Translation by Meaning and Style in LOLITA

Richard G. Morgan, Mark H. Smith & Sengan Short

University of Durham, UK
morgan@durham.ac.uk

Abstract

LOLITA is a large scale natural language engineering (NLE) system which has been under development at the Laboratory for Natural Language Engineering at Durham for the past eight years. The project is based on the principle of building a large problem and domain independent NLE core, and the development of various applications on top of it. One of these applications is a prototype Italian to English translator. The method adopted for the translation prototype is the 'meaning and style' idea: i.e. the content (expressed as a piece of normalised conceptual graph) and style (expressed as a set of parameters) is extracted from the original text, and then reconstructed in the target language, without any surface correlation between the two texts. Furthermore, a mixed grammar is adopted which allows the parsing of more than one language (e.g. English, Chinese, Italian etc.)

Our experience with this method has been very encouraging: to translate an Italian sentence such as *'se avessi saputo che la grossa moto veloce che mi haidato era posseduta da lei, mi sarebbe piaciuta, perche' lei mi piace davvero'*, only five rules had to be added to the grammar, and minor changes carried out to the normalisation and semantic analysis. Of course, no changes were needed to the pragmatics, nor to the English generation module. Furthermore, the method allows the output English text to be expressed from various points of view and in varying styles.

This paper is split into two main parts. The first discusses the NLE principles on which this system is based (section 1), the base LOLITA system (section 2) and its semantic network (section 3.1), parsing (section 3.2), analysis (section 3.3) and generation (section 3.4) sub-components. Then the paper turns to translation specific details and describes the operation and methodology behind the LOLITA translator (section 4), an example of its operation (section 4.1) and how the sub-components had to be modified to achieve this functionality (sections 4.2 to 4.5).

1 Natural Language Engineering

Natural language(NL) research at Durham University is concerned with NLE rather than the more traditional computational linguistics. Much computational linguistics orientated natural language processing (NLP) has concentrated on either trying to formulate universal theories that cover all aspects of language or developing very restricted theories which model small areas. The utilisation or expansion of these ideas to realistic systems which are not highly restricted by their task or domain has proved a great problem. Problems associated with other engineering disciplines which have to be considered in NL are:

Scale: The size of systems (e.g. grammar coverage, vocabulary size, word senses) must be sufficient to cope with realistic large-scale applications.
Integration: Components of a system must not be such that unreasonable assumptions are made about other parts. This is often the case when specific NLP problems are tackled in isolation. Components should be designed and implemented so that they assist other components.
Flexibility: There must be the ability to modify systems for different tasks in different domains.
Feasibility: For example, hardware requirements must not be too great and execution speeds must be acceptable. This process involves making the system and its components efficient.
Maintainability: There is a need for the system to be useful over a long period of time. The maintenance of a large system has proved to be an important aspect of the software life-cycle [LS80].
Usability: The system must be able to support the applications that end users want and be user-friendly.
Robustness: This is a critical aspect of large-scale systems. To quote [GSJ93] "While it [robustness] may not be a serious problem for any individual application, it has to be faced up to in general". This aspect concerns not only the linguistic scope of the system but also how it deals with input which falls outside of this scope.

The fact that there are a large number of systems and projects with very restrictive aims, and yet few that can claim to successfully address these issues, suggests that they have associated **intrinsic research problems of their own**.

The NLE method has foundations in the belief that it is not necessary to wait for complete linguistic theories covering all the problems associated with NL (which do not exist at present) before large, realistic and useful NL systems can be built. Instead a full array of artificial intelligence techniques is employed ranging from using well-developed linguistic and logic global theories (where they exist) to using more localised theories, corpora, knowledge based heuristics, adaptive techniques and, at the lowest level, ad-hoc rules. Incorporating this wide range of methods means that the development of the system does not get stuck due to the difficulty in following a particular logical or linguistic theory while the benefits of such well established theories can still be enjoyed. The result is a practical, working solution.

2 The LOLITA system

The LOLITA system has been developed over the last eight years at Durham University. It belongs to only a small group of systems which can claim to have addressed most of the properties required of large engineered systems as described above. The rarity of systems such as LOLITA is exemplified by the fact that NL system terminology defined in [GSJ93] has to be extended to define LOLITA's status; it is more than a generic system as it is not restricted to a single task type, but it is not, as it stands, a general purpose machine which can be used for any task in any domain. We extend the terminology by defining LOLITA as a *general purpose base*. Although demonstration prototypes have been built using LOLITA for various tasks and domains (see following section) no polished final application has yet been developed. This is because our research resources have been concentrated on the 'base' of the system and thus the task-dependent development has not resulted in such applications.

