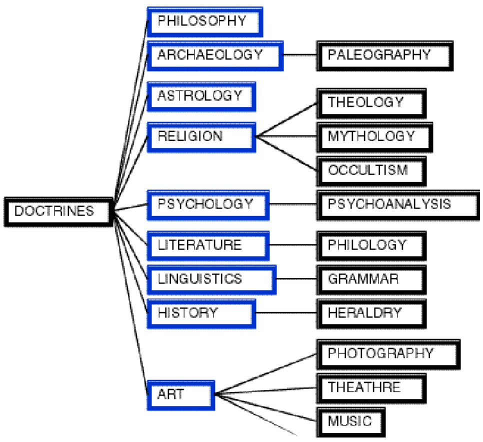

Figure 1: A sketch of the domain hierarchy

We have organized about 250 domain labels in a hierarchy (exploiting Dewey Decimal Classification), where each level is made up of codes of the same degree of specificity: for example, the second level includes domain labels such as BOTANY, LINGUISTICS, HISTORY, SPORT AND RELIGION, while at the third level we can find specialization such as AMERICAN_HISTORY, GRAMMAR, PHONETICS and TENNIS.

Information brought by domains is complementary to what is already present in WORDNET. First of all a domain may include synsets of different syntactic categories: for instance Medicine groups together senses from Nouns, such as `doctor#1` and `hospital#1`, and from Verbs such as `operate#7`. Second, a domain may include senses from different WORDNET sub-hierarchies (i.e. deriving from different "unique beginners" or from different "lexicographer files"). For example, Sport contains senses such as `athlete#1`, deriving from `life_form#1`, `game_equipment#1`, from `physical_object#1`, `sport#1` from `act#2`, and `playing_field#1`, from `location#1`.

3.3. Opposition of semantic fields

On the basis of well recognized properties of humor accounted for in many theories (e.g. incongruity, semantic field opposition, apparent contradiction, absurdity) we have modeled an independent structure of domain opposition, such as RELIGION vs. TECHNOLOGY, SEX vs. RELIGION, etc… We exploit these kinds of opposition as a basic resource for the incongruity generator.

3.4. Adjectives and Antonymy Relations

Adjectives play an important role in modifying and generating funny acronyms. So we gave them a thorough analysis. WORDNET divides adjectives into two categories. *Descriptive adjectives* (e.g. `big`, `beautiful`, `interesting`, `possible`, `married`) constitute by far the largest category.

The second category is called simply relational adjectives because they are related by derivation to nouns (i.e. `electrical` in `electrical engineering` is related to noun `electricity`). To relational adjectives, strictly dependent on noun meanings, it is often possible to apply similar strategies as those exploited for nouns. Their semantic organization, though, is entirely different from the one of the other major categories. In fact it is not clear what it would mean to say that one adjective "is a kind of" (ISA) some other adjective.

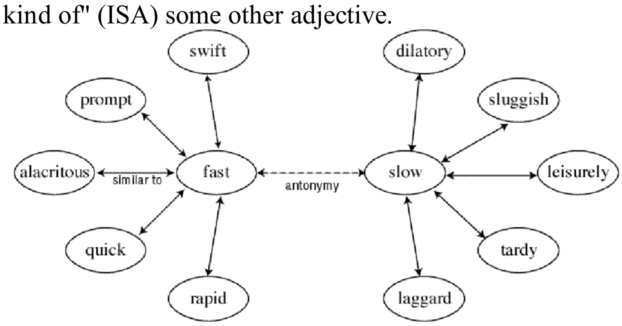

Figure 2: An example of adjective clusters linked by antonymy relation

The basic semantic relation among descriptive adjectives is antonymy. WORDNET proposes also that this kind of adjectives are organized in clusters of synsets associated by semantic similarity to a focal adjective. Figure 2 shows clusters of adjectives around the direct antonyms *fast/slow*.

3.5. Exploiting the hierarchy

It is possible to exploit the network of lexical and semantic relations built in WORDNET to make simple ontological reasoning. For example, if a noun or an adjective has a geographic location meaning, the pertaining country and continent can be inferred.

3.6. Rhymes

The HAHAcronym prototype takes into account word rhymes and the rhythm of the acronym expansion. To cope with this aspect we got and reorganized the CMU pronouncing dictionary (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) with a suitable indexing. The CMU Pronouncing Dictionary is a machine-readable pronunciation dictionary for North American English that contains over 125,000 words and their transcriptions.

Its format is particularly useful for speech recognition and synthesis, as it has mappings from words to their pronunciations in the given phoneme set. The current phoneme set contains 39 phonemes; vowels may carry lexical stress. (e.g. 0 No stress, 1 Primary stress, 2 Secondary stress).

The current phoneme set has 39 phonemes, not counting variations for lexical stress.

3.7. Parser and grammar

Word sequences that are at the basis of acronyms are subject to a well-defined grammar, simpler than a complete noun phrase grammar, but complex enough to require a a nontrivial analyzer. We have decided to use a well established non-deterministic parsing technique (ATN-based parsing). Ordinarily, an ATN parser has three components: the ATN itself, that represent the grammar in the form of a network, an interpreter for traversing it, and a dictionary (possibly integrated with a morphological analyzer). As obvious at this point for the third component we use WORDNET; integrated with an ad-hoc morphological analyzer. As far as the interpreter is concerned, we developed an ATN compiler that translate ATN's directly into Lisp code (i.e. Lisp augmented with non-deterministic constructs). Figure 3 sketches a portion of the acronym grammar.

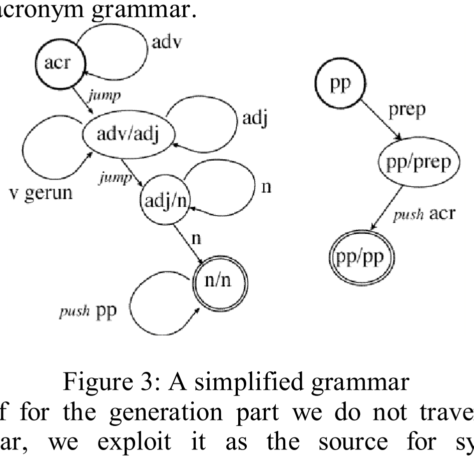

Figure 3: A simplified grammar

Even if for the generation part we do not traverse the grammar, we exploit it as the source for syntactic constraints also there.

3.8. Other resources

An "a-semantic" or "slanting" dictionary is a collection of hyperbolic/attractive/affective adjective/adverbs. This is a last resource, which some time can be useful in the generation of new acronyms.

In fact a slanting writing refers to that type of writing that springs from our conscious or subconscious choice of words and images. We may load our description of a specific situation with vivid, connotative words and figures of speech.

Some examples are: *abnormally*, *abstrusely*, *adorably*, *exceptionally*, *exorbitantly*, *exponentially*, *extraordinarily*, *voraciously*, *weirdly*, *wonderfully*.

This resource is hand-made, using various dictionaries as information sources.

Other lexical resources are: a euphemism dictionary, a proper noun dictionary, lists of typical foreign words commonly used in the language with some strong connotation.

4. Implementation

To get an ironic or "profaning" re-analysis of a given acronym, the system follows various steps and relies on a number of strategies. The main elements of the algorithm can be schematized as follows:

- acronym parsing and construction of a logical form
- choice of what to keep unchanged (for example the head of the highest ranking NP) and what to modify (for example the adjectives)
- look for possible, initial letter preserving, substitutions using semantic field oppositions;
- reproducing rhyme and rhythm (the modified acronym should sound as similar as possible to the original one);
- for adjectives, reasoning based mainly on antonym clustering and other semantic relations in WORDNET.

Figure 4 shows a sketch of the architecture.

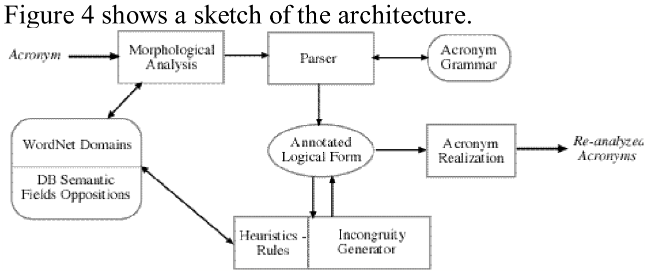

Figure 4: Acronym Reanalysis: a sketch of the architecture

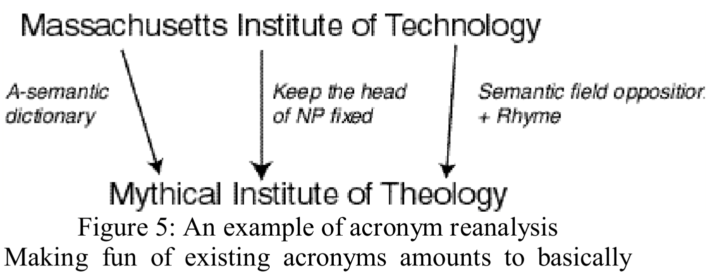

Figure 5: An example of acronym reanalysis

Making fun of existing acronyms amounts to basically using irony on them, desecrating them with some unexpectedly contrasting but otherwise consistently sounding expansion.

As far as acronym generation is concerned, the problem is more complex. We constrain resulting acronyms to be words of the dictionary. The system takes in input some concepts (actually synsets, so that input to this system can result from some other processing, for instance sentence interpretation) and some minimal structural indication, such as the semantic head. The primary strategy of the system is to consider as potential acronyms words that are in ironic relation with input concepts. Structures for the acronym expansion result from the specified head indication and the grammar. Semantic reasoning and navigation over WORDNET, choice of specific word realizations, including morphosyntactic variations, constrain the result. In this specific strategy, ironic reasoning is developed mainly at the level of acronym choice and in the incongruity resulting in relation to the coherently combined words of the acronym expansion.

5. Examples and Evaluation

Here below some examples of acronym re-analysis are reported. As far as semantic field opposition is concerned, we have slightly biased the system towards the domains FOOD, RELIGION AND SEX. For each example we report the original acronym, the re-analysis and some comments about the strategies followed by the system.

`CCTT` - Close Combat Tactical Trainer (Army second generation virtual trainer.)
Cold Combat Theological Trainer
This is an example of two changes: antonym strategy for the first adjective and semantic opposition found in the RELIGION domain that modifies `Tactical' into `Theological'.

`CHI` - Computer Human Interface
Computer Harry_Truman Interface
An unexpected result, mainly achieved exploiting rhyme.

`DMSO` - Defense Modeling and Simulation Office.
Defense Meat_eating and Salivation Office
The two modifications are coherent according to the FOOD semantic field. In general the system can choose either to keep coherence among modifications or to exploit contrast, picking them from different 'opposite' semantic fields, as in the following example:

`IST` - Institute for Simulation and Training.
Institute for Sexual_abstention and Thanksgiving

Here are a couple of examples of automated generation of new acronyms, starting from the themes: “Future” “Emerging” “Technology”

`GONE` - *Gushingly Organized Next Engineering_science*
`USED` - *Unmerchantable Subject for Engineering_science Discipline*

And from the themes “humorless” “computational” “intelligence” the system proposed:
`COMIC` – *Computational On-line Meditative Intelligence for Computers*
that is the title of the this paper.

The system was subjected to a successful evaluation. You can find some details in (Stock and Strapparava, 2003). A curiosity that may be worth mentioning: HAHAcronym participated to a contest about (human) production of best acronyms, organized in December 2002 by RAI, the Italian National Broadcasting Service. The system won a jury's special prize.

References

[Attardo 1994] Attardo, S. Linguistic Theory of Humor. Mouton de Gruyter, Berlin, 1994

[Attardo and Raskin 1991] Attardo, S. and Raskin, V. Script theory revis(it)ed: joke similarity and joke representation model. Humour, 4(3):293-347, 1991.

[Binsted and Ritchie 1994] Binsted, K. and Ritchie, G. An implemented model of punning riddles. In Proc. of the 12th National Conference on Artificial Intelligence (AAAI-94), Seattle, 1994.

[Ephratt 1990] Ephratt, M. What's in a joke. In Golumbic, M., editor, Advances in AI: Natural Language and Knowledge Based Systems, pages 406-450. Springer Verlag, 1990.

[Fellbaum 1998] Fellbaum C. *WordNet*. MIT Press 1998.

[Giora and Fein 1999] Giora, R. and Fein, O. Irony: Context and salience. Metaphor and Symbol, 14:241-257, 1999.

[Hofstadter et al. 1989] Hofstadter, D., Gabora, L., Raskin, V., and Attardo, S. Synopsis of the workshop on humor and cognition. Humor, 2(4):293-347, 1989.

[Holstijn and Nijholt 1996] Hulstijn, J. and Nijholt, A. (editors) Proc. of International Workshop on Computational Humour (TWLT 12), University of Twente, Enschede, The Netherlands

[Katz 1993] Katz, B. F. A neural resolution of the incongruity and incongruity-resolution theories of humor. Connection Science, 5(1):59-75, 1993.

[Magnini et al. 2002] B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. The role of domain information in word sense disambiguation. *Journal of Natural Language Engineering*, 8(4):359–373, 2002.

[Miller 1990] Miller, G. An on-line lexical database. *International Journal of Lexicography*, 13(4):235-312, 1990.

[Minsky 1980] Minsky, M. Jokes and the logic of the cognitive unconscious. Technical report, MIT Artificial Intelligence Laboratory. AI memo 603, 1980.

[Ritchie 2001] Ritchie, G. Current Directions in Computational Humour. *Artificial Intelligence Review* 16(2):119-135, 2001.

[Stock et al. 2002] Stock, O., Strapparava, C. , Nijholt, A. (editors) Proceedings of the The April Fools’ Day Workshop on Computational Humour (TWLT 20) April 15-16, 2002, Trento, Italy, 2002

[Stock and Strapparava, 2003] O. Stock, C. Strapparava. Getting Serious about the Development of Computational Humor. Proceedings of IJCAI 03, Acapulco, Mexico, 2003

[Utsumi 1996] Utsumi, A. Implicit display theory of verbal irony: Towards a computational theory of irony. In Hulstijn, J. and Nijholt, A., editors, Proc. of International Workshop on Computational Humour (TWLT 12), University of Twente, Enschede, Netherlands, 1996.

Interpreting noun-noun compounds: exploiting lexical resources to create a truly large-scale model

Jer Hayes*, **Tony Veale***, **Nuno Seco***

*Department of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland
{jer.hayes, tony.veale, nuno.seco}@ucd.ie

Abstract

Creativity in language has often been treated in terms of toy systems that inevitably do not have a large coverage. A paragon of this research approach is the study of concept combination, concept combination being the cognitive process by which noun-noun compounds are understood. Models of concept combination have exclusively been based around small closed systems, e.g. (Hayes, 2003). Although these types of models may deal adequately with the research problem within their system, they are not practically scaleable. The knowledge in these systems is hand-coded and would require a huge deal of work to cover even a small portion of the English lexicon. Ideally models of noun-noun compound interpretation should make use of existing linguistic resources, e.g. machine readable dictionaries. We suggest that a core set of relations can be associated with the modifier and head of a compound and that these relations can be used to interpret compounds. We outline a model based on this relation-based approach. After a preliminary experiment we note some of the major problems in interfacing linguistic resources.

1. Introduction

Creative processes of language have often been analysed in terms of small systems, e.g. the treatment of metaphor in terms of the SAPPER model (Veale, 1995). These systems may work successfully for the small domain they cover but it is difficult to see how these systems can be succesfully extended to cover larger domains, nevermind a domain which covers a whole language. However, the existence of linguistic resources such as WordNet (WN) (Miller, 1995) and the large corpus that is the web, suggest that perhaps both can be exploited to create truly large-scale models of language creativity. We analyse this idea of integrating linguistic resources with respect to the interpretation of noun-noun combinations which has been exclusively treated in terms of small-scale models.

Combinations such as "arms race" and "web surfer" are made up of two different words but appear to form a linguistic construct that, to a first approximation, functions like a single word. They often refer to a single concept or entity. Linguistically such constructs are known as compounds (O'Grady and Aronoff, 1993). Concept combination is the process whereby novel nominal compounds are understood (Costello and Keane, 2002). The interpretation of compounds with an existing meaning is assumed to be done by recalling a meaning from memory, although many existing compounds are still compositional (e.g. "lamb curry" is "a curry made from lamb"). Noun compounds are found frequently in many types of text from technical writing (McDonald, 1982) to fictional prose (Leonard, 1984). The frequency with which these entities occur has made them a major focus of NLP research.

Concept combination can also be viewed as a process through which new knowledge can be created from existing knowledge (and so is clearly a creative process). Each noun-noun compound consists of a modifier, the first noun element, and a head, the last noun element. Generally, the noun-noun compound describes a type of the head noun, e.g. a "lamb curry" is a type of curry (in some way). Large scale models of noun-noun compound interpretation do not exist and the main reasons for this are problems related to data, data must either be found or created for almost every available noun in the English lexicon. Most models of concept combination involve hand-coded data and although this may work for toy systems such an approach is not practially scaleable. The data involved in small-scale systems is often rich in nature with a concept being described in terms of other related concepts. For example, in describing the concept *butcher* other concepts such as *cleaver*, *carcass*, *butcher shop*, *butchery*, *meat* are brought into play (Hayes, 2003). This richness of information, which is the strength of small-scale systems, is inevitably also the main factor which keeps them small-scale. One possible avenue of research toward creating large-scale models is to use available linguistic resources rather than hand-coding resources (and re-inventing the wheel as it were). The obvious starting point in using linguistic resources for noun-noun interpretation is to focus on machine-readable dictionaries. Unfortunately, though these dictionaries such as WordNet, have a large lexical coverage they are not directly interfaceable with most models of concept combination. The information in these dictionaries is not rich enough for most of these models. However, one strand of work on concept combination has attempted to find particular relations between the modifier and the head in a compound. Relation-based compound interpretation generally suggests that there are a core number of relations that link all compounds and that the interpretation of a noun-noun compound is a question of finding the correct relation. This is the general approach of (Downing, 1977), (Levi, 1977) and more especially

The interface between WN and the web involves using the web to find information on what relations are associated with a noun in WN. We will examine the relations associated with nouns in a modifier role and in head role. The core relations that link compounds can be turned into queries that include the noun we wish to examine. For example we can create the queries “made of steel”, “located at the mountain” - and put these to AltaVista and return a score (hits) for each of these as they occur as an exact phrase. Presumably, if “made of steel” has a high number scores compared to other queries based on other relations then the primary relation associated with the modifier, steel is “made of”. Thus we can rank the relations associated with a noun in terms of its relative frequency. To take a larger example if we wished to associate the relations ‘location’, ‘used for’, ‘made of’ with respect to the noun mountain we could create the following queries:

1. “located in mountain” [“located in the mountain”, “located in a mountain”]
2. “used for mountain”
3. “made of mountain”[“made of the mountain”, “made of a mountain”]

From these queries we find that query set 1 returns 6,784 documents. While query set 2 and query set 3 return 445 and 172 documents respectively. Taking these three relations in terms of frequency of occurrence ‘location’ occurs 91.66% of the time while ‘used for’ occurs 6.0% of the time and finally, ’made of’ occurs 2.3% of the time. Overall this suggests that the ‘location’ relation is strongly related to mountain when used as a modifier. But importantly given WN and a core set of relations we can create scores for every noun in WN using AltaVista. This allows for the creation of a truly large-scale model of noun-noun interpretation. It will also tell us something about how different linguistic resources should be applied and what some of the common pitfalls are.

1.1. Goal of the paper

In this paper we outline how different linguistic resources can be used to develop a large-scale model of a creative language process. Essentially, we describe how a given noun can have a core set of compound relations associated with it. These compound relations are divided into how the noun operates in a head or modifier role. In Section 2 we describe the relations we use which are directly taken from (Gagne & Shoben, 1997). This section also sets out how these relations can be converted into queries which are used to associate a value for a compound relation with a noun. These compound relations are used to interpret compounds, we outline this process in section 3. In section 4 we examine the results of an interpretation process which is based on the compound relations. Finally, we offer suggestions on future work and some insights into the possible dangers of intergating lexical resources.

2. Finding a Core Set of relations

The CARIN model (Gagne & Shoben, 1997) views concept combination as the process of finding the appropriate relation between the modifier and the head. The acronym stands for: competition among relations in nominals. For the CARIN model, the combination ‘mountain stream’ would be interpreted in terms of a location relation, “a stream located in a mountain”. This approach proposes a limited number of relations that all combinations will fall into. The first problem in developing a relation-based approach is the selection of relations. As a starting point we adopt the relations used by Gagne & Shoben (1997). In total, they list 14 relations [1]. We should emphasise that these are not the only possible compound relations as others could have been chosen. So we do not claim this list to be complete or exhaustive.

The complete list of relations is shown in Table 1 (under relation). Gagne and Shoben (1997) claim that these relations have been picked to cover the largest amounts of interpretation possible. They suggest that each nominal has a set of relations associated with it when acting as a modifier. Given a particular nominal, each native speaker has knowledge about the frequency of these relations. When presented with a novel combination they can choose the most frequently occurring relation first. However these relations do involve some words which are highly polysemous. For example, in “noun has modifier”, the relation has is quite polysemous as it is an inflected form of the verb *have*. In WN the verb *have* has 15 senses. So taking the example compound we create a more specific relation, “X contains Y”, where X is a modifier and Y is a head.

The web has been used a corpus for a number of traditional NLP tasks, e.g. example-based machine translation (Way and Gough, 2003), statistical-based translation (Kraaij and Simard, 2003) and likewise we use the web as a corpus for associating compound relations with nouns. As the relations are divided into the form “X Relation Y” we suggest that every noun can have relations associated with it when it is in a head role or a modifier role. The method of associating a relation with a noun is based on creating a query with respect to that relation and submitting this query to a search engine (AltaVista) where the query occurs as an exact phrase. The number of documents returned for this query is used to score the strength of the relation. The queries for both modifier and head roles are listed in Table 2. Given a noun *flu* if we wish to know the score of relation 1 with regards to *flu* in a modifier role then we submit the query “flu is caused by”. For each noun we test we are interested in the complete relationship between the scores for each compound relation.

