Modeling a Quite Different Machine Translation using Lexical Conceptual Structure

Nadia Luiza Dinca

Abstract— **The goal of this study is to outline the readability of an Example-Based Machine Translation for any pair of languages by means of the language-independent properties of the lexical conceptual structure (LCS). We describe LCS as a representation of traditional dependency relationships and use in experiments an isolated pair of verbs, extracted from Orwell's "1984" parallel English – Romanian texts. We discuss the mental models in terms of specific knowledge structures. Finally, we present LCS-Based Machine Translation from the point of view of a complex adaptive system and present our ongoing work in order to capture the neutral linguistic core of any mental model corresponding to the real world.**

***Index Terms*—Lexical conceptual structure, machine translation, readability, complex adaptive system.**

I. Introduction

THE paradigm of 'translation by analogy', used to characterize the Example-Based Machine Translation, proposes the use of an unannotated database of examples (possibly collected from a bilingual dictionary) and a set of lexical equivalences simply expressed in terms of word pairs. The matching process is focused on checking the semantic similarity between the lexical items in the input sentence and the corresponding items in the candidate example. In fact, an Example-Based Machine Translation database is used for different purposes at the same time: as a source of sentence frame pairs, and as a source of sub-sentential translation pairs.

In this paper, we aim to present a new direction in designing the structural Example-Based Machine Translation. We are expanding the original Nagao's model [1] in order to obtain the translation of a complete sentence by utilizing more than one translation example and combine some fragments of them. Usually, the translation examples are represented as dependency trees with correspondence links between sub-trees. I propose here an issue to replace the traditional representation of these translation examples with lexical conceptual structure (LCS), a kind of a compositional abstraction with language-independent properties that transcend structural idiosyncrasies [2].

For an input sentence, there is a matching expression, naming a pointer to a translation unit, i.e., a lexical conceptual structure to be found in a manually constructed database of examples. The pointer is optionally followed by a list of commands for deletion/replacement/adjunction of nodes dominated by the node pointed to. The replaced or adjoined elements are other matching expressions. The data encapsulation of the translation examples is related to the modularity demands of the sub-sequences that inherit the features of the dominating units.

It is obviously that a machine translation system requires a substantial amount of translation knowledge, typically embodied in bilingual dictionaries, transfer rules, example databases or statistical models. Our approach seeks to obtain as much of this knowledge as possible by expressing translation examples in LCS- dependency trees.

The real value of this LCS–Based Machine Translation is offered by readability, since the machine captures the mental models of any language and therefore, isolates the correspondence links between the translation units.

II. Lexical Conceptual Structure

Traditionally, a translation example contains three parts:

- Dependency tree, adapted for the source language;
- Dependency tree, created for the target language;
- Correspondence links.

In this paper, the dependency trees are replaced with lexical conceptual structures, built by hand, for English-Romanian linguistic project. We have isolated pairs of verbs, extracted from Orwell's "*1984*" parallel English-Romanian texts.

The verbs were characterized from the point of view of syntagmatic and paradigmatic relations, using the verb classes and alternations described by Levin [3] and Visdic [4]**,** using a multilingual ontology editor.

The semantic properties of a lexical item are totally reflected in a number of relations associated to different types of contexts. These affinities developed by a word regarding a context are syntagmatic or paradigmatic. Lexical semantic relations are essentially paradigmatic, even if they can be combined directly with or be based on, some analytical elements or expression of properties, as in WordNet.

According to [5], a lexical conceptual structure is a directed graph with a root, where each root is associated with a special kind of information, including a *type*, a *primitive* and a *field*. The types name are Event, Path, Manner, Property, Thing; the fields refer to Locational, Possessional, and Identificational values. The primitive of a LCS node is splitted into structural primitive (e.g., *go, cause, act*) and constants (e.g., *reduce+ed, slash+ingly, face+ut, caine+este*). For example, the top node in the root LCS of the verb *follow* ("*the eyes follow you*") has the structural primitive ACT_ON in the locational field. Its subject is a star-marked LCS with the restriction of being a type thing. The number "1" specifies the thematic role of the

Manuscript received July 22, 2008. Manuscript accepted for publication October 30, 2008.

