Towards a Dynamic Linkage of Example-Based and Rule-Based Machine Translation [1]

Michael Carl (IAI), Leonid L. Iomdin (IPPI), Catherine Pease (IAI) and Oliver Streiter (IAI)

IAI Institute for Applied Information Sciences, Martin-Luther Straße 14, 66111 Saarbrücken, Germany

carl,cath,oliver@iai.uni-sb.de

IPPI Institute for Information Transmission Problems, Bol'shoi Karetnyj Pereulok 19, Moscow, 101447, Russia

iomdin@iitp.ru

In order to ensure a better performance of a machine translation system, most importantly to improve the quality of translation, and to make the MT systems easier to tune to the needs of different users, IPPI and IAI are combining the advantages of two machine translation ideologies, those of inductive and deductive MT, into one system. The objective of this activity is to investigate the consequences of this linkage and to determine the types of linguistic entities that can be dynamically transferred between the different components without introducing additional translation errors. Extensive research in this area will contribute to a better understanding of translation as a human activity and help to optimize the general paradigm of machine translation.

[1] The paper was supported in part by a grant (No96-06-80346a) from the Russian Foundation of Basic Research to whom the authors express their gratitude.

Contents

1 Introduction

The expertise gained by the world's leading NLP producers over the last 20 years has demonstrated beyond any doubt that all of the individual approaches to the task of MT/NLP which have been employed so far have their strengths and weaknesses. It is very unlikely that an entirely new, "ideal" approach may be proposed and implemented on a sizeable scale in the foreseeable future. Substantial progress in the field can therefore probably be achieved only by combining the strengths of different approaches. Proceeding from this assumption, IAI and IPPI started to look for strategies of integrating different MT paradigms into one framework. The paradigms we are considering for such integration are rule-based MT (RBMT) and corpus-based MT (CBMT).

As stated in [Car98b], the difference between RBMT and CBMT can be also described as **deductive** vs. **inductive** MT. The fundamental difference between deductive MT and inductive MT is the source of knowledge that eventually determines the behavior of the system. Deductive MT systems rely on linguists and linguistic engineers, who create or modify sets of rules in accordance with their knowledge, expertise, and intuition. In inductive MT systems, the rules are derived by the system itself and rely on a given set of translation examples. The adjustment of new translation requirements and improvement of an inductive MT system can thus be achieved by adding new translation examples.

This paper describes the interaction of two CBMT approaches, an example-based machine translation (EBMT) system and a tagged Translation Memory system, with two concrete RBMT systems. It is hoped that the linkage of different MT approaches will essentially improve the quality of machine translation, ensure a better performance, and make it easier to tune to varying needs. The objective of this activity is to investigate the consequences of this linkage and to determine exactly what kind of linguistic entities to be translated (syntactic constructions, lexicographic types, collocations etc...) can be dynamically exchanged between the different components without producing additional translation errors. Extensive research in this area will contribute to a better understanding of translation as a human activity and help to optimize the general paradigm of machine translation.

In order to combine different MT approaches in such a way that the new integrated system outperforms each individual component, we first establish weaknesses and strengths of the respective approaches. This will be done in section 2. In this way, we will be able to find out if, and how, a weakness of one component can be counterbalanced by a strong feature of the other component. This analysis will guide the strategies for linkage. These strategies will be

explained in section 3 where we strongly argue in favor of a dynamic linkage, since we believe that a static form of linkage cannot combine the advantages of the approaches in the best way. Concrete experiments and a fine tuning phase will show the scope of the newly created integrated system. Accordingly, we present the components to be linked in the experiments in section 4. These are, in the order of presentation, ETAP-3, CAT2 and EDGAR. The operation of the integrated systems is described in section 5. A discussion of the achieved results and future prospect (section 6) concludes the paper.

2 Parameters of Performance

Deductive and inductive approaches represent diametrically opposed strategies to MT. Neither of these has so far given a satisfactory response to the world's increasing need for automatic translation. The problem is that both have, beside their obvious advantages, a number of serious drawbacks. To list but a few, a TM, even a very large one, is unlikely to translate correctly a completely new sentence, let alone a new text. In their turn, RBMT systems generally do not learn (i.e. do not store translation results to be re-used later) and are difficult to adapt to new domains. In a way, both approaches are cumbersome and lack flexibility: they can hardly be expected to switch strategies in compliance with the changing requirements. Moreover, what happens to be an advantage in one approach may turn into a disadvantage in the other. So, if texts to be processed show great variance, the generally higher coverage of RBMT systems is of advantage. On the other hand, a CBMT system may be preferred to an RBMT system if text variation is limited because in this case they usually show greater reliability of the produced translation. This means that the advantages should be combined depending on the user's needs and the sorts of texts that are to be translated. Therefore, if we want to find a linkage which allows for a mutual compensation of the weaknesses to the maximum extent possible, we have to have a closer look at some parameters which characterize the performance of the two paradigms.

The parameters we discuss are of three different types. The first two describe external requirements and the third presents system internal requirements. The first type refers to the **quality of translation**. The second type describes the **coverage**, i.e. the range of texts or text types an MT system can handle without deteriorating its performance. The system internal parameters are **adaptability** and **reliability/tunability**. In the last subsection we shall discuss the interrelation between these parameters.

2.1 Translation Quality

In what follows we distinguish five quality levels of translation. An **indicative** translation tells the user what the text is about. An **informative** translation allows the user to understand more or less the content of the source text. A **literal** translation provides a translation for each unit of the source text in a correct grammatical form. A **reliable** translation is a translation which not only retains the meaning of the source text but is idiomatically and stylistically correct. A **user-oriented** translation is a translation which is correct from the standpoint of a particular user. User requirements may vary according to text type, terminological preferences, personal style, etc.

2.2 Coverage

All MT approaches may achieve a high quality translation for a corpus they have been tuned to. Any changes in this corpus, however, may entail an impairment in the quality of translation. We define therefore the coverage of an MT system as the extent to which various types of source texts can be translated into the target language (TL) without affecting the translation quality. Depending on a concrete MT application, a high coverage may be a compulsory feature or not. When a system deals with closed subject domains such as weather forecasts a high coverage is not necessary, while a translation tool operating in an Internet surfing machine has to handle a wide variety of domains equally well, in which case a high coverage is a must.

2.3 System Internal Parameters

The above quality levels of translation can be linked directly to some of the internal parameters of an MT system, which are listed below.

2.3.1 Recall

Recall refers to the matching between the source text and translation units available in the MT system. Different MT approaches do not differ principally with respect to their recall, i.e. all may achieve a 100% recall for a reference corpus, be it a set of reference translations in the case of CBMT systems or a set of test sentences for RBMT systems. Changes in this reference corpus however may play havoc with the recall value. It is obvious that if an MT system has a low recall, it can hardly be expected to yield anything but an

indicative translation. In order to promote the quality from an indicative to at least an informative translation, a high recall is required, i.e. a high percentage of the source text must be mapped onto translation units.

2.3.2 Adaptability

In order to achieve a literal translation, adaptations of the target side of the translation unit (TU) are required. This means that translation units must be rearranged and modified so that a well-formed text is produced. In other words, adaptability refers to the extent in which the MT system can properly incorporate the target side of a translation unit in its target context. To give a simple example, consider a German/English EBMT system which contains the following three TUs.

1	*die Brille*	$\leftrightarrow$	*the eyeglasses*
2	*ist billiger*	$\leftrightarrow$	*is cheaper*
3	*in Russland*	$\leftrightarrow$	*in Russia*

These examples provide a 100% recall for the sentence *Die Brille ist billiger in Russland*. The mere concatenation of the target TUs, however, yields the ungrammatical string **The eyeglasses is cheaper in Russia*. For a correct adaptation, TL peculiarities such as case assignment, the choice of prepositional forms, agreement in case, number and/or person, the choice of the part of speech etc. must be taken into account.

MT approaches differ in the complexity of adaptations which they can perform. The degree to which an MT system can perform adaptations will be referred to as **adaptability**. A requirement for adaptability is that TUs are described in such a way as to allow the adaptation both between and within these units. This means that the adaptation power of an MT system depends on the linguistic richness of its internal representation. In MT systems which lack linguistic knowledge this ability is limited.

