ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

Diplôme de Recherche Technologique
Communication Homme-Machine

Présenté et soutenu publiquement le 10 novembre 2000
par

Eric CRESTAN

Improvement of French generation for the KANT machine translation system

Composition du jury :

Eric Gaussier	XEROX, Grenoble	Rapporteur
Paul Sabatier	LIM-CNRS, Marseille	Rapporteur
Eric Nyberg	CMU, Pittsburgh	Examinateur
Jeffrey Allen	MIT2-Softissimo, Paris	Examinateur
Henri Meloni	LIA, Avignon	Examinateur
Marc El-Bèze	LIA, Avignon	Directeur de recherche

Language Technologies Institute
Carnegie Mellon University

Laboratoire d'Informatique d'Avignon

Abstract:

The Carnegie Mellon University' KANT system is a knowledge-based interlingua machine translation system developed to translate English document into a wide range of languages. It is a high quality machine translation system requiring controlled English sentences as input. First, we give an overview of machine translation. Then we describe the KANT project and the architecture of the system. Third, we present the largest part of our work on improving French generation, including work on gerund translation and examples of lexical selection rules. These rules have been written under a formalism developed at the Center for Machine Translation. This formalism has been conceived in order to achieve the constitution of F-Structures from Interlinguas. Finally, we propose the utilization of a unilingual statistical language in order to correct erroneous determiners and prepositions in French sentences generated from the KANT system. We illustrate the behavior of the model through experimental results.

Résumé:

Le système KANT est un programme de traduction à base de connaissances. Il est destiné à la traduction de documents techniques rédigés en anglais vers une variété d'autres langues. Son fonctionnement s'appuie sur une représentation universelle intermédiaire dénommée Interlingua. Si ce système de traduction atteint un haut niveau de qualité, ceci est entre autres dû au fait qu'il a été conçu pour traiter des textes sources rédigés en anglais contrôlé. Nous donnons tout d'abord un aperçu du domaine de la traduction automatique. Puis, nous nous intéressons plus particulièrement au projet KANT et détaillons l'architecture du système. Ensuite, nous présentons l'essentiel de notre travail : plusieurs améliorations apportées à la génération du français, dont notamment les travaux effectués sur la traduction des formes *-ing* anglaises, mais également des exemples de règles de sélection lexicale Ces règles ont été écrites dans un formalisme développé par l'équipe CMT de CMU en charge d'assurer une transduction en F-structures de phrases représentées selon les formes appropriées de l'Interlingua. Pour finir, nous proposons l'emploi d'un modèle de langage statistique unilingue, destiné à corriger les phrases générées en français par le système KANT lorsqu'elles contiennent des prépositions ou des déterminants erronés. Nous illustrons le comportement de ce modèle au travers de quelques résultats expérimentaux.

CONTENTS

Preface

His biographer report that 19th-century mathematician Charles Babbage convinced British government officials to finance his research on a "computing machine" by promising, among other things, that it one day would lead to the automated translation of spoken languages. Although Babbage today is recognized as the creator of many ideas that led to the computer, he was never able to perfect his own machine, nor to fulfill his promise of machine translation.

By Jeff Moad
January 23, 1998

Acknowledgements

I would like to thank KANT project managers, Dr. Teruko Mitamura and Dr. Eric H. Nyberg 3rd, for their guidance and advisement through this work. They provided me a pleasant and friendly work environment, and gave me the approval in order to expand my research. I also would like to thank my office maids, Mahlon Stoutz and Enrique Torrejon, for their help in understanding the subtleties of the English language.

I would like to express my appreciation to the other members of the Center for Machine Translation for their support and kindness along the 18 months I spent among them.

I would like to thank Pr. Marc El-Bèze to providing me with the support and guidance needed to develop a coherent presentation of the research.

I would finally like to thank my companion, Andrea Wattky, for all the difficulties she had to overcome in order to join me in Pittsburgh while she was carrying on remotely her study in France; and for all the support she provided me during this period.

1 Introduction

Since the beginning of humanity, mankind has been dreaming of a common language among them. Nevertheless, all the attempts to impose such a language, even recent, have failed. The twentieth century and the apparition of computers opened new possibilities, not in imposing a common language but in creating translation tools.

A huge evolution in the quality of translation has been made since the beginning of the century, but most actual machine translation systems are only good enough in order for a user to get the basic meaning of a document, not an accurate translation. Some others, like the Carnegie Mellon University's KANT system, are able to achieve a satisfactory quality of translation by applying different constraints, such as controlled input language.

Along this report, we give an overview of machine translation (MT), starting with a history of MT and followed by its different approaches. Then in section 2, we describe the CMU's KANT-KANTOO project. As well as a history of the project, this section contains a description of the architecture of the interlingua-based MT process. In section 3, we present some recurring problems of English into French translation. In addition, we explain the porting process that was used in order to convert the system from KANT to KANTOO (Object-Oriented) technology. Then in section 4.3, we present some representative examples of improvement made on French generation. We conclude this section by displaying the results obtained with the latest version of the system. Finally, in section 4 we describe an experimentation with statistical language models made in order to reduce the postediting on determiners and prepositions in French translation. At the end, we produce the results obtained on two sets of test corpus and we conclude this section by submitting several propositions for improvement.

2 Overview of Machine Translation

2.1 History

The idea of machine translation is not new, already during the 17th century Descartes and Leibniz were speculating on the creation of mechanical dictionary dictionaries (Hutchins and Somers 1992). Nevertheless, their attempts remained only on a theoretical level such as the interlingua elaborated by Wilkins in his "Essay towards a Real Character and a Philosophical Language" (Wilkins 1668).

At the end of the 19th century and the beginning of the 20th century, several proposals of creating a universal language (Esperanto 1887, Interlingua 1903) have been made to overcome the translation problems. The two first mechanized translations appeared in 1933 when Frenchman George Artsouni clamed he had designed a storage device on paper tape, which could be used to find the equivalent of any word in another language. At the same time, a Russian proposal, based on a three stages mechanical translation, was presented by Petr Smirnov-Troyanskii. His approach was more ambitious and used a first step where an editor knowing only the source language was to undertake the "logical" analysis. Then, the second step was a machine transforming base forms extracted from the previous step into equivalent sequences in the target language. Finally, another editor, knowing only the target language, was to convert this output into the normal form of the target language.

From the apparition of computers in the mid-40s and until the 60s, numerous projects have been held around the globe with machine translation for objective, with the first public demonstration of MT system in Jan. 1954. Developed at Georgetown University by Leon Dostert in collaboration with IBM, the system was able to translate 49 Russian sentences into English, using a 250 words restricted vocabulary and only six grammar rules. That had a very favorable effect, because large-scale funding of MT research had been stimulated. Several centers of theoretical research were created like the MIT, the Harvard University, the University of Texas, the University of California at Berkeley, the University of Leningrad, at Cambridge Research Language unit (CLRU), and at the University of Milan and Grenoble.

In 1964, the government sponsored the Automatic Language Processing Adviser Committee (ALPAC), in order to examine the prospects of MT in the USA. This leaded to the very controversial 1966 report that concluded that MT is slower, less accurate and twice as expensive as human translation. That had as effect a drastic cutback of large-scale funding for many years.

During the following decade, MT research mainly took place in Canada and in Western Europe, but barely in the United States. The few research projects on MT were concentrated on translation of scientific and technical Russian documents into English. In Canada and Europe, efforts were held of other languages, such as English-French translation.

In 1976, the Commission of European Communities decided to use an English-French MT system, called Systran. In fact, this system was not new; it has been developed by Peter Toma and has been used since 1970 for Russian-English translation. The 1970s showed an important development of other language pairs, such as English-Italian and English-German. At the end of the 1970s, an ambitious research project was founded to develop a multilingual system for all the Community languages. This project took fully advantage from previous work held at Grenoble and Saarbrücken on designing an interlingua-based system for Russian-French translation.

Because of disappointing results obtained with interlingua-based MT systems, several research centers started to develop instead transfer-based MT system. As examples, we can refer to the METAL system developed at the Linguistic Research Center (LRC) at Austin, Texas, the Ariane system at Grenoble and the Mu transfer system for Japanese-English translation at Kyoto University.

During the 1980s, new ideas joined the interlingua approach, as it was done with the knowledge-based systems at Carnegie Mellon University, Pittsburgh. The principal idea was to integrate additional information, not purely linguistic (syntactic and semantic), in order to achieve a higher level of understanding.

More recently, new alternative techniques have emerged, such as the statistical approach for MT, borrowed from speech recognition. One of the most advanced statistical MT systems has been developed at the IBM Laboratory at Yorktown Heights, New York, (Brown 1990).

A new horizon appeared recently with the boom of commercial MT systems. American Products such as ALPSystems, Weider and Logos were joined by many other Japanese systems (Fujitsu, Hitachi, Mitsubishi, NEC, Oki, Sanyo, Sharp, Toshiba), followed in the later 1980s by Globalink, PC-Translator, Tovna, METAL and several other in-house systems. However, in order to achieve an acceptable level of translation quality, nearly all the systems required heavy post-editing.

2.2 Architectures

2.2.1 Direct Architecture:

The method used for the direct architecture is pretty straightforward, what generally provides very poor translation quality. Historically, this kind of architecture has been the first to be under development; that is why they were also called "first generation systems". However, it should be kept in mind that available computers in the late 1950s and early 1960s were very primitive and therefore very slow and low in resources. The direct architecture arises from a simple morphological analysis phase, where verb endings are identified in order to extract the lemmas. Using a bilingual dictionary, source language lemmas are translated into target language words. Some systems use reordering rules that would try to reorder locally some elements of the sentence like adjectives or verb particles. As a matter of fact, pair of languages with a significant discrepancy would result in an extremely low quality of translation.

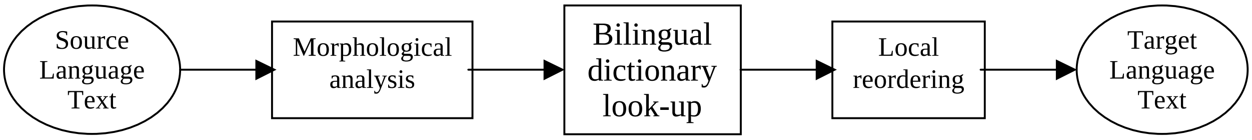

Figure 1: Direct MT system

It is obvious that this approach suffers from severe limitations. It can be assimilated as a word-to-word translation with some adjustments. It does not take into consideration any grammatical features or syntactic structures.

The failure of first generation systems led to the development of more sophisticated linguistic models, including deeper analysis of the source languages. Those are called indirect architectures.

2.2.2 Interlingua Architecture:

Disappointed by the results obtained with the direct transfer, research started to make its way toward an idealistic intermediate representation, which is the interlingua. It is issued from the analysis of a source text, then directly used to generate the target text. Interlinguas include all necessary information contained in the original sentence, it can be seen as an abstract representation of a source text as well as the target text (see section 3.2.2). That information should be sufficient in order to be able to regenerate the source sentence.

The idea of a universal representation, which is not language dependent, has been since left behind and interlingua systems are nowadays less ambitious.

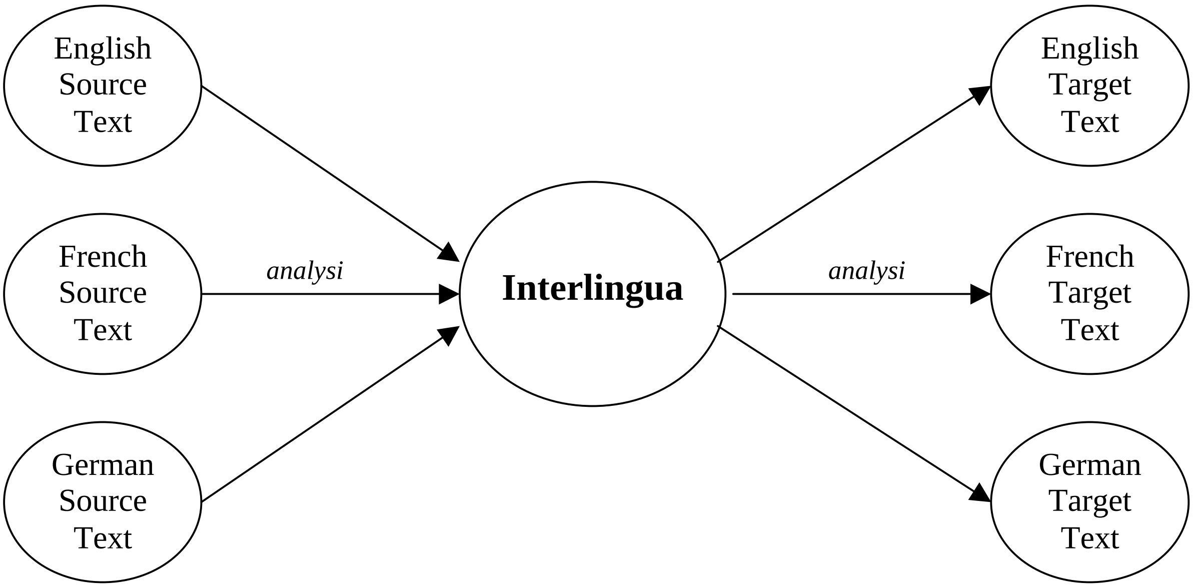

Figure 2: Interlingua MT system for six language pairs

The interlingua approach is very attractive because of the independence of its modules. Once the analysis is done, the same interlingua can be used to generate translations for multiple target languages. The choice of a target language or another will have no influence on the analysis process.