2.1 Applications of the LOLITA system

This section briefly mentions examples of applications other than translation for which the LOLITA general purpose base has been utilised.

2.1.1 Contents scanning

This involves building summarising templates from input texts (e.g. the MUC task [DAR91]). Input text is parsed and semantically analysed in order to build a full representation in the semantic network. An application (i.e. domain) dependent module, with the assistance of the general inference module, then searches the network for information relevant to each of the slots. This information, in the form of nodes in the network, is passed to the natural language generator, which produces the appropriate English for the template slot. For more information see [GMS93].

2.1.2 Chinese tutoring

The wide variety of possible applications which can be built on the LOLITA base is exemplified by the Chinese tutoring prototype [WG92]. The prototype involves tutoring students learning Chinese to overcome the problem of transfer errors caused by mother tongue influence.

2.1.3 Story application

This allows a user to interactively build paragraph pieces of text by passing a series of semantic network nodes, together with stylistic constraints, to the NL realiser. This is the basis of a project to aid in the generation of text for disabled users.

2.1.4 Dialogue analysis and generation

LOLITA contains a dialogue analysis component which generates interactive dialogues taking into account factors such as the user model, the situation and motivational information and constraints [JG93]. The dialogue can be optimised using an evolutionary programming algorithm [NG94] which exemplifies the eclecticism of methods used in the system.

3 LOLITA system components

The following subsections will describe the operation of the important sub-components of the LOLITA system. The descriptions are not application specific. When the prototype translation application is discussed in section 4, translation specific details will be given with special attention to the modifications which had to be made in order to allow the LOLITA base to be used as a translator.

3.1 Semantic network

LOLITA is dedicated to natural language processing and reasoning. It is based on the idea that thought is concept driven; concepts are created by people to model their environment. When people want to talk frequently about these concepts, they create new words in their language, each as a convenient short label, thus avoiding lengthy paraphrases (a topical example is 'yob'). Many concepts are not used frequently enough by the people who use a language to justify a separate word for each. Thus concepts far outnumber words.

This situation is reflected in the semantic network structure used by the LOLITA system: "concepts" are represented by nodes, and relations by arcs, just as in the Conceptual Graph representation [Sow84]. Some concepts are connected directly to words, whereas other concepts are only connected to words via other nodes.

This approach is unusual among NL systems. Earlier systems assumed there was a one-to-one mapping between words and concepts (for example, Conceptual Dependency Theory [Sch75]). More modern researchers have recognised that the word/concept mapping is not isomorphic but assume that the granularity of words is FINER than that of concepts [Ste94].

The semantic network structure of nodes and arcs is used to store LOLITA's knowledge. Although there are many other ways of representing knowledge, such as predicate calculus, the graph structure allows a far greater ease of expression for much of the knowledge that we communicate using natural language. This network is used by all of LOLITA's algorithms, which must search for the information they require. As search is a basic task in such a framework, the representation used must be efficient: relevant information is not accessed unless necessary, but is readily available. This is achieved by carefully choosing the arcs attached to nodes, by designing the representation to be as unique as possible, and by enforcing a normal form where this proves possible. Moreover, the net is an extended form of extensional Montague semantics. This set-based approach allows a higher efficiency of algorithms relying heavily on search, such as inheritance and disambiguation.

An interesting aspect of the semantic network is that the meaning of each node is given by the whole of the network. However, knowledge is distributed within the network, so that any segment of semantic network contains valid, albeit incomplete, knowledge.

LOLITA's semantic network now consists of 75000 nodes, and the knowledge it expresses is being constantly extended (for example, information from WORDNET [Mil90] is currently being integrated). Most of the labelled nodes have an English label – the word corresponding to the concept in English, but there are also about 1000 Chinese, 100 Spanish and 500 Italian labels, each with its associated linguistic information.

As an example, Figure 15 - 1 shows a much simplified semantic representation of an event.

3.2 The Parser

The parsing component of LOLITA involves morphological analysis and syntactic parsing. The morphological process extracts and labels the roots of the input words, making use of some of the grammatical information in the semantic network. Sometimes multiple extractions are required. For example, morphological analysis of the word 'unworthiness' will extract and label the word 'worth' by separating out the components 'ness' which makes an adjective into a noun, 'un' which indicates a negative, and 'y' which turns a noun into an adjective. There is also a facility at this preparation stage for recovering misspelt words and guessing unknown ones.