A closer examination of Table 2 suggests that the query for relation 7 is captured by the queries for relation 9. This will result in the same score for each query. However, the total score for relation 9 is the sum of the documents returned for each of the three queries. At this stage we propose to use the queries listed in Table 2 in this paper but we do note that there is room to develop these further, e.g. by including elements from a set of determiners in the query. We could create queries such as “located in a X” or “located in the X” and so on.

[1] A 15th relation, “like” appears in later work. We will not use this relation in this paper.

Relation	Example	Abstractions
1. “head causes modifier”	flu virus	X is caused by Y
2. “modifier causes head”	college headache	X causes Y
3. “head has modifier”	picture book	X contains Y
4. “modifier has head”	lemon peel	X is part of Y
5. “head makes modifier”	milk cow	X is made by Y
6. “head made of modifier”	chocolate bird	Y is made from X
7. “head for modifier”	cooking toy	Y is used for X
8. “modifier is head”	dessert food	X is a kind of Y
9. “head uses modifier”	gas antiques	Y uses X
10. “head about modifier”	mountain magazine	Y about X
11. “head located modifier”	mountain cloud	X located Y
12. “head used by modifier”	servant language	Y used by X
13. “modifier located head”	murder town	Y located X
14. “head derived from modifier”	oil money	Y derived from X

Table 1: Compound relations

Modifier-based queries of noun X	Head-based queries of noun X
1. X is caused by	1. is caused by X
2. X causes	2. causes X
3. contains X	3. X contains
4. X is part of	4. is part of X
5. is made by X	5. X is made by
6. is made of X, is made from X	6. X is made of, X is made from
7. is used for X	7. X is used for
8. X is a type of, X is a kind of	8. is a type of X ,is a kind of X
9. uses X, that uses X, is used for X	9. X uses, X that uses, X is used for
10. concerned with X, with regard to X	10. X concerned with, X with regard to
11. located in X, located on X, located by X	11. X located in, X located on, X located by
12. used by X	12. X used by
13. X occurs in	13. X occurs in
14. derived from X	14. X derived from

Table 2: Basic query structure

Taking the queries in Table 2 we can associate a score for each compound relation with every single lexeme in WN. There are 54,235 single lexeme nouns in WN (i.e. which are not compounds or multi-word expressions). A close inspection of Table 2 will show that there are 20 queries while we have only listed 14 relations. We suggest that some compound relations will have more than one query. For example, the location relation may be marked by the phrases “located in X”, “located on X”, “located by X” and so all of these are included.

Given a noun we propose to create a list of the total number of documents returned for each query for every single lexeme noun in WN. This creates the following raw data, where each word has twenty scores listed with it -

clamp [29, 1, 37, 248, 695, 692, 688, 1707, 1250, 171, 174, 110, 1707, 0, 0, 110, 107, 110, 823, 6, 25]
sympathy [61, 51, 15, 166, 572, 567, 572, 165, 124, 1569, 4, 19, 165, 2, 317, 5, 5, 5, 34, 10, 13]

From this data we can ascertain which relations are more associated with the word than others. For example the raw data for clamp when used in head role is converted into that information found in Table 3. Where ultimately there are 14 relations with associated strengths in percentage terms, e.g. relation 8 occurs 16.35% of the time. The relation numbers are those given in Table 1, so relation 8 is “modifier is head”. We can also see that the strongest compound relations associated with clamp when used in a head role are “head uses modifier”, “head made of modifier”and “modifier is head”.

2.1. Nouns with little or no relations

In attempting to associate relations with every single-lexeme noun in WN we discovered that some nouns could not be associated with any compound relation. The noun, *anoestrus*, returned no score for any of the queries listed in Table 2. This word appears in 916 documents in the document base indexed by AltaVista and so is not a frequently occuring word. This drawback points to a problem generic in research on the interpretation of noun-noun compounds, which is that the example compounds always use nouns which are well-known and have a high frequency of occurrence. This is a point we will return to in the conclusions.

Head-based queries of clamp	Scores	% Scores
1. “is caused by clamp”	29	0.3337169
2. “causes clamp”	1	0.01150748
3. “clamp contains”	37	0.42577675
4. “is part of clamp”	248	2.853855
5. “clamp is made by”	695	7.9976983
6. “clamp is made of”, “clamp is made from”	1380	15.880322
7. “clamp is used for”	1707	19.643269
8. “is a type of clamp” , “is a kind of clamp”	1421	16.352129
9. “clamp uses”, “clamp that uses”, “clamp is used for”	1991	22.911392
10. “clamp concerned with”, “clamp with regard to”	0	0.0
11. “clamp located in”, “clamp located on”, “clamp located by”	237	3.762946
12. “clamp used by”	823	9.470655
13. “clamp occurs in”	6	0.06904488
14. “clamp derived from”	25	0.28768697

Table 3: Compounds relations associated with *clamp* in a head role

3. Interpreting a compound

Given that a set of relations can be associated with a noun we must now describe how these relations can give rise to an interpretation. There are three possible scenarios we wish to cover:

(1) Where the modifier is known but the head is not
(2) Where the head is known but the modifier is not
(3) Where both the head and the modifier are known

In the first two scenarios the interpretation is based on the strongest relation associated with the modifier in (1) or with the head in (2). Scenario (2) also covers cases where the head may be a larger phrase and is not just a single noun. In scenario (3) we need a mechanism to integrate the compound relations associated with both the head and modifier. Given a compound “X Y” this mechanism could work in three distinct ways:
(a) Prioritise head relations associated with Y
(b) Prioritise modifier relations associated with X
(c) Create a score based on both head and modifier relations

For both (a) and (b) we merely take either the highest ranked head relation or the highest ranked modifier relation. In (c) we need to integrate both the head and modifier relations. If compound relations exist for X as a modifier and for Y as head then the relative percentages for all our found. The percentages are then added and divided by 2 with the largest percentage being suggested as the best relation. This should become clearer through the use of an example. Consider the compound “news report”. To interpret this compound we need information on the compound relations associated with the modifier weather and the compound head and this information can be found in Tables 4 and 5.

Prioritising the modifier relations first a “news report” could be interpreted as “the report is made of news”. Taking the head relation a “weather report” could be described as “a report that contains news”.

The approach we adopt to combining scores is the simplest one, we add the percentages for each related relation and divide this sum by 2. The larger the percentage the more favourable the relation, from Table 4 and 5 the largest percentage is “made of”. So a “weather report” is “the report is made of news”. This interpretation would perhaps parse better if we said “the report is made up of weather”.

4. Experiment

The experiment was divided into two stages. In the first stage 20 queries were created for every single-lexeme noun in WN. For each single-lexeme noun two lists of of 20 scores was associated with each word. One list represented the compound relations for the noun as a modifier the other list represented the compound relations for noun as a head. These lists were in the form of the raw data listed for *clamp* and *sympathy* in Section 2. From these lists percentage scores for each of the 14 relations (see Table 1) were associated with every single-lexeme noun. In stage 2, having associated the compound relations with every noun in WN the interpretations were generated for 20 compounds. This is not a large number of comounds but we wish to test the approaches in Section 3 to ascertain if they are at least workable. We drew on 20 compounds from Nastase (Nastase and Szpakowicz, 2003) as these compounds have already have a relation associated with them. However, we must map the associated relation into our set of relations. Thus, the relations in Table 6 marked as actual were judge by the authors of this paper. For each compound we applied the following strategies: (1) giving priority to the modifier relations, (2) giving priority to the modifier relations (3) selecting the best relation between both modifier and head relations. These were the three approaches to generating interpretations we outlined in Section 3. In Table 6, ‘Mod.’ refers to the strategy of prioritising compound relations based on the modifier role, ‘Head’ to the strategy of prioritising compound relations based on the head role. Finally, ‘Both Mod & Head’ refers to the strategy of combining compound relations from the head and modifier role.

The results in Table 6 show that our relation-based approach does not provide useful interpretations in the majority of cases. The strategy which gave priority to modifier relations only had 7 correct interpretations. The strategy which gave priority to head relations only had 5 correct

Head-based queries of noun report	Scores	% Scores
1. “is caused by report”	8396	0.3337169
2. “causes report”	119	0.01150748
3. “report contains”	83321	0.42577675
4. “is part of report”	8897	2.853855
5. “report is made by”	38837	7.9976983
6. “report is made of”, “report is made from”	77583	15.880322
7. “report is used for”	16719	19.643269
8. “is a type of report” , “is a kind of report”	46714	16.352129
9. “report uses”, “report that uses”, “report is used for”	31230	22.911392
10. “report concerned with”, “report with regard to”	3489	0.0
11. “report located in”, “report located on”, “report located by”	3012	3.762946
12. “report used by”	7562	9.470655
13. “report occurs in”	321	0.06904488
14. “report derived from”	535	0.28768697

Table 4: Compounds relations for *report* in a head role

Modifier-based queries of news	Scores	% Scores
1. “is caused by news”	1359	0.7135169
2. “causes news”	425	0.22313811
3. “news contains”	36608	19.22033
4. “is part of news”	11272	5.9181476
5. “news is made by”	28742	15.090437
6. “news is made of”, “news is made from”	57852	30.374086
7. “news is used for”	10457	5.4902477
8. “is a type of news” , “is a kind of news”	17124	8.990628
9. “news uses”, “news that uses”, “news is used for”	12368	6.493582
10. “news concerned with”, “news with regard to”	1718	0.90200293
11. “news located in”, “news located on”, “news located by”	1343	0.70511645
12. “news used by”	10384	5.45192
13. “news occurs in”	683	0.3585961
14. “news derived from”	130	0.06825401

Table 5: Compounds relations for *news* in a modifier role

Compound	Mod.	Head	Both Mod & Head	Actual
1. weather report	6.	3.	6.	10.
2. summer morning	6.	6.	6.	4.
3. ice crystal	6.	6.	6.	6.
4. water droplet	9.	9.	9.	6.
5. air current	9.	9.	9.	1.
6. lightning strike	6.	6.	6.	2.
7. steel frame	6.	6.	6.	6.
8. tv antenna	6.	8.	6.	7.
9. tobacco leaf	9.	6.	9.	?*
10. telephone wire	6.	9.	6.	7.
11. cirrus cloud	9.	6.	6.	8.
12. cumulus cloud	9.	6.	8.	8.
13. west coast	11.	6.	6.	11.
14. ocean side	11.	6.	6.	11.
15. rain maker	9.	6.	6.	5.
16. wing tip	9.	6.	6.	4.
17. metal body	6.	6.	6.	6.
18. metal airplane	6.	6.	6.	6.
19. wool scarf	6.	6.	6.	6.
20. water vapour	9.	9.	9.	6.

Table 6: Experiment Results

interpretations. The compounds "west coast" and "ocean side" had modifiers which scored very highly for the 'location' relation, relation 11, and this may have favoured the strategy which gave priority to modifier relations. The weakest strategy overall was the strategy which gave priority to head relations. The strategy which used the combination of the both types of compound relation had 6 correct interpretations.

In classifying the intepretations in terms of the relations we found that compound 9 in Table 6, "tobacco leaf", could perhaps be classified as "tobacco derived from a leaf". This interpreation falls outside of the relations listed in Table 1. There is a relation 'head derived from modifier' but not 'modifier derived from head'. However, it could also be interpreted as "a leaf used for tobaccoo" which does fit our relation set.

The results in Table 6 are surprising as the relations 6 and 9 seem to dominate and we did not expect this. On closer inspection it as was found that the AltaVista search engine was not carrying out searches for exact phrases correctly. For example, when we manually searched for the phrase "is made of steel" we were returned documents which listed "made of" and "steel". This was an unexpected occurrence as AltaVista has proved useful in other areas of research, e.g. (Hayes et al., 2004) and (Seco et al., 2004). This failure raises important questions in dealing with linguistic resources which we will deal with in Section 5.1. The poor results may be a result of this unexpected problem and this needs to be investigated further.

A simple transfer to just using another search engine may not be possible, however. The advantage of AltaVista is that is does not block users who send a large amount of queries to the engine. To find the modifier relations we sent 54,235 * 20 queries to AltaVista, this is over a million queries from one source in a relatively short space of time. But as we discuss in Section 6.1 exploiting available linguistic resources can lead to difficulties where there is a problem with an existing linguistic resource.

5. Conclusions

We outlined a model of noun-noun compound interpretation which attempted to associate a set of relations with each single lexeme noun in WN. This model was intended to demonstrate the usefulness of integrating various linguistic resources. However, the brief experiment in Section 4 clearly shows that the model failed in generating adequate interpretations for the majority of the 20 compounds. On closer examination the problem may result from queries which were sent to AltaVista as exact phrases but which return documents in which this queries did not occur as exact phrases.

This should be only a temporay setback and it emphasises the fact that a researcher should not be overly dependent on one linguistic resource, esp. where other similar linguistic resources exist.

It was also noted that not every single lexeme in WN can have compound relations associated with it. This is not surprising as the less a word occurs in a corpus the less likely it is to occur the form that we hope will match the queries we create. However, this does raise an important point about the overall approach taken here. Even with access to a large corpus such as the web not every noun can be covered. We could remedy this slightly by suggesting that where such a word occurs in a noun-noun compound if the other element has compound relations assocciated with it then these relations should be used.

5.1. Linguistic resources

The danger of working at the interface of various linguistic resources is that you place yourself at the mercy of more than one master. Using WN the researcher has complete control, however, using the web as a data source we have only the same control that any other user does. If a part of the actual interface to the search engine or a component of the search engine is malfunctioning then the researcher is in difficulty. Previous attempts at using AltaVista have been unproblematic but where a problem does occur the researcher has to wait for the resource to be fixed or must adapt to use a new resource.

5.2. Future work

We propose to re-run the experiment in Section 4 with the following changes: (1) Use a larger set of compounds of at least 50 plus. (2) Use at least two search engines to associate the compound relations with nouns. The larger data set should better detect which strategies are more effective. The use of different search engines allows us to test which is the most effective in finding the information we require to associate compound relations with nouns.

6. References

Costello, F.J. and M.T. Keane, 2002. Efficient creativity: constraint-guided conceptual combination. *Cognitive Science*, 24:299–349.

Downing, P., 1977. On the creation and use of english compound nouns. *Language*, 53:810–842.

Hayes, J., 2003. A structural alignment model of noun-noun compound interpretation. *MSc. Thesis*, Dublin City University, Ireland.

Hayes, Veale T., J., , and N. Seco, 2004. Enriching wordnet via generative metonymy and creative polysemy.

Kraaij, Nie J., W. and M. Simard, 2003. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3):381–419.

Leonard, R., 1984. *The interpretation of English noun sequences on the computer*. North-Holland, Amsterdam.

Levi, J., 1977. *The syntax and semantics of complex nominals*. New York: Academic Press.

McDonald, D., 1982. Understanding noun compounds. *PhD. Thesis*, Carnegie Mellon University.

Miller, G., 1995. Wordnet: A lexical database of english. *Communications of the ACM*, 38:39–41.

Nastase V. and S. Szpakowicz, 2003. Exploring noun-modifier semantic relations. *Fifth International Workshop on Computational Semantics (IWCS-5)*, The Netherlands:285–301.

O'Grady, Dobrovolsky M., W. and M. Aronoff, 1993. *Contemporary linguistics : an introduction*. New York: Longman.

Seco, Veale T., N., , and J. Hayes, 2004. Concept creation in lexical ontoligies.

Veale, T., 1995. Metaphor, memory and meaning: Symbolic and connectionist issues in metaphor interpretation. *PhD. Thesis*, University of Dublin, Dublin, Ireland.

Way, A. and N. Gough, 2003. webmt: Developing and validating an example-based mt system using the world wide web. *Computational Linguistics*, 29(3):421–457.

Knowledge Seeking Activities for Content Intelligence[1]

Key-Sun Choi [1], Yeun-Bae Kim [2]

[1] Computer Science Division, Korterm, KAIST, 373-1 Guseong-dong Yuseong-gu Daejeon 305-701 Korea
kschoi@cs.kaist.ac.kr
[2] Human Science Division, NHK STRL, 1-10-11 Kinuta Setagaya-ku Tokyo 157-8510 Japan
kimu.y-go@nhk.or.jp

Abstract. This paper proposes a problem on the knowledge seeking activities compared with information seeking and knowledge discovery from text. A knowledge seeking activity is accomplished by a dynamic linkage of contents that is called "content intelligence." An algorithm called "crossover" of knowledge units is proposed.

1 Introduction

"Content" can be anything that is conveyed or contained by a medium with proper handling method(s) or algorithms. We assume that the content is a text-based entity, e.g. Web documents, semantic web, captions for video data, dialog text, speech-recognized audio data with metadata. By the term "content intelligence", "content" itself will be able to acquire and apply knowledge from other chunks of "content", and will be capable of self-reasoning and being autonomous.

For example, suppose that we have a series of lecture files (e.g., PowerPoint slides), whose file names are f_1, f_2... f_n. Each file has its own segmentations. That is, each file contains several subtopics: m subtopics for the file f_1 are $f_{1,1}$, $f_{1,2}$... $f_{1,m}$. If a student asks a question $q(t)$ about a topic t, the answering against $q(t)$ can be found in several lecture files (say, f_1, f_3, f_5) and the right answers can be assembled from subtopics inside of each file, with the consideration of the student's prior-knowledge about the topic t. For example, a right answer is a sequence $f_{1,1}$, $f_{3,5}$... $f_{5,2}$. Here we need to assume that there is an effective way to extract or be aware of the student's prior-knowledge on the topic t. A "causality" relation among answering segments (e.g., $f_{1,1}$, $f_{3,5}$... $f_{5,2}$) is one of obligatory properties for justification of answers.

"Content intelligence" is enabled by metadata attached to the content, topic-specific ontology, resource ontology (to denote the real resource linked to nodes in ontology), and the methods (or programs) to handle them. "Method" means the

[1] This research was supported by Ministry of Science and Technology, Korea under Brain NeuroInfomatics Research Program for "Knowledge Base Prototype Construction and Its Application for Human Knowledge Processing Modeling" (2001~2004). The major work also had been supported by NHK Science & Technology Research Laboratories (STRL) during the period from March/2002~Feb/2003 while the first author stayed in NHK STRL.

reasoning faculty as well as the knowledge acquisition/application capacity. A method constitutes the "explanation" capabilities through the "causality" tracking.

"Networked content intelligence" assumes that "content" features (1) format diversity as well as (2) the distributedness of their location. In this context, we will say: first, "intelligence" in a content (say, c_i) wants other relevant contents in the network probed and merged in order to answer the question $q(t)$ through the causal reasoning based on its own knowledge:(say, $\{c_j|$ relevancy(c_i,c_j) is important or causal-related$\}$). Second, this process does not assume the physical integration of ontologies in different contents, but it assumes the integration of logical causal ontologies, depending on the question and its intention. This means that we can escape from any noise of totally physical integration between two different knowledge spaces.

First, some of the most precious things we get from content intelligence are: explanations about facts or incidents with multi-aspectual proofs based on solid contents (e.g., multimedia), and discovery of new facts and rules/patterns from networked contents by exploring causality residing inside of contents. *Second*, "causal justification" is important in knowledge-seeking activities. There are two different concepts: knowledge-seeking and knowledge-discovery. The act of seeking requires an explicit circumstance but discovery is an implicit, passive and natural act. Some discovery activities may require more than causal justification (maybe some factual or belief patterns). *Third*, causal justification has been studied in the knowledge-based logical inference as well as probabilistic causal reasoning, for example, based on Bayesian belief net [1]. But the "why"- or "how"-type question-and-answering has not been solved fully.

2 Relevant Works and Problem Definition

Two relevant problems about information seeking and knowledge discovery from text will be discussed in order to define our problem "knowledge seeking". Then an example is shown.

2.1 Comparison and Definition

"Information seeking problem" assumes "resource ontology" associating query components with the resources including the information that is searched for. "Resource ontology" lets the users (or the program) know the exact location of the relevant information for the specific type of query. For example, consider the question "How far is it from Mascat to Kandahar?" [2]. The resource ontology directs to the map information (longitude/latitude) resource for two locations ("Mascat and Kandahar") and the geographical formula resource for "How far." For a given query, the resource ontology guides the query's seeking goal to find the relevant set of information, which will be synthesized to an answer for the given question. The query type is not limited to any specific one but it covers resource-relevant questions (e.g., "how far", "yes/no") rather than "why" or "how"-type question.

On the other hand, “knowledge discovery from text” (hereafter “KDT”) is to automatically identify linguistic patterns used to extract information relevant to a particular task (e.g., knowledge about “causal relation”) from a collection of documents [3]. This problem is different from the information seeking such that KDT does not require any resource ontology but discovers the mapping between query terms and lexical patterns.

Our problem about “knowledge-seeking” assumes that a set of lexical knowledge bases[2] is available. While “information-seeking” focuses on the query decomposition to use resource ontology, “knowledge-seeking” focuses on how to link virtually lexical knowledge bases depending on the question type[3] for a given query. While the knowledge discovery problem focuses on the identification of linguistic patterns for a given semantic relation (e.g., causality), the problem of knowledge-seeking assumes that linguistic patterns have been absorbed in the lexical knowledge bases.