The advantage is that the addition of a new language to the system requires the creation of just an analysis module and a generation module. In addition to that, the developer of the new modules does not need to have any knowledge of other languages, at least in theory. However, in fact, it is a bit more complicated than that because such 'universal' representation does not exist, mostly due to structural differences between languages.

This was the reason why several projects were reoriented towards a less idealistic approach, which is the indirect transfer.

2.2.3 Transfer Architecture:

Although all translation systems involve a "transfer" of some kind, the paradigm *transfer method* has been used to describe systems that interpose bilingual modules between intermediate representations. It has a strong language dependency, because unlike interlinguas, the representation generated from the analysis is an abstract representation of the source text. In the same way, the representation that is issued from the transfer is an abstract representation of the target language. Therefore, three steps are needed: the analysis of the source text, the transfer from the source text representation to the target text representation, and the generation of the target text from this intermediate representation.

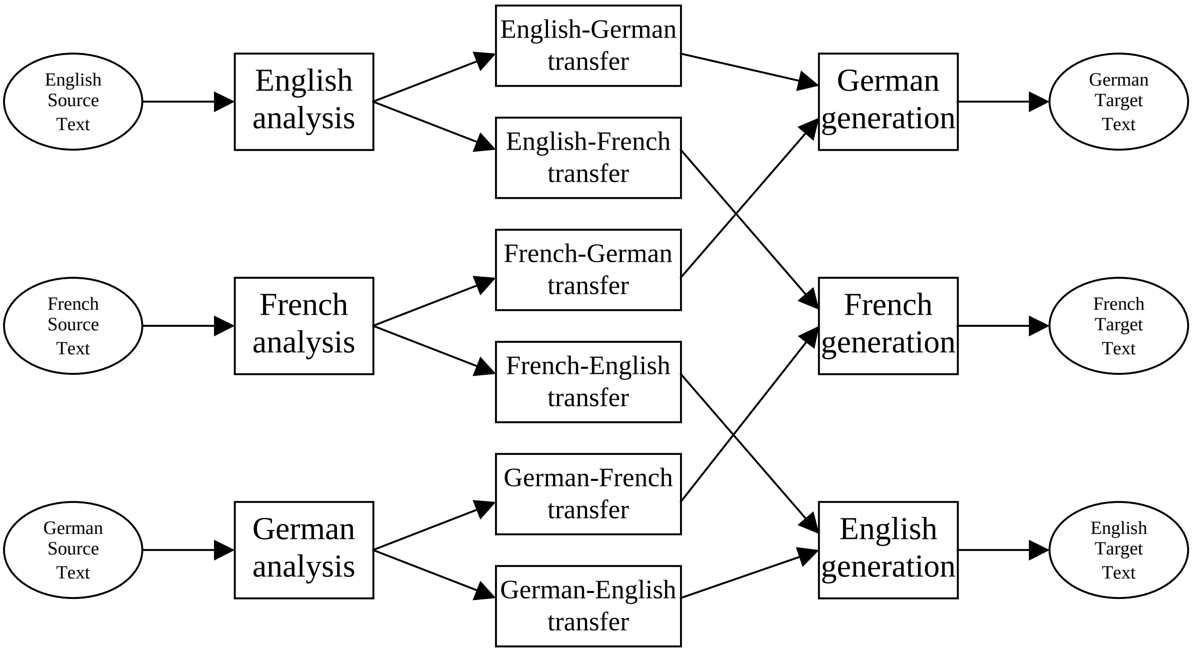

Figure 3: Transfer-based MT system for six language pairs

The major disadvantage of this method versus the interlingua method lies in the addition of new languages. While the addition of a new language with the interlingua approach would required the development of only two modules, with transfer approach it would require not only the development of an analysis and generation module, but also a transfer module.

But in spit of this disadvantage, transfer systems are still widely used. The first reason for this is that it is very difficult to create a truly language-independent representation. The second is the complexity of analysis and generation grammars that are required in order to obtain this "universal" representation.

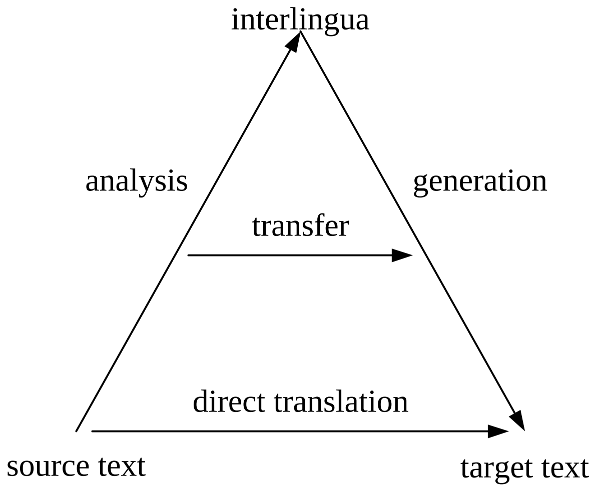

Figure 4: Vauquois Pyramid

To draw a conclusion from the three different architectures shown above, we can use the well-known Vauquois pyramid (see figure 4). This diagram illustrates the amount of required transfer regarding the amount of performed analysis. Therefore, the segment for direct translation is the longest, because of a succinct analysis, when the interlingua-based translation has the largest amount of analysis and the smallest amount of transfer.

2.3 Knowledge-Based Machine Translations

The paradigm of Knowledge-Based Machine Translation (KBMT) relies on explicit representation of world knowledge, which means a complete understanding of the meaning of source texts (Nirenburg et al. 1992). From an architectural point of view, KBMT belongs to the class of interlingua-based systems. However, the reciprocal is not true because systems like CETA (Vauquois and Boitet 1985), DLT (Wilkam 1983) and Rosetta (Landsbergen 1989) use interlinguas, but they are not knowledge-based.

The first KBMT system was developed in 1973 by Yorick Wilks at Stanford University, followed by Jaime Carbonell, Rich Cullingford and Anatole Gershman at Yale University (Carbonell et al. 1981) and by Sergei Nirenburg, Victor Raskin and Allen Tucker at Colgate University (Nirenburg et al. 1986). Since then, larger-scale development works has been done in this field, including ATLAS (Uchida 1989), PIVOT (Muraki 1989), ULTRA (Farwell and Wilks 1991), he KBMT system for doctor-patient communication (Tomita et al. 1987), KBMT-89 (Goodman and Nirenburg 1991) and DIONYSUS (Carlson and Nirenburg 1990).

The focus of KBMT paradigm is the development of knowledge-intensive morphology, syntactic and semantic data for a lexicon. In general, research in this field has been on the elaboration of underlying conceptualized representation. High-quality translation has been provided by recent systems, however, the amount of required information to provide a fully automated translation constrains developer to narrow the domain, to use controlled language and/ or manual disambiguation.

2.4 Other Approaches:

2.4.1 Example-Based Method:

The fast development of computer technology has opened new possibilities for machine translation. Hence, access to faster computers, larger memories and large data storage hardware allows MT researches based on large corpora of bilingual documents. The principle of example-based MT is simple: use bilingual text databases in order to find or recall analogous examples.

This method can be used as a substitute of traditional knowledge-based MT or can be used as a supplementary aid. Example-based methods split in two branches: the strict match type (Translation Memory systems) and the fuzzy match type, such as the Pangloss system (Brown, 1996) developed at CMU, Pittsburgh. Example-based MT systems are also widely used by free-lance translators.

Similar functions are also employed to compensate incomplete matches due to a lack of entries in the bilingual corpora (it is utopist to have a database containing all possible source language sentences). Those similarity functions depend on some measures of distance of meaning (e.g. classification of semantic items in semantic hierarchies).

Although it is a natural assumption that Example-based methods work best with structured sets of bilingual texts, the experiments at IBM show that correspondence of units in source and target texts can also be established alone by statistical means. However, to what extent this extreme position is proved valid has yet to be demonstrated.

2.4.2 Statistical Method:

The idea of a statistical machine translation goes back as far as the creation of the first computers. However, it was quickly left aside because of the amount of computation resources needed to complete the process. In the late 1980s early 1990s, serious research was done at the IBM research center (Yorktown Heights, NY), using approaches previously developed for speech recognition (Bahl et al. 1983), lexicography (Sinclair 1985) and natural language processing (Baker 1979; Ferguson 1980; Garside et al. 1987; Sampson 1986; Sharman et al. 1988).

The approach is simple; assigning to every pair of sentences (*S*, *T*) a probability *Pr*(*T*|*S*), to be interpreted as the probability that a translator will produce the sentence *T* in the target language when presented with *S* in the source language. The expectation is to have very small probability for unrelated pairs of sentences and high probability for pairs of source-target translation. Then, given a sentence *T* in the target language, we seek the sentence *S* from which the translator produced *T*. Thus, we have to choose the sentence *S* that maximizes the probability *Pr*(*S*|*T*).

Using Bayes’ theorem, we can write:

$$Pr(S|T) = \frac{Pr(S)Pr(T|S)}{Pr(T)}$$

Because Pr(*T*) does not depend on *S*, the best sentence *S* will be the one that maximizes the product Pr(*S*)Pr(*T*|*S*).

Even if the theory looks simple, there are many difficulties to face. First, a bilingual parallel corpus has to be built and aligned, which was not very easy 10 years ago because of the lack of bilingual corpora. Second, it is difficult to have a good estimation of the several parameters for the different models.

IBM continued to work on the subject until 1995 when all funding were withdrawn. The project has been alleged of failure by people in the domain of MT, such as Yorick Wilks (Wilks 1993). Pure statistical method appeared inappropriate for machine translation. However, the statistical approach was not definitively put aside. In recent years, hybrid systems have appeared conciliating the symbolic and the statistic pragmatics.

2.5 Controlled Language

The last 10 years have shown a significant increase in development of controlled language systems. Several companies have understood the advantage to use controlled language for authoring purpose, such as Boeing (Wojcik et al. 1990). Before presenting the advantages that charmed professionals, we need to define what a Controlled Language is.

A controlled language is an explicitly defined restriction of a natural language that specifies constraints on lexicon, grammar and style (Nyberg et al. in process). Especially if authored sentences are used for automatic machine translation, the restriction on the lexicon is considered as necessary.

Among the lexicon restrictions, it is common to limit the allowable parts of speech to the minimum necessary for adequate expression in the domain. This is however not possible when the domain becomes more general. In order to the limit ambiguity, there is often a limitation on the number of meanings per word in a particular domain. An example would be to allow the term ‘car’ only when it carries the meaning of “railroad carriage” in the specific domain of mining industry. It is also frequent to limit the semantic domain model by restrictions on the possible fillers of semantic roles (Mitamura et al. 1991).

Beyond the lexicon control, grammar should be controlled as well to solve several ambiguity problems. It is important to reduce attachment ambiguities when using a MT system, which will prevent us from having multiple parses. The coordinated structures can be also restricted for the same reasons as mentioned above.

Although, it could be frustrated for an author to have such restrictions on his authoring skills, controlled languages have a large positive impact on editing. First of all, it provides a high level of consistency while authoring a document, even if several authors are involved in the process. Second, because of this consistency, it will be easier to translate the documents into other languages by a MT system.

3 Presentation of the KANT-KANTOO Project

3.1 History of the KANT Project

The KANT project has emerged in 1991 from extensions and refinements of an earlier system (KBMT-89) developed at the Center of Machine Translation (CMT) at Carnegie Mellon University, Pittsburgh (PA). KBMT-89 was a knowledge-based, interlingua-style machine translation system developed at CMT for translation of IBM PC installation manuals (English-Japanese). Previously to this system, a prototype has been developed in 1986, called Doctor-Patient, which was the first KBMT. It was designed to translate English into Japanese in the doctor-patient domain. Then, it was extended, in collaboration with the University of Stuttgart, in order to handle German as well.

The growing success of machine translation brought Caterpillar Inc. in 1991 to fund the development of a KANT (Knowledge-based Accurate Natural language Translation) application for their domain (e.g., heavy machinery, computer equipment, etc.). This version of the KANT system translates technical English, written in controlled language, into Spanish, French and German.

The first KANT application was deployed for the Union Electrica Fenosa in 1994. This application translates texts in the domain of power utility management, and has an English/Spanish vocabulary of about 10,000 words.

Since previous step of this large-scale KANT application development, several languages have been added to the list, including Portuguese, Italian, Russian and Chinese. A re-implementation of the whole system has been done recently towards an Object-Oriented architecture, where the appellation KANTOO (KANT Object-Oriented) comes from.

3.2 Overview of the KANTOO System

The KANTOO system is an interlingua-based translation system, containing several knowledge sources. Two distinctive steps are required to translate a sentence from a source language into a target language. The first step consists to produce an interlingua representation by analysis of the input sentence. The interlingua, which is the same for all target language, is a tree-like representation with syntactic and semantic information retrieved from the leaf nodes of the domain Hierarchy called DMK (Domain Model Kernel). The next step is a generation of the target text from this intermediate representation.