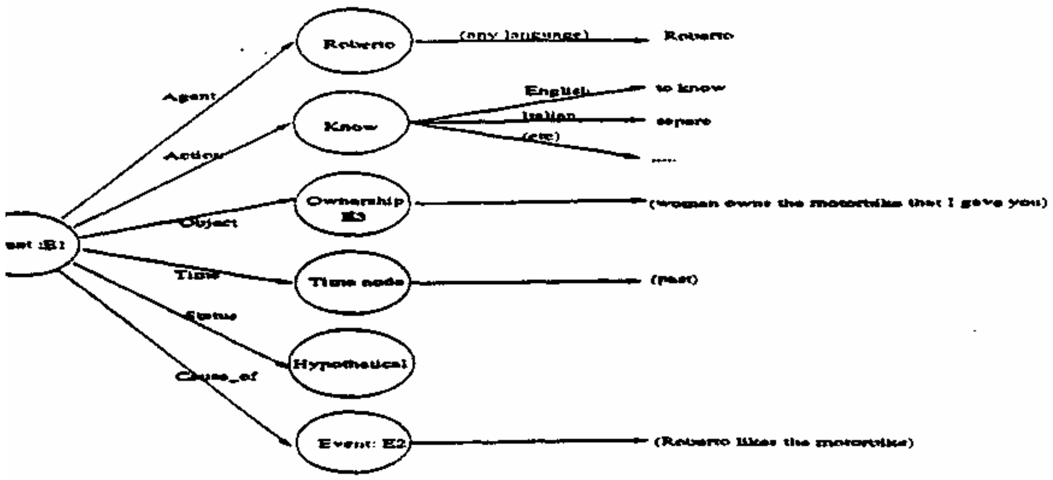

Figure 15 - 1 : Example of a semantic network event

The parsing is achieved using a 'deterministic grammar and parser' model. A deterministic parser is one that operates on the input in a left to right manner and is able to correctly classify every word and construct every syntactic category as it appears (using some sort of limited look-ahead facility). This prevents the parser from being misled by local ambiguities which may lead to the inefficient processes of backtracking. At the moment, the grammar is written in a BNF style, augmented with feature and determinism rules. Since the insertion of determinism rules in the grammar is a complex process, the current parser is not completely deterministic. However, there is a project currently underway in the Laboratory for Natural Language Engineering at Durham to automatically transform grammars each into a deterministically parseable form [EGM83]. Methods have already been developed to perform this task and are in the process of being implemented.

The grammar required for analysis of real-life text is extremely large. The LOLITA grammar has been built to account for special 'turns of phrase' which may or may not be accepted to be grammatically correct but are nevertheless often used and must therefore be handled. This means that the grammar must be much larger than one found, for example, in a grammar book. Another feature of the LOLITA's parsing system is its ability to cope with many types of error in the input text. Possible missing words or constructs at the beginning or end of each sentence or clause can be investigated. (The parser does not attempt to identify missing words in the middle of clauses as often even humans cannot understand these). This error handling feature again requires a much larger search space. The huge size of the grammar required to cope with these possible additional constructs and errors is the reason why deterministic parsing is needed. The fact that the grammar is not yet completely deterministic is reflected in the fact that an erroneous sentence requires more parsing time than a grammatically correct one.

The LOLITA parser can produce the best parse tree or a list of possible parses representing the deep grammatical structure of the input. Each parse tree has all word features extracted (e.g. verb root rather than third person singular etc.), errors (structural or feature caused) printed out, missing parts inferred and un-parseable parts isolated. Figure 15 - 2 shows the parsing of the sentence 'The cow jumped over the moon' whilst figure 15 - 3 shows an example of parsing the ungrammatical sentence 'and I likes him own'.

```
The cow jumped over the moon

 detph
   det THE
   commoun COW [Sing,Female,Per3]
 auxphrase_advprepph
   compintransv JUMP [Past]
   prepp
     prep OVER
     detph
       det THE
       commoun MOON [Sing,Neutral,Per3]
```

Figure 15 - 2 : Example of parsing

```
and I likes him own

subsen_phrase
join AND
sen  * clash: Per3 *
defpronoun I [Sing,Sexed,Nom,Per1]
sentvbph
  sentverb LIKE [Pres,Per3]
  sen  * clash: NoPer3S *
    defpronoun HIM [Sing,Male,Nom,Per3] * clash: Acc *
    transvp
      comptransv OWN [Pres,NoPer3S]
      conjtermph * MISSING *
```

Figure 15 - 3 : An example of 'parsing' a grammatically incorrect sentence

3.3 The Semantic Analysis

The semantic analysis in LOLITA creates the representation of natural language statements in the semantic net. To achieve this, it uses the grammatical parse tree of the statement. Each type of branch or leaf of the parse-tree has a label expressing its grammatical nature. This grammatical nature corresponds to only a few possible meanings in natural language. Thus, by assigning a rule to each label which is capable of determining the particular meaning, a set of rules is created which, when applied to a parse-tree, creates the corresponding semantic representation.