The complexity of the adaptation correlates with the size of TUs, i.e. the complexity can be reduced to a minimum by choosing large TUs. Secondly, the complexity of the adaptation depends on the linguistic nature of the TUs chosen. If TUs are chosen in such a way that they are only minimally affected by the linguistic context, the complexity is reduced.

2.3.3 Reliability/Tunability

In order to raise the quality status from literal translation to a reliable or user-oriented translation, the MT system has to be adapted to the requirements of the TL in general and/or to those of a particular user. In principle there are two ways to achieve this: a) modify the adaptation mechanism itself or b) modify the size of the TUs by using longer translation examples. In most of the cases it is easier to do the latter because modifying the adaptation mechanism generally requires profound knowledge of the internal structure of the system, whereas translation examples can normally be added to the system by an average system user. In addition, modification of the adaptation mechanism has a natural limit. If the complexity of the system grows beyond a level that is manageable or calculable, the modification proves counter-productive. The more adaptations the system applies to the translated chunks, the higher the probability of errors and uncontrolled output. In other words, of two MT systems having the same coverage the one using larger TUs is preferable.

2.4 Interaction of Parameters

2.4.1 Reliability vs. Coverage

MT approaches ranging from TMs over EBMT to RBMT differ with respect to the relative importance they attach to the TUs (e.g. lexicon, Term Bank, translation examples) and the adaptation mechanism (e.g. grammar). Whereas adaptation is the mechanism by which a high coverage (i.e. a stable translation quality across different text types) of an MT system is achieved, the nature of TUs is responsible for the reliability of the system. While the inductive approaches rely on shallow adaptation mechanism and huge amounts of bulky lexical material, RBMT systems rely on a complicated adaptation mechanism and a relatively compact lexicon where the entries describe small parts of the text and the potential contexts are coded with intricate feature structures. In principle, both techniques are beneficial: whereas adaptation enhances the coverage of the system, the use of large TUs increases its reliability. It is obvious, however, that the two requirements, i.e. to be reliable and to have a high coverage, are mutually exclusive if applied to one and the same MT paradigm. A combination of MT paradigms helps resolve this contradiction: the deductive component guarantees a high coverage and the inductive component raises the reliability level.

2.4.2 Adaptability vs. Translation Quality

The complexity of the adaptation mechanisms is directly related to the probability of producing internal errors. After an internal error occurs, the output can no longer be controlled so that the correctness and reliability of the translation can be badly affected by incorrect word choice, incorrect word order or incorrect morpho-syntactic assignment and labeling. In addition, a translation produced with large amounts of adaptations, even though it may be correct, is unlikely to be reliable, since minor changes in the source text or in the lexicon may drastically alter the output and render it unreliable.

2.4.3 Coverage vs. Translation Quality

In practice, when applied to real texts, high coverage and reliability are mutually exclusive. This is due to the fact that ample coverage can be achieved easier with short TUs. Short TUs, however require a complicated adaptation mechanism. With short TUs and a cumbersome adaptation mechanism, however, the best translation quality level that can be achieved is the literal translation. On the other hand, the longer the TUs are, the more reliable a MT system may become. However, with longer TUs the recall and coverage is likely to decrease because longer TUs are less likely to be found.

3 Strategies for Linkage

3.1 Dynamic Linkage

As significant progress in the field of MT is unlikely to be achieved by refining one single MT approach, we combine different approaches dynamically, so that with each parameter which defines the translation settings (text type, user, translation examples available), the system changes the mode of translation in order to make full use of the modules' respective advantages. This approach contrasts with earlier static attempts to combine MT approaches. The attempts, for example, to run different translation engines in parallel (e.g. Pangloss, [Bro96]) actually offer little help, as only the final outputs of different engines are compared. On the other hand, it has been shown that an integration of different MT approaches may yield better results than those achieved by an individual system. An instructive example is the experience of Verbmobil where the use of the complementary strengths of various MT approaches in

one framework (deep analysis, shallow dialogue-act based approach and simple translation memory technology) improves the performance of the system (e.g. [N$\ddot{9}$7]).

3.2 Preferability of Large TUs

¿From what we have seen above, an optimally combined system should have the high recall and coverage of a RBMT system and the good tunability and good translation quality of MT approaches working with large TUs. This optimized system segments the source text into large TUs wherever possible, as is typical for inductive systems. If the TUs matched by the CBMT are too short or if parts of the source text cannot be matched, the TUs of the RBMT system are chosen since they are coded using richer linguistic data and are better suited for adaptation. In such an architecture, the use of large TUs should reduce the need for adaptation to a minimum and supply the user with fast and reliable translations. If, however, adaptations are required, they should be triggered and performed mostly by the RBMT component, which is actually intended for this purpose.

4 Existing MT Resources

Proceeding from these considerations, we conceived and staged two experiments. One experiment is the linkage of the RBMT system ETAP-3 with a TM prototype. The other experiment, performed on similar (but not identical!) principles, is the linkage of the RBMT system CAT2 with the EBMT system EDGAR. In the following three subsections we describe ETAP-3, CAT2 and EDGAR in more detail.

4.1 ETAP-3

4.1.1 General Layout

ETAP-3 (see e.g. [ABI$^+$89],[ABI$^+$92], [ABI$^+$93], [LL97]) is a large software system developed in IPPI for versatile NLP purposes, primarily for high quality machine translation. The system is based on the Meaning $\Leftrightarrow$ Text linguistic theory proposed by Igor Mel'čuk [Mel74] [Mel95] and makes use of dependency

trees for the representation of syntactic structures. ETAP-3's main working languages are Russian and English, for which full-scale morphological and syntactic parsers and generators, as well as 60,000-strong high level syntactic and semantic lexicons, have been developed. Within a number of INTAS-sponsored projects, a prototype Russian-to-German MT version has been developed in cooperation with IAI. ETAP-3 also has a test version of French-to-Russian MT system and an experimental version of Russian-to-Korean machine translation developed in cooperation with South Korean researchers. In addition, ETAP-3's major modules are used in an experimental natural language database interface and a Russian-to-UNL[2] converter.

The English-Russian and Russian-English modules of the ETAP-3 MT system translate scientific and technical texts belonging to the subject domains of computer science, electrical engineering, and material management.

The translation, performed sentence by sentence, resorts to a number of processing phases. All of these make use of a variety of rules types and a number of lexicons. The rules are written in a specially designed formalism, called FORET, which is based on a unique three-valued logic.

During the first phase of processing, a context-free morphological analysis module produces a **morphological structure** (MorphS) of the source sentence, i.e. a sequence of morphological representations for each of the words of the sentence. Each representation consists of a lexeme name and a set of inflectional features.

If a word is lexically and/or morphologically ambiguous, it obtains several representations, which are included into the (united) MorphS. E.g. the English sentence *Wishes father thoughts* receives the following MorphS:

WISH1, V, sg, 3-prs	FATHER1, S, pl	THOUGHT, S, pl
WISH2, S, pl	FATHER2, V, mf	

The output of the morphological analyzer, i.e. the MorphS, is sent to the parser, which produces a **syntactic structure** (SyntS) of the source sentence. SyntS is a dependency tree whose nodes correspond to the words of the source sentence and whose arcs are labeled with names of syntactic relations (of which ETAP-3 uses about 60). Normally, the top node of SyntS is the verbal predicate of the sentence, although almost any word can be a top node. If a source sentence

[2]UNL, or Universal Networking Language, is an English-based interlingua developed by the United Nations University in Tokyo for the purpose of offering Internet users a chance to semi-automatically translate Internet documents from and to a host of languages. Within the framework of the UNL project, linguistic teams from different countries, including IPPI and IAI, are developing lexicons and linguistic tools aimed at creating a pilot Internet natural language communication system.

is lexically and/or syntactically ambiguous, it obtains several SyntS. E.g. the English sentence *Change options* will obtain two SyntS:

```
(1) CHANGE1, V, mf --1-compl--> OPTION, S, pl
(2) CHANGE2, S, sg <--compos-- OPTION, S, pl
```

The parser is the most important module of ETAP-3. The higher the quality and adequacy of the SyntS, the higher the overall quality of translation will be. It is for this reason that the syntactic rules and the underlying lexicons are developed with utmost care.

