8

A New Hybrid Approach Enabling MT for Languages with Little Resources

PETER DIRIX, VINCENT VANDEGHINSTE AND INEKE SCHUURMAN

Centre for Computational Linguistics, K.U.Leuven

Abstract

In this paper, we combine techniques from rule-based and corpus-based MT in a hybrid approach. We only use a dictionary, basic analytical resources and a monolingual target-language corpus in order to enable the construction of an MT system for lesser-resourced languages. Statistical and example-based systems usually do not involve a lot of linguistic notions. Cutting up sentences in linguistically sound subunits improves the quality of the translation. Demarcating clauses, verb groups, noun phrases, and prepositional phrases restricts the number of possible translations and hence also the search space. The sentence chunks are translated using a dictionary and a limited set of mapping rules. By bottom-up matching the different translated items and higher-level structure with the database information, one or more plausible translated sentences are constructed. A search engine ranks them using the frequencies of occurence and the matching accuracy in the target-language corpus.

Proceedings of the Sixteenth Computational Linguistics in the Netherlands.
Edited by: Khalil Sima'an, Maarten de Rijke, Remko Scha and Rob van Son.

8.1 Introduction

Since its introduction in the 1950s, machine translation (MT) has been the holy grail of computational linguistics. The first word-by-word systems were soon succeeded by rule-based systems. Despite their numerous limitations, these systems are nowadays still the most used. Their main bottleneck is the almost infinite number of rules you have to construct to get a good translation. Furthermore, the processing time was a problem for a very long period until computers got fast enough. You also need advanced resources such as syntantic (and maybe semantic) parsers.

In the 1980s new techniques, mainly borrowed from speech recognition, gave birth to statistical machine translation (SMT). Twenty years later, there are not a lot of commercial systems available yet, although Google announced to launch an SMT system in 2007. The main disadvantages of SMT are the need of a parallel text corpus and data sparsity: the parallel corpus used is in fact never large enough! Such parallel text corpora (or *bitexts*) are hardly ever available for most language pairs and terminological domains, especially for general language. The same disadvantages apply to example-based machine translation (EBMT).[29]

The METIS-II system[30] is under development at a consortium formed by the Institute for Language and Speech Processing (ILSP) in Athens, the Universitat Pompeu Fabra in Barcelona, the Institute of Applied Information Sciences (IAI) in Saarbrücken and the Centre for Computational Linguistics (CCL) of the K.U.Leuven. This system makes use of a target-language corpus only, and therefore by-passes the bitext problem. On the other hand, it needs a bilingual dictionary, a limited set of translation rules and a basic (shallow) source-language analysis.

The rationale for this approach is that for many, especially smaller, EU languages little digital resources are available (cf. the BLARK initiative[31]). Parallel corpora for language pairs of which at least one language does belong to this set of smaller languages are very scarce (even when the other language is English).

The fact that there are huge amounts of documents waiting to be translated, involving all kinds of language pairs for which only limited resources are available (e.g. no full parser, no large enough parallel corpus) made us investigate whether a machine translation technique can be developed for use under these conditions. So, although for the languages involved in the METIS-II project these more advanced tools are available[32], we refrain from using them in order to mimic the situation lots of *low-resource* languages are faced with. Hence, we are not claiming that our approach is better than the ones generally used in SMT and EBMT, when a large (huge) parallel corpus for a specific subdomain and a

[29] For a description of recent techniques, see Carl and Way (2003)

[30] Supported by the 6th European Framework Programme, FP6-IST-003768. It is the successor of the METIS-I project (Dologlou, Markantonatou, Tambouratzis, Yannoutsou, Fourla and Ioannou 2003), which confirmed the feasibility of this approach.

[31] For Dutch, a report was drawn up by Daelemans and Strik (2002).

[32] But note that even for the pair Dutch-English a large parallel corpus in the general domain does not yet exist!

specific language pair is available. The only *advanced* resource we are using is a bilingual dictionary, consisting of lemmas and their part of speech in both languages.[33]

The introduction of mapping rules could resolve some linguistic issues that arise with SMT and EBMT techniques. The combination of rule-based and statistical/example-based methods leads to a *hybrid* system, which seems the way to go (Thurmair 2005), to avoid the intrinsic obstacles of both the statistical and rule-based methods. The system uses a basic group of resources and a very limited set of rules, and uses the target-language corpus as the main resource for translation candidate selection and word order.

The use of this methodology enables us to construct an MT system for low-resource languages, on the condition that they possess a certain minimal set of linguistic tools, including a target-language corpus.

8.2 The METIS-II System

This general-domain MT system is being constructed for four language pairs: from Dutch, Modern Greek, German, and Spanish to English (Vandeghinste, Schuurman, Carl, Markantonatou and Badia 2006). Nevertheless, the system is designed in such a way that most parts are language-independent, whereas language-dependent modules can be plugged in when needed. Not all language pairs use the same resources[34], and this shows that the system can be used with a variety of resources, depending on the availability for the languages at hand, although this will have an effect on the translation accuracy. Of course, every partner institution is using its own tools for dealing with the source-language input.

At this stage, all partners developed their own expanders and search engines, albeit using the same ideas and paradigms. In addition, all are using the tagged and lemmatised versions of the target-language corpus, in this case the British National Corpus (BNC).

We will start with a general overview of the three stages in the translation process, called the language models. We will also give a short introduction to the general scoring mechanism. Next, each of the language models will be discussed in detail, describing the different modules that form the system and the way they score the building blocks of the translated sentence.