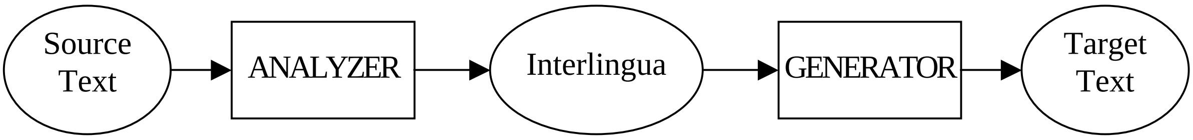

Figure 5: Interlingua-based Translation

3.2.1 Analyzer

The analyzer is a tool that takes a source text sentence as input, and brings an interlingua representation output for the sentence. Thanks to its useful feedback, the analyzer can also be used as a grammar checker, declaring any sentence as grammatical or ungrammatical. In order to come to a tree-like representation (interlingua) of a source sentence, the input string is processed through several modules. Each module adds a new level of abstraction over the text with semantic abstraction as the final level.

Several kinds of knowledge are also required in order to perform this analysis. The DMK (Domain Model Kernel) contains important knowledge about all concepts (see lexical analysis module). The DTD (Document Type Definition) defines a specific SGML markup language that was defined by Caterpillar Inc. and CMU. The Domo (Domain Model database) is used for disambiguation purpose. Finally, grammar rules are used for parsing purpose (see syntactic analysis module).

Source language sentences are processed through a succession of five modules in order to provide correct interlingua representations (IR). The sentence is first passed through the tokenizer module, which divides the sentence into individual words (tokens). Those are then passed to the lexical analysis module, which assigns definitions to words, numbers, and multi-word idioms. The syntactic analysis module receives these tokens with associated definitions, and combines them to form one or more tree-like structures, called Feature Structures (F-Structures). Next, the disambiguation module prunes ambiguous F-Structures by using heuristics or human manual disambiguation. Finally, an interpreter module completes the analysis by mapping each F-Structure slots into an interlingua structure.

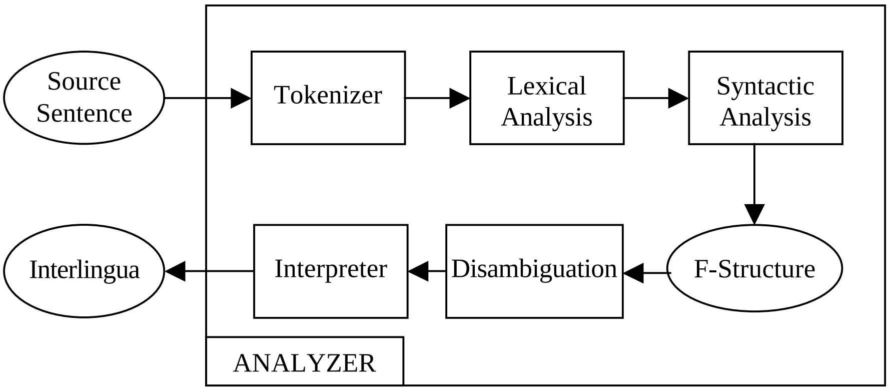

Figure 6: Analyzer module

Tokenizer module:

The Tokenizer is a small module using its own built-in grammar to parse source text sentences in order to output a sequence of token. It has to deal with words, numbers, punctuation and tags.

Lexical analysis module:

The lexical analysis module takes a list of tokens as input and generates a sequence of frames, which contain the definition for one token or sub list of tokens. In the case of ambiguous sentences, the frames (hence definitions) may overlap.

A morphological analysis is also performed to yield morphemes. They are used to extract the definitions from the DMK. The output frames contain therefore morphological information, such as gender, number, tense, etc.

Syntactic analysis module:

From a set of meanings, the syntactic analysis module outputs a tree-like syntactic structure. The Tomita parser (Tomita 1986), parses the lexical analysis module output using a grammar rule database in order to generate one or more parse trees. The Tomita Parser is an extension of the basic deterministic LR-parsing algorithm to handle non-deterministic languages.

Disambiguation module:

The bottom line of this module is to output an unambiguous interlingua form from the F-Structure produced by the syntactic analysis module. This module is designed to handle several types of ambiguity:

- Lexical ambiguity: This type of ambiguity occurs in the case of multiple possible concepts for one morpheme. This is common in the case of multiple meanings for a term. For example, the noun *bank* has at least two meanings, *bank of a river* and *bank as a financial establishment*.
- Structural ambiguity: This type happens when two or more syntactic structures are possible to generate from the same set of meanings. The problem here could be an adverb attachment with a sentence containing two verbs, for example.
- Part-of-Speech ambiguity: When the part of speech of a word cannot be determined by parsing, a categorical ambiguity is present. An illustration of this ambiguity can be found in the phrase: *liquid flows*, where *flow* can be a plural noun or a verb.
- Anamorphic ambiguity: This occurs when a pronoun can refer to more than one preceding noun.

Along the disambiguation process, the Domo provides information, which are used for heuristic disambiguation.

Interpreter module:

The interpreter module is a very simple module, which applies a set of mapping rules in order to convert a F-Structure representation into an interlingua representation. Rules are designed to turn each frame of F-Structures into English independent forms of knowledge (see section 3.2.2).

The analysis phase is very important in a machine translation process. A small error in the analysis of a sentence can generate a complete incorrect translation. The disambiguation step is of primary importance, because it clarifies the sense of the sentence. On previous KANT systems, most of the disambiguation was done by interactively questioning the author.

Nowadays, less and less questions are asked to authors, the analyzer uses heuristics in order to auto-disambiguate the sentences.

3.2.2 Interlingua:

Up to the present, several kinds of interlingua have been used in machine translation systems employing this approach. These interlinguas have a common point: they try to express the meaning of a sentence using a symbolic representation, where the relations between the symbols (concepts) are displayed.

The Interlingua Representation (IR) exhibits the source text as a sequence of frames with "codes" that indicate semantic, tense, aspect, case, and morphology, along with the syntactic relationships and punctuation of each sentences. Interlingua is not English, Chinese, German or Hindi: it is a special language designed to represent abstract concepts and relationships common to all natural languages.

Open the door.

```
(*A-OPEN-1
   (argument-class agent+theme)
   (mood imperative)
   (punctuation period)
   (tense present)
   (theme
      (*O-DOOR
         (number singular)
         (reference definite))))
```

Ouvrir la porte.

Figure 7: Interlingua for "*open the door*."

The KANT interlingua is sentential; that means it is designed for a sentence-by-sentence source text processing. Each interlingua is essentially a case frame, which is composed of a head concept, features and semantic roles. The head of the syntactic constituents is usually a concept (e.g., **A-OPEN*, **O-DOOR*, etc.) followed by zero or more feature-value pairs or semantic roles. The fundamental meanings of an utterance, such as grammatical information, are usually represented by features containing atomic values (e.g., tense, mood, form, etc.). Semantic role slots contain embedded interlingua expressions headed by the concept associated with the head of a syntactic constituent (e.g., theme, agent, q-modifier, etc.).

Each concept has a suffix that describes its part of speech, for example **A-* stands for action, and therefore for verbs. This information helps to classify them into the lexicon, and reduces the time needed for updates.

The domain model contains for each verb a set of possible argument-class. This feature is very useful for the translation, because it predicts the structure used by the verb (Mitamura 1989).

3.2.3 Generator

The Generator is composed by a sequence of three modules, which takes an interlingua representation as input, and outputs a target language text sentence. The generation process is on many parts similar to the analysis process, except for the order of the modules.

First, the interlingua is mapped into a F-Structure. In order to perform this conversion, three sources of knowledge are employed (see mapper module). Next, a grammar-based module breaks down the F-structure into a set of frames. At this level, the word order is already determined. Then, the morphology (agreement, verb inflection, etc.) can be applied by using a set of morphological rules.

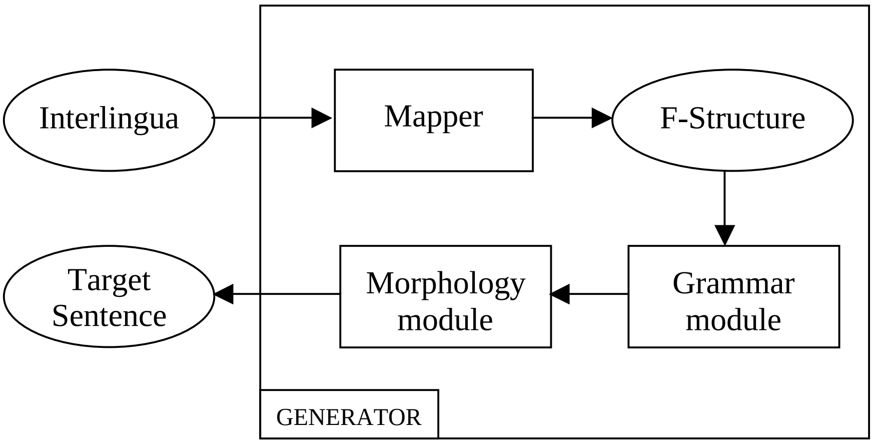

Figure 8: Generator module

<u>Mapper Module</u>:

The Mapper is the most knowledge-intensive module, including lexical translation, semantic and syntactic databases, but also mapping and lexical selection rules. Each database needs to be updated according to the target language.

Two kinds of knowledge can be differentiated. The *passive knowledge* can be seen as databases with no direct action on the interlingua mapping. The *active knowledge* builds piece by piece the F-Structure by consuming little by little the interlingua.

Passive Knowledge:

- Lexical Nodes: Database containing translations for all the concepts. It has to be updated regularly in accordance to the customer requirement.

- Semantic Tree: Database containing semantic information about parents of concepts. A concept can have 0, 1 or more parents. For example the concept **O-WATER* has two parents: *SPREADABLE-SUBSTANCE* and *LIQUID-GAS*. This database is useful when lexical selection rules are written (see Lexical Selection Rules).

- Syntactic Lexicon: Database containing the syntactic representation of each translation in a F-Structure-like format. This database contains also some useful information like the positioning of an adjective according to a noun (e.g. "*tuyau cylindrique*", "*long tuyau*") and invariability of some words (e.g. "*portes avant*").

Active Knowledge:

In order to write selection rules and mapping rules in an easy way, a pseudo-interpreted code has been developed internally to CMU. Called PATRICK (PAThname Resolution Interpreter Code for KANTOO), it relies on a set of predefined functions used in order to perform tests, to map slots and to navigate through interlinguas.

- Lexical Selection Rules: Used for disambiguation or re-phrasal purpose, they are manually developed in order to provide correct translations and correct structures for a given concept. An example of use of lexical selection rule for re-phrasal purpose:

  Eng: "Check the pipe for leakage."
  Fre: "Vérifier s'il y a une fuite dans le tuyau."

  In the case of multiple meanings, a lexical selection rule can be written to take into account the context of a word.

  Eng: "*Turn off* the power supply."
  Fre: "***Couper*** l'alimentation."

  and

  Eng: "*Turn off* the light."
  Fre: "***Eteindre*** la lumière."

  The previous example shows usage of a lexical selection rule with the verb-concept **A-TURN-OFF*. The lexical selection rule will generate a different translation for the verb *to turn off* according to its context.

- *Mapping Rules*: Heart of the Mapper module, the mapping rules are written in order to map every slot from an interlingua into the corresponding target language F-Structure. For each part-of-speech, a set of mapping rules is associated, which are aimed to map every possible slot of an IR. Mapping rules are intended to not evolve often, only in the case of modification in the interlingua format or in the case of new requirements expressed by the customer (e.g., request to change passive voice into active voice for a specific verb).

Grammar Module:

At the opposite of the parser, the grammar module takes a F-Structure form and decomposes it into a sentential frame representation. The grammar has to handle not only text and number, but SGML tags as well. SGML tags should have a very specific order in each target language, which is usually different from the order in English.

The output frames contain information about spacing between words, parts of speech and agreement for noun, verb, adjectives, etc.

Morphology module:

The morphology module applies morphological rules to each frame of the sequence composing the sentence. A sequence of tokens is then output, morphologically modified (e.g., "*ouvrir*" at the 3rd person of the indicative present becomes "*ouvre*"). Special morphologies, such as irregular verbs, have to be handled separately.

The sequence of tokens is finally processed by a small module, which joins the tokens together and takes care of things like elision and word spacing.

3.3 Other Developed Tools:

In addition to the analyzer and the generator, several other tools have been implemented for knowledge maintenance purpose:

- Knowledge Maintenance Tool (KMT) is a graphical user interface under Java language, which allows real-time browsing, editing, and incremental update of the knowledge sources used during analysis and generation (lexicon, grammar, domain model, lexical selection rules, mapping rules, etc.)
- Lexicon Maintenance Tool (LMT) is a PC-based Oracle database and forms application for rapid development and efficient maintenance of source language vocabulary (Caterpillar Technical English terminology)
- Language Translation Database (LTD) is an Oracle Forms interface for rapid update of target language technical terminology, by developers and end-users. The use of RDBMS technology supports efficient maintenance of large-scale terminology for commercial applications.

Caterpillar currently uses those tools in order to update the knowledge for further release of the KANTOO system.

4 Towards an Improvement in Quality of French Generation

4.1 From KANT to KANTOO, Story of a Porting

Since its beginning, the KANT system has been developed under Lisp code. The reason for this choice was of several orders. At the time of the first encoding, lisp was still widely used at universities. It was also appropriate for handling frames and tree-like structures. However, new imperatives appeared during the last years that carry new goals for the system to meet:

- Lowering cost and time for terminology maintenance (better database management tools)
- Lowering cost and time for system knowledge updates (troubleshooting tools, modular design)
- Improving the general robustness and maintainability (porting Lisp to C++)
- Improving the portability (to different platforms including Microsoft Windows, Unix...)