Each of the SyntS produced by the parser is sent to the transfer module. The latter consists of four consecutively applied submodules that modify the SyntS in order to supply its equivalent in the TL. Finally, the TL SyntS is sent to the morphological generator that produces a normal sentence (or sentences) in the TL.

4.1.2 Interactive Term Recognizer

The ETAP-3 system has a specially designed parsing tool, called **interactive term recognizer**, which has been successfully used on a broad scale in ETAP-3 lexicographic work. As this tool is used in the hybrid system described below, we consider it necessary to describe it in some detail.

The operation of the interactive term recognizer tool can be summarized as follows.

If we wish to enter into the dictionary an idiomatic translation pair that cannot be obtained using regular lexical entries, we launch the ETAP-3 parser first for the source and then for the target expression. After the parser has found a source structure, it is displayed to the linguist, who must either reject or approve it. If the linguist rejects the first structure offered, the parser iteratively offers other structures until the linguist is happy with one, whereupon it moves to the TL parser and performs a similar course of action. The two resulting structures are compared with a rather large list of template SL/TL correspondences. If one of these correspondences matches the two structures, a reference to a template translation rule is produced and semi-automatically entered into the dictionary. If the source and the target structures are so different that no match can be found in this list, the recognizer produces a new straightforward translation rule, which is shown to the linguist and, if approved by him, is entered into the dictionary. Both template and straightforward rules are applied during the

transfer phase of the ETAP-3 operation. In addition, the parse assigned to the example is supplied with the tag specifying the top node and, wherever necessary, the information on what elements of the example other than the top node may have syntactic daughters and what types of these they may have.

4.2 CAT2

CAT2 is an NLP formalism developed for the purpose of multilingual MT. Within this formalism different grammars and lexicons have been developed by different project groups. At IAI, CAT2 is used in a German/English/French MT component, which can be compiled for a variety of subject domains. Additionally, CAT2 is used in the UNL Project as a lexicon server and as a German-to-UNL precoder. [3].

In this subsection we shall describe the translation strategy assumed by CAT2, which will be illustrated step-by-step by a translation example. The example is a German sentence

(1) *Der Sprachwissenschaftler hat bei der Arbeit grosse Angst vor ununterscheidbaren Morphemen*

which is translated into English.

The result of the morphological analysis of sentence (1) looks as follows:

(2)
```
{lu=d_art,c=w,sc=art,spec=def,ehead={g=f,nb=sg,case=dat;gen}
;{g=m,nb=sg,case=nom};{nb=plu,case=gen}}
{c=noun,lu=sprachwissenschaftler,ehead={nb=sg,case=nom;dat;acc,g=m}
;{nb=plu,case=nom;gen;acc,g=m}}
{lu=haben,c=w,sc=verb,vtyp=fiv,tns=pres,mode=ind,per=3,nb=sg}
...
```

A syntactic and semantic analysis is applied to this morphological representation, based on HPSG-like schemes of composition (see Figure 1). Besides the syntactic functions (subject, direct object, indirect object, modifier (mod) and function words (f) etc.), semantic roles are identified (a=agent, t=theme, g=goal). In our example, *Angst* 'fear' has been recognized as a predicative noun of the support verb *haben* 'have', *der Sprachwissenschaftler* 'the linguist' as the person who feels (=experiences) the fear and *ununterscheidbare Morpheme* 'undistinguishable morphemes' as the source of the fear.

The syntactico-semantic representation depicted in Figure 1 (referred to as **constituent structure**, or CS) is reshaped to facilitate the translation into the

[3]Descriptions of the CAT2 formalism can be found in [Sha94], [SS95], and [Str96]

Figure 1: CS Structure of Source Sentence

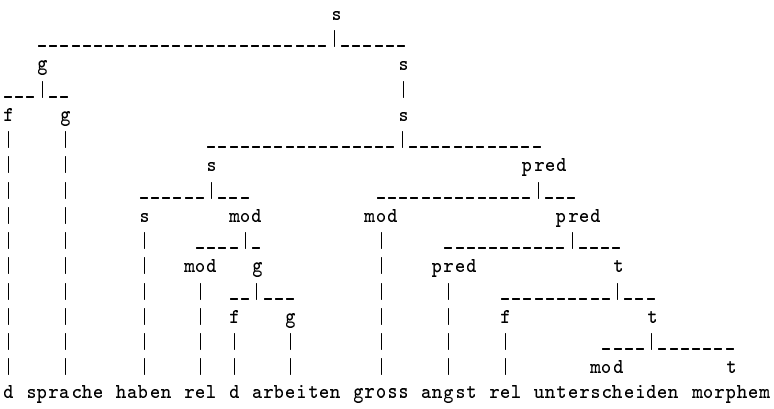
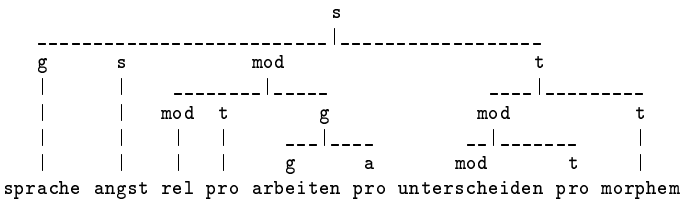


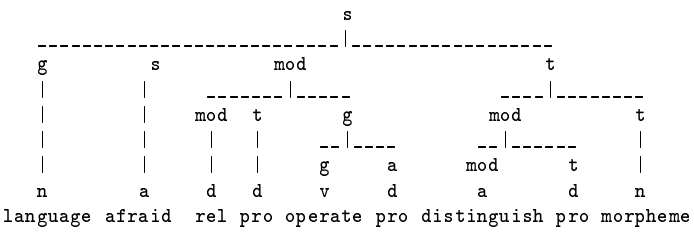
Figure 2: IS Structure of Source Sentence



TL as shown in Figure 2. First, all function words (determiners, case marking prepositions, degree words, auxiliaries etc.) are removed from the structure. The adjective *gross* 'big' is removed as well, as it acts in this case as marker of a high degree of *Angst*. Then, the binary syntactic structures are transformed into flat (multiple-branching) structures. In addition, pronouns are introduced as internal arguments of modifier relations (they may eventually appear in the target text in the form of relative pronouns). Support verbs (e.g. *haben*) and copula verbs are removed and the element bearing the argument structure (e.g. *Angst*) is moved into the position of the (former) copula. Such a restructuring is necessary if support verb constructions or copulative constructions are to be translated into simple verb constructions or if the TL (like Russian) has a zero copula. The output of the transformation, referred to as **interface structure**, or IS) is shown in Figure 2.

During transfer, the source IS is translated into the TL by replacing the lexical atoms of the SL by the corresponding lexical atoms of the TL. The choice

Figure 3: IS Structure of Target Sentence



of the lexical atom and its morphological derivation is constrained through the transfer of semantic information. Since the semantics but not the part of speech is controlled, the predicative noun *Angst* can be translated with the adjective *afraid*. The lexical transfer successively invokes different translational options until an overall integration of these options becomes possible. In our example the following options are created up to the moment the integration is possible:

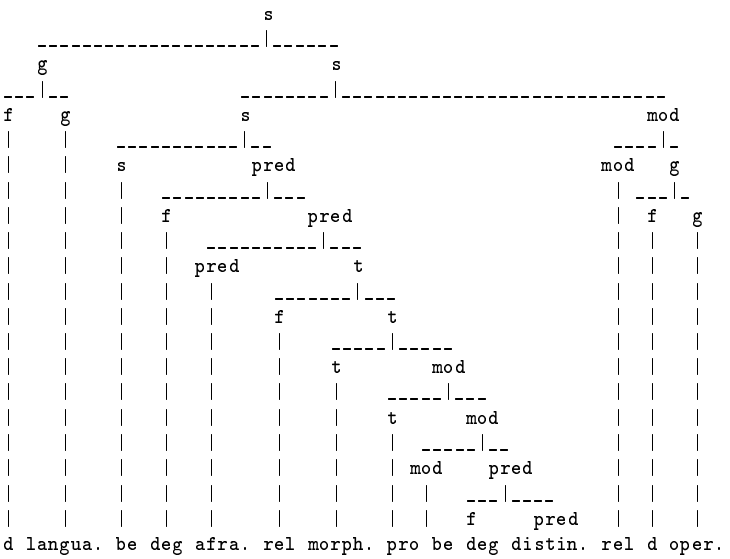
(5)
```
{slex=sprachwissenschaftler} => {lex=language,slex=linguist}
{slex=angst}                 => {lex=afraid,slex=afraid}
{slex=bei}                   => {lex=rel,slex=during}
{slex=angst}                 => {lex=pro,slex=?}
{slex=arbeit}                => {lex=operate,slex=operation}
{slex=someone}               => {lex=pro,slex=someone}
{slex=ununterscheidbar}      => {lex=differ,slex=different}
                                {lex=differ,slex=differ}
                                {lex=differ,slex=differently}
                                {lex=distinguish,slex=distinguishable}
{slex=someone}               => {lex=pro,slex=someone}
{slex=morphem}               => {lex=morpheme,slex=morpheme}
```

Throughout (5), slex denotes the lemma as it is found in the dictionary and lex refers to a generalized notion of lexeme as used in CAT2.

The integration of these hypothetical translations which satisfies all the constraints is the following target IS (3). In addition to the semantic roles we specify in this tree also the part of speech of the lexemes to be generated (n=noun,v=verb,a=adjective).

As can be seen in Figure 3, the German adjective *ununterscheidbar* is translated into the adjective *distinguishable* (just because the adjective *undistinguishable* is not present in the English lexicon of CAT2). The negation expressed by the German adjective must then be expressed in English by syntactic means. Support verbs and copulative verbs are inserted into the structure according to the specifications of the lexical items involved. In our example the copula *be* is introduced. The word order, which until now has remained that of the source

Figure 4: CS Structure of Target Sentence



language (SL), is rearranged, and function words (including negation markers) are generated as depicted in Figure 4.

The morphological generation produces the final translation of the sentence:

(8) *The linguist is very much afraid of morphemes that are not distinguishable during the operation.*

As can be seen in the above discussion, the translation is achieved with small TUs via complicated procedures of analysis and adaptation. Minor changes in the lexicon or the grammar would produce a completely different translation. If we wish to change the output of the system, a complex coding of lexical or grammatical rules is required.

4.3 EDGAR

EDGAR [Car98a] is an EBMT system prototype which relies on morphologically analyzed translation examples. EDGAR's main strategy is to decompose and "generalize" the sentence to be translated by matching it against a set of reference translations. A simple subject-verb-object sentence, for example, can be translated using a subject-verb-object template from the set of reference translations via linguistic generalization. The generalized input sentence is then specified (i.e. correct linguistic information is gathered) and "refined"

in the TL - in other words, the surface strings of the target text are generated. A prerequisite of the translation process is the existence of a case base which contains a set of reference example translations.

This section describes the basic ideas underlying the creation and compilation of the case base and outlines the generalization and refinement process which an input sentence undergoes in the translation process. It is shown how a case base can be created from a set of (sub-sententially aligned) reference translations and their generalizations.

4.3.1 Case Generalization and Compilation

The compilation of a case base involves the following steps:

- morphological analysis and lemmatization of examples
- sorting examples by chunk size (number of words)
- generalization and reduction of examples
- indexing of the examples and their generalizations

Morphological analysis yields for each word a feature bundle containing pairs of attribute/values which describe the (morphological) interpretation of that word. In this section, the results of the morphological analysis of are shown in italics.

Case generalization is seen as a sort of grammar induction derived from translation examples. It is presumed that each case may be viewed as a set of features that can be divided into two subsets: fixed features which are case-specific (e.g. lexical instantiations) and variable features (e.g. information such as grammatical case and number) which are typical for a whole range of similar cases. Generalization consists in replacing a sub-sequence of an example with a constraint variable. Generalized cases disregard the fixed features while keeping track of the variable ones. For instance, from French/English translation examples (1) and (2) below a generalized case (2g) can be inferred.

	French expression		English expression
(1)	*(ski)* $_{NOUN}$	$\longleftrightarrow$	*(ski)* $_{NOUN}$
(2)	*(station de ski)* $_{NOUN}$	$\longleftrightarrow$	*(ski station)* $_{NOUN}$
(2g)	*(station de* $\mathcal{X}_{NOUN}$*)* $_{NOUN}$	$\longleftrightarrow$	*(*$\mathcal{X}_{NOUN}$ *station)* $_{NOUN}$

Case (2g) will match a number of chunks such as *station de sport, station de taxi, station de métro, station de terre* etc. where the fillers of the slot $\mathcal{X}_{NOUN}$ are constrained by a set of features to be shared with *ski*. In this case only the feature *NOUN* must be shared. In the absence of full matching cases these sequences would be translated (sometimes incorrectly!) into *sport station, taxi station, metro station* and *ground station*, respectively. Generalizations are similar to grammar rules in a conventional NLP system. The difference is that generalizations are generated from examples and no explicit grammar rules are specified.

More than one reduction within a generalization is possible if different sequences are matched in an example. The ordering of the variables may be inverted on the TL side with respect to the SL side in order to account for the cases of word (or constituent) order discrepancies between SL and TL (e.g. Xacc essen Ynom $\Leftrightarrow$ Y eat X, as in *Spinat essen Kinder* $\Leftrightarrow$ *Children eat spinach*).

4.3.2 Translation by Generalizing and Refining a Sentence

In the translation process, (the result of the morphological analysis of) a new sentence is matched against the examples in the case base. Those sequences of the new sentence which match one or more examples are reduced to one node. The newly created node keeps track of the external constraints of the matching example(s) i (and/or the matched chunk of the input sentence) marked with subscript values such as *NOUN* or *ACC* and the internal constraints of the examples i marked with superscript values such as 1. Whereas the external constraints are visible in the generalization, the internal constraints only serve in the refinement step to determine the internal structure of the TL chunk to be generated. [4] The sentence – thus generalized – is then cyclically matched against the case base until no more reductions can be performed or until the entire sentence is reduced to a single node.

To give an example, let us assume that the case base below has been produced from the following English-German reference translations:

The big man eats a green apple. $\longleftrightarrow$ *Der grosse Mann ißt einen grünen Apfel.*
The small boy eats a red tomato. $\longleftrightarrow$ *Der kleine Junge ißt eine rote Tomate*

[4] Internal constraints are actually implemented as pointers to the TL side of the matching example.

Figure 5: Decomposition and Generalization

The English sentence *The small boy eats a green apple* is decomposed and reduced to generalization $\mathcal{Z}^{7/4,5}_{S,FIN}$

$$\begin{array}{ccc} \underbrace{\textit{The small boy}} & \textit{eat} & \underbrace{\textit{a green apple}} \\ \downarrow & \downarrow & \downarrow \\ \mathcal{X}^4_{DP,NOM} & \textit{eat} & \mathcal{Y}^5_{DP,ACC} \end{array} \tag{1}$$

$$\begin{array}{c} \underbrace{\mathcal{X}^4_{DP,NOM} \quad \textit{eat} \quad \mathcal{Y}^5_{DP,ACC}} \\ \downarrow \\ \mathcal{Z}^{7/4,5}_{S,FIN} \end{array} \tag{2}$$

For each example of the case base, a tag showing its type is provided:

(3) $(\textit{the big man})_{DP} \longleftrightarrow (\textit{der grosse Mann})_{DP}$

(4) $(\textit{the small boy})_{DP} \longleftrightarrow (\textit{der kleine Junge})_{DP}$

(5) $(\textit{a green apple})_{DP} \longleftrightarrow (\textit{einen grünen Apfel})_{DP}$

(6) $(\textit{a red tomato})_{DP} \longleftrightarrow (\textit{eine rote Tomate})_{DP}$

(7) $(\mathcal{X}_{DP,NOM}\ \textit{eat}\ \mathcal{Y}_{DP,ACC})_S \longleftrightarrow (\mathcal{X}_{DP,NOM}\ \textit{essen}\ \mathcal{Y}_{DP,ACC})_S$

The input sentence to be translated into German is the English sentence *The small boy eats a green apple*. As shown in Figunre 5, in the first generalization step (1) the sentence is decomposed into three chunks: */The small boy/ /eat/*, and */a green apple/*. The generalization "$\mathcal{X}^4_{DP,NOM}$ *eat* $Y^5_{DP,ACC}$" in (1) is computed from the examples 4 and 5 of the case base. The reductions $\mathcal{X}^4_{DP,NOM}$ and $Y^5_{DP,ACC}$ are single nodes which represent respectively the reduced chunks */The small boy/* and */a green apple/*. The reduced nodes include the external constraints $_{DP,NOM}$ and $_{DP,ACC}$ of the type of the matching example (*DP*) and the case of the reduced chunk (NOMINATIVE and ACCUSATIVE).