However, this does not take into account ambiguities. The parse trees provided by the grammar express ambiguity by special OR branches. These alternative meanings, or complete grammatical structures, are built lazily, in such a way that the common semantic structures are not rebuilt: only the word building structures which are later judged incorrect by disambiguation algorithms will be lost. Moreover, the stress is on disambiguating as early as possible in order to minimise the search space of possible meanings.

The resulting semantic structures produced are then normalised as required by section 3.1. This process involves transforming composite verbs into their conceptual equivalents, transforming expressions into corresponding concepts and ensuring the set-based properties of the net are maintained. An example is the conversion of the composite verb "go off" into the concept "go_off", a synonym of "explode".

Semantic disambiguation corresponds to disambiguation that can be achieved through valid reasoning. In the current version of LOLITA, disambiguation is limited to application of selectional restrictions. An example of this is: "John hit a lamp post. It happened very suddenly." At a grammatical level, "it" could refer to the event or the lamp post, however a selectional restriction can be used to disambiguate, since events can "happen" but concrete objects can't. Work is currently being conducted into more advanced methods such as semantic distance for establishing preferential meanings. It is important to note that the semantic analysis deals only with literal meaning; currently metaphors and jokes are either not believed if they contradict selectional restrictions, or are accepted at face value.

Certain features of language require a deep analysis to extract their meaning in such a way that it is usefully expressible in the semantic net. One such feature is tense: many NL systems use a model of tense based on that by Reichenbach [Rei66]. LOLITA however performs a complete analysis into the time primitives used to express any other temporal relation. This allows temporal reasoning to use the information provided by tense while only dealing with one temporal model.

3.4 The Generator

One of the most important factors which determines the design, scope and success of a Natural Language Generation (NLG) module is the input it assumes. While other NLG modules take as input static databases (e.g. [McK82]), customised specification languages in varying degrees of detail (e.g. NIGEL's SPL [Man83]) or specific formalisations (for example function descriptors, e.g. [MEF+90]), the LOLITA generator [Smi94] [SGM94] starts from a semantic network representation of the meaning to be expressed. Comparison between different generators is thus very difficult and, despite recent attention, standard methods for evaluating a generator's output do not exist [GSJ93].

Notable generating systems which do take similar input representations to that assumed by the LOLITA generator are those which generate from conceptual dependency representation e.g.[Gol75] [Hov88], conceptual graphs e.g. [Sow83] [NZ92], the SNePS representation e.g. [Sha82], and those based on an MTM model e.g.[IKP88]. The representation on which these systems are based, however, make different assumptions on the granularity of concepts and words and the mapping between them (see section 3.1 and [Ste94]).

The task of the LOLITA generator (and those which assume similar input) is to express the meaning of a node in surface NL. Despite the fact that theoretically a node's meaning is given by the whole semantic network, the generator is never required to generate the whole network. Depending on whether the input node is a generic concept, an instantiation of a concept, a simple event or a complex event, the generator will produce utterances of lengths ranging from single words, simple noun phrases, simple sentences, complex sentences to a few separate sentences. Thus the generator is partially message directed [Met93]; the type of nodes and presence of arcs to other nodes affects the generator's operation. However, control is also provided by the grammar of the language to be produced and is thus also partially grammar directed [Met93]; grammar dictates the order in which particular arcs are followed from node to node. Finally, control is also provided by a set of realisation parameters which can be set by the underlying application to create stylistic variation. These parameters can, for example, control the length of the utterance, the rhythm (the length of individual sentences) and grammatical styles (e.g. active/passive, dative/non-dative etc.).