A complete module re-implementation has been done according to a more modular design. Each module can be run independently from the other, that allows better traceability and debugging. For the knowledge porting, Perl scripts have been developed in order to convert the Lisp-like knowledge representation into the PATRICK-like representation. However, because of the differences in how the KANTOO (KANT Object-Oriented) system handles interlingua forms versus the KANT system, some manual work had to be done. Furthermore, callout functions, which were implemented in Lisp, had to be manually converted into PATRICK code.

The Spanish system has been the first to be ported to the PATRICK code; however, all the knowledge maintenance was still done under Lisp-like format until the first release of the Spanish KANTOO system. Scripts were used in order to translate all knowledge into the new format at the time of the system release. The first Spanish MT system under C++ technology has been released in June 1999. Since its release, the Spanish KANTOO system has demonstrated a higher translation quality than previous systems.

At the opposite, German and French MT system have been ported first to PATRICK code and then were maintained and updated. Because new target language leaders were not familiarized with either Lisp or PATRICK knowledge representation, it was better to convert the data first and then to update them in order to spare the training period.

4.1.1 Problems Encountered during the Porting:

Even if the PATRICK language has many similarities with Lisp (slot handling, interpreted code, etc.), it has some differences that required changes in the knowledge rules structure. The major variation was the absence of functions like *car* and *cdr* in PATRICK language, this prevents from branching in an interlingua or a F-Structure tree without knowing the name of the child leaf. For this reason, the nominalization function had to be redesigned because it was designed to navigate through the complete F-Structure tree to nominalize (change *gerund* into *noun*, see section 4.2.1) all it can.

Although the PATRICK language does not implement basic Lisp functions, it works at a higher level, which provides more efficient code representation and faster access through tree-like structures.

Some bugs were found in the porting scripts while porting French MT system. The problems occur because the scripts were designed with according to Spanish knowledge. Unfortunately, French knowledge had some none conventional mapping rules that have not been updated through time, when Spanish knowledge has been regularly updated.

4.1.2 State of French Generation Module in March 99

The French generation has been one of the first MT system released by the KANT project. Several technical leaders contributed to its development (D. Lonsdale 94-95, R. Chadel 95-97). The French MT system was accepted for the first time by the translation department at Caterpillar in December 1996, that means translated outputs were good enough to use the system in production.

Two years have passed since last French technical leader has worked on the system and little documentation was present. Although the level of the French output was good, many truncations remained present, due to erroneous mapping rules, bad terminology or grammar failures.

4.2 Problems Encountered in French Generation

Although a lot of English vocabulary comes from French, English is closer to German as for its sentence structures. For this reason, machine translation from English into French requires some heavy development in order to produce an acceptable level of translation. In the next section, some standard issues in English-French translation are presented.

4.2.1 Gerunds:

Unfortunately, the *-ing* gerund form in English does not always correspond to the French *-ant* form. However, several patterns of translation can be identified between the two languages. As an example, in most cases a gerund will be translated as an infinitive in French behind a preposition:

Eng: "Reinstall four bolts without ***using*** any washers."

Fre: "Remonter quatre vis sans ***utiliser*** de rondelles."

The English gerund can be translated in various ways such as using a subordinate clause or a noun phrase. This can increase the complexity on the translation process.

Eng: "***Measuring*** the amount of drift will determine if there is a need to check the travel brake."
Fre: "***La mesure*** de la quantité d'affaissement déterminera s'il y a un besoin de contrôler le frein de translation."

In the previous example, a noun would be preferred as translation for the gerund *measuring*.

4.2.2 Stative vs. Passive:

Especially within technical documents, the passive voice is widely used in English, while the French language uses more often active constructions. However, excessive use of passive voice in French is not critical and does not have an influence on comprehension of a text. More of a concern, is the ambiguity of English sentences between stative and passive constructions, which can result in a misleading translation:

Stative: "The window was ***broken*** and the rain could get in."
Passive: "The window was ***broken*** by the driver."

The first sample sentence illustrates a stative construction where "broken" expresses a state. The second presents a passive voice that can be changed into active voice:

Active: "The driver ***broke*** the window."

There would be no problem if the French language would keep the same ambiguity as English, but it is not the case.

Stative: "La fenêtre était ***brisée*** et la pluie pouvait rentrer."
Active: "La fenêtre a été ***brisée*** par le conducteur."

Even if it is easy to differentiate both constructions in this example, it is not always the case. This problem increases the complexity of analysis and requires extra information (more empirical), not included in the sentence, in order to differentiate between both structures.

4.2.3 Determiners (and Partitive):

If physically present in the sentence, English determiners can easily be translated into French. However, they are more difficult to generate when they are implied in the source language. For example:

Eng: "Power goes from the torque converter to the transfer gears."
Fre: "***La*** puissance est transmise du convertisseur de couple aux engrenages de

transfert."

Some translations can even require partitive structures:

Eng: "Leakage of the crankshaft seal can occur."
Fre: "***Des*** fuites risquent de se produire au niveau du joint de vilebrequin."

The problem with such a structure is that the English sentence does not contain the information needed for the generation of a determiner. We have to look at a more semantic level in order to extract the necessity information.

4.2.4 Prepositions:

Another typical problem of English-French machine translation is the translation of prepositions. Locative prepositions are a classical example of this problem (Japkowicz and Wiebe 1991):

Eng: "The man gets ***on*** the bus."
Fre: "L'homme monte ***dans*** le bus."

Eng: "The man gets ***on*** the table."
Fre: "L'homme monte ***sur*** la table."

This example shows how locative perception could be different. For a given preposition *on* in English, we can have two different translations in French. This demonstrates how much the context is important.

4.2.5 Other Issues:

Many other issues can be found to show the problems that encounter teams in the field while building machine translation systems. Those could be syntactic, semantic or even stylistic problems. To illustrate that last point, let us consider the following example:

Eng: "The truck ***is*** 3.5 m ***wide***."
Fre: "Le camion ***a une largeur*** de 3,5 m."

When in English an adjective is used as measurement attribute, a noun is preferred in French. It would not be incorrect to use the same structure in the target language as in the source language, but it is stylistically better to use the structure in the translation shown above.

4.3 Improving French Output:

Besides the porting, several modifications have been carried over the French knowledge in order to improve the accuracy of the translation process. Most of this work has been done on mapping and lexical selection rules, grammar generation and data representation. Other knowledge, as the one used for morphology purpose, was quiet stable and reliable, and did not require major updates.

4.3.1 Problem Detection:

A French output review cycle process has been withheld in order to extract existing problems. The first step consisted in running a set of sentences through the French MT system and sending the output to Caterpillar for review. Technical translators at Caterpillar reviewed the output and extracted a list of problems that needed to be fixed. Next, technical leaders at Caterpillar and CMT were having phone-calls in order to decide the actions to take about issues that should be fixed and those that were not worth to be fixed. Then, updates were made and the new outputs were tested before running the next set of sentences.

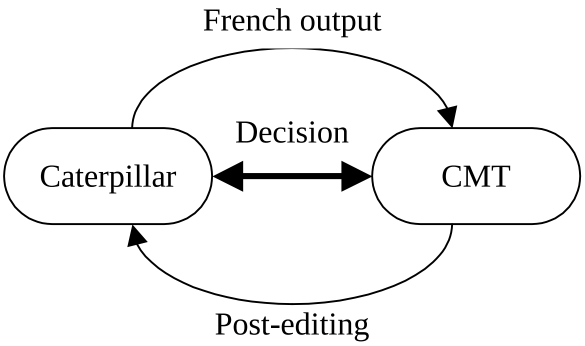

Figure 9: Problem reports

This kind of process has two main advantages:

- Excellent dialog with the customer: allows the customer to see improvements of the output along the process,
- Direct feedback from system end-users: translations are closer to what Human translators would expect.

4.3.2 Lexical Selection Rules:

Lexical selection rules are used for two purposes: the disambiguation of a term by looking at its context and the structure modification in order to generate a “non-conventional” structure. In this section, we will present both cases.

4.3.2.1 Lexical selection rule for ***A-CONNECT**:

The verb *to connect* can be translated in different ways depending on its context.

According to the "Dictionnaire technique général[1]", eight different translations are possible for this single verb. In this case, only three of them were retained which are ***accrocher***, ***raccorder*** and ***brancher***. The other translations were put aside because they were synonyms or because they were not allowed in the Caterpillar domain. The following piece of PATRICK code shows how the lexical selection rules work:

```
(node "?a-connect"
        :parent "?verb"
        :rule (
              (*TRY* ((*TEST* #test-concept ((concept %(ir theme CONCEPT))
                                             (choices (*OR*
                                                        *O-MACHINE
  1)                                                    *O-TRAILING-EQUIPMENT
                                                        *O-TRAILING-UNIT
                                                        *O-TRAILER-UNIT))))
                         (%() <= #lex-once ((env %()) (lex "accrocher")))))

              (*TRY* ((*TEST* #test-concept ((concept %(ir theme CONCEPT))
                                             (choices (*OR*
  2)                                                    *O-PIPE
                                                        *O-TURBINE
                                                        *O-HUB))))
                         (%() <= #lex-once ((env %()) (lex "raccorder")))))

  3)          (*TRY* ((%() <= #lex-once ((env %()) (lex "brancher")))))))
```

Figure 10: Lexical selection rule for **A-CONNECT*

The first remark is that the *theme* or *patient* slot (direct object) is usually the point of attention while trying to disambiguate a verb. Furthermore, the direct object determines through its characteristics the kind of actions (hence the verb) that can be applied on it. For example, we can *bend* or *cut* a pipe, but we cannot easily *turn on* a pipe.

Lexical selection rules are made from a sequence of **TRY** statements. Each **TRY** statement will be evaluated. If it fails, the environment will be restored as it was before the evaluation started, otherwise the rule will work and modification will be made on the F-Structure. In the case of the verb *to connect*, it has three possible translations, distributed in three **TRY** statements. 1) and 2) are composed of a test statement followed by a function that extracts syntactic features for a translation. In both cases, if the test fails, the **TRY** statement will fail. 3) is the default translation. If either 1) nor 2) was successful, the default translation will be selected. By looking at the *choices* slot in the test statements, we can determine:

- ***accrocher*** is employed with *machine, trailing equipment, trailing unit* and *trailer unit*,

Eng: "Connect the *machine* to the trailing equipment."
Fre: "***Accrocher*** la machine à l'équipement tracté."

[1] Dictionnaire technique général, anglais-français - J.Gérard BELLE-ISLE, Éditions Beauchemin, 1977.

- ***raccorder*** is employed with *pipe*, *turbine* and *hub*,

Eng: "Connect the *hoses* to the machine."
Fre: "***Raccorder*** les flexibles à la machine."

One can wonder why ***raccorder*** has been selected as translation because the object is *the hoses* which is not included in the choices list. This has to do with the semantic tree, which tells us that a *hose* is a kind of *PIPE*, and *PIPE* is in the list.

Lexical selection rules for disambiguation of verbs are the most frequent situations. It is less common to write a rule for nouns or adjectives disambiguation.

4.3.2.2 Lexical selection rule for **A-CHECK*:

The verb *to check* is a good example to illustrate a more complex lexical selection rule that implies a different structure than in English in the generated F-Structure. Among the different meanings of this verb, only one is used in Caterpillar Technical English (CTE) (Kamprath et al. 98):

> **check**^{2} *v* 1 [I (**for**, **on**, UP); T] to test, examine, or mark to see if something is correct, true, in good condition, etc.[2]

It can be used intransitively or transitively, and with or without complement.
The following PATRICK code sample shows the **TRY** statement that handles usage of *to check* with a complement introduced by the conjunction *that*:

```
(*TRY* ((%(ir argument-class) =c agent+complement)
          (*TEST* #test-concept ((concept %(ir complement extent CONCEPT))
                                  (choices *CONJ-THAT)))
          (%() <= #lex-once ((env %()) (lex "s'assurer")))))
```

Figure 11: Part of the rule for verb **A-CHECK*

If the argument-class is agent+complement and the conjunction is **CONJ-THAT*, then the translation of *check* will be ***s'assurer***:

Eng: "*Check* that the door of the truck is closed."
Fre: "***S'assurer*** que la porte du camion est fermée."

Several other cases have to be considered in order to translate in an accurate way every possible sentence using the verb *to check*. Although the three next sentences share exactly the same meaning (according to the action), it is not possible to have the same translation in all of them.

Intransitive verb + *for* + adverbial complement:
Eng: "The truck driver is checking ***for*** holes."

Dictionary of contemporary ENGLISH, Longman Group UK Limited 1987 (Second Edition).

Fre: "Le conducteur vérifie ***s'il y a*** des trous."

Transitive verb + direct object:

Eng: "The truck driver ***is checking*** the fuel tank."
Fre: "Le conducteur **contrôle** le réservoir de carburant."

Transitive verb + object + *for* + adverbial complement:

Eng: "The truck driver is checking the fuel tank ***for*** holes."
Fre: "Le conducteur vérifie ***s'il y a*** des trous ***dans*** le réservoir de carburant."

The verb ***vérifier*** has been preferred instead of ***contrôler*** when an adverbial complement introduced by the preposition *for* is present. In addition, the structure of the French sentence is different from the structure of the English sentence while using ***vérifier***. The rule is rephrasing the sentence as if it was:

Rephrased Eng: "The truck driver is verifying if there are holes."
Rephrased Eng: "The truck driver is verifying if there are holes in the fuel tank."

This would sound strange to say in English, but it is correct in French.