The internal constraints 4 and 5 are indices of the matching examples which are used in the refinement step to specify the appropriate TL chunk. Apart from the two reductions, the generalization in (1) contains the unreduced word *eat*.

A second level of generalization is shown in (2). Here, the English input sentence

Figure 6: Specification and Refinement

The generalization $\mathcal{Z}^{7/4,5}_{S,FIN}$ is refined into the German sentence *Der kleine Junge ißt einen grünen Apfel* .

$$\overbrace{\mathcal{X}^{4}_{DP,NOM} \quad essen \quad \mathcal{Y}^{5}_{DP,ACC}}^{\mathcal{Z}^{7/4,5}_{S,FIN}} \tag{3}$$

$$\overbrace{\mathcal{X}^{4}_{DP,NOM}}^{\downarrow}\ \overbrace{essen}^{\downarrow}\ \overbrace{\mathcal{Y}^{5}_{DP,ACC}}^{\downarrow} \qquad \overbrace{Der\ kleine\ Junge}\ \ isst\ \ \overbrace{einen\ grünen\ Apfel} \tag{4}$$

can be reduced into one single node based on example 7. This is possible because the generalization of the input sentence (i.e. $\mathcal{X}_{DP} \quad eat \quad \mathcal{Y}_{DP}$) matches example 7.

Refinement takes place as shown in Figugr 6. The reduced nodes of a generalization are recursively specified according to the internal constraints of the reductions and refined by applying the external constraints. Specification of node $\mathcal{Z}$ in (3) retrieves the TL side of example 7 from the case base and adds the internal constraints 9 and 10 into its respective slots $\mathcal{X}$ and $\mathcal{Y}$. Refinement of node $\mathcal{Y}$ in (4) handles case agreement in the generated German noun phrase *einen grünen Apfel* in accordance with the external constraint ACC. On the other hand, the verb *ißt* is inflected according to the internal constraints specifying the number of the subject node $\mathcal{X}$.

In the generalization step one can define two (possibly overlapping) sets of features which are used to match an input sentence against the case base: the fixed and the variable features. The fixed (lexical) set of features describes the specific characteristics of the matching example which show no variation and includes the lemma LU, noun gender G and part of speech C. The variable (morpho-syntactic) set of features includes tense TNS, verb form VTYP, number NB, case CASE, definiteness SPEC and prepositional form PFORM:[5]

[5]For technical reasons, the part of speech C is included into both sets of features.

Lexical features:	LU, G, C
Morpho-syntactic features:	C, TNS, VTYP, NB, CASE, SPEC, PFORM

It should be noted that some of the morpho-syntactic features are stored in the CB and therefore must agree with the source text, while other such features (e.g. CASE) are not stored but taken directly from the text. The former approach enhances the overall reliability of the system, whereas the latter contributes to a higher recall.

Depending on the type of the example, different features are percolated into the external constraints of the reduced nodes as shown in the following table.

phrase type	tag type	external constraint
adverbial phrase	*ADV*	—
adjective phrase	*A*	NB, CASE
proper name	*PROPER*	NB, CASE
noun phrase	*NOUN*	NB, CASE
determiner phrase	*DP*	NB, CASE, SPEC
prepositional phrase	*PP*	NB, CASE, SPEC, PFORM
sentence	*S*	TENSE, VFORM

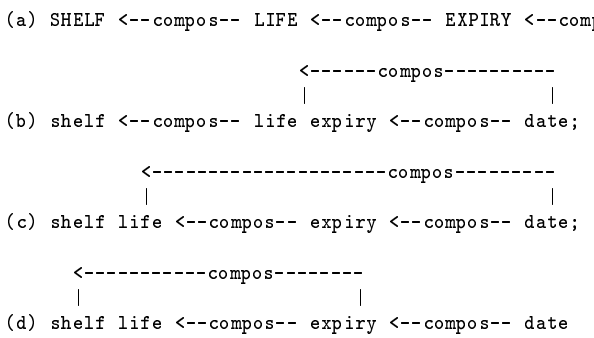
Part-of-speech information is percolated from the example matched by the lexical set of features into the reduced node. The features TNS, VTYPE, NB, CASE, SPEC, PFORM and their respective values are percolated from the example matched by the morpho-syntactic set of features into the reduced node.

5 Integrated System Architecture

5.1 The ETAP-3–TM hybrid prototype

As has been mentioned, the syntactic parser of ETAP-3 generates, for any sentence processed, a dependency tree structure, which is then sent to the transfer module. However, the parser does not distinguish between new sentences or phrases and those contained in a translation memory. In particular, it does not distinguish between free word combinations and terminological units likely to be present in a TM. If such units are syntactically and/or lexically ambiguous, or if they generate ambiguity when used in a broader context, this ambiguity persists in the parsing phase. Accordingly, the parser does not consider the

Figure 7: Ambiguous Dependency Structures



restrictions that could be used if the term was given one, and only one, particular parse and generates multiple parses, which are processed by the transfer module one after another.

It is therefore not surprising that, rather frequently, the first TL equivalent to be produced by the parser is far from being adequate. In order to optimize the search of equivalents, the parser has been supplemented with a preference mechanism, which ensures that multiword units **present in the dictionary** are processed first (i.e. that they receive an adequate and compact subtree representation).

However, even this improvement does not provide a fully satisfactory solution. To give an example, consider a relatively simple technical term *shelf life expiry date*. Such a term is likely to appear in any TM for a related subject domain (such as e.g. material management or warehousing) but may hardly be expected in an MT system dictionary of any reasonable size. As any other chain of English nouns forming a composite construction, this term can receive several analyses in dependency structures as depicted in Figure 7.

For any sentence containing this term, our parser will first supply a syntactic structure that includes subtree (d) in Figure 7, as the dictionary of ETAP-3 contains the two multiword expressions - (i) *expiry date* and (ii) *shelf life*. Accordingly, the transfer module will yield a Russian translation of the form **A of B**, where A and B are, respectively, (idiomatic) translations of the expressions (i) and (ii). However, such a translation is totally inadequate: since the translations of both *shelf life* 'srok xranenija' (which actually manages without the ideas of 'shelf' or 'life') and *expiry date* 'srok godnosti' are roughly synonymous, the target text is 'srok godnosti sroka xranenija'. Ridiculously,

this means something like 'the duration of validity of the duration of storage'.

In a way, this example demonstrates a precision limit that an MT system cannot exceed on a mass scale without resorting to auxiliary tools like translation memory. What we do in order to allow the resources of a translation memory to be processed in ETAP-3 is the following.

Any example contained in the TM case base is assigned ONE syntactic parse, which must be stored together with the example. This can be done rather easily with the help of the interactive term recognizer, described in 4.1.2.

During translation, a source text is checked against the TM case base. If any of the examples contained there are found, all syntactic links that involve the example concerned are forcefully established prior to regular parsing operation, irrespective of whether the same links would later be obtained or not. All links that contradict those established for the example are overridden, including the links that originate from those words of the example that are not allowed to have daughters.

If we proceed with our illustration *shelf life expiry date* (which in the TM is likely to be translated into Russian as *minimal'nyj srok xranenija*, i.e. minimum duration of storage), it will be assigned one parse (any one of Figure 7a to 7d) will do) and the tag saying that *date* is the top node. No other elements will be tagged as capable of having syntactic daughters. The same parsing procedures, as well as procedures of top node assignment and possible syntactic daughters slots definition, must be applied to translation equivalents of all items contained in the TM case base.