To generate an expression for an event such as that in figure 15 - 1, the realiser will operate by following arcs (e.g. agent, action, object) from this event to other nodes in the network and recursively generating expressions for these nodes. If the default realisation parameters are being used, the events will be generated in the active voice with a rhythm of one relative clause for each entity. The output for this example may be 'If Roberto had known that the woman whom he loves owned the big fast motorbike that I gave him then he would like it'. (Not all the information required to produce this utterance is shown in the diagram.) An unsimplified portion of the semantic network may, of course by more richly populated. There may for example be many more arcs from the node representing 'Roberto' which link to more information about him. If planning instructions (which vary according to the underlying application) indicate that this information should be expressed, it is likely that the realiser will have to split the utterance into separate sentences. Events which are encountered by the realiser which cannot be immediately expressed (because the resulting sentence will be too long) are placed on a stack so that they can be expressed as separate sentences. Heuristics are used to order this stack of events so that coherent focus and decipherable anaphoric references are maintained. (The development of these heuristics is ongoing). Another source of variation comes from the choice of starting point for generation. If, for example, the realiser was passed the node representing the event E2 instead of E1 then the 'story' will be realised from a different 'angle'. This method of producing different utterances by varying the starting node is utilised in the translation prototype (section 4).

Because of the underlying representation in which the granularity of concepts is smaller than that of words, it is possible that a concept does not have a single word which can be used to express it. In this case the realiser must produce a paraphrase expression by 'generating around' the particular concept node. This is especially important in translation and will be examined in more detail in section 4.5.

4 Translation using the LOLITA system

The components of the LOLITA system described in the previous sections have been designed to be application independent so as to form a generic base on which applications can be built. As prototype applications have been built, any new functionality or modification of existing functionality have been analysed carefully and, where they have been considered to play a more general role in NLP, have been realised in the generic part of the system rather than in the application itself. So for example, the Chinese tutor system's need for Chinese labels on nodes rather than English labels gave rise to a semantic network structure in which nodes representing concepts may be linked to nodes representing words by a link which specifies the language of the word node. Clearly, this structure is also useful in the context of a translation application.

Given the ability of the generic part of the system to move from text to concept and back again, an obvious approach to translation with the LOLITA system is to move from text in one language to the corresponding semantic network representation and then realise this semantic network representation in some other language using the generator. Although for some applications this approach might be adequate, many applications will also require the propagation of stylistic aspects of the original texts. To achieve this the system would need to be capable of analysing the style of the source text, as well as its meaning, thereby producing separate concept and style intermediate forms which could then be rendered by the generator.

To test this general approach and also the flexibility of the LOLITA system, we have built a prototype Italian to English translator. Currently this work is very much at an experimental stage, and is certainly not a completed translation system. As noted previously, the semantic analysis is limited to literal meaning. Thus translation of literary texts is for all intents and purposes excluded. The current implementation of the generator can only generate English, and extending it to other languages would require rewriting it. Moreover the generator produces an English language expression for every entity and event concept gleaned from the input text. Thus LOLITA's translator does not produce one polished translation, but rather a set of utterances which view events in the text from different angles. Also the analysis of style has not yet been implemented, although the generator is capable of producing various styles.

Despite these shortcomings the translation system is interesting because of the ease with which it has been implemented, the main work lying in the addition of an Italian vocabulary, morphology and some Italian grammar. So far, the approach of reusing the techniques and decisions of other applications throughout the medium of the generic core of the LOLITA system has paid off. Many of the decisions in the basic design were made for the purpose of reasoning and searching within a natural language framework. These decisions have revealed themselves to be remarkably adaptive to the field of translation. The next sections will analyse them in detail, giving an example of the translator in operation and then discussing the modifications which had to be made to each of the sub-components introduced in section 3.

4.1 Example Translation

This section will give an example of the capabilities of the prototype Italian to English translator. Although the coverage and data for the Italian language are not large (currently only about 500 Italian word labels being incorporated in the LOLITA semantic representation), it is hoped that the example will illustrate that complicated grammatical structures and sentences can already be translated. Figure 15 - 4 shows an Italian sentence entered by the user together with translated utterances provided by LOLITA. As mentioned above, a polished single translation is not produced but each of the utterances can be used either separately or together to build up the meaning of the input sentence.