2.2 An Example [4]

Consider the question: "Tell me whether mad cow disease (BSE) causes human brain disease." In Fig. 1, how can you justify (or refute) its corresponding hypothesis (b4) about the “causal” relation between mad cow disease and human brain disease? The hypotheses (b) resides in knowledge space (a2), but the justifying (a3) facts are located in databases (a1) of contents such as TV programs, books or other digital media. The hypotheses (b) could be proved or disproved based on facts acquired from databases (e.g., D1, D2 in a1). The question is how to link the components (e.g., “cow disease”, “human disease” in b1) of the hypothesis to their appropriate database units. In (c1) of disease hierarchy, BSE is a disease. Consider the hypothesis (b4) of "causal link between BSE and human brain disease." The justifying facts are acquired from two different supporting databases relevant to BSE and human brain disease (in a1). If a TV program in the database justifies the hypothesis that mad cow disease causes a human brain disease, the program is one "referent" (a4) of the fact. This hypothesis is an instance (d1) of a "causal" relation (d2) between disease (under c1) and food (under c2). Human brain disease can be caused by eating beef, a food (under c2), that is made edible (d4) from cow, infected (d3) by the disease, BSE (under c1), according to the ontologies. Ontology means all the relations linking concepts in the knowledge space.

Two different ontologies of living things (c3) and food (c2) are connected by the "edible" link (d4) between cow and beef. The "causal" link (d2) between food (c2) and disease (c1) represents the possible causality between the two ontologies. Because these links (e.g., d2, d3, d4) help connect differently classified databases, they will be called "contextual ontologies". Different databases (in a1) for human disease (D3, D4) and cow disease (D1, D2) are connected to different concepts, but a

[2] For example, consider HowNet [5]. A typical representation of “lexical knowledge base” is a set of triplets (relation, node1, node2) that represents the “relation” between “node1” and “node2”, where node1 or node2 may be a concept or a non-concept term. For example, if “human is an animal”, then it is represented by (is-a, human, animal).

[3] For example, “why”-type question to ask about the causality.

contextual ontology links them. Examples of contextual ontology are CAUSE in (d2) and INFECTED in (d3) of Fig. 1.

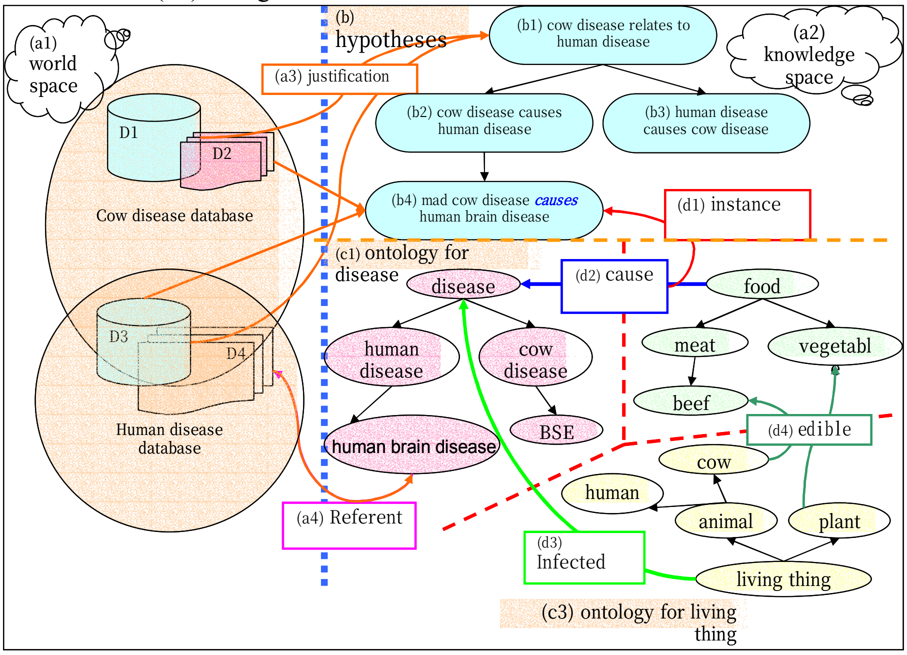

Fig. 1. World (resource) space and knowledge space

3 Hypothesis

Varieties of knowledge bases make physical ontology integration more challenging [6]. The idea behind this paper is to virtually integrate various ontologies according to question types and intention. Consider the question: "Why does the patient pay money to the doctor?" The answer is not found in the lexical dictionary, but the component of the query is in the dictionary. We found that the causality (for "why") answering is possible to integrate the relevant components. See the paths in Fig. 2 where a symbol * stands for AGENT, $ for OBJECT or PATIENT, and # for RELEVANT. Follow the path from (2): "doctor cures patient", "doctor is relevant to occupation", "occupation is to earn (the money)" in (4), and "(patient) giving money is equal to (doctor) taking money" in (6). We will call this path-finding algorithm "crossover". In the following, three hypotheses are shown:

Hypothesis 1: Dynamic virtual ontology integration is effective and transparent in the local pragmatics for the question type and intention.

We have performed the construction of ontology integration as well as lexical mapping between word senses and ontologies. As shown in [6], they eclipsed the nonresolved higher hierarchies in case of direct plugging, whose physical integration is not a complete hypothesis. It has much risk unless it serves ontological purposes. Our hypothesis is to make a virtual integration of various ontologies by different linking algorithms for ontological use (say, causality).

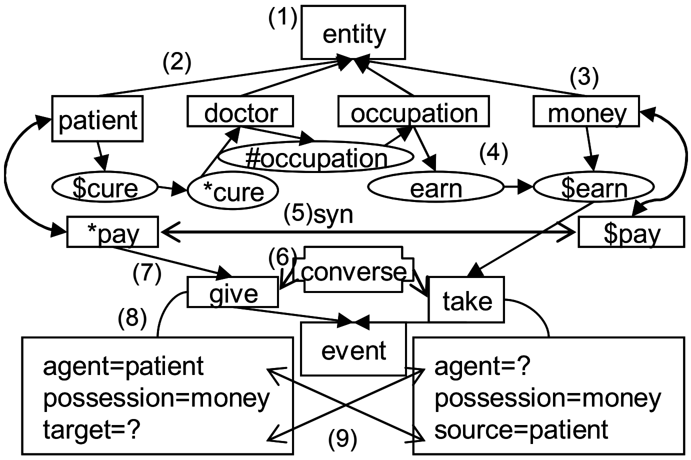

Fig. 2. Virtual linking of knowledge bases

Hypothesis 2 (Crossover Similarity): Causality relation between two nodes can be calculated whether they are causally related or not.

As shown in Fig. 2 for a query "Why does doctor cure patient?", two pairs of a partially shared unit between "doctor=*cure" and "patient=$cure" are used to form one unit of knowledge "cure(doctor, patient)". We will call this "concept crossover" for units of partially shared concepts under some principle. This thinking is just similar to "weighted abduction". Abduction is inference to the best explanation [7]. This achieves the goal on how to seek the ontologies for a hypothesis. Lexical chain [8] is a relevant theory, but limited to topical-relatedness in general.

Hypothesis 3 (Knowledge Uploading): Any pair of syntactically related words in a document is uploaded to knowledge space.

This is intended to solve problems with "prior knowledge" construction. The question is how to easily absorb the information in world space into the knowledge space.[4] Our hypothesis is to respect the syntactical binding. For example, if we find "patient pay money to hospital" in context, the new fact "patient paying to hospital, not to doctor" will be absorbed, e.g., $pay=hospital.

[4] Truth-maintenance is necessary when absorbing (or acquiring) knowledge.

4 Virtual Integration of Underlined Knowledge Bases

Some issues on ontology integration have been discussed from various points of view. Pinto et al. [9] classified the notions of **ontology integration** into three types: **integration**, **merging** and **use/application**. The term **virtually integrated** means the view of ontology-based use/application. The followings are excerpted from [10].

4.1 Example: A Snapshot of Virtually Integrated Knowledge Base

Each marked numbering in Fig. 2 has the following meaning:

(1) Entity hierarchy: **entity** is the top node in the hierarchy of entities.
(2) **entity** is the hypernym of **patient**, **doctor**, **occupation**, and **money** in the line (3).
(3) Concepts or word entries are listed in this line. All concepts and word entries represent their definition by a list of concepts and marked pointers.
(4) A concept (or word) in (3) features definitional relations to a list of concepts. For example, a **doctor** definition is composed of two concepts and their marking pointers: **#occupation** and ***cure**. Pointers in HowNet represent relations between two concepts or word entries, e.g., "#" means "relevant" and "*" does "agent".
(5) **syn** refers to the syntactic relation in the question "Why do patients pay money to doctors?"
(6) **converse** refers to the converse relation between events, e.g., **give** and **take**.
(7) Event hierarchy: For example, the *hypernym* for **pay** is **give** and the *hypernym* of **give** is **event**.
(8) Event role: Now, event roles are partially filled with entities, e.g., **patient** and **money**.
(9) Event role shift: The *agent* of **give** is equalized to the *source* of **take**.

An overview of each component of the knowledge base is in Figure 2, where three word entries **why**, **patient**, and **money** are in the *dictionary*. The four *concept facets* of **entity**, **role**, **event**, and **converse** are described in this example, mainly as part of linguistic knowledge.

4.2 Interpretation of Lexical Knowledge

Consider the following three sentences:

1. Doctors cure patients.
2. Doctors earn money.
3. Patients pay money.

One major concern is to find *connectability* among words and concepts. As shown in Fig. 3, the following facts are derived:

4. Doctor is relevant (#) to occupation.
5. Occupation allows you to earn money.

Because a converse relation exists between **give** and **take**, their hyponyms **earn** and **pay** also fall under converse relation. It is something like the following social commonsense as shown in Fig. 3: "If someone X pays money to the other Y, Y earns money from X." We humans now understand the reason for "why patients pay money." The answer is that "doctors cure patients as their occupation allowing them to earn money." The following is a valid syllogism where Y is being instantiated to **doctor**:

6. If "X pays money to Y" is equivalent to "Y earns money from X" by converse relation, and "a doctor earns money from X", then "X pays money to the doctor".

Consider the next syllogism:

7. If "a doctor cures X" and "doctor is an occupation" and **Axiom 1**, then "the doctor earns money from X".

Axiom 1 is needed to make such a syllogism that "*If Y cures X and Y is an occupation, then Y earns money from X*." Then our challenge is to find out this **Axiom 1** from the lexical knowledge bases. It is a commonsense and thus there is a gap in the lexical knowledge base.

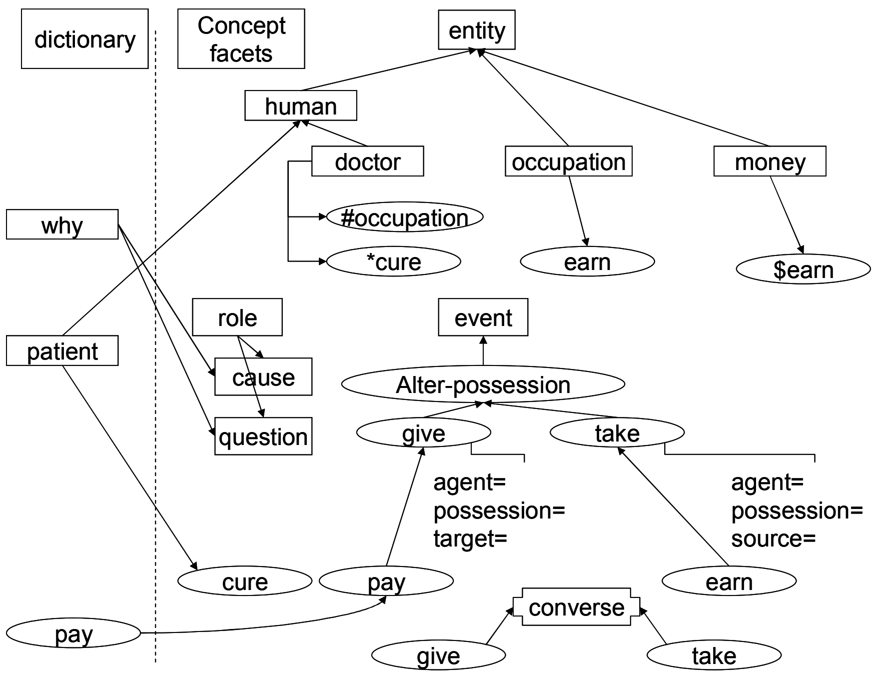

Fig. 3. An Example of Dictionary and Concept Facets in HowNet Architecture [5]

5 Connectability

Consider the query “Why do doctors cure patients?” Tracing Fig. 3 back through Fig. 2 leads to obtaining logical forms from (8) through (11). The best connectable path is planned from the first word (say, “why”) of the question.

8. sufferFrom(patient,disease)
9. cure(doctor,disease)
10. cure(doctor,at-hospital)
11. occupation(doctor)
12. cure(doctor,patient)

For each pair of words, the function called "**similar**(*,*)" will be estimated to choose the next best tracing concepts (or words). **similar**'s missions are summarized as (1) checking the connectability between two nodes[5], (2) selecting the best sense of the node,[6] (3) selecting the best tracing candidate node in the next step. Finally, following the guidance by **similar** allows us to explain the question.

5.1 Observation and Evidence of Topical Relatedness

Let's try to follow the steps 8-12 given in the logical forms. In the question “Why do doctors cure patient?” that focuses on three words **doctor**, **cure**, and **patient**, we can trace some keywords given in example sentences as follows: **patient ~ disease ~ cure ~ doctor ~ occupation ~ earn ~ pay ~ patient**.

What kind of lexical relations are relevant to each pair of words or concepts? Their observation can be summarized as follows:

- The relation between **patient ~ disease** is a role relation of **sufferFrom**(**patient**, **disease**).
- A sequence of **cure ~ doctor ~ occupation ~ earn** lets us infer the relation among **cure ~ earn**, which are closely linked by their *relevance* relation to **occupation**. Furthermore, **earn** and **cure** shares a common subject of these two events.
- The sequence of **earn** ~ **pay** is the result of a converse event relation between **earn** and **pay**.
- **pay ~ patient**: The agent of **pay** is a generic **human**. In other words, **pay** is a hyponym for the **act** of **human**, one of whose hyponym is **patient**.

Consider again the match between the tracing sequences of concepts and the knowledge base. Going into more details, notations with footnotes will be given to each example. At this point, we will give *names* and *formalization* based on the observed characteristics.

---

[5] A **node** means either concept or word.
[6] It is similar with word sense disambiguation.

Feature comparison: To find the role relation among patient ~ disease, search the definition of entities (referring to patient and disease) in ways that two entities share the same event concept (referring to cure):[7]

```
patient ⊃ human ∧ $cure ∧ *sufferFrom.
disease ⊃  medical ∧ $cure ∧ undesired.
```

Interrelation: To find the event interrelation among cure ~ earn, two possible paths are presented as follows.

- **Inverse interrelation**: Two event's role entities can be found by searching all of entities using ***earn ~ *cure** that share the same subject, and using ***earn ~ $cure** where the subject of **earn** is the object of **cure**.
- **Sister interrelation**: The following logical form can be derived from Fig. 3:

```
doctor ⊃ *cure ∧ #occupation.
occupation ⊃ earn.
```

Because **cure** and **occupation** is in the definition of **doctor**, a *probable* (~) logical implication can be derived as follows:

```
*cure ⊃ ~#occupation
```

Converse/antonymy: earn and **pay** have their respective hypernyms **take** and **give**. There exists a converse relation between these two hypernyms.

Inheritance: The relation among **pay** ~ **patient** is represented as follows: ("<" stands for "is a hyponym of")

```
pay < act
human ⊃ *act
patient < human
```

5.2 Rationale of Connectability

In the former section, we summarized four characteristics of causality (relatedness)-based path finding: feature comparison, interrelation, converse/antonymy in their hypernym's level, and inheritance. Among search spaces available, it is necessary to find out a measure of guiding the optimal path tracing.

We will call such a measure **similar** which will be defined according to the four characteristics just mentioned. Further details about the calculation formula will be presented again later.

---

[7] According to HowNet convention, "$" represents patient, target, possession, or content of an event, and "*" represents agent, experiencer, or instrument. "⊃" means **implies** or **has features**.

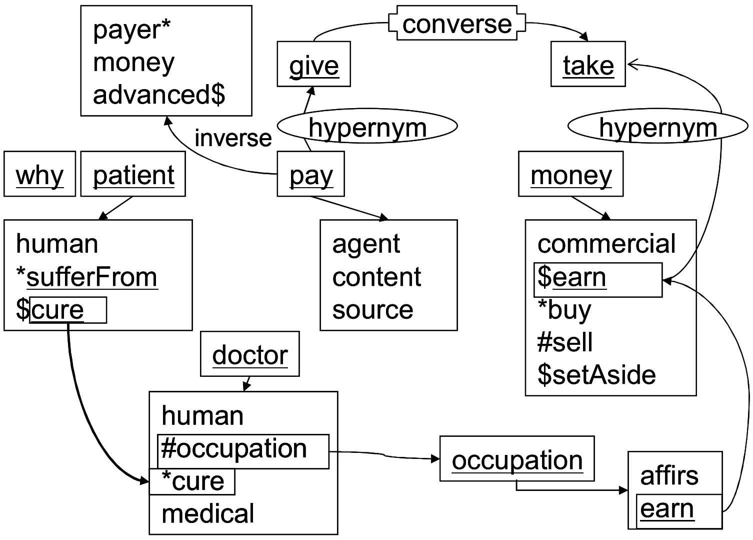

Fig. 4. Definition of Words and Virtual Linking by Crossover. “pay” is defined by two ways: one from case frame (agent, content, source), and the other from the objects in arguments of “pay”. “payer” is an agent of “pay” such that payer*=pay or pay.agent=payer.

Feature comparison:
The measure **feature similar(X,Y)** defines the notion of similarity between the features in **X** and **Y**.

Two interrelations:

- For “inverse interrelation”, **inverse similar(X,Y)** calculates how much similarity exists between $X'\theta$ and $Y'\theta$ in a manner that $X'\theta = \{Z \mid Z \subset \theta X\}$, where θX is an abstraction of role-marked concepts like *X, $X, #X, etc. Thus **inverse similar(X,Y)** = **similar(X’θ,Y’θ)**. In Fig. 4, “payer*” means that “payer” can be an agent role of “pay”.
- For “sister interrelation”, the measure **sister similar(X,Y)** means the following two situations: First, X and Y are features to define one concept (say, W). Second, one of them, say, Y's definitional feature concepts (referring to Z) are *similar* with X such that X and Z are similar if $W \supset X \wedge Y$ and $Y \supset Z$.

Converse or antonymy:
The converse relation **converse(X,Y)** can be found by the measure **feature similar**. **converse(X,Y)** is formulated by $X \subset \theta Y$ and $Y \subset \theta X$ where θ = converse.

Inheritance:
Using inheritance property in the concept hierarchy, relations between hypernym of concepts X and Y are inherited to X and Y in a way that X and Y is similar if there exist X' and Z such that $X<X'$, $Z \supset \theta X'$, and $Y<Z$ where θ is a pointer or null. This inheritance tracing can be determined by how much similar X and Y are in terms of their path upward based on the relation of hypernym. We will define path similar. But tracing the path upward following hypernym links is to be described later according to the algorithm.

5.3 Algorithm CROSSOVER

The main idea of algorithm **Crossover** is obtained by switching over the role pointers θ whenever tracing is performed. [10] Consider again the question "Why do patients pay money to doctors?" As shown in Fig. 2, the best trace is **$cure ~ *cure ~ *earn ~ $pay**. It provides an explanation for the statement that "patients are cured by doctors ~ doctors earn money ~ patients pay money to doctors". This minimal path is obtained by crossing **$cure** over to ***cure**. By crossover operation, **patient** and **doctor** are meaningfully and causally linked through **cure**. Note the following equations:

```
*cure = {doctor, medicine}
$cure = {patient, disease}
```

6 Conclusion

The proposed "justification probing" puts a new frontier line forward to the Turing test[8] about machine intelligence, as well as the current open problem in why/how-type question-answering area. But, although the linguistic knowledge bases have been developed enormously during the last several decades, we have few applications to use them for the knowledge-based reasoning. The reusability of knowledge resources is very important in the sense that we can merge and use the already available knowledge resources[9]. Such knowledge bases required too much cost and human labors. They have to be reused in ways that meet our needs.

The concepts "content intelligence" and "networked content intelligence" are proposed. Content itself is adapted to environment with its own methods. One of methods is investigated under the term "knowledge seeking". It is to use the already made knowledge bases and to link them virtually whenever they are necessary to keep the content intelligent. This approach has advantages over other approach in aspects of dynamic use of already made online ontologies and why-type question handling as shown in Table 1.

[8] The Turing test is that the computer is interrogated by a human via a teletype, and passes the test if the interrogator cannot tell if there is a computer or a human at the other end. [11]

[9] For example, electronic dictionaries, online encyclopedias, electronic usage databases, Web, SemanticWeb resources, eJournal, etc, .

Table 1. Comparison of Three Approaches

	Goal direction	Prior knowledge	Prior knowledge use	Query types
Information seeking	from hypotheses[10] to world space	resource ontology	static, physical	what, where, when, who
Knowledge discovery	from world space to ontology	lexical ontology	static, physical	5W1H
Knowledge seeking	from hypotheses to ontology	networked ontologies, from world space to ontology	dynamic, virtual integration	5W1H (how, why)

References

1. Pearl, J.: Causality: Models, Reasoning, and Inference, Cambridge University Press (2001).
2. ACQUAINT: http://www.ai.sri.com/aquaint/.
3. Girju, R., Moldovan, D.: Mining Answers for Causation Questions. Proc. of AAAI – Spring Symposium (2002).
4. Choi, K.-S.: Knowledge and Standards – A Scenario toward Personalized Interactive Television Services -. Broadcast Technology. No. 12, Autumn. NHK STRL, Tokyo (2002).
5. Dong, Z. and Dong, Q.: HowNet, http://www.keenage.com/. Beijing (1999~2003).
6. Magnini, B. and Speranza, M.: Merging Global and Specialized Linguistic Ontologies. Proceedings of Ontolex 2002 (Workshop held in conjunction with LREC-2002), Las Palmas (2002).
7. Hobbs, J.R., Stickel, M., Appelt, D., Martin, P.: Interpretation as Abduction. Proceedings of the Conference on 26th Annual Meeting of the Association for Computational Linguistics (1988).
8. Moldovan, D. and Novischi, A.: Lexical Chains for Question Answering. Proceedings of COLING 2002, Taipei (2002).
9. Pinto, H. S., Gómez-Pérez, A. and Martins, J.P.: Some Issues on Ontology Integration, Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm (1999).
10. Choi, K.-S., Kim, J., Miyazaki, M., Goto, J., Kim, Y.-B.: Question-Answering Based on Virtually Integrated Knowledge Base. Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages (IRAL2003), Sapporo (2003).
11. Russel, S., Norvig, P.: Artificial Intelligence - A Modern Approach. Prentice-Hall (1995).