After the syntactic tree of the sentence processed is ready, it is sent to the transfer component, where the equivalent side of the example is substituted for the fragment corresponding to the source side of the example. In the simplest case, the (only) syntactic link coming into the top node of the source example is replaced by a link coming into the top node of the target example, whereas all links originating from the elements of the source example are represented as ones originating from the top node of the target example. (More complicated cases, which require the consideration of links originating from target example elements other than the top node, are being investigated now.) In accordance with the above strategy, if ETAP-3 has to translate a sentence like

(5) *Products with shelf life expiry date close to present date must be withdrawn at once*,

the following sequence of action will be applied:

1 TM case base is consulted and example *shelf life expiry date* is found and

activated.

2 Sentence (5) is morphologically analyzed and sent to the parser.

3 After the first phase of parser operation is finished and all hypothetical syntactic links are chosen and amassed, the result is compared with the activated structure from the TM Case base. All syntactic links of the set generated by the parser that contradict this structure are deleted. Also deleted are all extra lexical and/or grammatical homonyms of words that occur in the example: in our case, these may include *shelf* as sea shelf, *date* as a verb, an exotic fruit, a meeting etc. As a consequence, the set of possible syntactic links is noticeably reduced.

4 Normal parsing procedure is continued.

5 The obtained syntactic structure is sent to a pre-transfer component that replaces the syntactic fragment corresponding to TM case base example with its TL equivalent. The pre-transfer component acts in much the same way as the rules produced by the ETAP-3 interactive term recognizer.

Roughly, the output of the pre-transfer phase for sentence (5) looks as follows:

```
                ------prepos----> -----modif--->
                |               | |            |
(5') Products with minimal'nyj srok godnosti close to
     present date must be withdrawn at once.
```

(In (5'), only links going to and from the substituted TL example are shown).

6 The transfer procedure is continued until the sentence is fully translated.

As shown by the example, we have combined a RBMT system with a TM based on translationally equivalent dependency structures. For this purpose, the TM must be transformed into a linguistically rich instrument, which is able of supplying material for future operations of adaptation. The two components interact so that the strong sides of the RBMT (high recall, high coverage and at least literal translation quality) can be maintained - while the TM component increases the reliability of the system. As both components work with identical structure types, the adaptation is taken over completely by the RBMT component which can operate on structures coming from the TM. Structures which are not handled or not recognized by the TM are taken

over by the RBMT component which guarantees the high recall and coverage. Priority however is given to the TM component which by its recognition or non-recognition of structures controls which parts of the text are translated based on the TM and which parts are translated by rules.

5.2 The CAT2-EDGAR hybrid prototype

5.2.1 Architecture Outline

In the CAT2-EDGAR experiment IAI linked the RBMT system CAT2 dynamically to the EBMT system EDGAR in such a way that EDGAR comes into play after the morphological analysis and before the syntactic analysis performed by CAT2 (during the analysis phase) and, during generation, after the syntactic generation and before the morphological generation. In such an architecture, EDGAR serves for CAT2 as an intelligent multiword and phrase translation front end, whereas CAT2 for EDGAR performs the translation of linguistic structures which are beyond the capabilities of EDGAR.

The two systems implement linguistic theories of varying "richness". Whereas EDGAR makes use of morphological and syntactic information only, CAT2 implements a semantic theory of the languages involved. Due to the simple, example-driven approach, EDGAR is easy to customize and easy to extend to a new domain. On the other hand, CAT2 focuses on semantic principles which underlie the languages involved. CAT2 is thus capable of achieving a high coverage. When EDGAR fails to find an appropriate translation example, CAT2 comes into play and generates a literal translation. If a user prefers a translation different from the literal translation he can simply add a suitable translation example to the case base. Subsequent translations will then use this translation example instead of the literal translation.

For instance, a user might prefer translation 1b to 1a. The only thing he would need to do is to add 1b to the case base, without having to add any relevant semantic pieces of information.

1a. *(concrete building)* $_{NOUN}$ $\leftrightarrow$ *(konkretes Gebäude)* $_{NOUN}$
1b. *(concrete building)* $_{NOUN}$ $\leftrightarrow$ *(Betonbau)* $_{NOUN}$

EDGAR matches the (morphologically analyzed) input text against the CB, whereby the chunks that match an example in the CB are reduced to single nodes and tagged with type information of the matching example. There are three possible outcomes when a sentence is matched against the CB:

1 The entire input text is segmented into "self-sufficient" chunks (e.g. whole sentences). In this case, the (reduced) chunks need not pass through CAT2 at all.

2 No chunks could be found in the input text. In this case, the source text is transmitted to CAT2 unchanged to be processed as usual.

3 The input text matches the CB partially. In this case, both the identified chunks and the remaining unrecognized text elements are transmitted to CAT2. In generation, EDGAR re-generates only those target language parts that it has reduced during the analysis phase.

Since the output of EDGAR is fully determined by the examples of the CB, CAT2 is either simply assisted with the translation of terminology, multiword expressions, or proper names, or else completely circumvented when large parts of the input text are matched in the CB. In other words, our hybrid MT system operates in a dynamic manner, switching translation strategies according to the status of the CB and the text encountered. While a complete match of cases in a sentence converts the system into a TM, in the next sentence the system may return to a purely rule-based treatment, or combine the two approaches.

As for the chunks obtained from EDGAR, they remain "lexically sealed" for CAT2, much in the same way as are multiword expressions of the TM for ETAP-3. This means that CAT2 considers the TUs that come from EDGAR as single nodes, disregarding their internal lexical structures. CAT2 may or may not assign some grammatical features to the target side of the chunks in order to guide adaptation. The lexical content of these TUs remains unchanged and thus does not affect translation reliability. As a side effect, CAT2 is bound to operate faster and in a more robust way, if for no other reason then simply because it has fewer units to handle.

In the English-to-German translation experiments with the hybrid EDGAR-CAT2 MT system we have used reference translation examples of the following types: noun phrases *NOUN*, determiner phrases *DP*, prepositional phrases *PP*, adverbial phrases *ADV*, and entire sentences *S*. The solutions offered for the treatment of feature percolation are based on these types of linguistic data. Future experiments will involve other syntactic types (e.g. subordinate clauses).

5.2.2 Examples of Operation

In order to show how the hybrid EDGAR-CAT2 system translates different phrase types, we make up a sample CB containing the following examples.

1	*(man)* $_{NOUN}$	$\leftrightarrow$	*(Mann)* $_{NOUN}$
2	*(newspaper)* $_{NOUN}$	$\leftrightarrow$	*(Zeitung)* $_{NOUN}$
3	*(a man)* $_{DP}$	$\leftrightarrow$	*(Ein Mann)* $_{DP}$
3g	*(a* $\mathcal{X}_{NOUN}$*)* $_{DP}$	$\leftrightarrow$	*(Ein* $\mathcal{X}_{NOUN}$*)* $_{DP}$
4	*(The newspaper)* $_{DP}$	$\leftrightarrow$	*(Die Zeitung)* $_{DP}$
4g	*(The* $\mathcal{X}_{NOUN}$*)* $_{DP}$	$\leftrightarrow$	*(Der* $\mathcal{X}_{NOUN}$*)* $_{DP}$
5	*(The old man)* $_{DP}$	$\leftrightarrow$	*(Der alte Mann)* $_{DP}$
5g	*(The old* $\mathcal{X}_{NOUN}$*)* $_{DP}$	$\leftrightarrow$	*(Der alte* $\mathcal{X}_{NOUN}$*)* $_{DP}$
6	*(for the man)* $_{PP}$	$\leftrightarrow$	*(für den Mann)* $_{PP}$
6g	*(for* DP*)* $_{PP}$	$\leftrightarrow$	*(für* $\mathcal{X}_{DP}$*)* $_{PP}$
7	*(The old women)* $_{DP}$	$\leftrightarrow$	*(die alten Frauen)* $_{DP}$
8	*(secretary of state)* $_{NOUN}$	$\leftrightarrow$	*(Staatsminister)* $_{NOUN}$
9	*(on the table)* $_{PP}$	$\leftrightarrow$	*(auf dem Tisch)* $_{PP}$
10	*(day after day)* $_{ADV}$	$\leftrightarrow$	*(Tag für Tag)* $_{ADV}$
11	*(The man reads the newspaper every day.)* $_{S}$	$\leftrightarrow$	
	(Der Mann liest jeden Tag die Zeitung.) $_{S}$		

Below, the contribution of the two modules will be shown as follows:
Units which are recognized by EDGAR are underlined as in *the car*; all the remaining units are translated by CAT2. Processing phases written in small capitals (such as CHUNKING are tackled by EDGAR. Phases represented with standard font are handled by CAT2.