Input Sentence:

```
se avessi saputo che la grossa moto veloce che mi hai dato era posseduta da
lei, mi sarebbe piaciuta, perche' lei mi piace davvero.
```

Output Utterances:

```
If I had known that the woman whom I love owned the big fast motorbike
that you gave me then I would like it.

I love the woman who owned the big fast motorbike that you gave to
me. I would have liked it if I had known this.

I would have liked the big fast motorbike that you gave me if I knew
that the woman owner whom I love owned it.

The woman owner whom I love owned the big fast motorbike that you gave to
me. I would have liked it if I had known this.
```

Figure 15 - 4 : Example of LOLITA's translation capability

4.2 The Parser for Translation: Adding Italian Morphology and Grammar

Since Italian makes much greater use of morphology, a separate morphological analyser is employed. However even this makes use of the same structures and basic algorithm as the English morphology; only the rule set is completely different.

In contrast to this, the Italian grammar has been written using a 'mixed grammar' approach. This involves using a single grammar for all languages with which the system deals, but marking some of the rules to indicate their validity (or otherwise) for particular languages. Unless otherwise marked, a rule applies to all languages (but it is quite possible that an unmarked rule is never referenced by any applicable rules in a particular language).

This approach is particularly suited to situations in which substantial parts of the grammar can be shared among languages, and has allowed us to very quickly build an Italian grammar capable of parsing the complex sentence we are using in our example (Figure 15 - 4). In fact, the Italian grammar makes use of the English grammar to such an extent that only five new rules needed to be added.

These new rules all involved the high level structure of Italian and at the low level it was found that the majority of rules already existed for English.

As an example, we take the rule that deals with relative sentences which contain a missing part (covering cases such as "you bought" in "I like the dog you bought"). In simplified form, this rule is written as follows:

```
rel_sentence
  = jointermph (rel_auxphrases | rel_adv_preph)
```

It allows a full jointerm phrase to be followed by a relative auxiliary phrase or a relative adverbial prepositional phrase either of which could have a missing part.

This effectively allows only a missing object, which is correct for English. However, in Italian, it is also possible to have a missing subject in the jointerm phrase, since the morphology carries information about the subject (e.g. 'il cane che hai comprato'). The rule has therefore been extended as follows:

```
rel_sentence
 =   jointermph (rel_auxphrases | rel_adv_preph)
 | ital_gate ital_rel_sentence

ital_re_sentence
 = soft_imp_jointermphrase (re_auxphrases | re_adv_preph)
```

The interesting point about this example is that the components of the `ital_rel_sentence` rule are already defined in the grammar so, although we have had to modify the grammar to allow for a different structure of relative sentence, the lower level components belong to both the Italian and English cases.

An added benefit of the mixed grammar approach is that it allows the system to parse sentences from one language that incorporate rules from another. This is particularly useful in the Chinese tutor application where the system can detect negative transfer in the form of English grammar rules in Chinese sentences input by the student. The parser initially looks for a parse using only Chinese grammar but, if no such parse can be found, it relaxes the restriction and allows an increasing number of English rules (since the system is intended for English students of Chinese). In the latter case, the student is told of his/her error and the tutor takes appropriate remedial action.

4.3 The Semantic Representation for Translation

LOLITA's semantic net has a number of features which render it particularly suitable for translation.

As well as containing nodes which represent concepts, there is a separate linguistic layer of nodes which correspond to words and contain both the string of characters and the grammatical categories of the words. Each such node is linked to its corresponding concept by a 'concept' arc, and concept nodes are linked to word nodes using the appropriate language arcs. This not only allows concepts to be expressed by individual words in different languages, but also allows synonyms which may happen to have different grammatical features, such as different gender, to be mapped onto the same concept by the semantic analysis. Moreover, this feature is readily exploitable within the context of many languages: as a general rule the language dependent parts of LOLITA access the linguistic nodes, and the concept dependent ones the conceptual nodes. By assigning a particular arc to each language so as to connect concepts with the relevant linguistic nodes, the linguistic information is always readily available, but does not interfere with the correct execution of the concept using algorithms. A further advantage of this separation of linguistic word nodes and concept nodes is that it allows LOLITA to talk and reason about words as distinct from the concepts they represent.

Phenomena such as tense are deeply analysed to allow reasoning. This is important as the structure of tense is language dependent, whereas the corresponding structure within the semantic net is not. Thus Chinese, which does not have any conjugation at all but uses explicit temporal references and three special particles to express the time at which an event occurred, will be analysed into the same temporal representation as a language with a full system of tenses such as ancient Greek.

4.4 Semantic Analysis for Translation

The semantic analysis module changed very little in the translation experiment. None of the original code was modified for this purpose, and only one rule for a grammatical construction specific to Italian had to be added (less than 0.1% of the code). This illustrates the advantage of the approach: since the semantic construction rules depend on meaning of the grammatical structure derived by LOLITA's parser, they did not change. Only new grammatical structures will require the addition of such rules. Later operations such as disambiguating, and ensuring the set properties of the semantic net are preserved, are all either conceptual or structural, and are language independent.