In order to fulfill this task, the following selection rules had to be written:

```
(*TRY* ((*TEST* #test-concept ((concept %(ir q-modifier CONCEPT))
                                          (choices *Q-SOUGHT_FOR)))
```

Extraction and Mapping of adverbial complement

```
IR:    (q-modifier
            (*Q-sought_FOR
                    (case
                        (*K-FOR))
                    (object ...
```

with for as preposition of the complement clause

```
FS:    (comp-clause (
                (obj (
                      (agr ((gender m) (number pl) ( person 3)))
                      (cat noun)
                      (det ((root "un")))
                      (root "trou")
                      (tgtlex-class nom-m)))
```

Extraction and Mapping of other complements (if necessary)

Mapping of patient as prepositional phrase with preposition ***dans***

```
(%fs = ((comp-clause ((subord-conj ((root "si")))
                           (mood indicative) (tense present) (voice active)
                           (verb-attachment +) (cat verb)
                           (root "avoir")
                           (modifier ((root "y") (midadv +) (reftype pro)))
                           (impers +)))))

(%() <= #lex-once ((env %()) (lex "vérifier")))))
```

Figure 12: Lexical selection rule for verb **A-CHECK*

This rule is designed to handle sentences including the preposition *for* (*Q-sought_FOR) attached to the verb *to check*. The corresponding interlingua for the previous sentence is:

The truck driver is checking the fuel tank for holes.

```
(*A-CHECK
    (agent
        (*O-TRUCK-DRIVER
            (number
              (:OR mass singular))
            (reference definite)))
    (argument-class agent+patient)
    (mood declarative)
    (patient
        (*O-FUEL-TANK
            (number singular)
            (reference definite)))
    (progressive +)
    (punctuation period)
    (q-modifier
        (*Q-sought_FOR
            (case
              (*K-FOR))
            (object
              (*O-HOLE
                        (number plural)
                        (reference no-reference)))
            (role sought)))
    (tense present))
```

Figure 13: Interlingua representation with verb **A-CHECK*

While mapping the verb *check*, the previous rule will be "tried". First, it will be tested if the semantic role concept **Q-sought_FOR* is present. Considering that this is true, the object of the adverbial complement is mapped to the object slot of a complement clause. Next, the *patient* (direct object **O-FUEL-TANK*) is mapped as a prepositional phrase with ***dans*** as a preposition. Then, additional features are added to the complement clause in order to produce ***s'il y a***, and finally, verb features for ***vérifier*** are retrieved from the syntactic lexicon.

After the mapping is completed and the F-Structure is generated (see Figure 14), the grammar module has to generate a correct frame order. Information, like (verb-attachment +), are useful in order to know where the clause should be positioned in regard to the verb. In the present example, the subordinate clause has to go immediately behind the verb that means in front of the prepositional phrase with ***dans***.

```
(
        (subj (
            (cat noun) (agr ((gender m) (number sg) (person 3)))
            (det ((root "le")))
            (root "conducteur")
            (tgtlex-class nom-m)))
        (cat verb) (mood indicative) (tense present) (voice active)
        (root "vérifier")
        (comp-clause (
            (subord-conj ((root "si")))
            (imper +)
            (modifier ((midadv +) (reftype pro) (root "y")))
            (cat verb) (mood indicative) (tense present) (voice active)
            (root "avoir")
            (verb-attachment +)
            (obj (
                (cat noun) (agr ((gender m) (number pl) (person 3)))
                (det ((root "un")))
                (root "trou")
                (tgtlex-class nom-m)))))
        (pp (
            (prep ((root "dans")))
            (p-obj (
                (cat noun) (agr ((gender m) (number sg) (person 3)))
                (det ((root "le")))
                (root "réservoir")
                (syn-pp ((root "de carburant")))
                (tgtlex-class phr)))))
        (punctuation ((root "period"))))
```

Le conducteur vérifie s'il y a des trous dans le réservoir de carburant.

Figure 14: F-Structure generated from interlingua in *figure 13*

The development of the lexical selection rule for the verb **A-CHECK* has significantly improved the French AMT output, because this structure is extensively used in a technical domain like Caterpillar. Besides the rule for the verb **A-CHECK*, several other lexical selection rules were developed for special verbs, like for the verb **A-CAUSE*, to provide a better translation.

4.3.2.3 Conclusion on Lexical selection rules:

Lexical selection rules are mostly used for verb disambiguation purposes, especially when an English verb is translated as a phrasal verb in French. Such cases are frequent, because some English terms do not have their equivalence in French, as it is the case for the verb *to face*, which is translated as ***faire face à***.

We also have to write lexical selection rules when a verb needs a causative translation for a transitive usage. For example, the verb *to circulate* will be translated as ***circuler*** when intransitively employed, and ***faire circuler*** when transitive employed. In fact, manual disambiguation allows reducing the number of lexical selection rules needed by selecting the concept with the appropriate sense for a context.

4.3.3 Mapping Rules:

The KANT mapping rules have been under development for years in order to match the complexity of the French generation. After all the efforts, a quiet stable system has been developed, achieving high quality translation. Since the porting from Lisp-like representation into PATRICK-like representation, several generation problems have been fixed, always in accordance with customer requirements. As a matter of fact, the customer requirements about translations can be really different from what one can expect. A good example to illustrate this aspect is the requirement expressed by the Caterpillar translation department to translate the modality *should* as an obligation *must*.

Eng: "You ***should*** stop the truck."
Fre: "Vous ***devez*** arrêter le camion."

All those requirements have to be considered while developing mapping rules.

Although some structures are straightforward to translate, some others require much more work. As presented from examples in section 4.2.1, gerunds are very difficult to deal with for machine translation developers. Whatever technique used (e.g. transfer, interlingua...), no system achieved a high quality translation for gerunds.

In this section, we will present the approach used in order to solve this problem in the KANT system.

Gerund Mapping:

Remis Chadel developed the first mapping rule dedicated to gerund translation in 1996. It was designed as a top-level grammar generation function, which was "nominalizing" whenever necessary. The "nominalization" is the process used to transform a verb (in its gerund form) into a noun. During the porting of the system, this rule had to be moved within the mapping module because of some differences between the Lisp code and the PATRICK code (see section 4.1.1). Since then, it works as a top-level rule in the mapping tree.

In order to illustrate this problem, we can look how well the MT system, called Systran (by Systran Corp.), performs on three different sentences:

1) Eng: "This will prevent the towed machine from ***rolling***."
Systran: "Ceci empêchera la machine remorquée du ***roulement***."

2) Eng: "***Removing*** the rear tires is not necessary."
Systran: "***Retirer*** les pneus arrière n'est pas nécessaire."

3) Eng: "After ***tightening*** the bolt to the correct torque, install the plastic cap over the bolt."
Systran: "Après ***serrage*** du boulon au couple correct, installez le chapeau en plastique au-dessus du boulon."

The sentence 1) contains a gerund behind a preposition. Systran system incorrectly analyzes this verb as a noun. In the sentence 2), the gerund is used as a verbal noun, because, as noun, it is the subject of the verb *to be*, and, as verb, it is the verb of the direct object *the rear tires*. The strategy of Systran is to translate this gerund into a verb with the infinitive form. Even if the result looks pretty close to the English, we can argue that it would sound strange in French to say something like that in a technical document. As for the sentence 3), the gerund is preceded by the temporal preposition *after*. The translation proposed by Systran system is a noun, which would be better with a determiner, but is also acceptable without.

It is interesting to see that even a MT system such as Systran is not able to overcome the problem of gerund translation several years of experience in the field.

Before presenting the results obtained by the KANT system, we are going to describe how the system deals with gerunds.

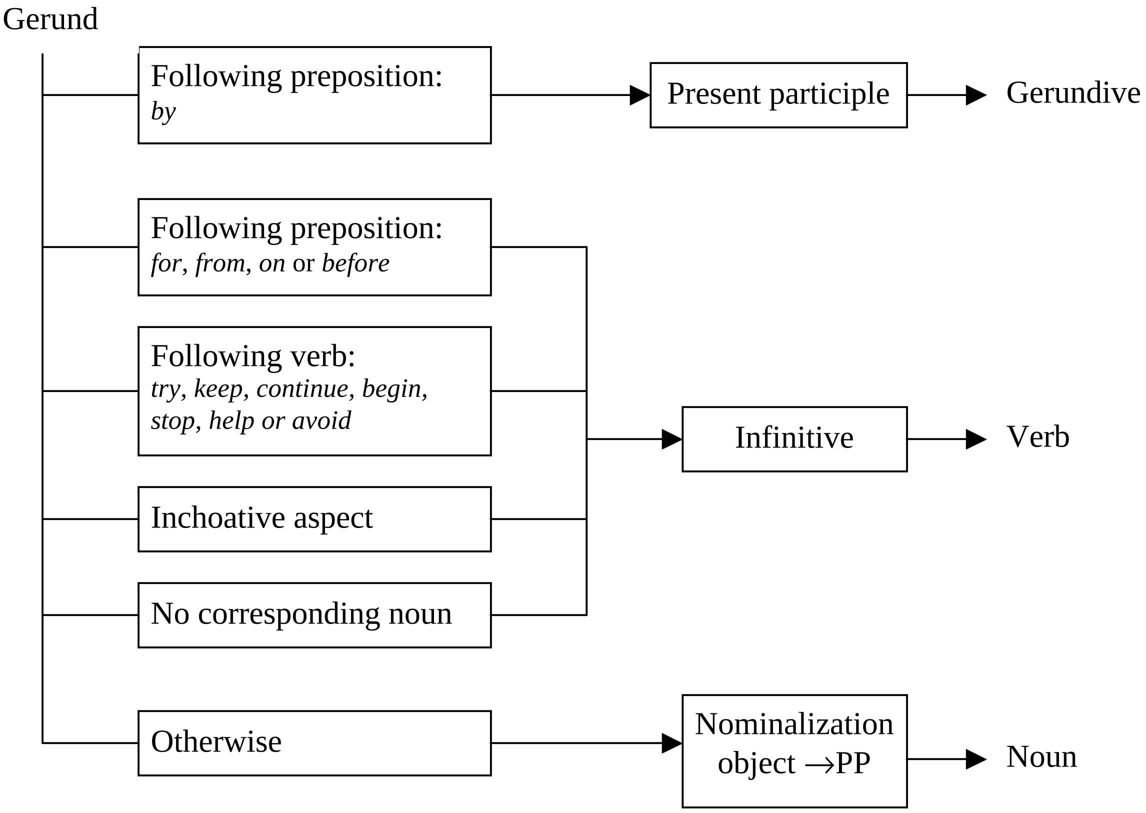
Gerund
Following preposition: *by*
Present participle
Gerundive
Following preposition: *for, from, on* or *before*
Following verb: *try, keep, continue, begin, stop, help or avoid*
Infinitive
Verb
Inchoative aspect
No corresponding noun
Otherwise
Nominalization object →PP
Noun

Figure 15: Nominalization algorithm

In order to obtain a correct gerund translation, the process relies on a good analysis of the source language sentence. The parser should be good enough to distinguish between nouns (e.g. "the blocking") and gerunds (e.g. "blocking the lever...") to prevent misrepresentations. The first step of the process is to identify gerundives and translate them as the present participle of the verb. This is simply done by looking if the gerund is preceded by the preposition *by*. In many cases, French language requires an infinitive verb. When the gerund follows a preposition such as *for*, *from*, *on* or *before* or a verb such as *try*, *keep*, *continue*, *begin*, *stop*, *help* or *avoid*, the infinitive verb form will be preferred. The list of verbs given above is an exhaustive list of verbs, which can govern another verb (e.g. "keep driving the truck"). The infinitive will also be chosen if the verb has an inchoative aspect (a grammatical aspect, by which the beginning of an action is specified, e.g. “to be about to do something”) or if there is no corresponding noun for a specific French verb (e.g. ***creuser***).

In other cases, the verb will be “nominalized”. The object of this verb, if any, will be transformed as a prepositional phrasal introduced by the preposition ***de*** (e.g.

“removing the tires” becomes “the removal of the tires”).

In order to nominalize, a corresponding noun (and features like gender and number) had to be provided for each verb in the French lexicon, whenever possible.

Eng: "This will prevent the towed machine from ***rolling***."
KANT: "Cela empêchera la machine remorquée de ***rouler***."

Eng: "***Removing*** the rear tires is not necessary."
KANT: "***Le retrait*** des pneus arrière n’est pas nécessaire."

Eng: "After ***tightening*** the bolt to the correct torque, install the plastic cap over the bolt."
KANT: "Après ***le serrage de*** la vis au couple correct, monter le couvercle plastique au-dessus de la vis."

Currently, the translation quality of gerunds into French by the KANT system has reached a satisfying level as shown by the translations above. However, some structures continue to cause troubles, and particularly when an adverb is modifying a gerund:

Eng: "After ***quickly*** ***tightening*** the bolt to the correct torque, install the plastic cap over the bolt."

The presence of the adverb *quickly* prevents from nominalizing the verb *to tight* shown in the previous example. The reason is that adverbs cannot modify a noun. It is also not possible to translate a gerund as a verb in the infinitive because of the preposition *after* which does not accept an infinitive verb. Two solutions could be implemented in order to solve the problem:
The first solution is the introduction of the ***avoir*** auxiliary:

Fre: “Après ***avoir*** ***rapidement*** ***serré*** la vis au couple correct, monter le couvercle plastique au-dessus de la vis."