Example 1
The sentence *The old man is selling the secretary of state's car.* undergoes the following transformations.

CHUNKING:	*The old man* is selling *the secretary of state's* car .
GENERALIZATION:	$\mathcal{X}^{6}_{DP,NOM;ACC,DEF,SG}$ is selling $\mathcal{Y}^{5g/8}_{DP,GEN,DEF,SG}$ car .
translation:	$\mathcal{X}^{6}_{DP,NOM,DEF,SG}$ verkauft den Pkw $\mathcal{Y}^{5g/8}_{DP,GEN,DEF,SG}$.
REFINEMENT:	*Der alte Mann* verkauft den Pkw *des Staatsministers*.

The chunk *The old man* matches CB example 5 and is reduced into the node $\mathcal{X}^{6}_{DP,NOM;ACC,DEF,SG}$. The chunk *the secretary of state's* is recognized in two successive steps of generalization. During the first step *secretary of state's* is matched with example 8 and is reduced into the node $\mathcal{Y}^{8}_{NOUN,GEN,SG}$. Notice that *state* and *state's* differ only in case. As outlined above (Section 4.3.2), the CASE feature (here, *GEN*) is taken from the source chunk and percolated into the reduction. During the second step, the chunk *the* $\mathcal{X}^{8}_{NOUN,GEN,SG}$ matches the generalized CB example 4b. Since no more reductions can be computed,

the resulting generalization is passed to CAT2 for translation.

CAT2 identifies the subject of the sentence and disambiguates the CASE feature. It parses the $\mathcal{Y}$ node as a pre-nominal modifier, which can be realized in German as a post-nominal genitive, and identifies the progressive tense *is selling*, translating it into the German present tense *verkauft*. The resulting structure is then passed back to EDGAR for specification and refinement of the reduced nodes.

Example 2
The sentence *The old men sell cars* is processed as follows:

CHUNKING:	*The old men* sell cars .
GENERALIZATION:	$\mathcal{X}^{5-7}_{DP,NOM;ACC,DEF,PLU}$ sell cars .
translation:	$\mathcal{X}^{5-7}_{DP,NOM,DEF,PLU}$ verkaufen Autos.
REFINEMENT:	*Die alten Männer* verkaufen Autos.

In contrast to the previous example, the chunk *The old men* matches CB example 5 lexically (because *men* and *man* have different number values) and CB example 7 with respect to morpho-syntactic features. For this reason, both examples 5 and 7 are used as reference translations. Note that both indices are stored in the reduced node. CAT2 translates the remaining items $\mathcal{X}^{5-7}_{DP,NOM;ACC,DEF,PLU}$ *sell cars.* and dictates the case of the nominal chunk *NOM*. EDGAR then merges the lexical and the morphological features of the TL reference translations and refines the merged chunk in accordance with the dictated case.

Example 3
The sentence *The old woman is waiting for the old man* is translated as follows:

CHUNKING :	*The old woman* is waiting *for the old man* .
GENERALIZATION:	$\mathcal{X}^{7-5}_{DP,NOM;ACC,DEF,SG}$ is waiting $\mathcal{Y}^{6g/6}_{PP,NOM;ACC,DEF,SG}$
translation:	$\mathcal{X}^{7-5}_{DP,NOM,DEF,SG}$ wartet $\mathcal{Y}^{6g/6}_{PP,ACC,DEF,SG,auf}$
REFINEMENT:	*Die alte Frau* wartet *auf den alten Mann*.

The chunk *The old woman* is found in a way similar to *The old men* in the previous example, i.e. by combining the same two examples. This time,

however, the morpho-syntactic features are matched with CB example 5 and the lexical features with CB example 7. *For the old man* is chunked in two generalization steps. First, *the old man* is matched with example 5 which yields the reduction $\mathcal{X}^{5}_{DP,NOM;ACC,DEF,SG}$. Secondly, the chunk matches the CB example 6g. The sentence, thus reduced to four nodes, is passed over to CAT2.

CAT2 translates the progressive *is waiting* into simple German Present tense as in the previous example. Further, the node $\mathcal{Y}^{\mathcal{Y}//}_{\mathcal{PP},\mathcal{NOM};\mathcal{ACC},\mathcal{DEF},\mathcal{SG}}$, which represents the prepositional phrase *for the old man*, is assigned the semantic role *THEME* as a valency argument of *wait*. The German translation requires for this valency the preposition *auf* and the accusative case. When refining the TL side of the example 6g *(für* $\mathcal{X}_{DP}$*)* $_{PP}$, EDGAR replaces the preposition *für* with the preposition *auf* based on the information provided from CAT2. In this way the correct preposition can be produced if the prepositional phrase is a verbal argument despite the fact that CB example 6g represents a modifier PP.

Example 4
The sentence *The man put the book on the table.* is treated as follows:

CHUNKING:	*The man* put the book *on the table* .
GENERALIZATION:	$\mathcal{X}^{4g/1}_{DP,NOM;ACC,DEF,SG}$ puts the cup $\mathcal{Y}^{9}_{PP,NOM;ACC,DEF,SG}$.
translation:	$\mathcal{X}^{4g/1}_{DP,NOM;ACC,DEF,SG}$ stellt die Tasse $\mathcal{Y}^{9}_{PP,ACC,DEF,SG}$.
REFINEMENT:	*Der Mann* stellt die Tasse *auf den Tisch*.

The chunk *The man* is reduced to one node in two generalization steps: first, *man* is reduced based on CB example 1 and then CD example 4g is used to match the entire chunk. *On the table* has a complete match in the CB example 9. The reduced sentence is then translated in CAT2 where the node $\mathcal{Y}$ receives the semantic role *DIRECTION* and, given that the preposition *auf* is already known, can be assigned an unambiguous case *ACC*. In contrast to the role *THEME*, no specific preposition is dictated by CAT2 for *DIRECTION* (as well as other roles such as *LOCATION* or *PROVENANCE*). Accordingly, the default preposition is taken from the CB.

Example 5
The sentence *Day after day the man buys a newspaper* is processed as follows:

CHUNKING:	*Day after day* *the man* buys *a newspaper*.
GENERALIZATION:	$\mathcal{X}^{10}_{ADV}$ $\mathcal{Y}^{4g/1}_{DP,NOM;ACC,DEF,SG}$ buys $\mathcal{Z}^{3g/2}_{DP,NOM;ACC,INDEF,SG}$.
translation:	$\mathcal{X}^{10}_{ADV}$ $\mathcal{Y}^{4g/1}_{DP,NOM,DEF,SG}$ buys $\mathcal{Z}^{3g/2}_{DP,ACC,INDEF,SG}$.
REFINEMENT:	*Tag für Tag* kauft *der Mann* *eine Zeitung*.

Day after day is recognized as an adverbial phrase *ADV* and *the man* and *a newspaper* are recognized as determiner phrases *DP* as they match the examples 10, 4g/1 and 3g/2, respectively. CAT2 translates the unreduced item in the sentence and generates an appropriate word order of the elements in the target language.

Example 6
The sentence *The man reads the newspaper every day* is treated very simply.

CHUNKING:	*The man reads the newspaper every day .*
GENERALIZATION:	$\mathcal{X}^{11}_{S}$
translation:	$\mathcal{X}^{11}_{S}$
REFINEMENT:	*Der Mann liest jeden Tag die Zeitung.*

There is a complete match for the whole sentence in the CB example 11, so no unreduced part is left for CAT2. The entire translation is taken from the CB base.