[10] "Hypotheses" stands for (b) in Fig. 1.

Creativity in Natural Language: Studying Lexical Relations

Mateus Mendes , Francisco C. Pereira†, Amílcar Cardoso†

Escola Superior de Tecnologia e Gestão de Oliveira do Hospital
R. General Santos Costa, 3400-124 Oliveira do Hospital, Portugal
mmendes@estgoh.ipc.pt

†Departamento de Engenharia Informática da FCTUC
Pólo II, Pinhal de Marrocos, 3000 Coimbra, Portugal
{camara, amilcar}@dei.uc.pt

Abstract

There are already many systems provided with the capacity of automatically generating sentences. Most of them were developed for reliability, others for creativity. Dupond uses lexical relations to transform a sentence, following certain criteria. It is able to produce new sentences keeping the original meaning. It was developed as part of a larger project whose goal is to understand how lexical relations can be used to influence creativity in natural language. But Dupond is suitable for use in applications such as chatter bots and other sentence generators.

1. Introduction

Due to the generalised use of computers, the problem of automatic text generation has become of crucial relevance in recent years. Many systems have already been developed which generate natural language, but most of them invariably produce well known sentences based on rigid templates or other strict rules that make them repeat themselves with little variance. Some systems produce novel sentences, but these don't usually limit their output to a given topic. Dupond was built with the purpose of studying how lexical relations can be used to achieve some creative automatic discourse. It is able to produce different sentences to express the same idea.

Before a truly creative sentence generator can be built, it is necessary to understand what creativity in natural language is. Then we can go further to mimic it. Dupond can presently be fed a sentence and, using selected lexical relations, translate it into another one. Ideally, this new sentence should express the same idea carried by the original one.

Below is a short review of some related work. Section 3. explains the system's theoretical principles, based on the properties of natural language. Section 4. describes the system's capabilities. Sections 5. and 6. contain a short description of its internal modules and how they work. Finally, section 7. discusses some preliminary results.

2. Related Work

2.1. Random sentence generators

Random sentence generators are the simplest ones and don't usually require a complete and well structured knowledge base. They simply pick-up random words or phrases and fit them together in a particular, grammatically correct, order. They are not at all reliable, and their interest, on a scientific view, is very limited. The most frequent practical applications for random sentence generators are word games. Spew and Yak (Schwartz, 1999) are examples of these kind of generators. They are simple word-fitting systems, built just for play. Hypercard Random Sentence Generator (Kelly, 1993) is another example, with the particularity that it applies the theory of random text generation to language teaching.

2.2. Straight sentence generators

Straight sentence generators produce their output in a carefully studied way, and their reliability makes them suitable for many different purposes. Their creativity is very limited, if it ever exists at all. Long interactions with these systems are often boring, and they are not supposed to be used as creativity-aid tools. They are very useful for tasks such as translation, question-answering, report and letter writting, or summarising.

The simplest strict sentence generators are template-based. They contain a set of templates with empty slots that can be filled with known pieces of information. This approach is widely used, because of its low complexity. Most modern text editors are good examples of these systems, since they provide the user with template-based letters, reports and other documents. Another example is Eliza (Weizenbaum, 1966), a computer program built in the sixties, which emulates the discourse of a psychotherapist. Eliza is considered the first great automatic chatterer. *She* works based on tricks like string substitution and canned responses triggered by keywords.

More complex systems usually produce the sentences from formal specifications and grammatical rules. Penman (Matthiessen, C.M.I.M. and Bateman, 1991) is one of the most well known systems of this kind. It receives as input a formal specification of a sentence and translates it into words using the theory of Systemic Functional Linguistics. Internally Penman consists of a network of over 700 nodes, each node representing a single minimal grammatical alternation. In order to generate a sentence, Penman traverses the network guided by its inputs and default settings. At each system node Penman selects a feature until it has assembled enough features to fully specify a sentence. After constructing a syntax tree and choosing words to satisfy the features selected, Penman then generates the sentence.

Straight sentence generators have long been used for many different purposes. Examples include the CO-OP paraphraser (McKeown,), the AGILE (Hana, 2001) translator, the SummariserPort (Oliveira, Paulo et al., 2002) text summariser and the IDAS (Reiter, Ehud et al., 1992) documentation writer, among others.

3. Creativity, Natural Language and Fluency

Natural language is usually analysed in three different layers: syntax, semantics and pragmatics. Creativity can be spotted in any of these layers.

At the syntactic level creative sentences can arise from an original sentence form or an irregular word or phrase ordering. Since syntax in most languages is ruled by well known grammatical rules, creativity at this level is limited to either respecting these rules and have little liberty or breaking them and produce ungrammatical sentences - either meaningless or not. At the semantic level creativity can be the product of using some word or expression to mean something unusual. Poets and some writers do it all the time, producing the literary discourse. Since semantic rules are not as strict as the syntactical ones, it's easier to work on creativity at the semantic level. Pragmatics relates to the *context*, and can be exploited to disambiguate words and make semantic shifts meaningful and useful. Creativity in this level depends on things such as one's culture, values and education.

Writers exploit both syntax, semantics and pragmatics' properties to achieve a fluent discourse, through the use of **figures of speech**. Most of the figures of speech are the product of conceptual relations (metaphor and simile, for instance) and require knowledge and careful reasoning about the world. So far, Dupond doesn't use figures of speech theory in order to produce its output.

The use of lexical relations is another way to express the same idea in different ways. Lexical relations are the following: antonymy, hypernymy/hyponymy, antonymy, homophony, homonymy, polysemy, metonymy and collocation (Yule, 2001). Collocation is an aspect of language which characterises words which tend to occur with other words. For instance, many people associate the pairs *salt-pepper* and *table-chair*. This is just a characteristic that seems of little use for Dupond. Metonymy is a whole-part relation between some words (*car-wheels*, *house-roof*) that makes possible the use of one for replacing another. Most examples of metonymy are highly conventionalised and easy to interpret. However, many others depend on an ability to infer what the speaker has in mind. Thus, this interchangeability requires pragmatic analyses and a good database of knowledge. Polysemy can be defined as one form of a word having multiple meanings, which are all related. For example the words *head*, meaning something or someone *on top* of something. Homonymy can be defined as one form of a word having multiple meanings, but which are not related. For example, *race [speed]* and *race [ethnic group]*. Homophony happens when two differently written words have the same pronunciation (*bare-bear*, for instance). Polysemy, homonymy and homophony make it possible to do some language tricks, but the latter is only suitable for oral speech, and the formers shall not be used if one wants the system to be reliable. Antonymy occurs when two words have opposite meanings, and it is mostly convenient for us to transmit meaning. For instance, our natural explanation for *dirty* is *not clean*. But antonymy is not a general relation we can use in all the situations. Consider the word *beautiful*. Searching the WordNet [1] (Fellbaum, 1998) for antonyms we find *ugly*, but we cannot say the sentence *It's a beautiful morning* is the same as *It's a not ugly morning*. Antonymy is good for explaining relationships with other words in many different situations, but its use requires some common-sense knowledge, so that one knows where to use it.

Hypernymy/hyponymy relations happen when the meaning of one word is included in the meaning of another. A typical pair is *dog-animal*, where dog is an hyponym of animal and the later is a hypernym of the former. One can replace any word in any sentence by one hypernym without changing the original idea. At most the result is an odd sentence or a general, ambiguous sentence. For example, consider the word *girl*. Searching WordNet for hypernyms we find *girl* is a kind of *woman*, *woman* is a kind of *female*, and there are 4 more relations before getting to the top word *entity*. All these words are semantically valid replacements for *girl*. In practise, though, replacements above 1 or 2 levels usually sound unnatural.

Synonymy is the most simple relation one can use, once the correct sense of a word is found. The vast majority of the words can be replaced by synonyms in almost all the contexts, although the result can be an odd sentence, or a different sentence in terms of formality. For instance, consider the sentences *Cathy had one answer correct on the test* and *My dad bought a bigcar*. Using synonyms for replacing words we can get to: *Cathy had one reply right on the examination*, which sounds odd, and *My father purchased a large automobile*, which sounds more formal.

4. Dupond's Features

For now, Dupond is able to disambiguate words, replace words by synonyms and hypernyms and suppress unnecessary words. Each of its features can be configured from *wanted* (always do that, if possible) to *not wanted*. In the middle level it is expected to do that half the time.

4.1. Disambiguating words

Disambiguation is done in function of the context. For instance, in the sentence *That woman is a dog*, the meaning of *dog* is probably *{dog,frump}*, and the system finds it realising that the word *woman* is found in the sentence and the WordNet gloss for the sense *{dog,frump}*. If the word *dog* has never been used in this sense in the current session, Dupond accepts this sense with a given confidence. If it has been used in another sense, then the previous confidence is

[1] Wordnet is available at http://www.cogsci.princeton.edu/˜wn/.

pondered and the preferred sense is a function of the previous confidence and the confidence in the current disambiguation. If the word cannot be disambiguated in function of the context but it has been used before, then the sense with the highest rank is accepted as its current meaning. If the word has not been used before and cannot be disambiguated, then the most frequent sense is preferred with no confidence at all.

4.2. Selecting replacement words

Once we have a word and a set of synonyms in a given context, there are various possible criteria to choose a valid replacement.

One possible criterion is to pick the one with less senses, thus minimising the probability of misinterpretation. Another possible criterion is to pick the one with more senses, thus maximising the probability of that being a known word. Dupond can follow any of the criteria or simply pick a synonym randomly. It can also use hypernymy relations, up to 7 levels, to find valid replacements.

4.3. Other features

Dupond can also be configured to prefer previously used replacements and/or replacement methods, thus producing a more coherent discourse. For instance, consider it chose the word *miss* to replace *woman*. If it is configured to reuse previous replacements, the following occurrences of *woman* will always be replaced by *miss*.

Another feature is its ability to suppress unnecessary words. For instance, consider the sentence *John ate cookies and Mary [ate] cake*. The word in square brackets can be suppressed without the sentence loosing significance and it becomes simpler.

The fact that the system is based on the product of probabilities gives it an infinite flexibility.

5. Dupond's architecture

The system's architecture is as shown in figure 1. All the processing is coordinated by the server module, which receives sentences and orders from its clients through a message queue, performs the necessary steps and sends the new sentences and responses back to them. Users are not expected to interact with the server directly. There is a web client interface where the users can set their preferences and send their sentences in a comfortable way. The client then communicates with the server through the message queue. The server can attend many different clients at the same time. That led to the need of a module for user authentication. When a client sends his first message, it is assigned an identification number and a data structure is created for it. User preferences and some data about the ongoing dialogue are stored, for better performance.

After receiving a sentence, the very first step the server performs is to parse it. A sentence which cannot be parsed, either because it is ungrammatical or for some other reason, is not translated. For parsing Dupond uses Link Grammar Parser[2], a free parser based on link grammar (Sleator and Temperley, 1993). Once the sentence is successfully parsed, the server obtains an equivalent tree-structure which contains all the necessary information about it. The grammatical category of each word and its connection with other words in the same sentence should be well known. Figure 2 illustrates an example parse tree. Suffixes indicate the grammatical category of each word. For instance, ".n" is appended to nouns, and ".v" is appended to verbs.

Presently the parse tree is not used - only the tags attached to each word. In the future the tree may be used to replace phrases or other portions of the sentence.

But knowing the grammatical category of a word is not enough for this system. Consider the word"girls": we need to know not only that it is a name but also that it's in the plural form. To solve this problem there's an additional module, named **Morphy**. Morphy can be interfaced in two different ways. If it is given as input a word in its context it returns complete information about it. For instance, when asked for the word *girls*, morphy would find it's a plural noun and its base form is *girl*. On the other hand, it can be asked what the plural form for the noun *girl* is, and the output would be *girls*.

The disambiguation module tries to find the correct sense of a word, based on the present context and any previous concepts. For example, consider the sentence "The bird went to the market". Searching the WordNet for *bird* we find 5 senses for the noun and 1 for the verb. Since we parsed the sentence we know *bird* is a noun. When asked for the correct sense of this noun in this context, the disambiguator module would return sense 3, indicating that *bird* refers to a girl with an acceptable confidence. If we had been using the noun *bird* in sense 1 *(warm-blooded egg-laying vertebrates characterised by feathers and forelimbs modified as wings)* for a long time before, the disambiguator would most probably return sense one with little confidence. If it cannot disambiguate the word, the module returns the most frequent sense with no confidence.

The **Replacer** module receives the disambiguated word and the set of user preferences. In function of the user's preferences, it picks an appropriated word that could replace the original one. The server uses all these modules to parse the sentence, disambiguate each word, get its base form, find a valid replacement word, put it in the correct grammatical form and rebuild a new sentence.

6. Finding valid replacements

Dupond is controlled internally by "state words". This state words represent sets of probabilities whose values the user can change in order to get different behaviours. Figure 3 shows the system's interface.

If the state words are null, all the probabilities are zero and the system's output is equal to the input.

Once the sentence is parsed, the first optional step Dupond can perform is to disambiguate each word. For this step the user can choose between disambiguation in function of the context, picking the most frequent sense or pick a sense randomly. If the user assigns 7 to the "Disambiguate words" option, the system will always try to disambiguate. 0 means Dupond should never disambiguate, and try any of the other options if they are selected. Once a sense is selected for a given word, it's necessary to choose a valid replacement for it. For example, considering the

[2] http://www.link.cs.cmu.edu/link/

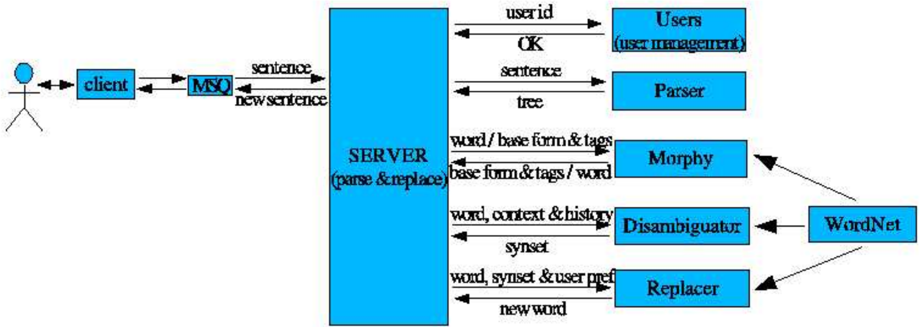

Figure 1: System Overview.

```
  +-Dmc-+---Sp--+--PP-+---Op--+
  |     |       |     |       |
the girls.n have.v got.v flowers.n
```

Figure 2: Parse tree for the sentence "The girls have got flowers".

word *confess* in the last sentence shown in figure 3, the disambiguation process would return the sense 1: *{confess, squeal, shrive}*. The replacement module should then select a valid word from this setfor replacing *confess*. If the user had assigned 7 to the "Prefer synonym with less senses:" option, Dupond would always select the verb *shrive*, since this word contains only one sense and *confess* and *squeal* contain more. The "Trust memory and acquired concepts" option tells the system to repeat previous replacements. If this was selected in the example above, the word *progress* in this sense would always be replaced by *advancement*. The option "Prefer previously used methods:" intends to make the system be more coherent with past behaviour. It tells Dupond to reuse previously applied methods. For instance, if it explored an hypernymy relation to replace a noun (*e.g. dog* -> *canine*), it should use hypernymy to find replacements for subsequent nouns (*e.g. cat* -> *feline*).

7. Preliminary Results

The main goal of this project is to study how important the lexical relations may be to produce sentences in an original way. This involves two steps: 1) build a system able to receive a sentence and, using lexical relations, produce a different one with an equivalent meaning; 2) study how different, meaningful and interesting this automatically rebuilt sentences are for the people. Dupond was built for performing step 1. It can be fed English sentences and rebuild them in function of the user's preferences.

Figure 3 shows a sample session, using sentences selected from the first paragraphs of the book "The return of Sherlock Holmes"[3], with the options shown in the figure.

8. Conclusions

Sentence generators are being used more and more in modern intelligent systems. Creativity will play an important role if one wants to overcome the present limitation that makes machines' speech sound unnatural and repetitive. Dupond is an automatic word replacer ready for being used in the study of natural language and/or other applications. Namely, it may be adapted for automatic chatter bots, documentation and letter writers, message generators and similar systems. Indeed, its main limitation is that it isn't a stand-alone system, thus not suitable for any purpose on its own.

In future work Dupond will be used to study how lexical relations may be used to improve the creativity of natural language generation systems. Possible questions to be answered are: "Do people prefer the more common or the less common words? What makes a sentence look like odd? Do people prefer words with more or less senses?".

Dupond may also be improved for dealing with some figures of speech, replacing phrases and sets of words as well as working on the syntactic and pragmatic levels.

9. References

Fellbaum, Christiane (ed.), 1998. *WordNet: An Electronic Lexical Database*. USA: Bradford Books.

Hana, Jirí, 2001. The agile system. In *PBML*. Praha, pages 39–67.

Kelly, Charles, 1993. A hypercard random sentence generator for language study. *Bulletin of Aichi Institute of Technology*, 28, Part A:51–55.

Matthiessen, C.M.I.M. and J.A. Bateman, 1991. *Text Generation and Systemic-Functional Linguistics*. London: Pinter.

McKeown, Kathleen R. Paraphrasing using given and new information in a question-answer system. In *17th Annual Meeting of the Association for Computational Linguistics*. ACL.

Oliveira, Paulo, Khurshid Ahmad, and Lee Gillam, 2002. A financial news summarisation system based on lexical cohesion. In *Workshop at the International Conference on Terminology and Knowledge Engineering*.

Reiter, Ehud, Chris Mellish, and John Levine, 1992. Automatic generation of on-line documentation in the idas

[3]"The return of Sherlock Holmes", by Arthur Conan Doyle. Downloaded from the project Gutenberg: http://gutenberg.net.

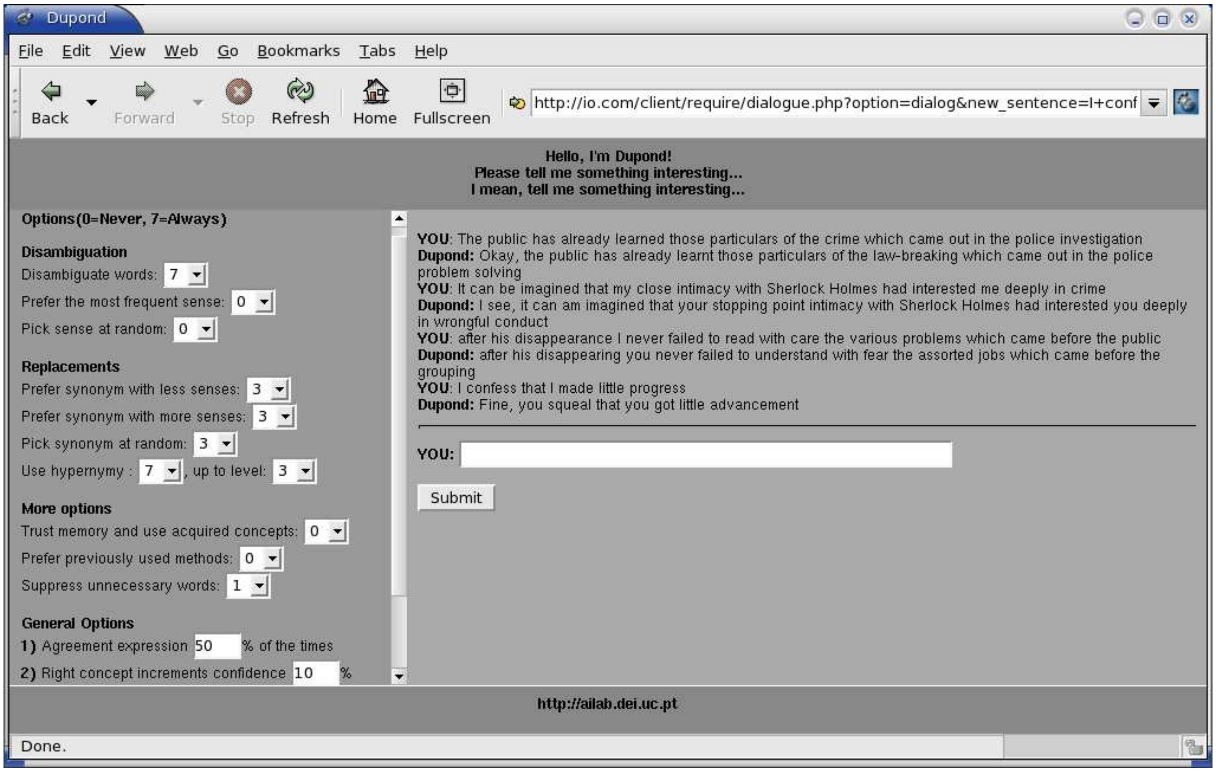

Figure 3: Screenshot showing Dupond's client after a short session.

project. In *Procs of the 3rd Conference on Applied Natural Language Processing*. Trento, Italy.

Schwartz, Randal, 1999. Writing randomly. *Linux Magazine*. Http://www.stonehenge.com/merlyn/LinuxMag/col04.html.