The representation's useful features for reasoning also turn out to be useful in building semantics for new, previously unconsidered, languages. An example is the Chinese handling of tense discussed in the previous section. Had the conceptual representation been closer to that of European languages, it might have assumed that tense is a feature of the action, and have encoded the tense information there. This would have complicated the semantic code significantly. Fortunately, reasoning algorithms were more efficient if events were connected to temporal nodes expressing the time of their occurrence.

Another feature used for reasoning is the set-like behaviour of concepts. This turns out to be also very useful in translation. For instance, in Italian the word "drizzle" does not exist, and the concept is expressed by the paraphrase "sparse rain", or "pioggia rada". A literal translation is unwieldy. But the semantic analysis tries to analyse the source natural language text into the least ambiguous representation possible, given the available information. Thus, whether the sentence were in English or in Italian, it would be disambiguated to the concept "drizzle". This is achieved by using set based information that drizzle is a subset of rain and is qualified by the fact that the raining phenomenon is of lower intensity than the expected norm. Moreover sparseness is an element of the set of all properties expressing a lower intensity of a phenomenon. Thus sparse rain is deduced to be an instance of drizzle, and the paraphrase can be eliminated.

However we must temper this picture of success by pointing out that some of the language dependent parts of the semantic analysis will require minor change.

- The analysis of tense is currently the same, irrespective of the language being analysed. Although the relative times at which events occur do not vary in English, French or German, their aspect does. Thus the French present tense does not convey a likelihood of repetition, whereas the English does. This is illustrated by "Que fais tu? Je mange une pomme" sounding correct, but "What you do? I eat an apple" sounding awful. The English would use the progressive in this sense.

- Various analyses can be extended to use the different information provided by other languages: an example is the use of pronouns for grammatically sexed but conceptually inanimate entities.

- The language dependent normalisations need to be adapted: an example is the treatment of composite verbs: a German literal translation of "go off" does not exist. Similarly many German composite forms do not exist in English. Another example is the treatment of expressions. It is of interest that not only do some expressions exist in different languages, but even if they do not, their normalisation may each provide the best interpretation of an otherwise meaningless phrase: for instance, "to make a threat" has a literal translation in German, but not in French. However the best interpretation of "faire une menace" would be to threaten.

4.5 The Generator for Translation

The natural language generation module has not had to be modified for the prototype Italian to English translator described here. Currently, however, the generator can only generate English output and thus English is the only possible target language. In order to develop a translation system for other target languages, the generator will have to be extended by adding different rules of grammar and morphology. As it stands, English rules of grammar are embedded into the generator and modification for other languages would require a great deal of work.

Although the problem of finding an expression for a concept which does not have a corresponding word in a language must be handled as part of the generator's normal functionality, it is particularly applicable in machine translation. Whereas the task of semantic normalisation is to map input text onto the most specialised concept, it is the task of the generator to express specialised concepts for which there is no direct link to a surface word.

This task is assisted by the fact that the semantic network is encoded so that relevant concepts are topologically 'near' each other. The concept of topological distance within the semantic net corresponds to the minimum number of arcs that must be traversed to reach a destination node. This principle of locality is essential to the efficiency of all LOLITA's algorithms and is useful to the generation process: For instance, the concept "unpleasantly soft" is expressed in Italian by the word "molle". This word will be connected to the "unpleasantly soft" concept in the semantic network, but this concept node will not have a link to any English word. However, it will be connected to the concepts "soft" and "unpleasant" in such a way as to indicate that it is the intersection of all things soft and of all things unpleasant. The generator can therefore move up from the "unpleasantly soft" concept to the "unpleasant" and "soft" concepts and generate the appropriate paraphrase.

5 Implementation Details

LOLITA is implemented in the functional language Haskell. It comprises a total of approximately 32,000 lines of source code equivalent to approximately 300,000 lines of imperative code. The modifications in the code required for the translation prototype amount to about 0.5% of this total. LOLITA runs on a 48Mb Sparc workstation. The translation in the example given in this paper takes a few seconds to achieve.

6 Conclusion

Although LOLITA was not initially designed to be a translation system, this experiment has shown that with relatively little effort a promising Italian to English prototype has been developed. Although the coverage and robustness of the translator is by no means great, it has been shown that complicated Italian sentences can already be translated.