The other solution is to nominalize the verb and to use the adjective corresponding to the adverb (e.g. “rapidement” becomes “rapide”):

Fre: "Après ***le serrage rapide*** de la vis au couple correct, monter le couvercle plastique au-dessus de la vis."

However, It is simpler to implement the first solution because it does not require supplying the corresponding adjectives for each adverb in the lexicon.

4.3.4 Syntactic Lexicon Representation:

The syntactic lexicon includes the internal representation for each translation from the lexicon. It is not directly linked to a concept, but to a translation string. Entries could be words such as nouns, adjectives or adverbs, or phrases such as prepositional phrasal, noun phrasal, adjective phrasal, etc…

Single words are not especially difficult to handle, although we have to take a special care of multiple possible part-of-speech for a word (e.g. ***large*** is an adjective and a noun). Most of the efforts have to be focused on phrasal entries in order to provide a suitable representation. They come most of the time from the translation of phrasal noun (e.g. *accelerator control lever*), but also other parts of speech, even single words (e.g. *forward* translate is translated as ***vers l'avant***).

Phrasal translations could be of several forms:

noun + adjective + prepositional phrase:
**O-ACCIDENTAL-TRANSMISSION-ENGAGEMENT* = "engagement accidentel de la boîte de vitesses"

noun + relative clause:
**O-ACCEPTABLE-RING-GROOVE-TOOL* = "outil qui convient aux gorges de segment"

noun + reduced relative clause:
**O-ENGINE-DRIVEN-ACCESSORY* = "accessoire entraîné par le moteur"

The first approach to this problem has been to provide a full representation for the translation. A deep analysis of the text string was be performed in order to extract all the information. The major disadvantage was the size of the obtained representation; hence increasing the size of the F-Structure representation and the processing time. This method was correct, but too heavy to handle. For this reason, we decided to reduce the size of the syntactic lexicon by using a pseudo representation instead of a full representation.

The idea was simple: "we don't need to provide a complete representation for parts that cannot get morphologically inflected ". Prepositional phrases are a good example of sub-string that does not need any inflection.

The translation for the concept **O-ACCIDENTAL-TRANSMISSION-ENGAGEMENT* with a plural translation would be:

Translation: "engagements accidentels de la boîte de vitesses"

The sub-string "*de la boîte de vitesses*" is invariable in this context and can be interpreted as a single unit.

Old representation:

```
[       [NOUN engagement
        [ADJ accidentel]
        [PP [   [PREP de]
                [OBJ [  [DET la]
                        [NOUN boîte]
                        [PP [   [PREP de]
                                [OBJ [  [NOUN vitesses]]]]]]]]]
```

New representation:

```
[       [NOUN engagement
        [ADJ accidentel]
        [PSEUDO-PP de la boîte de vitesses]]
```

Figure 16: Difference between old and new syntactic lexicon representation

This method allows cutting off drastically the size of the representation tree as we can see in figure 16. It is even more effective that French language uses widely prepositional phrases as translation for English phrasal nouns. Another advantage is that the level of analysis required for the translations is highly reduced.

4.4 KANT System Evaluation

The results presented in this section are coming from the regression test corpus used at CMT for regression test purposes. The French target language leader has performed the scoring in order to identify promptly the regressions (correct sentences that become incorrect in the new tested version) that could occur between two versions of the KANT MT system. The first remark is that post-editors at Caterpillar did not do the scoring, presented in Table 1. Therefore, the scores are probably higher than what a post-editor would obtain if he had scored the corpus. In fact, what seems acceptable for a developer is not always good enough for a translator's point of view, which one has more experience with translation of technical vocabulary. Because the regression test corpus has not been fully scored before, we were not be able to make a score comparison.

The regression test corpus contains 19,294 sentences selected in a way that it is representative of the Caterpillar technical documents domain. The level of *perfect output* obtained with the KANT system for French translation reaches 70% of the total number of sentences in the corpus, as shown in Table 1.

	Sentence Number	**Per cent of corpus**

Perfect output	13670	70.85%
Incorrect translation	1995	10.34%
Minimal post-editing	1291	6.69%
Incorrect preposition	604	3.13%
Bad authoring	461	2.39%
Word order	382	1.98%
Truncated output	214	1.11%
Grammar problem	193	1.00%
Incorrect case	151	0.78%
Incorrect Interlingua	112	0.58%
Incorrect morphology	107	0.55%
No interlingua	37	0.19%
Incorrect voice	25	0.13%
Domain model problem	19	0.10%
Incorrect tense	18	0.09%
Other problem	15	0.08%
	19294	

Table 1: Regression test corpus score

The second row of Table 1 corresponds to the number of *incorrect translation* that were generated, including bad lexical selections, incorrect structure, etc…

The *minimal post-editings* are minor generation problems that can be quickly corrected manually, like a missing/erroneous determiner. Those problems are not considered as crucial for the translation, and will be manually handled by post-editors. Next comes the number corresponding to the *incorrect prepositions*. They represent about 3% of the sentences. Some work still needs to be done on prepositions generation, but it will not be possible to correct all preposition errors with the help of lexical selection rules. The translation of prepositions from English into French has too many exceptions to rely on lexical selection (see section 4.2.4).

Among the remaining errors, we should say that 2.4% of the sentences were not authored in a correct way (unauthorized structure or wrong vocabulary choice). In addition, we notice that about 2% of the sentences had a word order problem and about 1% of the sentences were truncated (mostly missing adverb or tag generation problem).

Those results (even if optimistic) are interesting because they illustrate the way the errors are distributed. In addition, they show that in 70% of the cases, post-editors would not make any modifications to the sentence and only small modifications in 10% of the cases. Hence, it could be a huge gain of time in the translation process, if an automatic correction component was available. Our work is a contribution to such an improvement.

5 Potential of Statistical Language Model for Improving French Generation

Although pure statistical approaches were proved rather inadequate for machine translation (Wilks 1993), statistics were not completely banished from the domain. More and more MT systems use statistics in order to improve the generation or the analysis phase, like for automatic disambiguation (Carbonell et al. 1992).

The current KANT system does not use any statistical tool for generation purpose, but only information provided by lexicons and rules. In this chapter, we will present an attempt to use Statistical Language Models (SLM) in order to improve the quality of translation obtained in French generation. We focused our efforts on determiners and prepositions replacement and insertion as an automated postediting process. An attempt in automated postediting for determiner insertion in English has been already proposed in (Knight et al. 1994), but the approach was rule-based.

The proposed system is not dealing with English source sentences, but is based only on words context in the French sentences. The reason for this choice is of different orders. First, the system was developed in order to be used as post-module of the KANT generation process. This position does not give an access to the source sentence in the current system. Second, it is interesting to see how much information is provided by the context in a French sentence. Last, the time available for this research was limited and did not allow a large-scale study and development.

5.1 Problem Presentation

Along the work done on the KANT system in collaboration with the translation department at Caterpillar, we noticed that most minimal post-editing (small modification of translation by a post-editor) were due to incorrect or missing determiners or prepositions.

English:	"The wiring insulation must be *in* good shape."
AMT French:	"L'isolation du câblage doit être ***dans*** bon état."
Post-edited French:	"L'isolation du câblage doit être ***en*** bon état."

In this case, the preposition *in* was incorrectly translated by the AMT system as ***dans***, when it should be ***en***. A solution for such a problem would be to write a selection rule for the preposition *in* in order to translate it as ***en*** when followed by *good shape*. However, if this method was chosen, it would be very long and exhaustive to list all specific cases where *in* has to be translated as ***en***. Another solution to this problem is to learn these regularities from a corpus, through SLM. This solution has the advantage to be less time consuming for target language developers because based on machine learning.

5.2 Idea

Statistical language models have been widely used in the last decade in different domains, especially where language generation is needed (like natural speech processing). The purpose is always the same: trying to catch the regularities of a text stream (a corpus). The *n*-gram tells how much context is taken into consideration, where *n-1* is the length of this context. The entropy of a model is useful to estimate how much a model is able to catch these regularities. The more regularities the model catches, the less the entropy would be.

The nature of SLM corresponds highly to our needs, because we are dealing with text stream and because of the nature of the text we handle. The KANT MT system uses controlled English sentences as source; hence, the target sentences will have a certain degree of control as well (Allen 2000), controlled in its structure, but also in its vocabulary. On this last point, the vocabulary is even more controlled that we are dealing with a very specific domain (Caterpillar manuals for heavy machinery).

5.3 Principle

Our first objective was to reuse existing parts with the purpose of sparing the development time. From this principle, we used available tools in order to create the SLM. The Cambridge-CMU toolkit was a good candidate for our needs. Developed mainly at Carnegie Mellon University by Ronald Rosenfeld (Rosenfeld 1994), the kit allows, through a sequence of modules, to build an *n*-gram model stored as an ARPA standard format ASCII file. The main idea is to use this *n*-gram as a source of knowledge, a record of possible contexts for determiners/prepositions. For this research, 3-gram has been used for different experiments described further. Then we built a tool devoted to extract the information included in the SLM in order to correct or insert erroneous determiners/prepositions like illustrated in *figure 17*.

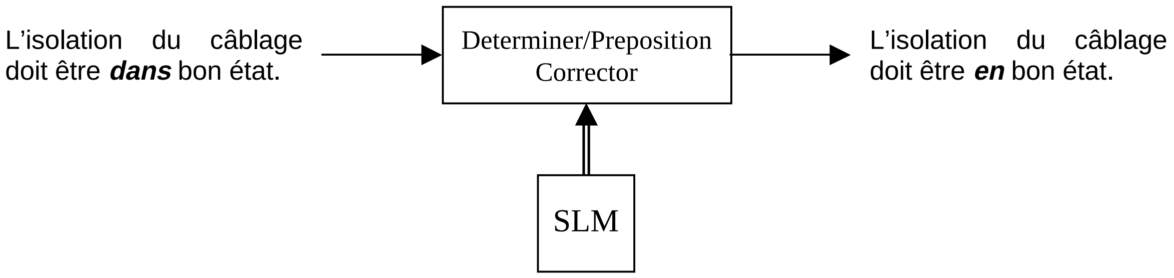
L'isolation du câblage doit être ***dans*** bon état.
Determiner/Preposition Corrector
L'isolation du câblage doit être ***en*** bon état.
SLM

Figure 17: Principle of the determiner/preposition corrector

5.4 Building the Model

An aligned corpus provided by Caterpillar has been the basis corpus. This corpus contains 111,047 unique sentences (no duplicate sentences). One can think that because we are using a corpus containing only unique sentences the model will be biased. However, this assumption is only partially right. In the present case, we are only interested by determiners/prepositions used as transition between words. Another element in the same direction is that small sentences would be more likely to be duplicated than long sentences. To have them missing do not really affect the model, because those sentences would be more likely to contain fewer determiners/prepositions.

The corpus had to be the most representative possible according to the MT output in order to take into consideration a vocabulary as large as possible.

5.4.1 Corpus Cleanup

We used Perl scripts to cleanup and to add some special context cue in the corpus. Spaces were inserted around all punctuation, tags and words. Two kinds of tag were added as context cue. The first one *<s>* is used to wrap sentences (containing a terminal punctuation), and the second one *<p>* is used to wrap phrases (without terminal punctuation). It is very important to differentiate between sentences and phrases because sentences are more likely to have a determiner as first word, when phrases do not. Some special words were also added in order to replace numbers and codes as single terms (*NUM* and *CODE*). Hence, we are more interested to know that the word ***numéro*** precedes a number than to know that the word ***numéro*** precedes the number *3*.
The two following sentences illustrate the cleanup procedure:

Original: "Le régime moteur est entre 15 tours par minute (rpm)."
Modified: "<s> le régime moteur est entre NUM tours par minute (rpm) . </s>"

Original: "<code>GA-4</code> <codedesc>Fuel Level</codedesc>"
Modified: "<p> <code> CODE </code> <codedesc> Fuel Level </codedesc> </p>"

In order to consider partitive structures ("*de la*") as single elements, they had to be joined in the cleanup task.

de la → de_la
de l' → de_l'

This was done in order to reduce the complexity of the decision criterion needed for determiners and prepositions replacement (*section 5.4.1*) and insertion (*section 5.4.2*).

5.4.2 Creation of the Language Model

The CMU-Cambridge Statistical Language Modeling Toolkit (version 2) is an open source set of UNIX software tools making easy the construction and testing of 4-gram, 5-gram and over, language models. Because designed only to compute the perplexity of a text, the evaluation tool was inappropriate for our experiment, and we had to develop our own tools.

The following steps are required in order to generate a language model in ARPA format containing an *n*-gram model:

Text → Extract word frequency → Select Vocabulary → Vocabulary
(*text2wfreq*) (*wfreq2vocab*)

Text + Vocabulary → Extract *n*-gram frequency → id-Ngram file
(*text2idngram*)

id-Ngram + Vocabulary → ARPA format file generation → Language Model
(*idngram2lm*)

The cleaned corpus had 1,876,168 words with 15,771 unique words. These numbers are interesting because they confirm our hypothesis about the degree of control of the used vocabulary. The number of unique words is very small comparing to the total number of words in the whole corpus. After removing the words appearing only once in the corpus, the final size of the vocabulary is 11,061 words.

This vocabulary was the basis for the generation of a 3-gram SLM as an ASCII file in ARPA standard format.