Nadia Luiza Dinca is with Research Institute for Artificial Intelligence, Bucharest, Romania (hnadia_luiza@hotmail.com).

agent. The second child node is an argument position and needs to be of type thing, too; its number "2" represents the clue of the theme:

(DEF_WORD: "follow"
LCS: (act_on loc (thing 1) (* thing 2)))*

The format for thematic roles is the following:

1. Any thematic role preceded by an underscore (_) is obligatory.
2. Any thematic role preceded by a comma (,) is optional.
3. Prepositions inside parentheses indicate that the corresponding phrases must necessarily be headed by the specified prepositions.
4. An empty set of parentheses ( ) indicates that there necessarily must be a prepositional head, but it is left unspecified.
5. The difference between the main communication and the incident constructions referring to a second level of speech is marked by indices "1" for the first level, and "2" for the second one.

In the notation we used, the DEF_WORD, THEM_ROLES and LCS represent the Dorr's description attributes [2], [6]. We added TE, CLASS, SYNSET attributes with the values of *translation equivalent*, *semantic* and *synonymic classes*. The LCS specifies a star marker (*) for very explicitly realized argument and modifier. The star marker forces logical constituents to be realized compositionally at different levels.

Consider the sentence (1a). This can be represented as shown in (1b), glossed as (1c):

1a. I walk to cinema.
1b. (event go loc
(thing I+)
(path to loc
(thing I+)
(position at loc (thing I+) (thing cinema+)))
(manner walk +ingly))
1c. 'I move (location) to the cinema in a walking manner.'

The next figure shows the lexicon entry for the contextual sense of the English verb 'walk' with several pieces of information, such as the root form of the lexical item, its translation equivalent, the semantic verb class and the synset, introduced by the fields: DEF_WORD, CLASS, SYNSET and TE. The thematic roles appearing in the root LCS entry are classed in a canonical order that reflects their relative surface order: first available in this case is theme, with the obligatory specification; the last two optional roles are source and goal:

(DEF_WORD: "walk"
TE: "merge"
CLASS: "51.3.2"
SYNSET: "walk: 4", "jog: 3",
"run: 29"
THEM_ROLES: "_th ,src() ,goal()"
*LCS: (event go loc (* thing 2)*
*((* path from 3) loc (thing 2)*
(position at loc (thing 2) (thing 4)))
*((* path to 5) loc (thing 2)*
(position at loc (thing 2) (thing 6)))))

The field LCS introduces the uninstantiated LCS corresponding to the underlying meaning of the word entry in the lexicon. The top node for '*walk'* has the structural primitive *go* in the locational field. Its subject, marked with a star "*", indicates that the node must be filled recursively with other lexical entries during semantic composition. The only restriction is that the filler must be of type '*thing'*. The second and third child nodes are in argument positions filled with the primitives FROM and TO; the numbers 3 and 5 mark the source and goal particle; the numbers 4 and 6 stand for source and goal.

III. Lexical Conceptual Structure-Based Machine Translation

The main problem concerning an Example-Based Machine Translation is how to use a translation example for translating more than one source sentence. The solution described here uses the lexical conceptual structure as a representation of traditional dependency relationships. The words are introduced in the dictionary with the specification of semantic class, the synset and LCS –for the verb lexical entry– and with the notation of thematic roles and LCS, for all other parts of speech lexical entries (i.e., nouns, pronouns, numbers, adverbs, adjectives, prepositions).

The basic properties of LCS are: idempotence, reflexivity and compositionality:

- *Idempotence*: a LCS multiplied by itself, gives itself as a result. For example, the root LCS for "*follow*" can combine with any members of the semantic class "51.6" ("*watch*: 2", "*observe*: 7", "*follow*: 13") and the resulted LCS has the same characteristics as the original one.
- *Reflexivity* means the act of self-reference. For example, the LCS created for the verb "*divide*" can refer to any members of the synset "*divide*: 1", "*split*: 1", "*split up*: 2", "*separate*: 4", "*dissever*: 1", "*carve up*: 1").
- *Compositionality* states that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. It can be considered the most important property because it allows to a translation equivalent to be used in order to translate more than one source sentence.

Let's consider the translation of the following sentence:

(1) He dipped the pen into the ink.