Sleator, Daniel and Davy Temperley, 1993. Parsing english with a link grammar. Technical report, Third International Workshop on Parsing Technologies, Carnegie Mellon University.

Weizenbaum, Joseph, 1966. Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):35–36.

Yule, George, 2001. *The study of language*, chapter 11. Cambridge, pages 114–125.

The Paradoxical Role of Similarity in Creative Reasoning

Nuno Seco, Tony Veale and Jer Hayes

Department of Computer Science
University College Dublin
Dublin 4, Ireland

Abstract

In this paper we present a semantic similarity metric that wholly relies on the hierarchical structure of WordNet which makes it amenable as a means of evaluating creativity when considering creative recategorizations of concepts in an Ontology (Veale, 2004). Many creative discoveries are only acknowledged long after their conception due to changes in the evaluation criteria (Bento and Cardoso, 2004), therefore evaluation plays a critical role in creative reasoning systems. We evaluate the similarity function and report a correlation value of 0.84 between human and machine similarity judgments on the dataset of (Miller and Charles, 1991), which is suggestively close to the upper-bound of 0.88 postulated by (Resnik, 1999). We then use the similarity metric as basis for evaluating some examples of creative categorizations. An extension of the metric is also suggested as a means of assessing analogical similarity by looking for analogical cues in the taxonomy.

1. Introduction

Creativity is a vexing phenomenon to pin down formally (Wiggins, 2003), which is perhaps why we tend to think of it in largely metaphoric terms. For example, creativity is often conceived as a form of mental agility that allows gifted individuals to make astonishing mental leaps from one concept to another (Hutton, 1982). Alternately, it is popularly conceived as a form of lateral thinking that allows those who use it to insightfully cut sideways through the hierarchical rigidity of conventional categories (de Bono, 1994). Common to most of these metaphors is the idea that creativity involves recategorization, the ability to meaningfully move a concept from one category to another in a way that unlocks hidden value, perhaps by revealing a new and useful functional property of the concept. For example, psychometric tests such as the Torrance test of creative thinking (Torrance, 1990) try to measure this ability with tasks that, e.g., ask a subject to list as many unusual and interesting uses of old tin cans as possible.

The ad-hoc nature of creativity is such that most ontologies do not and can not provide the kinds of lateral linkages between concepts to allow this kind of inventive recategorization. Instead, ontologies tend to concentrate their representational energies on the hierarchical structures that, from the lateral thinking perspective, are as much a hindrance as an inducement to creativity. This is certainly true of WordNet (Miller et al., 1990), whose **isa** hierarchy is the most richly developed part of its lexical ontology, but it is also true of language independent ontologies like Cyc (Lenat and Guha, 1990), which are rich in non-hierarchical relations but not of the kind that capture deep similarity between superficially different concepts. It is connections like these that most readily fuel the recategorization process.

Withal, (Veale, 2004) has suggested several ways of detecting these lateral linkages in WordNet by exploiting existing polysemies. Polysemy is a form of lexical ambiguity in which a word has multiple related meanings. The form of polysemy that interests us most from a creativity perspective is function-transforming polysemy, which reflects at the lexical level the way concepts can be extended to fulfill new purposes. For instance, English has a variety of words that denote both animals and the meat derived from them (e.g., chicken, lamb, cod), and this polysemy reflects the transformation potential of animals to be used as meat.

(Veale, 2004) further points out that if one can identify all such instances of function-transforming polysemy in WordNet, we can generalize from these a collection of pathways that allow a system to hypothesize creative uses for other concepts that are not so entrenched via polysemy. For example, WordNet defines several senses of *knife*, one as an *edge tool* used for cutting and one as a *weapon* used for injuring. Each sense describes structurally similar objects (sharp flat objects with handles) with a common behavior (cutting) that differ primarily in function (i.e., slicing vs. stabbing). This polysemy suggests a generalization that captures the functional potential of any other *edge tool*, such as *scissors* and *shears*, to also be used as a *weapon*.

Some recategorizations will exhibit more creativity than others, largely because they represent more of a mental leap within the ontology. We can measure this distance using any of a variety of taxonomic metrics, and thus rank the creative outputs of our system. For instance, it is more creative to reuse a *coffee can* as a *percussion instrument* than as a *chamberpot*, since like *tin can* the latter is already taxonomized in WordNet as a *container*. Any similarity metric (called σ, say) that measures the relative distance to the Most Specific Common Abstraction (MSCA) will thus attribute greater similarity to *coffee can* and *chamberpot* than to *coffee can* and *tympan*. This reckoning suggests that the creative distance in a recategorization of a concept c_1 from α to φ may be given by $1 - \sigma(\alpha, \varphi)$.

Of course, distance is not the only component of creativity, as any recategorization must also possess some utility to make it worthwhile (e.g., there is a greater distance still between *tin cans* and *fish gills*, but the former cannot be sensibly reused as the latter). In other words, a creative product must be unfamiliar enough to be innovative but familiar enough to be judged relative to what we know already works. This is the paradox at the heart of ontological creativity: to be creative a recategorization must involve a significant mental leap in function but not in form, yet typically (e.g., in WordNet), both of these qualities are ontologically expressed in the same way, via taxonomic structure. This suggests that the similarity σ must be simultaneously maximized (to preserve structural compatibility) and minimized (to yield a creative leap).

Fortunately, polysemy offers a way to resolve this paradox (Veale, 2004). If a creative leap from α to φ is facilitated by a polysemous link between β and χ where β is a hyponym of α and χ is a hyponym of φ, the sensibility of the recategorization of c_1 can be measured as $\sigma(c_1, \beta)$ while the creativity of the leap can be measured as $1 - (\alpha, \varphi)$. The value of a creative product will be a function of both distance and sensibility, as the former without the latter is unusable, and the latter without the former is banal. The harmonic mean is one way of balancing this dependency on both measures:

$$value(c_1, \varphi) = \frac{2 \times \sigma(c_1, \beta) \times (1 - \sigma(\alpha, \varphi))}{1 + \sigma(c_1, \beta) - \sigma(\alpha, \varphi)} \quad (1)$$

Considering the example of an *ax* being categorized as a *weapon* would lead to the following instantiation:

- c_1 = *ax*
- α = *edge tool*
- β = *knife* (the edge tool sense)
- χ = *knife* (the weapon sense)
- φ = *weapon*

It is precisely the issue of Semantic Similarity (SS) that this paper will address. We present a wholly intrinsic measure of similarity that relies on hierarchical structure alone. We report that this measure is consequently easier to calculate, yet when used as the basis of a similarity mechanism it yields judgments that correlate more closely with human assessments than other, extrinsic measures that additionally employ corpus analysis. Given the hierarchical nature of our metric we argue that it is an ideal candidate for the role of σ presented in equation 1.

This paper is organized in the following manner; in section 2. we provide a brief overview of some of the approaches that we believe are increasingly relevant to our research and that base themselves on the notion of Information Content (IC) (Resnik, 1995) which is the cornerstone of our metric. These approaches are usually dubbed Information Theoretic, a terminology that we will also employ in the present paper. The following section describes our method of deriving IC values for existing concepts in WordNet (Miller et al., 1990) along with the assumptions made and its formal definition. Section 4. presents the experimental setup and a discussion of the results obtained evaluating our metric against human ratings of similarity. When analyzing our results we also consider alternative approaches (i.e. non-information theoretic) in order to exhaustively evaluate our metric. In section 5. we suggest how this similarity metric may be used for evaluating creative recategorizations, possible extensions that may facilitate the assessment of analogical similarity according to the WordNet ontology are given in section 6. Comments regarding our similarity metric will conclude this paper.

2. Information Theoretic Approaches

A recent trend in Natural Language Processing (NLP) has been to gather statistical data from corpora and to reason about some particular task in the light of such data. Some NLP systems use a hybrid approach where both statistics and a hand-crafted lexical Knowledge Base, such as WordNet, is used. SS has been no exception to this trend. Despite this movement, we feel that these knowledge bases have not yet been fully exploited, and that there is still much reasoning potential to be discovered. Hence, we present a novel metric of IC that is completely derived from WordNet without the need for external resources from which statistical data is gathered. Experimentation will show that this new metric delivers better results when we substitute our IC values with the corpus derived ones in previously established formulations of SS.

Previous information theoretic approaches ((Jiang and Conrath, 1998), (Resnik, 1995) and (Lin, 1998)) obtain the needed IC values by statistically analyzing corpora. They associate probabilities to each concept in the taxonomy based on word occurrences in a given corpus. These probabilities are cumulative as we go up the taxonomy from specific concepts to more abstract concepts. This means that every occurrence of a noun in the corpus is also counted as an occurrence of each taxonomic class containing it. The IC value is then obtained by considering the negative log likelihood:

$$ic_{res}(c) = -log\, p(c) \quad (2)$$

where c is some concept in WordNet and $p(c)$ is its probability according to its frequency in a corpus. It should be noted that this method ensures that IC is monotonically decreasing as we move from the leaves of the taxonomy to its roots. (Resnik, 1995) was the first to consider the use of this formula, that stems from the work of (Shannon, 1948), for the purpose of SS judgments. The basic intuition behind the use of the negative likelihood is that the more probable a concept is of appearing then the less information it conveys, in other words, infrequent words are more informative then frequent ones. Knowing the IC values for every concept allows us to calculate the SS between two given concepts. According to Resnik, SS depends on the amount of information two concepts have in common, this shared information is given by the MSCA that subsumes both concepts. In order to find a quantitive value of shared information we must first discover the MSCA, if one does not exist then the two concepts are maximally dissimilar, otherwise the shared information is equal to the IC value of the MSCA. Formally, semantic similarity is defined as:

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} ic_{res}(c) \quad (3)$$

where $S(c_1, c_2)$ is the set of concepts that subsume c_1 and c_2.

Another information theoretic similarity metric that used the same notion of IC was that of (Lin, 1998). His definition of similarity states:

> ”The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and

the information needed to fully describe what A and B are."

Formally the above definition may be expressed by:

$$sim_{lin}(c_1, c_2) = \frac{2 \times sim_{res}(c_1, c_2)}{(ic_{res}(c_1) + ic_{res}(c_2))} \quad (4)$$

(Jiang and Conrath, 1998) also continued on in the information theoretic vein and suggested a new measure of semantic distance (if we consider the opposite[1] of the distance we obtain a measure of similarity) that combined the edge-based counting method with IC serving as a decision factor. Their model takes into consideration several other factors such as local density, node depth and link type, but for the purpose of this paper we will only consider the case[2] where node depth is ignored and link type and local density both have a weight of 1. In this special case, the distance metric is:

$$dist_{jcn}(c_1, c_2) = (ic_{res}(c_1) + ic_{res}(c_2)) - 2 \times sim_{res}(c_1, c_2) \quad (5)$$

Both Lin's and Jiang's formulation correct a problem existent with Resnik's similarity metric; if one were to calculate $sim_{res}(c_1, c_1)$ one would not obtain the maximal similarity value, but instead the value given by $ic_{res}(c_1)$[3]. This problem is corrected in both subsequent formulations, yielding that $sim_{lin}(c_1, c_1)$ is maximal and $dist_{jcn}(c_1, c_1)$ is minimal.

3. Information Content in WordNet

As was made clear in the previous section, IC is obtained through statistical analysis of corpora, from where probabilities of concepts occurring are inferred. Statistical analysis has been receiving much attention and has proved to be very valuable in several NLP tasks (Manning and Schütze, 1999). We feel that WordNet can also be used as a statistical resource with no need for external ones. Moreover, we argue that the WordNet taxonomy may be innovatively exploited to produce the IC values needed for SS calculations.

Our method of obtaining IC values rests on the assumption that the taxonomic structure of WordNet is organized in a meaningful and structured way, where concepts with many hyponyms convey less information than concepts that are leaves. We argue that the more hyponyms a concept has the less information it expresses, otherwise there would be no need to further differentiate it. Likewise, concepts that are leaf nodes are the most specified in the taxonomy so the information they express is maximal. In other words we express the IC value of a WordNet concept as a function of the hyponyms it has. Formally we have:

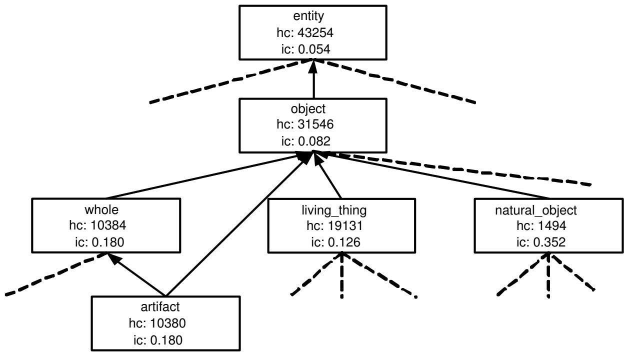

Figure 1: An example of multiple inheritance in the upper taxonomy of WordNet. **ic** and **hc** stand for Information Content and Hyponym Count respectively.

$$ic_{wn}(c) = \frac{log(\frac{hypo(c)+1}{max_{wn}})}{-log(max_{wn})} \quad (6)$$

where the function *hypo* returns the number of hyponyms of a given concept and max_{wn} is a constant that is set to the maximum number of concepts that exist in the taxonomy[4]. The denominator, which is equivalent to the value of the most informative concept, serves as normalizing factor in that it assures that IC values are in $[0, .., 1]$. The above formulation guarantees that IC decreases monotonically as we transverse from the leaf nodes to the root nodes as can be observed in figure 1. Moreover, the IC of the imaginary top node of WordNet would yield an information content value of 0.

As result of multiple inheritance in some of WordNet's concepts, caution must be taken so that each distinct hyponym is considered only once. Consider again the situation in figure 1, the concept *artifact* is an immediate hyponym of *whole* and *object*. Since *whole* is also a hyponym of *object* we must not consider the hyponyms of *artifact* twice when calculating the number of hyponyms of *object*.

Obviously, this metric gives the same score to all leaf nodes in the taxonomy regardless of their overall depth. As a consequence of this, concepts such as *blue sky* and *mountain rose* both yield a maximum IC value of 1 despite one being at a two link depth and the other at a nine link depth in the taxonomy, which is in accordance with our initial assumption. However, some counter examples do exist that disagree with the assumption; take the concept *anything* which is a leaf node thus yielding maximum IC. Qualitatively analyzing the amount of information conveyed by this concept may lead us to question the score given by our metric which indeed seems to over exaggerate. But yet another perspective may lead us to ask: "Why weren't any nodes considered as hyponyms of *anything*?" Whatever the answer may be, we must recognize that certain commitments had to be made by the designers of WordNet and

[1]Note that we avoid using the word *inverse* which may be misleading. If one were to simply mathematically inverse the distance this would alter the magnitude of the resulting correlation coefficient. Suppose w_1 and w_2 represent the same concept hence have a semantic distance of 0, consider also that between w_3 and w_4 there is a distance of 1. If one were to consider the mathematical inverse function this would profoundly alter the magnitude of comparison. In the distance scenario we have a difference of 1 between the two pairs; in the similarity scenario we obtain a difference of infinity between the two.

[2]Which is also the most widely observed configuration in the literature.

[3]Note that the MSCA that subsumes c_1 and c_1 is c_1.

[4]There are 79689 noun concepts in WordNet 2.0.

that these may not always match our present needs. Irrespective of this fact, in some NLP tasks like Information Retrieval where SS is essential, we will find that words like *anything, nothing , something, ...* which yield exaggerated IC scores are frequently stored in *stop word lists* and are ignored, which will somewhat attenuate these apparent contradictions.

4. Empirical Studies

In order to evaluate our IC metric we decided to use the three formulations of SS presented in section 2. and substituted Resnik's IC metric with the one presented in equation 6. In accordance with previous research, we evaluated the results by correlating our similarity scores with that of human judgments provided by (Miller and Charles, 1991). In their study, 38 undergraduate subjects were given 30 pairs of nouns and were asked to rate similarity of meaning for each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The average rating for each pair represents a good estimate of how similar the two words are.

In order to make fair comparisons we decided to use a independent software package that would calculate similarity values using previously established strategies while allowing the use of WordNet 2.0. One freely available package is that of Siddharth Patwardhan and Ted Pederson[5]; which implement semantic relatedness measures described by (Leacock and Chodorow, 1998), (Jiang and Conrath, 1998), (Resnik, 1995), (Lin, 1998), (Hirst and St-Onge, 1998), (Wu and Palmer, 1994) and the adapted gloss overlap measure by (Banerjee and Pedersen, 2003). Despite our focus being on SS, a special case of Semantic Relatedness, we decided to also evaluate how all of these algorithms would judge the similarity of the 30 pairs of words using WordNet 2.0. In addition to these we also used Latent Semantic Analysis (Landauer et al., 1998) to perform similarity judgments by means of a web interface available at the LSA website[6].

Table 4.1. presents the similarity values obtained with the chosen algorithms and their correlation factor with human judgments. Each of the capital letters heading each column represents a different semantic relatedness algorithm. The columns are organized in following manner:

- A — The data gathered by Miller and Charles Regarding human Judgments.
- B — The results obtained using the independent implementation of the Leacock Chodorow measure.
- C — The results obtained using the independent implementation of the simple edge-counts measure.
- D — The results obtained using the independent implementation of the Hirst St. Onge measure.
- E — The results obtained using the independent implementation of the Jiang Conrath measure.
- F — The results obtained using the independent implementation of the adapted gloss overlap measure.
- G — The results obtained using the independent implementation of the Lin measure.
- H — The results obtained using the independent implementation of the Resnik measure.
- I — The results obtained using the independent implementation of the Wu Palmer measure.
- J — The results obtained using the independent implementation of the LSA measure.
- K — The results obtained using our implementation of the Resnik measure.
- L — The results obtained using our implementation of the Lin measure.
- M — The results obtained using our implementation of the Jiang Conrath measure.

It should be noted that in two of the configurations, namely E and G, two word pairs were not considered in the correlation calculation. This is due to the fact that SemCor, a small portion of the Brown Corpus, was used in obtaining the concept frequencies to calculate the IC values. SemCor is a relatively small sized corpus which contains about 25% of the existing nouns in WordNet. The word *crane* (nor none of its hyponyms) that appear twice in the Miller dataset does not appear in the corpus, thus no IC value may derived for the word. Due to this fact we decided to ignore the entries that would need these values in their assessment and calculated correlation without considering them.

One last observation regarding our implementations must be made before we discuss the results. Using Resnik's and Lin's formulas yields results in $[0, .., 1]$ where 1 is maximum similarity and 0 corresponds to no similarity whatsoever. However, Jiang and Conrath's measure is a measure of semantic distance, in order to maintain the coherency of our implementations we decided to apply a linear transformation on every distance value in order to obtain a similarity value[7]. Yet this transformation will only yield similarity values instead of distance, so normalization factor was also required in order to constrain the output to values to $[0, .., 1]$. The resulting formulation is:

$$sim_{jcn}(c_1, c_2) = 1 - \left(\frac{ic_{wn}(c_1) + ic_{wn}(c_2) - 2 \times sim_{res'}(c_1, c_2)}{2}\right) \quad (7)$$

Note that $sim_{res'}$ corresponds to Resnik's similarity function but now accommodating our IC values.

4.1. Discussion of Results

Observing table 4.1. we see that the algorithms performed fairly well. Established algorithms for which there are published results regarding the Miller compilation appear to be the same. The results obtained using our IC

[5]This software can be downloaded at http://www.d.umn.edu/ tpederse/.

[6]The web interface can be accessed at http://lsa.colorado.edu/.

[7]This transformation will not change the magnitude of the resulting correlation coefficient, although its sign may change from negative to positive (Jiang and Conrath, 1998).

values in the information theoretic formulas (K, L and M) seem to have outperformed their homologues (H, G and E), which suggests that the initial assumption concerning the taxonomic structure of WordNet is correct. It should be noted that the maximum value obtained, using Jiang and Conrath's formulation, is very close to what (Resnik, 1999) proposed as a computational upper bound. Reproducing the experiment performed by Jiang and Conrath where they removed the pair *furnace — stove* from their evaluation claiming that MCSA for the pair is not reasonable[8], we obtain a correlation value of 0,87.

5. Similarity in Creative Recategorization

Considering the high correlation value obtained with configuration M and the hierarchical nature of the metric we believe that it is an ideal candidate to fulfill the role of σ presented in equation 1. As a starting point for the validation of the above hypothesis, we conducted an exploratory experiment in which we generate new recategorizations and then assess their creative value by substituting σ in equation 1 with the SS metric used in configuration M. The recategorizations are generated by a process dubbed **Category Broadening** (Veale, 2004).

As an example of this process imagine we want to broaden the WordNet category *weapon*. The members of this category can be enumerated by recursively visiting every hyponym of the category, which will include *knife*, *gun*, *artillery*, *pike*, etc. But by traversing polysemy links as well as **isa** relations, additional prospective members can be reached and admitted on the basis of their functional potential. Thus, the polysemy of *knife* causes not only *dagger* and *bayonet* but *steak knife* and *scalpel* to be visited. Stretching category boundaries even further, we may generalize that all *edge tools* maybe considered *weapons*, thereby allowing *scissors*, *ax*, *razor* and all other sharp-edged tools to be recognized as having weapon-like potential.

At the heart of the broadnening process is the use of polysemy links. Since WordNet does not contain these links explicitly a patchwork of polysemy detectors are needed. As such we implemented the polysemy detectors presented in (Mihalcea and Moldovan, 2001) and (Veale, 2004) to find the needed facilitating links. The new domain pointers of WordNet 2.0 were also used; basically we consider that if two senses of the same word belong to same domain then they are polysemous. We then applied the broadening process described above to the WordNet 2.0 noun hierarchy and divided the generated recategorizations into 3 groups according to their creative value:

- High — the creative value of the recategorization is in $[0.66, 1]$.
- Medium — the creative value of the recategorization is in $[0.33, 0.66[$.
- Low — the creative value of the recategorization is in $[0, 0.33[$.