Further work is being undertaken to increase the coverage of the Italian translator and to add data and rules for other source languages. Work has also been initiated to investigate the possibility of modifying the generating module so as to produce translations in other target languages. Finally, an analysis of style extraction techniques has already been performed with a view to implementing a style analysis module for LOLITA.

For any further details on the LOLITA project please do not hesitate to contact the authors.

References

[DAR91] DARPA. *Proceedings of the Third Message Understanding Conference*, 1991.

[EGM83] Nigel R. Ellis, Roberto Garigliano, and Richard G. Morgan. A new transformation into deterministically parsable form for natural language grammars. In *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, Netherlands, August 1983.

[GMS93] Roberto Garigliano, Richard.G. Morgan, and Mark H. Smith. The LOLITA system as a contents scanning tool. In *Proceedings of the 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language Processing*, Avignon, France, May 1993.

[Gol75] Neil M. Goldman. Conceptual generation. In Roger C. Schank and Christopher K. Riesbeck, editors. *Conceptual Information Processing*. American Elsevier, New York, NY, 1975.

[GSJ93] J.R. Galliers and K. Sparck-Jones. Evaluating natural langauge processing systems. Technical Report 291, Computer Laboratory, University of Cambridge, 1993.

[Hov88] Eduard H. Hovy. *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988. Based on PhD thesis, Yale University.

[IKP88] Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. Implementing a meaning-text model for language generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, August 22-27, 1988.

[JG93] C.E Jones and Roberto Garigliano. Dialogue analysis and generation: A theory for modelling natural English dialogue. In *EUROSPEECH '93 volume 2*, pages 951–954, September 1993.

[LS80] B.P Lientz and E.B Swanson. *Software Maintenance Management*. Addison-Wesley, 1980.

[Man83] William C. Mann. An overview of the Penman text generation system. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, pages 261–265, Washington, DC, August 22-26, 1983.

[McK82] Kathleen R. McKeown. The TEXT system for natural language generation: An overview. In *Proceedings of the 20th Annual Meeting of the ACL*, pages 113–120, University of Toronto, Ontario, Canada, June 16-18, 1982.

[MEF$^+$90] Kathleen R. McKeown, Michael Elhadad, Yumiko Fukumoto, Jong Lim, Christine Lombardi, Jacques Robin, and Frank A. Smadja. Natural language generation in COMET. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*, pages 103–139. Academic Press, New York, 1990.

[Met93] Marie Meteer. *Expressibility and the Problem of Efficient Text Planning*. Francis Pinter Publishers, London, 1993.

[Mil90] G.A Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.

[NG94] Dave J. Nettleton and Roberto Garigliano. Evolutionary algorithms for dialogue optimisation in the LOLITA natural language processor, Jan 1994. Seminar on Adaptive Computing and Information Processing.

[NZ92] J. Nogier and Michael Zock. Lexical choice as pattern matching. In T. Nagle, J. Nagle, L. Gerholz, and P. Elklund, editors, *Conceptual Structures: current research and practice*. Ellis Horwood, New York, NY, 1992.

[Rei66] H. Reichenbach. *Elements of Symbolic Logic*. Free Press, New York, 1966.

[Sch75] Roger C. Schank. *Conceptual Information Processing*. North Holland, Amsterdam, 1975.

[SGM94] Mark H. Smith, Roberto Garigliano, and Richard G. Morgan. Generation in the LOLITA system: An engineering approach. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 241–244, Nonantum Inn, Kennebunkport, Maine, June 21-24 1994.

[Sha82] Stuart C. Shapiro. Generalized ATN grammars for generation from semantic networks. *Computational Linguistics*, 8:12–26, 1982.

[Smi94] Mark H. Smith. *Natural Language Generation in the LOLITA System: An Engineering Approach*. PhD thesis, Department of Computer Science, University of Durham, 1994.

[Sow83] John F. Sowa. Generating language from conceptual graphs. *Computers and Mathematics with Applications*, 9(1):29–43, 1983.

[Sow84] John F. Sowa. *Conceptual Structures (Information Processing in Mind and Machine)*. Addison-Wesley, 1984.

[Ste94] Manfred Stede. Lexicalization in natural language generation: A survey. *Artificial Intelligence Review*, 1994.

[WG92] Yang Wang and Roberto Garigliano. *An Intelligent Tutoring System for Handling Errors Caused by Transfer*. Lecture Notes in Artificial Intelligence, 608. Springer-Verlag, Montreal, Canada, 1992.