If the translation database contains the translation examples (2) and (3), then we can translate sentence (1) into (4) by imitating examples and combining fragments of them:

(2) He dipped the pen into the blue liquid.
El isi inmuie penita in lichidul albastru.
(3) I bought an ink bottle.
Eu am cumparat o cutie cu cerneala.
(4) El isi inmuie penita in cerneala.

Formally, a translation example consists of three parts:

- Source LCS-tree (ELCS; English Lexical Conceptual Structure);
- Target LCS-tree (RLCS; Romanian Lexical Conceptual Structure);
- Correspondence links.

Each number prefixed by "*e*" or "*r*" represents the identifier of the sub-tree and each node in a tree contains a word (in root form), a thematic role and its corresponding part of LCS. A correspondence link is represented as a pair of identifiers.

If the translation of an identical sentence is not available in the bilingual corpus, the EBMT system makes use of some sort of similarity metric to find the best matching translation examples. Suitable sub-sequences are iteratively replaced, substituted, modified or adapted in order to generate the translation. While the replacement, substitution, modification or adaptation is rule-driven, the mapping of a source segment into an equivalent target segment is guided from translation examples.

According to [1], the concept *matching expression* (ME) is defined as in the following:

```
<ME> ::= [<ID> | <ME - Commands>]
<ME – Commands> ::=
        [ ]
or [<ME – Command> | <ME – Commands>]
<ME – Command> ::=
        [d, <ID>]                %% delete <ID>
or      [r, <ID>, <ME>]  %% replace <ID> with <ME>
or      [a, <ID>, <ME>]  %% add <ME> as a child of
                                 root node of <ID>
```

Under these assumptions, the LCS trees (a) can be represented by the matching expression (b):

```
(a) elcs_e ([e11, [dip, cause],
        // TE1
            [e12, [he, _ag, (* thing 1)]],
             [e13, [pen, _th, (go loc (* thing 2))]],
              [e14, [into, _goal (into), ([into] loc (thing 2))],
               [e15, [liquid, _goal (into), (thing 6)]
                [e16, [blue, _goal (into), (thing 6)]]]]])

        rlcs_e ([r11, [inmuia, cause],
             [r12, [el, _ag, (* thing 1)]],
              [r13, [penita, _th, (go loc (* thing 2))]],
                [r14, [in, _goal (in), ([in] loc (thing 2))],
                  [r15, [lichidul, _goal (into), (thing 6)]
                    [r16, [albastru, _goal (into), (thing 6)]]]]

 %% clinks: ([e11, r11], [e12, r12], [e13, r13], [e14, r14],
[e15, r15])

        elcs_e ([e21, [buy, cause_exchange],
        // TE2
             [e22, [I, _ag, (* thing 1)]],
             [e23, [bottle, rec, (go poss (* thing 2))],
          [e24, [ink, _th, (go poss (* thing 2))]]]]])

        rlcs_e ([r21, [cumpara, cause_exchange]
             [r22, [Eu, _ag, (* thing 1)]],
             [r23, [cutie, rec, (go poss (* thing 2))],
          [r24, [cerneala, _th, (go poss (* thing 2))]]]]])

  %% clinks: ([e21, r21], [e22, r22], [e23, r23], [e24, r24])

  (b) [e11, [(r, e15, [e24]), (d, e16)]  // for source language
```

The first step is matching fragments of the input sentence *He dipped the pen into the ink* against a database of real examples. Two sequences can be found: *he dipped the pen into the* and *ink*, respectively. The data encapsulation of the translation unit provides a kind of logical independence of the sub-sequences. Therefore, the complex translation unit, i.e., the entire sentence, is splitted into two sub-units; the first sub-unit encapsulates the second sub-sequence 'into the', while the last has lexicalized only the head and the specifier, and it waits for the noun corresponding to the input modifier.

TE_1 joins TE_2 and the result matches against the input only if the result has the same structure as the source sentence and its arguments have values for the same type, primitive and field as the input. The inheritance develops translation sub-units incrementally through defining new objects in terms of one previously object defined. It is applied only from noun to adjective, from preposition to noun, and the inherited features allow the matching by literals, instead of complex translation units.