Some examples from each of these groups are given in table 2.

6. Analogical Similarity

Analogy is regarded as an important creative reasoning mechanism, as such we feel that extending our metric to deal with analogical similarity is very appealing. Obviously, a simple taxonomic metric will not be able to capture some of the deep similarities of an analogical insight, but taxonomic cues do exist that may shed some light on a potential analogy. As suggested by (Veale, submitted manuscript), WordNet defines *seed* as hyponym of *reproductive structure* and *egg* as a hyponym of *reproductive cell*. Reproduction is thus the unifying theme of the analogy $\{$seed-plant; egg-bird$\}$. The strict taxonomic similarity between *seed* and *egg* is very low yielding a value of 0.37, as their lowest common WordNet hypernym is the root node *entity*. However, if *reproductive structure* and *reproductive cell* are treated as equivalent by considering the average of their IC values as the IC value of a hypothetical analogical pivot we obtain a value of 0.88. We feel this value indicates the analogical similarities between *egg* and *seed*.

7. Conclusion and Future Work

Obviously, the use of such a small dataset does not allow us to be conclusive regarding the true correlation between computational approaches of SS and human judgments of similarity. Nevertheless, when our IC metric is applied in previously established semantic similarity formulations, we find a very motivating quislingism. One major advantage of this approach is that it does not rely on corpora analysis, thus we avoid the sparse data problem which was evident in these experiments when judging pairs that contained the word *crane*.

Future work will consist of a more thorough evaluation of our metric regarding both its literal facet and also its potential to evaluate creative recategorizations. Another aspect that will also deserve our future attention is the application of our metric to other taxonomic knowledge bases (e.g. Gene Ontology), allowing us to conclude if our intuition about IC is generalizable to other taxonomic resources.

8. References

Banerjee, Satanjeev and Ted Pedersen, 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Acapulco, Mexico.

Bento, Carlos and Amilcar Cardoso, 2004. Studying creativity in ai. *Fachbereich Kunstliche Intelligenz der Gesellschaft fur Informatik*:45–46.

de Bono, E., 1994. *Parallel Thinking*. London: Viking Press.

Gangemi, Aldo, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider, 2002. Sweetening ontologies with dolce. In *Proceeding of the European Workshop on Knowledge Acquisition, Modeling, and Management*.

[8] We agree with their claim in that a more informative subsumer should have been chosen, but we also think that algorithms dealing with manually constructed knowledge bases must be able to deal with these situations as they are inescapable. Fortunately, some research has emerged that looks for these inconsistencies allowing a restructure of the taxonomy ((Veale, 2003), (Gangemi et al., 2002)).

Algorithm		A	B	C	D	E	F	G	H	I	J	K	L	M
car	automobile	3,92	3,47	1,00	16,00	0,00	9577,00	1,00	6,11	0,89	0,60	0,68	1,00	1,00
gem	jewel	3,84	3,47	1,00	16,00	0,00	2297,00	1,00	10,52	0,86	0,21	1,00	1,00	1,00
journey	voyage	3,84	2,77	0,50	4,00	4,95	192,00	0,69	5,82	0,92	0,43	0,66	0,84	0,88
boy	lad	3,76	2,77	0,50	5,00	3,41	154,00	0,82	7,57	0,80	0,43	0,76	0,86	0,88
coast	shore	3,70	2,77	0,50	4,00	0,62	336,00	0,97	8,93	0,91	0,40	0,78	0,98	0,99
asylum	madhouse	3,61	2,77	0,50	4,00	0,41	104,00	0,98	11,50	0,82	0,12	0,94	0,97	0,97
magician	wizard	3,50	3,47	1,00	16,00	0,00	976,00	1,00	11,91	0,80	0,29	0,80	1,00	1,00
midday	noon	3,42	3,47	1,00	16,00	0,00	152,00	1,00	10,40	0,88	0,59	1,00	1,00	1,00
furnace	stove	3,11	1,39	0,13	5,00	18,13	202,00	0,220	2,56	0,46	0,28	0,18	0,23	0,39
food	fruit	3,08	1,39	0,13	0,00	11,65	128,00	0,13	0,86	0,22	0,39	0,05	0,13	0,63
bird	cock	3,05	2,77	0,50	6,00	3,76	200,00	0,80	7,74	0,94	0,38	0,40	0,60	0,73
bird	crane	2,97	2,08	0,25	5,00	*	102,00	*	7,74	0,84	0,31	0,40	0,60	0,73
tool	implement	2,95	2,77	0,50	4,00	1,23	542,00	0,92	7,10	0,91	0,13	0,42	0,93	0,97
brother	monk	2,82	2,77	0,50	4,00	14,90	503,00	0,25	10,99	0,92	0,03	0,18	0,22	0,33
crane	implement	1,68	1,86	0,20	3,00	*	51,00	*	3,74	0,67	-0,05	0,24	0,37	0,59
lad	brother	1,66	1,86	0,20	3,00	12,47	28,00	0,29	2,54	0,60	0,24	0,18	0,20	0,28
journey	car	1,16	0,83	0,07	0,00	11,93	158,00	0,00	0,00	0,00	0,10	0,00	0,00	0,00
monk	oracle	1,10	1,39	0,13	0,00	17,42	35,00	0,23	2,54	0,46	0,06	0,18	0,22	0,34
cemetery	woodland	0,95	1,16	0,10	0,00	19,75	21,00	0,08	0,86	0,18	-0,01	0,05	0,06	0,19
food	rooster	0,89	0,83	0,07	0,000	15,19	38,00	0,10	0,86	0,13	0,03	0,05	0,08	0,40
coast	hill	0,87	1,86	0,20	4,00	5,37	123,00	0,71	6,57	0,67	0,05	0,50	0,63	0,71
forest	graveyard	0,84	1,16	0,10	0,00	18,70	25,00	0,08	0,86	0,18	-0,01	0,05	0,06	0,19
shore	woodland	0,63	1,67	0,17	2,00	17,00	78,00	0,14	1,37	0,44	0,14	0,08	0,11	0,30
monk	slave	0,55	1,86	0,20	3,00	15,52	73,00	0,25	2,54	0,60	-0,02	0,18	0,23	0,39
coast	forest	0,42	1,52	0,14	0,00	17,60	89,00	0,13	1,37	0,40	0,14	0,08	0,10	0,29
lad	wizard	0,42	1,86	0,20	3,00	13,60	13,00	0,27	2,54	0,60	0,20	0,18	0,21	0,32
chord	smile	0,13	1,07	0,09	0,00	14,86	31,00	0,27	2,80	0,44	0,05	0,25	0,28	0,35
glass	magician	0,11	1,39	0,13	0,00	18,07	57,00	0,13	2,50	0,36	0,14	0,18	0,20	0,31
noon	string	0,08	0,98	0,08	0,00	18,32	16,00	0,00	0,00	0,00	0,09	0,00	0,00	0,00
rooster	voyage	0,08	0,47	0,05	0,00	21,61	16,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00
Correlation		1,00	0,82	0,77	0,68	-0,81	0,37	0,80	0,77	0,74	0,72	0,77	0,81	0,84

Table 1: Results obtained evaluating correlation with human judgments using several algorithms and WordNet 2.0.

High	Medium	Low
dog collar isa *tie*	*cigar band* isa *necklace*	*dancing* isa *performance*
plane ticket isa *leave of absence*	*smoking room* isa *hiding place*	*coat* isa *plumage*
priest doctor isa *sorcerer*	*scissors* isa *weapon*	*outdoorsman* isa *worker*

Table 2: Some examples of creative recategorizations grouped by their creative value.

Hirst, Graeme and David St-Onge, 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, chapter 13. MIT Press, pages 305–332.

Hutton, J., 1982. *Aristotle's Poetics*. New York: Norton.

Jiang, J. and D. Conrath, 1998. Semantic similarity based on corpus statistics and lexical taxonomy.

Landauer, T. K., P. W. Foltz, and D. Laham, 1998. Introduction to latent semantic analysis. *Discourse Processes*:259–284.

Leacock, C. and M. Chodorow, 1998. Combining local context and wordnet similarity for word sense identification. In Christiane Fellbaum (ed.), *WordNet: An Electronic Lexical Database*. MIT Press, pages 265–283.

Lenat, Douglas B. and R. V. Guha, 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Reading, Massachusetts: Addison-Wesley.

Lin, Dekang, 1998. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA.

Manning, Christopher D. and Hinrich Schütze, 1999. *Foundations of statistical natural language processing*. MIT Press.

Mihalcea, Rada and Dan Moldovan, 2001. Ez.wordnet: Principles for automatic generation of a coarse grained wordnet. In *Proceedings of Flairs 2001*.

Miller, George, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller, 1990. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235 – 244.

Miller, George and W.G. Charles, 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6:1–28.

Resnik, Philip, 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*.

Resnik, Philip, 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of*

Artificial Intelligence Research, 11:95–130.

Shannon, C.E., 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.

Torrance, E. P., 1990. *The Torrance Tests of Creative Thinking*. Illinois: Bensonville.

Veale, Tony, 2003. The analogical thesaurus: An emerging application at the juncture of lexical metaphor and information retrieval. In *Proceedings International Conference on Innovative Applications of Artificial Intelligence*.

Veale, Tony, 2004. Pathways to creativity in lexical ontologies. In *Proceedings of the 2nd Global WordNet Conference*.

Wiggins, Geraint, 2003. Categorizing creative systems. In *Proceedings of 3rd Workshop on Creative Systems*.

Wu, Z. and M. Palmer, 1994. Verb semantics and lexical selection.

A Description Logic Ontology for Fairy Tale Generation

Federico Peinado, Pablo Gervás, Belén Díaz-Agudo

Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
fpeinado@fdi.ucm.es, {belend,pgervas}@sip.ucm.es
http://gaia.sip.ucm.es

Abstract

The combination of resources like Ontologies and an inference formalism such as Description Logics has proved very useful for generating semantically correct texts. However the possibilities of applying such combinations to obtain results in practical situations is restricted by the availability of ontological resources for the domains under consideration. This paper presents work on the development of an OWL ontology based on Propp's *Morphology of the Folk Tale* oriented towards automatic story generation. The ontology is designed so that it allows measurement of the semantical distance between narrative functions. We explain how to use this resource to generate creative and meaningful stories.

1. Introduction

Certain properties of structured domains, like the syntax of formal poetry, make them particularly suitable to modeling in terms that allow automatic generation of elements belonging to that domain. This may be achieved by applying formal techniques of knowledge representation like Ontologies and Description Logics (DL). We have found ontologies and description logics a very powerful combination as a resource for generating linguistically creative correct texts (Díaz-Agudo et al., 2002). However the possibilities of applying such combinations to obtain results in practical situations is restricted by the availability of ontological resources for the domains under consideration. This paper presents work on the development of an OWL ontology oriented towards automatic story generation.

Automatic construction of story plots has always been a longed-for utopian dream in the entertainment industry, specially in the more commercial genres that are fuelled by a large number of story plots with only a medium threshold on plot quality, such as TV series or story-based video games.

The work of russian formalist Vladimir Propp on the morphology of folk tales (Propp, 1968) provides a formalism to describe the composition of folk tales as a structured domain. In this paper we describe the conversion of Propp's morphology into OWL description logic format (Bechhofer et al., 2004). The choice of OWL as representation language provides the additional advantage, that it is designed to work with inference engines like RACER (Haarslev and Möller, 2003), and that it is easily connected with Protégé (Gennari et al., 2002). This constitutes an extremely powerfull development environment, well suited for exploring linguistic creativity, and we hope to use it for exploring issues of story generation.

The resulting resource is employed as underlying representations for a Knowledge Intensive Case-Based Reasoning (KI-CBR) approach to the problem of generating story plots from a case base of Propp functions. A CBR process is defined to generate plots from a user query specifying an initial setting for the story, using the ontology to measure the semantical distance between words and structures taking part in the texts.

2. Theories and Implementations of Plot Generation

The automatic generation of stories requires some representation for plot structure and how it is built up from primitives, a computational solution to generating stories from a given input, and the choices of some format for presenting the resulting plots that is easy to understand and to generate.

2.1. General Theories on Plot Generation

In the first chapters of Seymour Chatman's *Story and Discourse* (Chatman, 1986) there is a review of various classical theories about narrative structures. Janet Murray shows another short review in the seventh chapter of her popular book *Hamlet on the Holodeck* (Murray, 1997). For example, she mentions Joseph Campbell's morphology of the mythic "hero (Campbell, 1972).

Our work is based on the work of Vladimir Propp (Propp, 1968), because it is easy to understand and translate into a machine-processable representation (the author brings us his own formal naming system). However there are other theories (Lakoff, 1972; Barthes, 1966) that propose more complex grammars and "deeper representations.

Propp's original goal was to derive a morphological method of classifying tales about magic, based on the arrangements of 31 "functions". The result of Propp's work is a description of the folk tales according to their constituent parts, the relationships between those parts, and the relations of those parts with the whole. Propp's work has been used as a basis for a good number of attempts to model computationally the construction of stories.

The main idea is that folk tales are made up of ingredients that change from one tale to another, and ingredients that do not change. According to Propp, what changes are the names - and certain attributes - of the characters, whereas their actions remain the same. These actions that act as constants in the morphology of folk tales he defines as *functions*.

For example, some Propp functions are: Villainy, Departure, Acquisition of a Magical Agent, Guidance, Testing of the hero, etc. There are some restrictions on the choice of functions that one can use in a given folk tale, given by implicit dependencies between functions: for instance, to be able to apply the Interdicion Violated function, the hero must have received an order (Interdiction function).

The Proppian fairy tale Markup Language (PftML) (Malec, 2004) is an XML application developed by University of Pittsburgh's researchers based on Propp's work. PftML utilizes a Document Type Definition (DTD) to create a formal model of the structure of Russian magic tale narrative and to help standardize the tags throughout a corpus when analyzing it. As a test corpus, they have used a subset of the same Russian language corpus from which Propp drew, since it allows for an empirical test of the conclusions of Propp's initial analysis against the original data.

We have used PftML, together with Propp's original work, as the basic sources for building the ontology that underlies our system.

2.2. Computer Models for Narrative

There have been various attempts in the literature to obtain a computational model of story generation. Important efforts along these lines are presented in (Meehan, 1981; Rumelhart, 1975; Lang, 1997; Callaway and Lester, 2002).

Fairclough and Cunningham (Fairclough and Cunningham, 2003) implement an interactive multiplayer story engine that operates over a way of describing stories based on Propp's work, and applies case-based planning and constraint satisfaction to control the characters and make them follow a coherent plot.

Of particular interest is their definition of a plot as a series of character functions and a series of complication-resolution event pairs, where a complication occurs whenever a character performs a function that alters the situation of the hero. A case based reasoning solution is used for storyline representation and adaptation. They use 80 cases extracted from 44 multi-move story scripts given by Propp. These scripts are defined as lists of character functions. There are stories composed of one, two or more moves. A case is a move, seen asa story template, to be filled in by a constraint satisfaction system that chooses which characters perform the functions - *casting*.

2.3. Template-based Natural Language Generation

The natural format for presenting a plot to users is to describe it - or rather narrate it - in natural language. Obtaining a high quality natural language text for a story is itself a subject of research even if the plot is taken as given (Callaway and Lester, 2002). This paper is concerned strictly with the process of generating valid plots, and only the simplest sketch of a natural language rendition is attempted as means of comfortably presenting the results. This is achieved by means of natural language generation (NLG) based on templates. The conventionalized patterns that make up common texts are encapsulated as *schemas* (McKeown, 1982), template programs which produce text plans. The basic resource required to apply this type of solution is a set of templates, obtained from the analysis of a corpus of example texts.

As in template-based NLG, Case-Based Reasoning (CBR) relies heavily on reusing previous solutions to solve new probles, drawing on a *case base* of existing problem-solution pairs enconded as *cases*. In (Díaz-Agudo et al., 2002) poetry generation is chosen as an example of the use of the COLIBRI (*Cases and Ontology Libraries Integration for Building Reasoning Infrastructures*) system. COLIBRI assists during the design of KI-CBR systems that combine cases with various knowledge types and reasoning methods. It is based on CBROnto (Díaz-Agudo and González-Calero, 2000; Díaz-Agudo and González Calero, 2001; Díaz-Agudo and González Calero, 2003), an ontology that incorporates reusable CBR knowledge and serves as a domain-independent framework to develop CBR systems based on generic components like domain ontologies and Problem Solving Methods (PSMs).

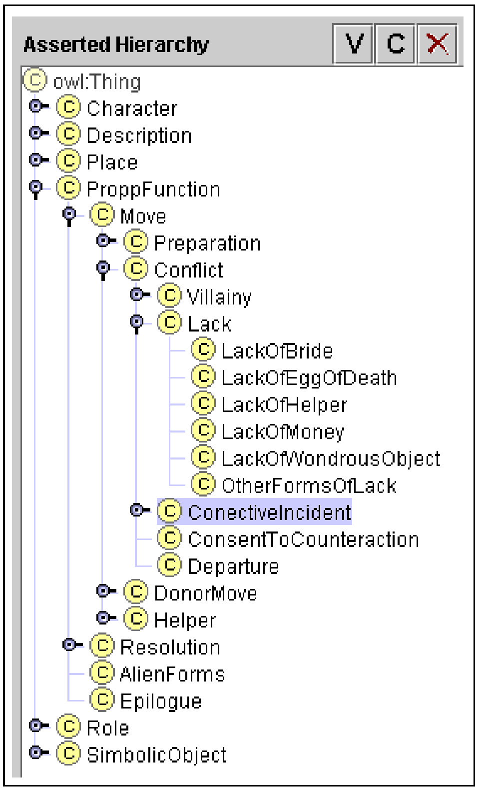

Figure 1: Function sub-hierarchy in the ontology as modelled in Protégé.

3. A DL Ontology for Fairy Tale Generation

Knowledge representation in our system is based on an ontology which holds the various concepts that are relevant to story generation. This initial ontology is subject to later extensions, and no claim is made with respect to its ability to cover all the concepts that may be necessary for our endeavour.

The ontology has been designed to include various concepts that are relevant to story generation. Propp's character functions are used as basic recurrent units of a plot. In order to be able to use them computationally, they have been translated into an ontology that gives semantic coherence and structure to our cases. A view of the top of the function hierarchy is given in figure 1.

Roles	Place	Character	Description	Simbolic Object
Agent Donor FalseHero Hero Prisoner Villain	City Country Dwelling	AnimatedObject Animal Human	Family description Human description Place description	Ring Towel

Table 1: Summary of additional subconcepts of the ontology

We have implemented this ontology using the last release of the Protégé ontology editor (Gennari et al., 2002), capable of managing ontologies in OWL(Bechhofer et al., 2004).

Although the functions of the *dramatis personae* are the basic components, we also have other elements. For instance, conjunctive elements, motivations, forms of appearance of the dramatis personae (the flying arrival of a dragon, the meeting with a witch), and the attributive elements or accessories (a witch's hut or her clay leg) (Propp, 1968).

This additional ontology provides the background knowledge required by the system, as well as the respective information about characters, places and objects of our world. This is used to measure the semantical distance between similar cases or situations, and mantaining a independent story structure from the simulated world. The domain knowledge of our application is the classic might-and-magic world with magicians, warriors, thieves, princesses, etc. The current version of the ontology contains a number of basic subconcepts to cover this additional domain knowledge that needs to be referred from within the represented function. Examples of these subconcepts are listed in table 1, including the character's roles proposed by Propp.

3.1. Propp's Terminology

In our approach, Propp's *character functions* act as high level elements that coordinate the structure of discourse. Each function has constraints that a character that is to perform it must satisfy. A view of the top of the function hierarchy is given in Figure 1.

The contents of a function are the answers to the Wh-questions: what (the symbolic object), when, where (the place), who (who are the characters of the function) and why.

Morphologically, a tale is a whole that may be composed of *moves*. A move is a type of development proceeding from villainy or a lack, through intermediary functions to marriage, or to other functions employed as a *denouement* (ending). Terminal functions are at times a reward, a gain or in general the liquidation of a misfortune, an escape from pursuit, etc. (Propp, 1968).

One tale may be composed of several moves that are related between them. One move may directly follow another, but they may also interweave; a development which has begun pauses, and a new move is inserted.

We represent tales and their composing moves using structured descriptions. A tale is related with an ordered sequence of complete moves. We represent the temporal sequence between these moves using the CBROnto temporal relations.

3.2. Background Knowledge

The ontology includes a significant amount of background knowledge needed for the successful application of the rest of its structure to the problem in hand.

Certain *locations* can be significant to the way a story develops (outdoors, indoors, country, city, lake, forest ...), and any sort of substitution during adaptation must take this into account. Our ontology must have the ability to classify such locations.

The roles in the story must be filled by *characters*. Each character is defined by a set of relationships with other characters, objects in his possession, location... These characters are one of the elements that the user can choose to customize a story.

The *descriptions* are represented in the ontology in such a way that their relations with the relevant concepts are modelled explicitly. This ensures that the inference mechanisms available can be employed to select the correct descriptions during the template-based NLG process which obtains a textual rendition of the plot.

The *properties or attributes of the characters* are the totality of all their external qualities: their age, sex, status, external appearance, peculiarities of this appearance,... These attributes provide the tale with its brilliance, charm and beauty. However, one character in a tale is easily replaced by another (permutability law) (Propp, 1968).

3.3. The Case Base

The case base is built up of texts from the domain of fairy tales, analyzed and annotated according to Propp's morphology. A selection of stories from the original set of the Afanasiev compilation originally used by Propp are taken as sources to generate our initial case base.

We use a structural CBR approach that relies on cases that are described with attributes and values that are predefined, and structured in an object-oriented manner. This structural CBR approach is useful in domains (like the one we are considering) where additional knowledge, beside cases, must be used in order to produce good results. The domain ontology insures that new cases are of high quality (regarding the ontology commitments) and the maintenance effort is low.