N-gram	Number
1-gram	11,061
2-gram	126,969
3-gram	354,539

Table 2: Number of N-gram for the 3-gram model

5.5 Sentence Correction Tools

The first step has been to look at some of these problems to identify the requirements. The post-edited sentences were a good place where to look at, because they represent the sentences with translation problems. We just had to extract sentences where the post-editing has been done on determiner or preposition. The first observation was that they were almost only incorrect or missing determiners/prepositions, but no extra ones. Starting from this observation, we decided to create two tools only, one to correct inaccurate determiners/prepositions, the other to insert them when missing. There was no need for a det/prep deletion tool. The tools have to be independent because they do not share the same decision criterion and do not modify the sentences in the same way. However, they both use information contained in the language model. All the N-grams are loaded in memory in such a way that they can be easily accessed to compute the criterion.

Before any process could be applied to a sentence, this last one should be cleaned in the same way it was done for the corpus. Principally, adding spaces around punctuation and adding context cue to wrap the sentences/phrases (as explained in section 5.4.1).

5.5.1 Determiner/Preposition Replacement

The goal is to replace erroneous determiners or prepositions by "better" ones. This statement brings two distinctive questions: how to recognize a determiner/preposition? and, how do we know that one is better that another one?

In order to identify determiners/prepositions (det/prep), we created a file containing the list of det/prep we are interested in checking. This file contains also the list of det/prep allowed as replacement.

dans	→	en
en	→	dans
le	→	la
la	→	le
un	→	une
une	→	un
à	→	en
à	→	au

This list has been build by looking at post-edited sentences, and is very restrictive to avoid completely incoherent replacements. For example, it is nonsense to add the following rule:

dans	→	de

Because the English term from where ***dans*** has been generated is probably *in* or *within*, it has almost no chance to be translated as ***de***. Another element against a non-controlled det/prep target is that there is possibly a better preposition to use instead of ***dans*** in a certain context, but the sense would be completely different and far from what we expect.

In addition, there is no possible ambiguity about the parts of speech of the handled det/prep because of the controlled language used as input. Hence, the Caterpillar Technical English does not allow employing the pronoun *it*, which prevents from having the pronoun *le* or *la* in the translated sentences. The same thing happens with the preposition *en*, which cannot have adverb or pronoun as part of speech.

Now that we know how to distinguish det/prep, we have to define a criterion in order to know if a det/prep is "better" than another one in a specific context. At least two elements of context can give some information about the det/prep. The first element is the word following the det/prep, giving the gender and number for a determiner for example. The second element of context is the word preceding the det/prep, especially in the case of prepositions where some verbs have to be followed by a specific preposition (e.g. "*demander à*"). We used this minimum context to defined a decision criterion based on counts of 3-gram, as illustrated by:

$$\frac{C(w_1, det_1, w_3)}{C(w_1, det_2, w_3)} \qquad (1)$$

Where det_2 is the current det/prep and det_1 is the candidate det/prep. This ratio would work only if the 3-gram was present in the corpus. This assumption cannot be verified in most cases. For this reason we have to modify this ratio. The conditional probabilities $P(w_3 / w_1, w_2)$ and $P(w_2 / w_1)$ can be rewrite as the following quotients of counts:

$$P(w_3/w_1, w_2) = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)} \qquad (2)$$

$$P(w_2/w_1) = \frac{C(w_1, w_2)}{C(w_1)} \qquad (3)$$

The formula (2) is the formula used by the CMU-toolkit to compute the *3*-gram probabilities of the SLM. It uses the same process (3) to estimate the 2-gram probabilities. The unigram probabilities *P(word)*are estimated by *count(word) / NbWords in corpus*.

Using the statements in (2) and (3), and the Bayes' theorem we can replace the counts to obtain the next ratio:

$$\frac{P(w_3/w_1, det_1)\cdot P(det_1/w_1)}{P(w_3/w_1, det_2)\cdot P(det_2/w_1)} \tag{4}$$

The new criterion derived from (1) uses probabilities instead of counts. The two main advantages of (4) are that it uses conditional probabilities as stored in the language model and that it allows to use back-off weights generated by the CMU-Cambridge toolkit. The back-off weights are used when there is no probability for a specific *n*-gram in a model. This means that the *n*-gram was not present in the training corpus while building the language model. For example, if the 3-gram (w_1, det_1, w_3) is not present in the model, the probability $P(w_3 / w_1, det_1)$ would be given by the probability of w_3 knowing det_1 pondered by the back-off weight of (w_1, det_1).

However, it is not desired to back-off as far as singleton, because we would loose all contextual information. Hence, a restriction is applied by imposing the presence of the bi-gram (det_1, w_3) in order to declare det_1 a potential replacement for det_2.

Because we have joined the preposition ***de*** with the determiner ***la*** and ***l'***, it is possible for the tool to correct errors such as:

Original: "Ceci provoque une confusion au sujet ***d'***évacuation du circuit de refroidissement."

Corrected: "Ceci provoque une confusion au sujet ***de l'***évacuation du circuit de refroidissement."

This prevents us to use 4-gram in order to catch contextual information around the couple (*de*, *l'*).

The ratio (2) obtained with an alternative det/prep has to be inferior to an empirically defined threshold in order to replace the current one. This threshold was fixed in order to be certain that the new det/prep is a much better candidate than the original one.

5.5.2 Determiner/Preposition Insertion

The goal for this tool is to insert mostly determiners where they are likely to be missing. The only preposition handled by this tool is the preposition *de* used in the very specific case of partitive structures (e.g. "*de l'eau*"). Other prepositions are not likely to be missing, but only incorrect. Then, the first tool described would handle them.

Like for the replacement tool, a list of determiner candidate is used for insertion:

l'	d'
le	de
la	du
les	des

un	de_la
une	de_l'

The approach for the decision criterion is a little bit different from the one used before. In the present case, there are no mark points to know where to try the det/prep insertion. We have to examine the possibility to insert a determiner between every word couples (w_1, w_2). Because of complexity of the insertion process, two criterions had to be defined:

First decision criterion:

As for the det/prep replacement, a criterion relying on a similar ratio is computed:

$$\frac{C(w_1, \mathrm{det}, w_2)}{C(w_1, w_2)} \quad (5)$$

Where *det* is the det/prep tested for potential insertion. Also based on counts, it is the ratio of the number of occurrences of the 3-gram (w_1, *det*, w_2) and the number of occurrences of the 2-gram (w_1, w_2).

By derivation of the ratio (5) using formulas (2) and (3), we obtain:

$$\frac{P(w_2/w_1, \mathrm{det}) \cdot P(\mathrm{det}/w_1)}{P(w_2/w_1)} \quad (6)$$

This criterion (6) is not sufficient by itself because if the couple (w_1, w_2) is not present in the training corpus, the probability $P(w_1, w_2)$ could be very small by backing-off on unigrams. Then, a det/prep would be incorrectly inserted.

Second decision criterion:

We needed a second criterion in order to fulfill the drawbacks of the first one, in other words, the lack of contextual information "analysis". The idea is to gather more information around the insertion point to see if the sentence "sounds better" with this new det/prep. The entropy function (7) is a good estimator for this purpose, because it quantifies the level of regularity of a sentence.

$$\mathrm{Entropy}_{3-\mathrm{gram}}(w_1, w_2, w_3, w_4) = -\frac{(\log[P\ (w_1)] + \log[P\ (w_2/w_1)] + \log[P\ (w_3/w_1, w_2)] + \log[P\ (w_4/w_2, w_3)])}{\mathrm{NbW}\mathit{ords}} \quad (7)$$

Where *NbWords* corresponds to the number of words in the sentence (4 in this example).

To reduce computation cost, the entropy is only computed on a bounded "window" around the insertion. After determining what is the best possible det/prep with the first criterion, we compare the entropy of the current window with the entropy of the same window with the inserted det/prep.

$$\mathrm{Entropy}(w_1, w_2, w_3, w_4) - \mathrm{Entropy}(w_1, w_2, \mathrm{det}, w_3, w_4) \quad (8)$$

If the entropy gain is more than a given threshold, the det/prep is inserted.

Like for the replacement tool, joined det/prep can be inserted as shown in the following example:

Original: "La poussière qui peut contenir amiante"
Corrected: "La poussière qui peut contenir ***de l'***amiante"

Both decision criterions are used successively, the first one determines which is the best determiner or preposition for a given context. Then, the second is the final decision criterion using the entropy function to decide if the sentence is "better" with or without the proposed det/prep.

5.5.3 Software Architecture

With the tools described above, several other functions had to be implemented. Among those functions, a tokenizer has been built, segmenting a sentence into words and adding extra tags like context cues. In order to use the probabilities stored in the language model, we also built a function able to load the language model file in memory and tools to access these probabilities. All the encoding has been done under C++ language.

5.6 Experimental Results

In this section, two kinds of results are presented: the results obtained with the development corpus (section 5.6.1), and the results obtained with a larger test corpus (5.6.2). The development corpus is small, but its results raise some interesting problems. The second corpus is much larger than the development corpus, and has been used after adjustment of the tools parameters in order to reduce the error rate.

5.6.1 Development Corpus

In order to test the tools while building it and setting the thresholds, we created a corpus composed with sentences translated by the KANT system. This corpus contains 438 French sentences.

A preliminary observation can be made about the corpus; it is not composed of randomly chosen sentences in the Caterpillar domain, but it is a sample extracted from a publication. This will have an influence on results, as we will point it out later in this section.

We used iteratively this corpus in order to set empirically thresholds. The sentence corrector tools processed the sentences, then, only modified sentences were analyzed. The scope was not to quantify the ability of the tools to fix all the det/prep problems in KANT French output, but to have a tool with a low error rate (<10%). As a principle, we decided that it is better to not replace/insert a det/prep than to replace/insert an incorrect one. For this reason we did not use the measures of recall and precision. Another reason is that the time available for this research was limited and did not allow the study of the test corpus to extract the number of possible right corrections.

The results introduced below were obtained as intermediate results, and were not produced with the final threshold. We present them because they raise some interesting remarks.

Among the 438 sentences of the corpus, 97 had been modified, which corresponds roughly to 22% of the total number of sentences. Some sentences had more than one modification. The Table 3 shows the repartition of those modifications between replacements and insertions. At the end, we have 112 modifications.

Modifications	Quantity	Per cent of total
Replacement	43	38.4%
Insertion	69	61.6%
Total	112	

Table 3: Repartition of modifications for development corpus

As we can see from Table 3, there were more insertions than replacements. This confirms the pre-analysis done on the post-edited sentences, where we discovered more missing det/prep than wrong ones. In Table 4, we show the percentage of correct and incorrect modifications. The last row of the table presents the number of modifications that had *no influence* on the sentence. As a matter of fact, the new sentences are not better neither worth than the original.

	Quantity	Per cent of total
Correct	71	63%
Incorrect	31	28%
No influence	10	9%

Table 4: Tool performances for development corpus

This is mostly due to bad translations or truncations in the output. Another thing to consider is that nine errors were due to tags, because the corpus is inconsistent on usage of det/prep with tags. This matter will be described in more details in section 5.6.1.2.

5.6.1.1 Examples of Correct Modifications

Missing determiners will occur principally when the number of a noun is "*mass*" in the interlingua. For example, the noun **O-DATA* has (number *mass*). As we can see in the following sentence, the determiner was missing in the KANT generated sentence:

Original:	"Charger données sur le cycle avant que les limites de mémoire soient obtenues."
Corrected:	"Charger ***les*** données sur le cycle avant que les limites de mémoire soient obtenues."

The determiner "***les***" was inserted thanks to the ratio:

$$\frac{C(\text{"changer","les","données"})}{C(\text{"changer","données"})} > cst$$

In this other example, the preposition ***dans*** was replaced by the preposition ***en*** which is correct. AMT error comes from the difficulty to translate the English preposition *in* as described in (Japkowicz 1991).

Original: "<title>Changement de position ***dans*** marche arrière avec le corps a se relevé</title>"

Corrected: "<title>Changement de position ***en*** marche arrière avec le corps a se relevé</title>"

5.6.1.2 Examples of Incorrect Modifications

The following example presents a bad insertion of the determiner ***un***. The principal reason for this incorrect insertion is the wrong translation of the noun **O-NUMBER* that should be translated as ***nombre*** in this case. Then, because the bigrams (*numéro,un*) and (*un,total*) are present in the training corpus, the insertion is made.

Original: "Dans ce mode, l'affichage à six postions montrera le numéro total des heures d'utilisation de la machine."

Corrected: "Dans ce mode, l'affichage à six postions montrera le numéro ***un*** total des heures d'utilisation de la machine."

In the next case, the problem is a little bit more meaning oriented. The adjective ***certain*** have two different senses. In this case, the sense is *sure* as *to be sure*. The other possible meaning for this term is *some*, which is an indefinite adjective in French.

Original: "Soyez certain qu'aucun personnel n'est près de l'aire de vidage."

Corrected: "Soyez ***un*** certain qu'aucun personnel n'est près de l'aire de vidage."

Errors can also occur when a word has multiple possible parts of speech. For example, the sentence below shows a miss-insertion of the determiner ***les*** before ***mises***. In this case, the word ***mises*** is the past participle form of the verb ***mettre*** (*to put*, *to lay*, *to place*, *to set*). The other possible part of speech for this word is noun, as we can find in the expression ***mises en garde*** (*warnings*).

English: "Make sure that the mounting bolts are put in position into the mounting support."

Original: "S'assurer que les vis de montage sont mises en position dans le support de montage."

Corrected: "S'assurer que les vis de montage sont ***les*** mises en position dans le support de montage."