In the example above, the preposition node for *into* requires a daughter node with the feature tuple (_goal (into), (thing 6)), instantiated by the lexical items *black liquid*. Also, the item 'black' lexicalizes the daughter node of the noun 'liquid' and it inherits its feature tuple. The inheritance is possible only if the selected literal has the same type as the input needed to match. The LCS-tree created for the second translation example shows the value of theme for the bi lexeme '*ink* ⇔ *cerneala*' and the type 'thing', so the literal is corresponding to the input word.

In the transfer step, the system replaces every identifier in the source matching expression with its corresponding identifier:

```
SME= [e11, [r, e15, [e24]]
TME= [r11, [r, r15, [r24]
```

In the composition step, the lexical conceptual structure-tree is composed according to the target matching expression:

```
TME = [r11, [r, r15, [r24]]
TLCS=   ([r1, [inmuia, cause],
            [r2, [el, _ag, (* thing 1)]],
            [r3, [penita, _th, (go loc (* thing 2))]],
            [r4, [in, _goal (in), ([in] loc (thing 2))],
             [r5, [cerneala, _goal (into), (thing 6)]]]]])
%% El isi inmuie penita in cerneala.
```

IV. Discussion

The EBMT idea is to translate by analogy [7], [8]. But what is happening when the translator finds only a synonym of a given input, instead of its matching in different associations? In our opinion, the solution is to consider a new mental model as we explain below.

Any translation means to acquire new knowledge about the real world by taking into consideration the older knowledge organized into mental schemata by the system. The synonymy relation, the classification into semantic verb classes, the verb arity and the thematic roles, the lexical-semantic representation in terms of primitive, fields and types- all represents mental schemata corresponding to different levels of the linguistic representation [9], [10]. Each of these schemata identifies distributed information and a starting point in understanding, learning and transferring the code from a source language to a target language.

When the system confronts with a new situation, i.e., the presence of a synonym, instead of its matching, it must unify all the distributed schemata and organize them into a new mental model, which is, in our case, the lexical conceptual structure tree representation of the translation examples. Therefore, while the schemata are generic pre-compiled knowledge structures, the mental models are specific knowledge structures, built in order to figure a new situation using this generic knowledge.

Lexical forms written in the example side may be a synonym for the matched input word and we must modify the input side before constructing a target structure. The matching is analogue to a reasoning issue in natural language, where an unknown word is translated depending on the associations with known items, participants in the same kind of context and used as pointers for the semantic interpretation of the item given.

Usually, the context acceptance means the words which occur with a lexical item in order to disambiguate it semantically. For this approach, the context refers to the semantic class and synonymy relation of the verb given in the lexical entry (e.g., the context of *jog* is the class “51.3.2.” and the synset “*run*: 29, *jog*: 3, *walk*: 4, *zigzag*: 1, *jump*: 1, *roll*: 12”, instantiated for the sentence *John jogged to school*). Formally, the source verb “a” may be translated into the target verb “b” when:

a. there is the appropriate equivalent in the translation database;

b. there is a source verb “c” which is:

b.1. in a synonymy relation with “a”;

b.2. in the same semantic verb class with “a”;

b.3. the translation equivalent of “a”.

The following example shows how to obtain a translation if the system has a translation example and its context:

Input Sentence: *He runs to school.*

Translation Example:

He jogs to school = El alearga la scoala.

Context:

jog: 3= run: 29

jog: <- class “51.3.2.”

jog: (synset) <- (run: 29, jog: 3, walk: 4, zigzag: 1, jump: 1, roll: 12)

Target Sentence: *He runs to school = El alearga la scoala.*

Even LCS is not a deep knowledge representation; it captures the semantics of a lexical item through a combination of semantic structure (which is something the verb shares with a semantic verb class) and semantic content (which is specific to the verb itself). The semantic structure relies also on the subcategorization level of linguistic representation by the fact that there are three ways a child node relates to its parents: as a subject (maximally one), as an argument, or as a modifier. Considering this relation between different levels of linguistic representation, I can define a well-formedness principle for LCS-based MT:

A translation is well-formed if:

i. There is an appropriate equivalent in the database examples;

ii. There is an appropriate context (semantic verb class and synset) for the input verb;

iii. The LCS children are lexicalized (if there is one minimally) and associated with information including a type, a primitive and a field.