Within the defined case structures we represent the plots of the fairy tales. Besides this structural representation of the cases we also associate a textual representation to each case that can be used to generate texts from the plot descriptions (see Section 4.2.).

Cases are built based on CBROnto case representation structure (Díaz-Agudo and González Calero, 2003) using the vocabulary from the domain ontology. The semantic constraints between scene transitions are loosely based on the ordering and co-occurrence constraints established between Proppian functions.

CBROnto provides a *primitive* concept CASE. System designers will have to define instances of different CASE subconcepts to represent any new types of cases. There are different level of abstraction that allow the description of cases that are part of other cases.

In our application each case represents a complete tale that is typically composed of one or more interrelated moves (that are also cases). For representational purposes, relation between moves are basically of two types: *temporal* relations (before, after, during, starts-before, ends-before, ...) or *dependencies* (meaning that a change in one of them strongly affects the other) like *place-dependency*, *character-dependency* and *description-dependency* (Díaz-Agudo and González Calero, 2001).

DLs allows representing hierarchies between relations (see Figures 2 and 3), which allows easy definition of reasoning methods (using the top level relation) that are applicable (and reusable) with all the sub-relations.

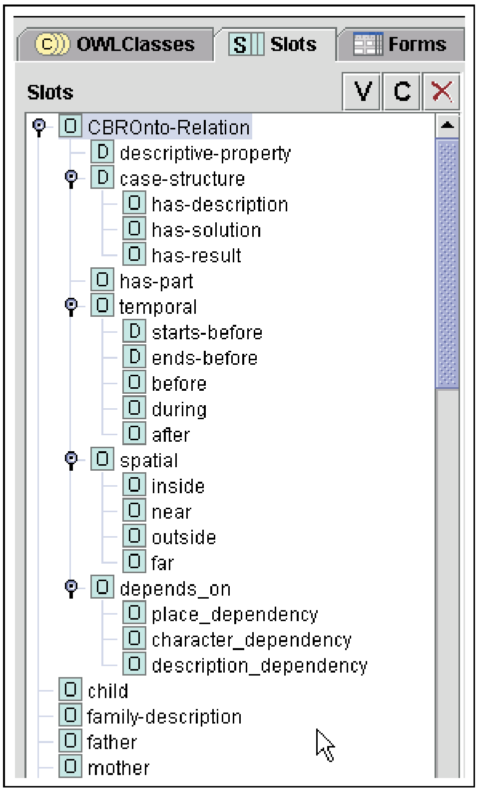

Figure 2: CBROnto relation hierarchy in Protege

As an example of the type of stories that are being considered, the following outline of one of the tales that Propp analyzes is given below [1]. The main events of the plot are described in terms of character functions (in bold) :

> *The Swan Geese (113 of Afanasiev Collection).* **Initial situation** (a girl and her small brother). **Interdiction** (not to go outside), **Interdiction violated**, **Kidnapping** (swan geese take the boy to Babayaga's lair), **Competition** (girl faces Babayaga), **Victory**, **Release from captivity**, **Test of hero** (swan geese pursue the children), **Sustained ordeal** (children evade swan geese), **Return**.

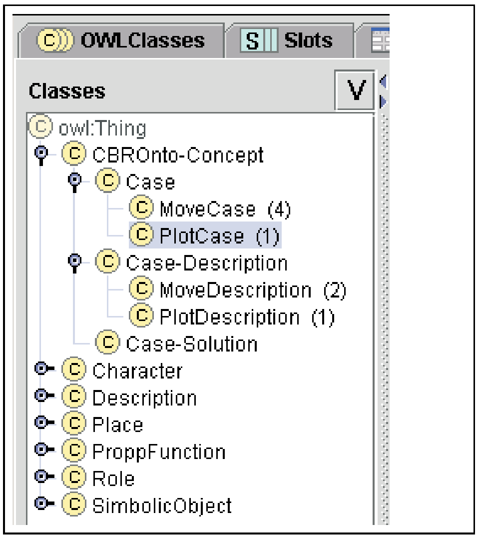

Figure 3: CBROnto concept hierarchy in Protege

4. Ontologies and Case Base Reasoning in Plot Generation

The resources that are described in this paper are applied to the problem of generating story plots in two phases: an initial one that applies CBR to obtain a plot plan from the conceptual description of the desired story provided by the user, and a final phase that transforms the resulting plot plan into a textual rendition by means of template based NLG.

4.1. The First Stage: Description to Plot Plan

We use the descriptive representation of the tale plots with a CBR system, that retrieves and adapts these plots in several steps using the restrictions given in the query.

A query determines the components of the tale we want to build. For example, its characters, descriptive attributes, roles, places, and the Propp functions describing the actions involved in the tale. Although there are roles whose existence (a character that plays that role) is mandatory in every plot, like the hero and the villain, they are not required in the query as they can be reused from other plots (cases).

In a query the user describes: the tale characters, roles, places, attributes, the set of character functions that are to be involved in the story,and (optionally) which characters take part in each function.

This is done by selecting individual (predefined instances) from the ontology (see Figure 1) or creating new

[1] Complete text in:
http://gaia.sip.ucm.es/people/fpeinado/swan-geese.html

ones (new instances of the types of characters or places given by the ontology). The knowledge in the ontology (and the associated reasoning processes) can help the user in this selection while maintaining the corresponding restrictions.

The system retrieves the most similar case to the query which constitutes a plot-unit template. The components of the retrieved case are substituted for information obtained from the context, i.e. the query, the ontology and other cases, during the adaptation process.

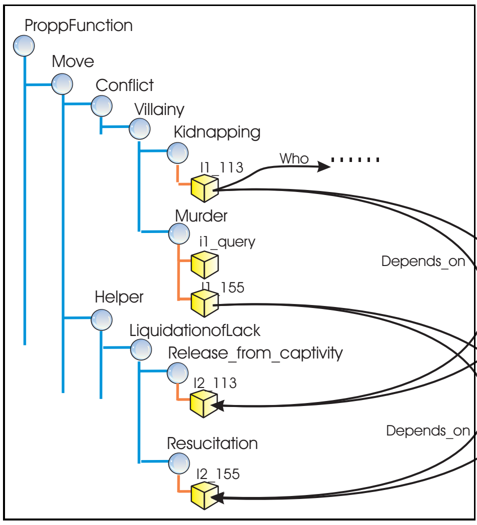

Figure 4: Substitution example

For instance, let us say we want a story about a *princess*, where **Murder** occurs, where an **Interdiction** is given and **Violated**, there is a **Competition**, and a **Test of the hero**. We can use that information to shape our query. The system retrieves the case story number 113, Swan-Geese (presented in the previous section).

Retrieval has occurred because the structure of this story satisfies straight away part of the conditions (interdiction, competition, test of hero) imposed by the query. No murder appears, but there is a *similar* element: a kidnapping. **Kidnapping** and **Murder** are similar because they are different types of villainies; so, they are represented as children of the same concept **Villainy** in the ontology.

The retrieval process provides the plot skeleton where the system makes certain substitutions. A basic and simple initial adaptation step is to substitute the characters given in the query into the template provided by the retrieved case. This is equivalent to Fairclough and Cunnigham's process of *casting*.

A more elaborate adaptation may be achieved by generating a solution as a mixture of the ingredients from various cases. During the adaptation of our *plot case*, we use additional retrieval steps (defining adequate queries) over the case base of *move cases* (that are part of the plot cases) to find appropriate substitutes maintaining the dependencies and temporal relations.

In our example, the system may suggest an adaptation where **Murder** is substituted for the **Kidnapping**. However, the **Kidnapping** in the retrieved case has *dependencies* with the **Release from captivity** that appears later on (which is a **Liquidation of lack** according to the ontology) (see Figure 4). To carry out a valid adaptation, the adaptation process is forced to define a query and retrieve cases in which **Murder** appears with a *similar* dependency (i.e. dependency with another **Liquidation of lack**).

The following case is retrieved (only a part of which is relevant to the issue):

> (*155 of Afanasiev Collection*). (...) **Absentation** of the hero (brother goes hunting), **Deception** of the villain (beautiful girl entices him), **Murder** (girl turns into lioness and devours him), (...) **Consent to counteraction** (other brother sets out), **Competition** (faces beautiful girl), **Victory** (kills lioness), **resuscitation** (revives brother), **Return**.

In this case there is a dependency between the **Murder** and the **Resuscitation**. The adaptation system can therefore substitute the kidnapping-release pair in the first retrieved case with the murder-resuscitation pair in the second, obtaining a better solution for the given query. Additional adaptations can be carried out to substitute the hero of the first case (the girl) or the prisoner (the boy) for the princess specified in the query. Besides, the swan-geese character in the retrieved case can be substituted for a similar element (for instance, another animal like the lioness that appears in the second retrieved case). The second part of *The Swan-Geese* story is not possible because of the lioness' death.

The resulting plot could be a story like this:

> *The Lioness (new fairy tale).* **Initial situation** (a knight and his beloved princess). **Interdiction** (not to go outside), **Interdiction violated**, **Murder** (a lioness devours her), **Competition** (knight faces the lioness), **Victory** (kills lioness), **Resuscitation** (revives the princess), **Return**.

4.2. The Second Stage: Plot Plan to Textual Sketch

A readable rendition of the plot plan is obtained by applying template-based natural language generation. The second stage takes as input a data structure satisfying the following constraints:

- The case that has been selected during retrieval, has been pruned or combined with other cases retrieved during adaptation and to make up a plot skeleton.
- The character functions, acting as templates for the basic units of the plot, have been filled in during adaptation with identifiers for the characters described in the query

A one-to-one correspondence can be established between character functions in the plot plan and sentence templates to be expected in the output and a simple stage of surface realization is applied to the plot plan. This stage converts the templates into strings formatted in accordance to the orthographic rules of English - sentence initial letters are capitalized, and sentences are ended with a colon.

The fact that we are using an ontology to represent concepts, and not a set of axioms encoding their meaning somehow restricts the degree of correctness that can be guaranteed by the substitution process. Any checking algorithm can only test for structural equivalence within the ontological taxonomy, and it cannot carry out proper inference over the meanings of concepts.

5. Conclusions

A major point of discussion that should be taken into account is whether Propp's formalism does constitute a generic description of story morphology. Without entering into that discussion here, it is still necessary to consider whether the procedure described in the paper enables the system to build new stories in a creative manner, or whether it simply allows reinstantiation of those in the original corpus with new elements. Unlike the uses of Proppian functions in other systems, our approach represents character functions with more granularity. This allows the establishment of relations between characters and attributes and the functions in which they appear. Using this facility, a coherent character set can be guaranteed throughout the story. Additionally, dependencies between character functions are modeled explicitly, so they can be checked and enforced during the process of plot generation without forcing the generated plots to be structurally equivalent to the retrieved cases.

The coverage of the ontology is an open issue dependent on whether one accepts Propp's set of character functions as complete. In the face of disagreement, the ontology is easy to extend, and, as mentioned before, it is not intended to be complete as it is. Under these conditions, the approach described in this paper may be extend to work in other domains.

Systems attempting to model linguistic creativity in the field of story generation would greatly benefit from incorporating semantic information in the form of a knowledge rich ontology such as the one described here. In future work we intend to address the specific problems of the natural language generation, involving the transition from plot plan to textual sketch, and to explore the possible interactions between the two stages.

6. Acknowledgements

The first author is supported by a FPI Predoctoral Grant from Complutense University of Madrid. The work was partially funded by the Spanish Committee of Science & Technology (TIC2002-01961).

7. References

Barthes, Roland, 1966. Introduction à lanalyse structurale des récits (in french). *Communications*, 8:1–27.

Bechhofer, Sean, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, and Andrea Stein, 2004. OWL web ontology language reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

Callaway, Charles B. and J. C Lester, 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252.

Campbell, Joseph, 1972. *Myths to live by*. New York: Viking Press.

Chatman, Seymour, 1986. *Story and Discourse : Narrative Structure in Fiction and Film*. Cornell University Press, 2nd edition.

Díaz-Agudo, Belén, Pablo Gervás, and Pedro González-Calero, 2002. Poetry generation in COLIBRI. In S Craw and A Preece (eds.), *ECCBR 2002, Advances in Case Based Reasoning 6th European Conference on Case Based Reasoning*. Aberdeen, Scotland: Lecture Notes in Artificial Intelligence. Springer.

Díaz-Agudo, Belén and Pedro Antonio González-Calero, 2000. An architecture for knowledge intensive CBR systems. In Enrico Blanzieri and Luigi Portinale (eds.), *Advances in Case-Based Reasoning*. New York: Springer-Verlag Berlin Heidelberg.

Díaz-Agudo, Belén and Pedro Antonio González Calero, 2001. A declarative similarity framework for knowledge intensive CBR. In *International Conference on Case-Based Reasoning*. Springer-Verlag.

Díaz-Agudo, Belén and Pedro Antonio González Calero, 2003. Knowledge intensive CBR through ontologies. *Expert Update*, 6(1).

Fairclough, Chris and Pádraig Cunningham, 2003. A multiplayer case based story engine. In *GAME-ON Conference*.

Gennari, J, M. A Musen, R. W Fergerson, W. E Grosso, M Crubézy, H Eriksson, N. F Noy, and S. W Tu, 2002. The evolution of Protégé: An environment for knowledge-based systems development. Technical report, Stanford University.

Haarslev, Volker and Ralf Möller, 2003. Racer user s guide and reference manual version 1.7.7. Technical report, Concordia University and University of Applied Sciences in Wedel.

Lakoff, George P, 1972. Structural complexity in fairy tales. *The Study of Man*, 1:128–150.

Lang, R. R, 1997. *A Formal Model for Simple Narratives*. Ph.D. thesis, Tulane University.

Malec, Scott A, 2004. Proppian structural analysis and XML modeling. http://clover.slavic.pitt.edu/ sam/propp/theory/propp.html.

McKeown, K. R, 1982. The text system for natural language generation: An overview. In *20th Annual Meeting of the ACL*. University of Toronto, Ontario, Canada.

Meehan, James R, 1981. Tale-spin and micro tale-spin. In Roger C. Schank and Christopher K. Riesbeck (eds.), *Inside computer understanding*. Hillsdale, NJ: Erlbaum Lawrence Erlbaum Associates.

Murray, Janet Horowitz, 1997. *Hamlet on the Holodeck. The Future of Narrative in Cyberspace*. Cambridge, MA: MIT Press.

Propp, Vladimir, 1968. *Morphology of the Folktale*. Austin: University of Texas Press.

Rumelhart, David E, 1975. Notes on a schema for stories. In Daniel G Bobrow and Allan Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*. New York: Academic Press, Inc, pages 211–236.

An Automatic Method for Lexical Semantics Transformation

Francisco Camara Pereira*, **Pablo Gervás**†

*Centro de Informática e Sistemas da
Universidade de Coimbra, Portugal
camara@dei.uc.pt

†Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
pgervas@sip.ucm.es

Abstract

In many cases the functionality of a system with linguistic capabilities is restricted by the coverage or the nature of its resources. Given this situation, it seems reasonable to assume that if any creativity is to be expected from a linguistic-capable system, a number of "creativity specific" demands will be placed on the resources it is using. An easy way to tackle this problem might be to introduce some version of dynamical pre-processing of the resources, such that each run of the system operates on a creatively different version of the resource. Such pre-processing could produce an appropriately modified version of the resource that is better prepared to tackle the required tasks creatively. This paper outlines the role that Divago, a system that generates novel concepts through *conceptual blending* of existing ones, can play in such a pre-processing stage.

1. Introduction

Most forms of linguistic-related computation are knowledge intensive and place heavy demands on the resources - world model, grammars, lexicon, dictionary... - they employ. In many cases the functionality of a system with linguistic capabilities is restricted by the coverage or the nature of its resources. Given this situation, it seems reasonable to assume that if any creativity is to be expected from a linguistic-capable system, a number of "creativity specific" demands will be placed on the resources it is using. It is not easy to define the way in which a grammar or a lexicon can be creative, though some efforts have been made to sketch various possibilities (Gervás, 2002). To expect linguistic creativity from a computational system imposes heavy demands on the linguistic resources that it uses. An easy way to tackle this problem might be to introduce some version of dynamical pre-processing of the resources. This could ensure that each run of the system operates on a creatively different version of the resource. Such pre-processing might be driven or guided by the input to the system, and could be designed to produce an appropriately modified version of the resource that is better prepared to tackle the required tasks creatively. This paper outlines the role that Divago, a system that generates novel concepts through *conceptual blending* of existing ones, can play in such a pre-processing stage.

2. Structure of a Linguistic Resource

Throughout the history of the development of NLP systems, many solutions have been employed to model the linguistic information that such a system requires to operate in a satisfactory fashion. In recent times there has been a drive towards standardization of specific alternatives to this problem in terms of generally available linguistic resources. It is beyond the scope of the present paper to describe and discuss the various alternatives that have arisen, but two of them are presented here to illustrate relevant points.

A well established classic is WordNet (Miller, 1995). Although this linguistic resource originated as a side result from a set of psychological experiments, it has been widely used in the field of NLP, possibly due to its availability with no costs. In WordNet, English nouns, verbs, adjectives and adverbs - no closed class words are included - are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets.

A different approach is followed in MikroKosmos (Lonergan, 2001). The set of linguistic resources developed for the KBMT Machine Translation project (Nirenburg, 1987) consists of an ontology, an English lexicon, and a Spanish lexicon. The ontology organizes primitive symbols used in meaning representation in a tangled subsumption hierarchy, and a rich system of semantic relations defined among the concepts further interconnects these symbols between them. For each of the operative languages of the system, a lexicon is built of lexical terms co-indexed with the concepts of the ontology. In this way, the ontology can be used as inter-lingua for representing meaning at intermediate stages during translation.

When we consider the application of the dynamic preprocessing envisaged in this paper to linguistic resources of this type, many possible alterations can be considered: new symbols can be added, new connections between symbols can be added, existing symbols or connections can be destroyed, or existing connections can be modified. It is clear that, however creative the results for one particular ocasion, this type of modification is in general not desirable as a long term modification of the original linguistic resource. For this reason, the type of pre-processing proposed here is intended as dynamic, applied each time to the original copy of the linguistic resource - or indeed just to a selected subset or subsets of it - to obtain a creatively-warped view of it to be used for a particular purpose.

```
isa(guitar, instrument).          isa(woman, female).
made_of(guitar, wood).            made_of(woman, flesh).
have(guitar, strings).            have(woman, body).
made_of(string, nylon).           have(woman, hair).
produce(guitar, sound).           have(woman, eyes).
have(guitar, neck).               have(woman, neck).
have(guitar, body).               property(woman, beautiful).
have(guitar, bridge).             can(woman, sing).
have(body, ressonance_hole).
```

Figure 1: Concept maps representing *guitar and woman.*

3. DIVAGO

We will now present Divago. Its general motivation is to be a system that can wander (i.e. diverge) in a search space for concepts in the same way humans sometimes do. We will give a general overview of the aspects that are relevant for this paper, therefore we will leave out some of its foundations and specificities for the reader to find elsewhere (Pereira and Cardoso, 2003b; Pereira and Cardoso, 2003a; Pereira and Cardoso, 2004).

For illustration, we are using as an example the association of *guitar* and *woman*, which has been present in many poems, melodies and paintings, in Iberian (Spanish and Portuguese) culture. Therefore all the examples will be related with these two. Examples (extremely simplified)of their concept maps are given in figure 1.

3.1. Knowledge Representation

Divago allows several different kinds of knowledge representation:

- **Concept maps** describe factual knowledge about a concept. A concept map is essentially a semantic network in which all arcs are binary (i.e. they connect exactly two different concepts). For example, the fact $have(guitar, string)$ could be part of the concept map for $guitar$ (see figure 2).

- **Rules** describe procedural knowledge about a concept or a domain. Rules are represented in first order logic format. A possible rule could be "If X is a stringed instrument and any of its strings gets plucked, then a certain musical note is played".

- **Frames** describe abstract concepts or procedures. They can be instantiated by the concept maps (when this happens one says that "the frame has been integrated" or that "the concept map accomplishes the frame". They are formally equivalent to rules (their representation is similar). An example of a simple frame could be "wooden instrument". If a concept map about a concept c instantiates this frame, then we can say that c is a "wooden instrument" (see figure 2). Frames are extremely important in Divago and they can be seen as information molds which can be used to *shape* new concepts.

- **Integrity constraints** are simple rules (with *false* consequent) that serve to identify inconsistencies (e.g. something cannot be made of "flesh" and "wood" at the same time). These constraints, however, do not imply the elimination of the concepts that violate them (e.g. a "living wooden" object, such as *Pinocchio*), rather they are pressures against these concepts.

- **Instances** are actual examples of the concepts (e.g. a drawing of a $guitar$) and their representation is free, but an effort should be made such that the names defined in the concept map are also applied (e.g. in a guitar drawing, one should use the names $neck$, $string$, etc. which are defined in the concept map).

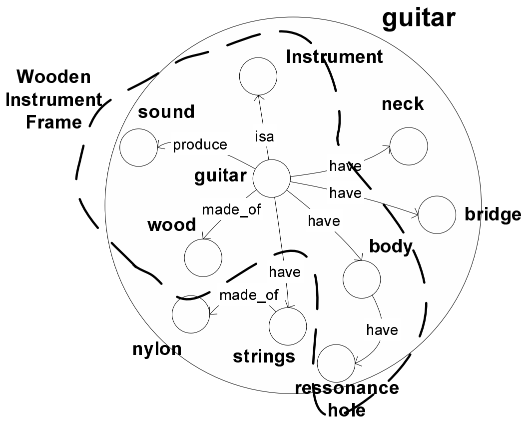

Figure 2: The concept map for guitar and the Wooden Instrument frame.