This example shows the limitation of the back-off process. The sequence of words ***sont les*** is not very likely to occur in CTE. However, because of their high frequency in the training corpus, they will get a score not too penalizing, and because the sequence ***les mises*** has a "high" probability to occur, the insertion is happening.
Errors like this one are difficult to fix without any knowledge of the part of speech.

The last kind of error is tags related problem. The following example shows an insertion of the determiner ***le*** where it should not.

Original: "Se référer au <pubref><pubtype>Guide d'utilisation et d'entretien</pubtype><ie-topic>..."
Corrected: "Se référer au <pubref><pubtype>***le*** Guide d'utilisation et d'entretien</pubtype><ie-topic>..."

Sown in this example, the French MT system is consistently adding the determiner in front of the tag sequence tags *<pubref><pubtype>*. However, the insertion has been made because the training corpus is inconsistent in determiner placement regarding to tags and because we are not taking into account enough contextual information. This amounts to process the sub-string:

Sub-string: "<pubref> <pubtype> Guide d'"

In this sub-string, not information of the determiner placed in font of the tags shows up, because positioned too far from the insertion point.

5.6.1.3 Remarks

The size of the development corpus and its origin bring a strong indication that the results are probably biased. We decided to present those results because they give a good example of SLM limitations. In order to reduce the error rate, we had to increase the constraints on replacements and insertions, those we did for the test corpus.

5.6.2 Test Corpus

This corpus is composed of 15,510 sentences coming from several publications in the Caterpillar domain. As we did for the development corpus, we looked only at modified sentences and not at the others. The first observation is that the tools modified 207 sentences. Compared to the whole corpus, this represents only 1.3%. We can immediately underline the higher level of constraints applied in this case. The modifications, shown in Table 5, are distributed in the same way as the precedent corpus according to replacements and insertions:

Modifications	Quantity	Per cent of total
Replacement	77	36.2%
Insertion	136	63.8%
Total	213	

Table 5: Repartition of modifications for the test corpus

However, the quality of modification has largely been improved as shown in Table 6.

	Replacement	Insertions	Total	Per cent of total
Correct	74	121	195	91.5%
Incorrect	3	15	18	8.5%
Total	77	136	213	

Table 6: Tool performances for test corpus

We have reached an acceptable level of incorrect modifications (<10%) of the French AMT output in order to not constrain the parameters anymore.

5.7 Conclusion and Prospects

It is obvious that the gain is fairly small because of only 1.3% of changes. Nevertheless, this experimentation was conducted in order to prove that a statistical language model could help in correcting erroneous output generated from a MT system, and this, from a very simple unilingual SLM. Now that this theory has been demonstrated, we present several ideas that can be applied in order to extend this work.

- Training corpus: The training corpus used for this experimentation contained less than 16,000 unique words though the Caterpillar domain contains more than 70,000 concepts. This brings a lack of training data to light. The bigger and the most representative the training corpus is, the better the SLM would be.
- Tags handling: Tags are a real problem because they do not carry intrinsic information for the meaning of the sentence, but they are markers of structural information. Two solutions can be conceived: removing them in order to have direct access to the contextual information or handling them in a special way, like to allow retrieving the words before of after some tags.
- Parts of speech: In order to avoid incorrect modifications due to multiple possible parts of speech, a *lemmatizer* and a *tagger* can be used. This will require the creation of at least two SLM, one for the lemmas and the other for the parts of speech. The difficulty, here, is to find/create the training corpus. The tool could even be integrated as a module after the grammar module, where lemmas and part of speech are available without any need for *lemmatizer* or *tagger*.
- Bilingual information: The English sentences can also provide useful information, especially in the case of prepositions. Even if statistical MT was proved inadequate for automatic machine translation, it could probably be excellent for punctual corrections in MT output.
- Threshold setting: Last improvement, but not the least, will be to implement a convergence algorithm to fix the different thresholds by training. That will also require a simple distance algorithm between what we want and what we have.

Implementing these propositions will not only require a complete restructuring of the software architecture, but also the conception of new tools in order to build the language models. Statistical tools should not be restrained to determiner and preposition correction, but can be employed in a larger context and for other languages. An excellent example would be the correction in Spanish generation of the auxiliaries’ *ser* and *estar* (*to be*), where it is almost impossible to write a lexical selection rule.

Conclusion

The field of commercial machine translation systems is a fast growing market, but commercially available translation accuracy leaves much to be desired. However, MT system developers have understood that a syntactic analysis is not enough to provide all the information required in order to generate a high quality translation. Therefore, knowledge-based systems, such as Carnegie Mellon University' KANT system, start to show up as the preeminent alternative for the future of MT. A colossal work is required to collect the syntactic and semantic knowledge for all the vocabulary within a specific language. For this reason, even long-term projects that have been working on the subject for years have not been able to cover completely the domain. More, they had to constrain the source language on the domain, the structure of the sentence or/and the vocabulary in order to achieve a high quality translation. On the other hand, the use of controlled input language allows imposing a certain degree of consistency on the source language, but also on the target language.

Since it was first release in 1996, the English-French translation generated from the KANT MT system has not ceased to be improved, thanks to successive French technical leaders that have added their contribution to the generation module. Even if many lexical selection rules have been added for the generation to meet the complexity level of the French language, many others still need to be written in order to increase the correctness and the style of the translations. However, our work on gerund mapping was of a great importance, because it impacts on a large range of sentences. The remaining problems with gerund generation can find a solution in the propositions we made. Nevertheless, it would be impossible to build a commercial system in order to fulfill everybody expectations, because they are different according to the customers. Actual MT systems should be customized in accordance with the user.

We also presented our contribution on reducing the size of the syntactic lexicon by moderating the depth of the stored representations for the translations.

The results we have obtained show that the KANT MT system for French generation has reached a high level of translation quality with about 70% of correctly translated sentences. In addition, we stated that about 10% of the remaining errors were due to missing or incorrect determiners and that about 3% of the prepositions were incorrect.

Then, we reported an experimentation we performed in order to improve the generation of determiners and prepositions by an automatic postediting system based on a statistical language model. Our strategy was to develop two separate tools for replacement and for insertion of determiners/prepositions. Although the results we obtained were limited, we put forward several approaches that can be used to enlarge the scope of this post-process.

Abbreviations and Acronyms

ARPA:	Advanced Research Project Agency
CMT:	Center for Machine Translation
CMU:	Carnegie Mellon University
CTE:	Caterpillar Technical English
Det/pre	Determiner/preposition
DMK:	Domain Model Kernel
FS:	Feature Structure
F-Structure:	Feature Structure
IR:	Interlingua Representation
KANT:	Knowledge-based Accurate Natural language Translation
KANTOO:	Knowledge-based Accurate Natural language Translation Object-Oriented
KBMT:	Knowledge-Based Machine Translation
KMT:	Knowledge Maintenance Tool
LMT:	Lexicon Maintenance Tool
LTD:	Language Translation Database
LTI:	Language Technologies Institute
MT:	Machine Translation
PATRICK:	PAThname Resolution Interpreter Code for KANTOO
SGML:	Standard Generalized Markup Language
SLM:	Statistical Language Models

References

Allen. J. and C. Hogan. (2000). "Toward the Development of a Postediting Module for Raw Machine Translation Output: A Controlled Language Perspective". In: *Proceedings of the Third International Controlled Language Applications Workshop*, Seattle, Washington.

Bahl, L. R., F. Jelinek and R. L. Mercer (1983). "A Maximum Likelihood Approach to Continuous Speech Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5 (2): 179-190.

Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer and. P. S. Roossin (1990). "A Statistical Approach to Machine Translation". In: *Computational Linguistics,* Vol. 16, Num 2.

Brown, R. D. (1996). "Example-Based Machine Translation in the Pangloss System". In: *Proceedings of the 16th International Conference on Computational Linguistics*. Copenhagen, Denmark: 169-174.

Baker, J. K. (1979). "Stochastic Modeling for Automatic Speech Understanding". In: Reddy, R. A. (ed.) *Speech Recognition*. Academic Press, New York, NY.

Carbonell, J. G., T. Mitamura and E. H. Nyberg 3rd. (1992). "The KANT perspective: a critique of pure transfer (and pure interlingua, pure statistics, …)". In: *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation,* Montreal, Quebec.

Carbonell, J. G., R. E. Cullingford and A. G. Gershman (1981). "Steps Towards Knowledge-Based Machine Translation". In: *IEEE PAMI,* Vol 3, Num 4.

Carlson, L., and S. Nirenburg (1990). "World Modeling for NLP". In: *Technical Report CMU-CMT-90-121,* Center for Machine Translation, Carnegie Mellon University, Pittsburgh, USA.

Farwell, D., Y. Wilks (1991). "ULTRA: a Multilingual Machine Translator". In: *Proceedings Machine Translation Summit III,* Washington, DC, 19-24.

Ferguson, J. D. (1980). "Hidden Markov Analysis: An Introduction". In: Ferguson, J. D. (ed.), *Hidden Markov Models for Speech*. IDA-CRD, Princeton, NJ.

Garside, R. G., G. N. Leech and G. R. Sampson, (1987). "The Computational Analysis of English: A Corpus-Based Approach". Longman, NY.

Gaussier E. (1995). "Modèles statistiques et patrons morphosyntaxiques pour l'extraction de lexiques bilingues". Thèse de doctorat en informatique fondamentale, Université Paris 7.

Goodman, K. and S. Nirenburg (1991). "The KBMT Project: A Case Study in Knowledge-Based Machine Translation". Morgan Kaufmann Publishers, San Mateo, CA.

Hutchins, W. J. and H. L. Somers (1992). "An Introduction to Machine Translation". Academic Press, London.

Japkowitz, N. and J. M. Wiebe (1991). "A System for Translating Locative Prepositions from English into French". In: *29th Annual Meeting of the Association for Computational Linguistics*: 153-160.

Kamprath, C., E. Adolphson, T. Mitamura and E. H. Nyberg 3rd (1998). "Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English". In: *Proceedings of the Second International Workshop on Controlled Language Applications: CLAW-98*, Pittsburgh.

Knight, K. and I. Chander (1994). "Automated Postediting of Documents". In: *Proceedings of the American Association of Artificial Intelligence AAAI-94*. Seattle, WA.

Mitamura, T. (1989). "The Hierarchical Organization of Predicate Frames for Interpretive Mapping in Natural Language Processing", PhD thesis, University of Pittsburgh.

Mitamura, T., E. H. Nyberg 3rd and J. G. Carbonell (1991). "An Efficient Interlingua Translation System for Multi-lingual Document Production". In: *Proceedings of the Third Machine Translation Summit*.

Muraki, K. (1987). "PIVOT: A Two-Phase Machine Translation System". In: *Machine Translation Summit - Manuscripts and Program*, Japan, 81-83.

Nirenburg, S., J. G. Carbonell, M. Tomita and K. Goodman (1992). "Machine translation: A knowledge-based approach". Morgan Kaufmann Publishers, San Mateo, CA.

Nirenburg, S., V. Raskin and A. Tucker (1986). "On Knowledge-Based Machine Translation". In: *Proceedings of International Conference on Computational Linguistics COLING-86*, Bonn, Germany.

Nyberg 3rd E. H., T. Mitamura and W. Huijsen (publication in process). "Controlled Language for Authoring and Translation". Submitted as a chapter for: *Computers and Translation: A Handbook for Translators*, Somers H. (ed.), Johns Benjamin Publishing.

Rosenfeld, R. (1994). "The CMU Statistical Language Modeling Toolkit, and its use in the 1994 ARPA CSR Evaluation". In: *Proceeding ARPA Spoken Language Technology Workshop,* Austin, Texas.

Sampson, G. R. (1986). "A Stochastic Approach to Parsing". In: *Proceedings of the 11th International Conference on Computational Linguistics*. 151-155.

Sharman, R. A., F. Jelinek and R. L. Mercer (1988). "Generating a Grammar for Statistical Training". In: *Proceeding of the IBM Conference on Natural Language Processing*, Thornwood, NY.

Sinclair, J. M. (1985). "Lexicographic Evidence". In: Ilson, R. (ed.) *Dictionaries, Lexicography and Language Learning*. Pergamon Press, New York, NY.

Tomita, M., J. G. Carbonell (1987). "The Universal Parser Architecture for Knowledge-based Machine Translation". In: *Proceedings of the 10th International Joint Conference on Artificial Intelligence*. Milan, Italy, 718-721.

Tomita, M. (1986). "Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems". Kluwer Academic Publishers. Boston, Massachusetts.

Uchida, H. (1989). "ATLAS-II: A machine translation system using conceptual structure as an interlingua". In: *Proceedings of the Second Machine Translation Summit*, Tokyo.

Vauquois, B. and C. Boitet (1985). "Automated translation at Grenoble University". In: Computational Linguistics, Vol. 11, Num 1: 28-36.

Wilkins, J. (1668). "An Essay Towards a Real Character and a Philosophical Language", London, England.

Wilks, Y. (1993). "Corpora and Machine Translation". In: *Proceedings of Machine Translation Summit IV*, Kobe, Japan.

Witkam, A. P. M. (1983). "Distributed Language Translation: Feasibility Study of a Multilingual Facility for Videotex Information Networks". BSO, Utrecht, Holland.

Wojcik, R. H., J. E. Hoard and Holzhauser (1990). "The Boeing Simplified English Checker". In: *Proceedings of the International Conference, Human Machine Interaction and Artificial Intelligence in Aeronautics and Space*. Toulouse, Centre d'Etudes et de Recherches de Toulouse, 43-57.