5.2.3 Discussion

A prerequisite for an effective integration of EDGAR and CAT2 is an appropriate level of adaptability of both systems. As shown in translation examples, in the refinement step EDGAR performs an adaptation on agreement features in determiner and prepositional phrases and replaces prepositions in a prepositional phrase according to the values dictated by CAT2. Such a minimal adaptability is required if sub-sentential chunks (such as determiner or noun phrases) are used sometimes in the subject position and sometimes in the object position or else, as in examples 3 and 4, implement a verbal valency or act as modifiers. In our integration scenario, however, we avoid to consider lexical forms other than prepositions in prepositional phrases as adaptable to the context.

The proposed technique of using linguistic data in EDGAR and CAT2 for translation can only be viewed as a tentative approach, because the experiments

carried out so far a) have not been extensive enough to make any concrete statements about the behavior of the system when huge amounts of data are involved and b) have been limited to certain phrase types so that we cannot make general statements about the source of their linguistic information and their subsequent treatment (such as percolation). In this connection, a couple of essential difficulties will have to be dealt with:

- **Transfer of category**

  In order to achieve a reliable translation quality it might sometimes be required to change the part of speech when transferring a TU into the TL. In the same way CAT2 dictates the case of a noun phrase or a specific prepositional form if a prepositional phrase is an argument of a verb, it might wish to dictate the part of speech for refinement, e.g. to translate *wait for the old man* into the DP *erwartet den Mann*. This might cause problems in EDGAR, due to the fact that EDGAR identifies its possible target structures prior to the intervention of CAT2.

- **Incorrect chunking**

  A sentence wrongly segmented by EDGAR cannot be treated by CAT2 correctly. If, for example, EDGAR recognizes the chunk *interesting story* in a sentence like *John read a very interesting story* and reduces it to a noun, the sequence *a very* $\mathcal{X}_{NOUN}$ cannot be reasonably analyzed by CAT2. Equally, if *la femme* in *la femme heureuse* is reduced to a *DP*, the sequence $\mathcal{X}_{DP}$ *heureuse* cannot be analyzed. A solution to this problem could consist in checking the reduction in the given for completeness and undo it if the check fails.

  It must also be stated that only those chunks are good candidates to be taken over by EDGAR that act independently at all levels of representation of the RBMT component and do not need more information than that supplied by EDGAR. If we look back at the CAT2 translation example of section 4.2, the word *Angst* in (1) is not a good candidate, because the RBMT component requires more information than could be supplied by EDGAR, e.g.. the reference of *Angst* to *haben* and *gross*. With *Angst* chunked by EDGAR, the translation would look differently. How EDGAR should decide exactly which chunks it must skip, is a problem that still awaits a proper solution. Nevertheless we assume that chunks of size one are better taken over by the RBMT system if no user-oriented translation is required.

6 Summary and Outlook

The paper describes the development and tentative implementations of a new MT scenario where a CBMT system works in combination with a conventional RBMT system. The main idea of the CBMT module is to introduce into the RBMT paradigm, which is based on sophisticated language models, a significant share of human translation experience accumulated in Translation Memories and Term Banks, which are, after all, relatively simple but very large and accurate collections of bilingual texts. These examples have to be enriched linguistically in order to be able to fully participate in the translation process of the RBMT system.

A shallow morphological enrichment chosen in EDGAR represents the minimal requirement, which, however, has the advantage of being performed fully automatically. A full syntactic representation of the translation examples in the TM–ETAP-3 prototype proves to be more satisfactory; it can, however, only be produced semi-automatically by trained linguists. In both experiments a dynamic architecture has been created which should outperform earlier attempts at static linkage.

The mechanism that monitors this dynamic interaction is the CBMT component. If this component recognizes a piece of the source text as a TU, it handles this piece itself; if not, it sends the piece to the RBMT component. This architecture ensures an optimal interaction of the two components where the full reliability of the TUs covered by the CBMT component is enhanced by the mutual adaptation of these TUs by the RBMT system, a high recall for TUs not covered by the CBMT and a high coverage even if the text is beyond the scope of the CBMT component.

Although the application's architecture described here has already been implemented, the evaluation and refinement of the components are still under discussion. In the near future, we may expect the following results.

- The efficiency of a CBMT module integrated into RBMT systems will be assessed more precisely. It will become clear to what extent the performance and translation quality can be improved by the linkage.
- It will become clear what types of word combinations or chunks should be introduced into a CBMT system in order to ensure a sizeable positive effect on the translation quality and performance.
- Nevertheless, it may prove that the communication between the components is still too rigid for an optimal interaction of the components. Instead, the components should negotiate the structures handled between

them, e.g. ask for differently chunked input or refuse a rule-based translation in which uncovered chunks are required.

- Finally, it may become necessary to merge the pieces of information coming from different components. In our examples, it may prove expedient to merge the representation of *Angst* coming from EDGAR and that coming from CAT2. In such a way, *Angst* would receive the user-oriented translation in a trivial context, but would resort to a part of speech transformation if required by a more intricate context.

References

[ABI+89] Jurij D. Apresjan, Igor M. Boguslavskij, Leonid L. Iomdin, Alexandre V. Lazurskij, Vladimir Z. Sannikov, and Leonid L. Tsinman. *Lingvističeskoe obespečenie sistemy ETAP-2*. Izdatel'stvo "Nauka", Moskva, 1989.

[ABI+92] Jurij D. Apresjan, Igor M. Boguslavskij, Leonid L. Iomdin, Alexandre V. Lazurskij, Vladimir Z. Sannikov, and Leonid L. Tsinman. ETAP-2: The linguistics of a Machine Translation system. *Meta*, 37(1):97–112, 1992.

[ABI+93] Jurij D. Apresjan, Igor M. Boguslavskij, Leonid L. Iomdin, Alexandre V. Lazurskij, Vladimir Z. Sannikov, and Leonid L. Tsinman. Système de traduction automatique ETAP. In P.Bouillon and A.Clas, editors, *La Traductique*. Les Presses de l'Université de Montréal, Montréal, 1993.

[Bro96] D. Ralf Brown. Example-Based Machine Translation in the Pangloss System. In *COLING-96*, 1996.

[Car98a] Michael Carl. A constructivist approach to Machine Translation. In *Proceedings of NeMLaP3/CoNLL98*, pages 247–256, Sydney, 1998.

[Car98b] Michael Carl. On the meaning preservation capacities of machine translation. In *Proceedings of the ESSLLI '98 Machine Translation Workshop*, 1998.

[LL97] L.L.Cinman and L.L.Iomdin. Lexical functions and Machine Translation. In *Proceedings of the Dialogue'97 International seminar in Computational Linguistics and Applications*, pages 291–297, Moscow, 1997. (In Russian. English summary).

[Mel74] Igor Aleksandrovič Mel'čuk. *Opyt teorii lingvističeskix modelej Smysl⇔Tekst. Semantika, sintaksis.* Izdatel'stvo "Nauka", Moskva, 1974.

[Mel95] Igor A. Mel'čuk. *Russkij Jazyk v modeli Smysl ⇔ Tekst.* Shkola "jazyki russkoj kul'tury" - Wiener Slavistischer Almanach, Moscow- Vienna, 1995. (In Russian, English, French and German.).

[NÖ97] Rita Nübel. End-to-end evaluation in verbmobil i. In *MT-Summit*, San Diego, 1997.

[Sha94] Randall Sharp. CAT2 Reference Manual Version 3.6. IAI WP n.27, IAI, Institut der Gesellschaft zur Förderung der angewandten Informationsforschung e.V. an der Universität des Saarlandes, 1994. URL: http://www.iai.uni-sb.de/en/cat-docs.html.

[SS95] Randall Sharp and Oliver Streiter. Applications in Multilingual Machine Translation. In *Proceedings of The Third International Conference and Exhibition on Practical Applications of Prolog, Paris, 4th-7th April*, 1995. URL: http://www.iai.uni-sb.de/en/cat-docs.html.

[Str96] Oliver Streiter. *Linguistic Modeling for Multilingual Machine Translation.* Informatik. Shaker Verlag, Aachen, 1996.