The concept maps and the frames are the ones more central for Divago. Indeed it works at the more abstract levels (of concept maps) rather than at specific, domain dependent, levels (of instances). For this reason, we will only focus on concept maps in this paper.

3.2. The Architecture

In figure 3, we show the architecture of Divago. The **Knowledge Base** comprises a set of concepts (the majority having solely a concept map) and a *generic domain*, which has generic background knowledge (e.g. an *isa* and a relation hierarchies based on the Generalized Upper Model (Bateman et al., 1995), a set of frames and integrity constraints). The first step for the invention of a new concept is the selection of the input knowledge, in this case a pair of concepts. Currently, this selection is either given by a user or randomly chosen. After given a pair of concepts, the **Mapper** builds a structural alignment between (the definitions of) them. It then passes the resulting mapping to the

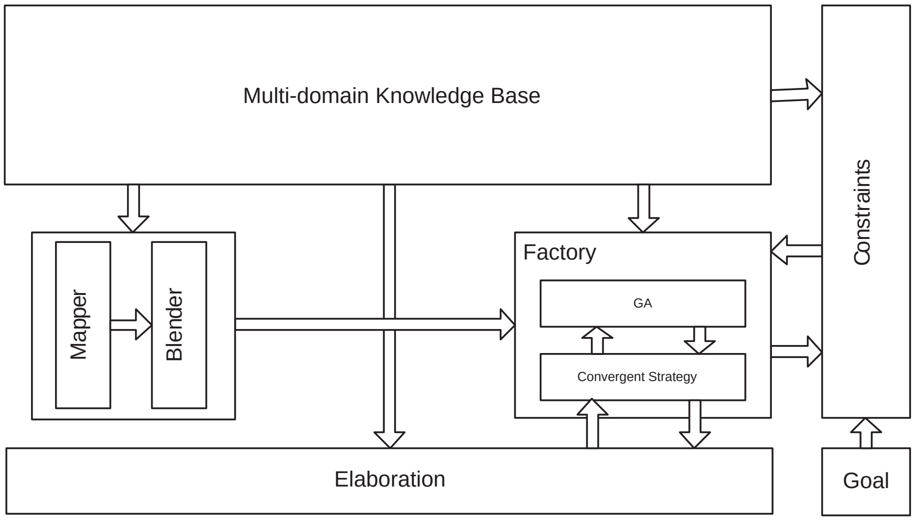

Figure 3: The architecture of Divago

Blender, which then proposes a set of conceptual combinations to be considered. These are resulting from the *projections* that implicitly define the set of all possible *blends*, or concept combinations. A projection is meant to be the "new existence" of each single part of the input concepts (e.g. in the blend of "guitar" and "woman", assuming that the properties of $woman$ get projected to guitar, the "new existence" of $flesh$ can be $wood$, while $hair$ can become $strings$). The set of all possible combinations of projections makes the search space for the reasoning mechanism, the Factory.

The **Factory** is based on a parallel search engine, a *genetic algorithm* (GA), which searches for the blend that best complies with the evaluation given by the Constraints module. Prior to sending each blend to this module, the Factory sends it to the Elaboration module, where it is subject to the application of domain or context-dependent knowledge. The GA thus interacts both with the Constraints and Elaboration modules during search.

The evaluation of a blend given by the **Constraints** module is based on an implementation of the Optimality Principles (Pereira and Cardoso, 2003b). Apart from the blend itself, our implementation of these principles also takes into account knowledge that comes from the Knowledge Base (namely integrity constraints and frames), as well as the accomplishment of a goal that comes in the form of a query. In section 3.4., we clarify this a bit more. The **Elaboration** module essentially applies rule-based reasoning (e.g. the application of rules such as the one given in section 3.1.). These rules are also part of the knowledge base.

After reaching a satisfiable solution or a specified number of iterations, the Factory stops the GA and returns the best solution it achieved. In some cases, this result is also the input of an **Interpretation** module, which produces an interpretation of the new concept. In previous versions of Divago, we made interpreters for 2D (Pereira and Cardoso, 2002) and 3D images (Ribeiro et al., 2003), as well as textual description of the blend (Pereira and Gervás, 2003). Of course, these several *modalities* were adapted to specific uses and therefore they are not guaranteed to work in different applications.

Both the Mapper and the Elaboration modules are optional, for different reasons. The mappings provided by the Mapper are essentially based on Metaphor research (Veale, 1995), however, in some situations, these mappings are very much restrictive. Thus, without having implemented alternative procedures, we allow an externally defined mapping (which, in some experiments, is user-defined). The Elaboration can also be bypassed for experimentation reasons. When analyzing results, the elaboration can hide the *real* results, i.e., it can fix problems by itself that we may need to watch in order to assess the functioning of the system.

3.3. Input and Output

Apart from the configurations (regarding the parameters for the GA, the weights of the evaluation function, among others), the input that Divago needs for generating a concept is a selection of a pair of concepts to bisociate (by default, it picks randomly from its knowledge base) and a goal to accomplish (by default, no goal is used and it generates without concerning with goal-satisfaction). The goal can consist of a set of frames and/or a set of relations that the concept map of the result is expected to contain (e.g. if we wanted to create a new concept of musical instrument, we must at least have the relation $produce(X, sound)$ in the query). The output of the system will be another concept map, which should be self explicatory.

3.4. Bisociation in Divago

The mechanism of bisociation of Divago follows the principle that, when one part of a concept is transferred to another concept, it gets a different meaning. For example, if we transfer the "body" of the concept of $woman$ to the concept of $guitar$, then the latter gains a different meaning, let us call it a $guitar|woman$, a bisociation of $guitar$ with $woman$. Divago uses a computational model of Conceptual Blending (Fauconnier and Turner, 1998; Pereira and Cardoso, 2003b) to determine which knowledge structures should be transferred at each time. The result is called *blend*. For any two concepts, there is an extremely large number of possible blends (Pereira and Cardoso, 2003a), indeed some of the steps of the conceptual blending are non-deterministic and, for this reason, Divago uses a genetic algorithm to select, from this large search space, the blends that best respect the goal given externally. It is not, however, guaranteed that it finds the best results.

Below, we summarize the bisociation algorithm used in Divago. Inside square brackets "[]", we discriminate the modules where the respective steps take place.

```
1. Find a mapping M between the two
   concept maps. [Mapper]
2. For each pair <a,b> of correspon-
   dences in M, determine all possi-
   ble combinations  {a, b, a|b   or
   void} that can be copied (projec-
   ted) to the blend.  These are ca-
   lled  candidate projections.  The
   set of all projections  is called
   blendoid. [Blender]
3. Pick a subset of projections from
   the blendoid B  and  generate its
   concept map, the blend. [Factory]
4. Check whether  the blend respects
   the goal and the constraints con-
   figured in the  system.
   [Constraints]
    4.1. If the blend is ok, finish
    4.2. If the blend does  not ful-
    fill the requirements, return to
    step 3.
```

Given the high complexity that arises, the steps 3 and 4 are performed by a genetic algorithm (GA). The *genotypes* of this GA correspond to strings of projections. Their *phenotype* correspond to the concept maps that are generated in step 3.

The verification of the results (step 4) is based on the satisfaction of the *optimality constraints* of conceptual blending (Pereira and Cardoso, 2003b), with particular attention to the goal given to the system (which is measured in one of these optimality principles, called *Relevance*). A goal can range from specific characteristics one expects the blend to have (e.g. "be an instrument with woman body, hair, eyes, and beauty property") to very abstract directives (e.g. "be the application of properties from the second input to the first input"). The mean term is more usual. For example, if we ask Divago for "a musical instrument with new properties", a possible result (with the above given inputs of $guitar$ and $woman$) is given in figure 4.

3.5. Applicability of Divago

Divago can be used to transform concept maps (or any analogous or equivalent representation, such as semantic networks, or binary relation-based ontology representations, as in WordNet) into novel concept maps that inherit aspects from the inputs, although having its own emergent structure. If appropriate cross-concept associations between the inputs (e.g. from a structure alignment algorithm) and frames are used, then the novel concept can bring surprising and meaningful semantics. For example, in a poetry

```
isa(guitar, instrument).
isa(guitar, female).
property(guitar, beautiful).
can(guitar, sing).
made_of(guitar, wood).
have(guitar, strings).
made_of(string, nylon).
produce(guitar, sound).
have(guitar, neck).
have(guitar, body).
have(guitar, bridge).
have(body, ressonance_hole).
```

Figure 4: Concept map for the *guitar/woman blend.*

generation system, one could establish an analogy between two concepts (the inputs) and let Divago propose new relationships (the blend). Above, we show a possible association, plenty of times used poetically, of guitar and woman, which could be used in the core of such a generative system.

A different application could be in an educational system, by explaining a concept by analogy with another (more familiar) concept, although it is clear that to achieve such a role, this system would have to be much more constrained.

4. A Hypothesis to Test: Conventional Use of Transformed Resources Becomes Creative

It is clear that applying Divago to the linguistic resources that feed a standard NLG process would result in some interference with the lexicon. Under certain interpretations, one could consider that interference to be creative. It would provide the sort of 'dynamic transformation' of a language resource that people perform daily to achieve their everyday feats of linguistic creativity that seem well beyond the abilities of current natural language processing systems.

A simple way of testing the applicability of this idea would be to take a linguistic task that an existing system solves by resorting to a given resource. Although the Divago system is fully operative as a creative resource, its applicability as proposed here has not yet gone beyond the stage of *gedanken* experiments. One such is outlined below to illustrate the concepts under discussion.

Assume the existence of a linguistic resource of the type described above and a text generation system that obtains a passable linguistic text version from a set of elementary facts, based on an ingenious recombination of the given facts with the knowledge embodied in the linguistic resource, by means of simple steps of inference/substitution. Take for instance the following set of facts:

```
hear(speaker,sound)
produce(guitar,sound)
attracted_to(speaker,sound)
```

By applying whatever NLG operations to the semantic and lexical knowledge encoded in the resource, the system might produce the following description to match those facts:

"I heard the attractive sound of the guitar".

Divago can be applied to creatively transform the resource into a modified version. According to our definition of dynamical pre-processing this would imply the following operations: select portions from the linguistic resource that are relevant (in our case, for instance, the concept maps for guitar and woman given in figure 1), warp them appropriately (taking the user's query into account). This results in the new concept map given in figure 4.

Having this new concept map available, even if it was not part of original linguistic resource, enables new inferences to be carried out for the same process of generating a textual description of the given facts. For instance, as a result of the appearance of the new concept map, the following data that were originally in the linguistic resource but unrelated to the input data have now become relevant, linked to the data by the additional facts in the new concept map:

```
isa(mermaid, mythical_creature).
isa(mermaid, female).
property(mermaid, beautiful).
can(mermaid, sing).
produce(mermaid,song).
attracted_to(men,song).
```

The reader may easily trace the relevant connections. These connections could be made more direct or efficient with a general ontology (that could associate, e.g. "song *imply* sound"). This new situation enables the system to generate alternative, more creative, descriptions for the same set of given facts. A range of possibilities could arise, from the straightforward direct substitution "mermaid=guitar" (1) to the more elaborated and computationally hard to get "mermaid song=attractive sound" (3), passing by an intermediate and feasible transformation "song=sound" (2).

1. "I heard the attractive sound of the mermaid".
2. "I heard the attractive song of the guitar".
3. "I heard the mermaid song of the guitar".

Thus, the transformations brought by Divago (as in figure 4) can become a middle space that suggests novel associations between concepts. This could be of use from a surface lexical substitution perspective (as in 1.) as well as from a conceptual change/figure of speech perspective (as in 3.).

The results obtained by the system in this revised version of the task can be evaluated to see how the changes affect it. This evaluation can be oriented towards locating any indications of linguistic creativity introduced by the process, possibly by applying metrics and analyses of creative activities that have progressively emerged over the recent years (Ritchie, 2001; Colton et al., 2001).

5. A Proposal for a New System

The experiment described above hints at a possible wealth of linguistic creativity waiting to be exploited at the junction between a linguistic resource, the Divago system, and an adequately configured natural language generation system. Although the specification and design of such a combination are well beyond the scope of the current paper, some requirements that such a system would have to fulfil and constraints on its operation may already be inferred from the discussion so far. The present section is intended as an exploration of the hurdles and rewards that might be found along that tempting path.

The following discussion exploits insights derived from an existing system that used jointly the conceptual blending abilities of Divago and a simple NLG system to generate textual descriptions of contextual blends (Pereira and Gervás, 2003).

Let us propose then a system (Don Divago) capable of reversing the roles in the original colaboration: instead of applying an NLG system to provide a description of the result of a conceptual blend, we can study the role - or roles, as will become immediately apparent - that a conceptual blending module might be able to play in close interaction with an NLG system.

5.1. The Architecture for Don Divago

One of the major moot points would be in deciding on the extent and the nature of the interactions between the conceptual blending module and the NLG module.

This requires some basic groundwork on elementary architecture of an NLG system. Research on natural language generation over the years has come to propose a pipelined architecture (Reiter, 1994) as the simplest engineering solution to generate texts meant to convey information. This solution is not optimal and the generation of other types of texts calls for different architectures (DeSmedt et al., 1995; Beale et al., 1998). However, for the purpose of the present discussion, the modularity presented by a pipeline architecture outweighs any other disadvantages that it may have. Its simple modular nature allows discussion of foreseen connections between modules at an abstract level. If more complex interconnections are required - which seems likely -, alternative more refined architectures can be considered at later stages.

The pipelined architecture establishes a number of basic tasks to be carried out when generating a natural language text: *content determination* - finding what to say -, *document structuring* - organising what is to be said -, *sentence aggregation* - grouping together the parts that allow it -, lexicalization - selecting the words that will realize each concept -, *referring expresion generation* - choosing the right expression to refer to each element in its actual context - and *surface realization* - turning the result into a linear natural language sentence. These tasks tend to be grouped into bigger modules that operate in sequence over the initial input: a *text planning* component that deals with content determination and document structuring, a *sentence planning* component that deals with aggregation, referring expression generation and lexicalization, and a *surface realization* component.

The example described in the previous section involves operations mostly at the level of lexicalization or referring expression generation. It is clear that in those cases the linguistic resource plays a significant role in as much as the lexical representation for a content specified in terms of semantic concepts must be decided. This, as described earlier, may be achieved by traversing the connections available in the linguistic resource with the aim of exploiting it to its best advantage. Therefore a plausible way of connecting the conceptual blending module to the NLG system might be as an auxiliary process to the lexicalization/referring expression generation tasks as carried out within a sentence plannnig module.

From the point of view of linguistic creativity, such an architecture would surely provide many interesting possibilities in need of exploration. However, it by no means exhausts the available alternatives. For the tasks of lexicalization and referring expression generation, an NLG system exploits its linguistic resources by operating on the fringe of them, right where the lexical tags appear as leaves of a graph/tree of semantic concepts. Linguistic creativity of the type shown above occurs when conceptual blending enables a new exciting path from the input concepts, following branches already in the tree, to leaves that were not available before and that produce valuable and surprising results.

Since the transformation that is taking place during blending actually occurs on the conceptual part of the linguistic resource, this discussion should also take into account the possible interactions of the blending module with those NLG tasks that are concerned strictly with the semantic part of the content to be expressed, such as content determination, document structuring, and certain types of aggregation. Though it is beyond the scope of this discussion to enter into the possibilities to any depth, results of such an interaction would involve not simply rephrasing a given message in terms of using unexpected words to render it, but rather to reconstruct it with a different content, a different structure, or a different way of grouping its ingredients. Furthermore, all of these possibilities might be combined in a single step of "creatively enhancing" the rendition of a given set of input facts. The possibilities that lie open for exploration are extremely promising.

5.2. Critical points

There are some critical points in this architecture:

- Mappings. What sort of mapping algorithm should be used? Would structural alignment suffice for the generation of productive blends? Some authors argue that structural alignment is very restrictive in analogical and metaphorical contexts (e.g. (Keane and Costello, 2001; Veale, 1997; Pereira and Cardoso, 2003a). Moreover, an initial problem arises: what should be the methodology for the selection of concept maps to blend? In other words, why blend "woman" and "guitar" instead of "potato" and "guitar"? One can imagine some odd results that would come from random associations. A possible solution can rely on the associations found within the concept maps (e.g. "women sing" and "guitar produces sound"; both have "neck" and "body", among others that could be found in richer concept maps).

- Queries. Which methodology should be followed for the generation of queries? Should these rely on templates (depending on context, user-defined, etc.) or should these be dynamically constructed? Although the latter choice may seem ideal, it is clear that a realistic approach must first rely on a template based methodology. For example, queries such as "the blend should consist of the first concept added with the diagnostic properties of the second", which could work with success in examples such as given above.

- Connections. In the examples given in section 4., it was left unclear which were exactly the paths followed, more precisely which connections between concepts were used and how this should be done. A first, perhaps trivial, choice would just follow the respective *isa* taxonomy (e.g. replacing the hyponym for the hypernym, or the interchange new *cousins*, such as "mermaid" and "guitar"). More elaborate solutions would include metonymy (e.g. "song" for "mermaid song") or causality-driven substitution (e.g. replace the effect by the cause, as in "song" for "sound").

- Knowledge richness. It is is unavoidable fact that Divago is highly dependent on the richness of its resources. In principle, the more detailed and complete the concept description, the more creative the blend can be (or at least, the higher probability of adding knew knowledge to the rest of the system). We plan to use the resources available today (e.g. WordNet, CYC, Mikrokosmos), however these may be found poor in some respects (e.g. WordNet is exclusively centred in a small set of relations, namely hypernym/hyponym).

6. Conclusions

The present paper sketches some of the linguistically creative effects that might be obtained by connecting a conceptual blending module (the Divago system) to a natural language generation system. The major envisaged connection involves using Divago to dynamically pre-process selected portions of the linguistic resource available to the NLG system, in order to warp them in such a way that their use by the system opens new and valuable possibilities in terms of sentence generation for a given input.

Although the effects presented here are merely sketched in the paper, they demonstrate a procedure that demands further exploration. In order to proceed along this line of research, a possible architecture for linking conceptual blending and language generation is described, corresponding to a new proposed system, Don Divago, that would attempt to exploit these effects.

A number of fundamental questions concerning the different ways in which these interactions might be controlled to enhance the value of the resulting sentences are posed in the final sections of the paper, and initial approximations to their resolution are proposed.

7. Acknowledgements

The work was partially funded by the Spanish Committee of Science & Technology (TIC2002-01961).

8. References

Bateman, John, Bernardo Magnini, and Giovanni Fabris, 1995. The generalized upper model knowledge base:

Organization and use. *Towards very large knowledge bases: knowledge building and knowledge sharing*.

Beale, S., S. Niremburg, E. Viegas, and L. Wanner, 1998. De-constraining text generation. In *Proceedings of the Ninth International worshop on Natural Language Generation, Niagara-on-the-Lake, Ontario*.

Colton, Simon, Alison Pease, and Graeme Ritchie, 2001. The effect of input knowledge on creativity. In Amílcar Cardoso, Carlos Bento, and Geraint Wiggins (eds.), *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

DeSmedt, K., H. Horacek, and M. Zock, 1995. Architectures for natural language generation: Problems and perspectives. In G. Adorni y M. Zock (ed.), *Trends in natural language generation: An artificial intelligence perspective*. Berlin: Springer.

Fauconnier, Gilles and Mark Turner, 1998. Conceptual integration networks. *Cognitive Science*, 22(2):133–187.

Gervás, P., 2002. Linguistic creativity at different levels of decision in sentence production. In G.A. Wiggins (ed.), *Proceedings of the AISB 02 Symposium on AI and Creativity in Arts and Science, 3rd-5th April 2002, Imperial College*. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Keane, Mark T. and Fintan J. Costello, 2001. Setting limits on analogy: Why conceptual combination is not structural alignment. *The Analogical Mind: A Cognitive Science Perspective*.

Lonergan, E., 2001. Lexical knowledge engineering: Mikrokosmos revisited. In *PACLING2001 - Pacific Association for Computational Linguistics 2001*. Kitakyushu, Japan.

Miller, G. A., 1995. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11).

Nirenburg, S., 1987. Knowledge-based machine translation, the CMU approach. In *Machine Translation: theoretical and methodological issues, Studies in Natural Language Processing*. Cambridge: Cambridge University Press.

Pereira, Francisco C. and Amílcar Cardoso, 2002. The boat-house visual blending experience. In *Proceedings to the Second Workshop on Creative Systems*. European Conference of Artificial Intelligence, ECAI'02.

Pereira, Francisco C. and Amílcar Cardoso, 2003a. The horse-bird creature generation experiment. *AISB Journal*, 1(3).

Pereira, Francisco C. and Amílcar Cardoso, 2003b. Optimality principles for conceptual blending: A first computational approach. *AISB Journal*, 1(4).

Pereira, Francisco C. and Amílcar Cardoso, 2004. A model of concept invention. *In preparation*.

Pereira, Francisco C. and Pablo Gervás, 2003. Natural language generation from concept blends. In *AISB'03 Symposium on AI and Creativity in Arts and Science*. SSAISB.

Reiter, E., 1994. Has a consensus NL generation architecture appeared, and is it psychologically plausible? In David McDonald and Marie Meteer (eds.), *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*. Kennebunkport, Maine.

Ribeiro, Paulo, Francisco C. Pereira, Bruno Marques, Bruno Leitao, and Amílcar Cardoso, 2003. A model for creativity in creature generation. In *Proceedings of the 4^{th} Conference on Games Development (GAME ON'03)*. EuroSIS / University of Wolverhampton.

Ritchie, Graeme D., 2001. Assessing creativity. In Geraint Wiggins (ed.), *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Sciences*. AISB.

Veale, Tony, 1995. *Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation*. PhD Thesis, Dublin City University.

Veale, Tony, 1997. Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic ”borrowing”. In *Proceedings of Computational Models of Creative Cognition (Mind II)*. Dublin, DCU.