The type of the LCS created for ‘*give*’ is *Event*, its structural primitive is *go*, which appears in many generalized movements, and the field which specifies the domain is *Possessional*. The thematic roles are organized into the grid: "_ag_th_goal(to)". If the sentence contains a form which doesn’t fill all the values of LCS, it won’t be lexicalized and the sentence won’t be generated:

Input: *He gives*.

TE$_1$: *He gives fruits to children*. ⇔ *El da fructe copiilor*.

TE$_2$*: He buys a kilo of fruits*. ⇔ *El cumpara un kilogram de fructe*.

LCS _input :

```
((cause (* thing 1)
  (go poss (* nill)
     ((* to 5) poss (nill) (at poss (nill) (nill))))
 (give+ingly 26))
```

V. Conclusion and Future Work

The LCS-Based Machine Translation can be a powerful linguistic tool because it allows clear readability of the results. Since the information contained in these lexical conceptual structures is language-independent, we consider them an interesting issue to capture the core of a machine translation, which is not centered on any particular pair of language.

In fact, the LCS-Based Machine Translation has the behavior of a *complex adaptive system*, which means:

- *Patterns of activity:* the linguist has to describe the LCS for every verb, considered a lexical entry in translation database;
- *Self-organization*: the structure receives a holistic interpretation, including a type, a primitive and a field;
- *Collective behavior*: the verbal core is extended, so that the root LCS accepts all the verbs which respect the same semantic class.

In conclusion, this paper aims to present the improvement of readability of an Example-Based Machine Translation in terms of lexical conceptual structure. It is only a beginning of a more profound study that will be developed in order to characterize a machine translation without the old centering on the particular pair of languages – source and target languages.

The future work involves the following practical tasks of investigation:

1. Creating LCS-lexicons by taking the most frequent verb pairs in "*1984*" corpus, Romanian and English versions (with a frequency threshold of 5 occurrences, minimally).
2. Considering French as a new language for translation and creating also LCS-tree representations.
3. Organizing the LCS-trees of English, Romanian and French languages into new mental models using the encapsulation and a top ontology. We will try to reduce the language differences and to isolate the neutral linguistic core of any mental model, corresponding to the real world.

References

[1] S. Satoshi and M. Nagao. Toward Memory- Based Translation. In: Proceedings of the 13th conference on Computational linguistics, Finland: Helsinki, 1990, pp. 247-252.

[2] B. J., Dorr. LCS VerbDatabase, Online Software Database of Lexical Conceptual Structures and Documentation. http://www.umiacs.umd.edu/~bonnie/LCS_Database_Documentation.html

[3] B. Levin. English Verb Classes and Alternations - A Preliminary Investigation. The University of Chicago Press, 1993.

[4] http://nlp.fi.muni.cz/projekty/visdic/

[5] A. N. Fazil and B. J. Dorr. Generating a Parsing Lexicon from an LCS-Based Lexicon. In: Proceedings of the LREC-2002 Workshop on Linguistic Knowledge Acquisition and Representation, Spain: Las Palmas, 2002, pp. 43-52.

[6] H. Nizar, B. J. Dorr and D. Traum. Hybrid Natural Language Generation from Lexical Conceptual Structures. Machine Translation, 18:2, 2003, pp. 81—128

[7] M. Carl and A. Way (eds.) Recent Advances in Example-Based Machine Translation. Kluwer Academic Publishers, 2003.

[8] H. Tanaka. Verbal case frame acquisition from a bilingual corpus: gradual knowledge acquisition. In: Proceedings of the 13th conference on Computational linguistics, Kyoto, Japan, 1994, pp. 727-731.

[9] R. Green, B. J. Dorr and Ph. Resnik. Inducing Frame Semantic Verb Classes from WordNet and LDOCE. In: Proceedings of the Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 96-102.

[10] A. N. Fazil and B. J. Dorr. Generating A Parsing Lexicon from an LCS-Based Lexicon. In: Proceedings of the LREC-2002 Workshop on Linguistic Knowledge Acquisition and Representation, Las Palmas, Canary Islands, Spain, 2002, pp. 43-52.