The different modules are integrated in an NLP engine that follows the flow presented in figure 16.

8.2.1 Three language models

The translation process is divided in three stages (see figure 16).

[33] If necessary, such a dictionary can be obtained by extending a basic vocabulary using a comparable corpus (Sadat, Déjean and Gaussier 2002), which is much easier to come by than a parallel corpus. Another possibility would be to use such a comparable corpus for translation purposes (in addition to a parallel corpus, or maybe even instead of such a corpus).

[34] Especially the Spanish team is trying to develop a system only using statistical means.

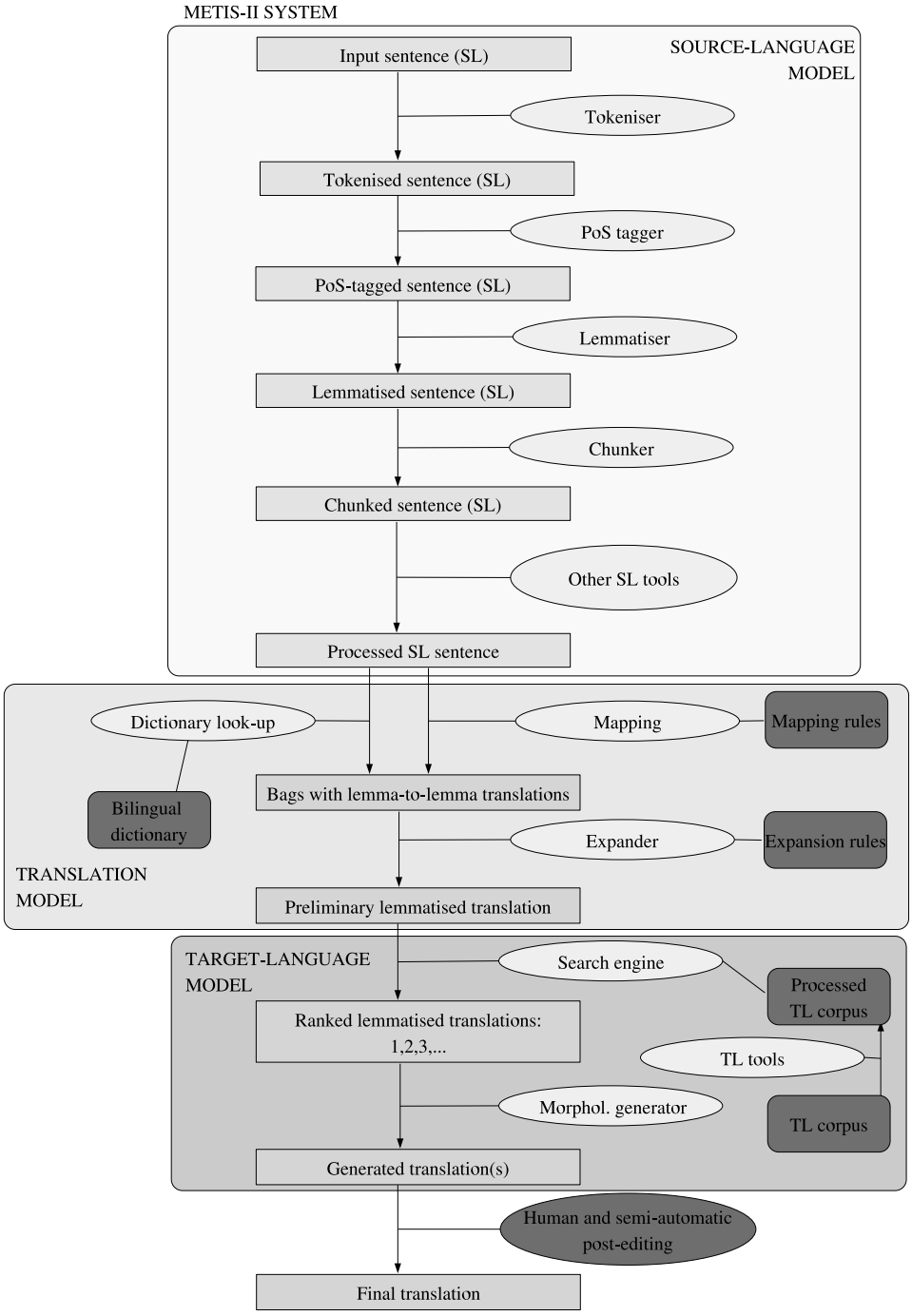


FIGURE 16 General data flow

First, a shallow *source-language model* (SLM) is constructed, using tools that analyse the input sentence. This sentence is tokenised, part-of-speech (PoS)-tagged, lemmatised and chunked into phrases. We also use additional tools like a subclause detector, and intend to use a subject detector in the near future.

Subsequently, the sentence needs to be translated. The *translation model* (TM) consists of a bilingual dictionary, a limited set of tag-mapping rules and grammatical rules to map the source structure to the target language. It enables the transition of the source-language lemmata to the target language and allows for reorganisation of the chunks in the sentence. Since modules in the SLM could generate various possible translations and structures, the system produces a list of possible translations.

The search engine compares this list with the *target-language model* (TLM), based on a target-language corpus, and chooses the (*n*) best translation(s). The fact that we are translating lemmata instead of tokens, simplifies the search by reducing the sparsity, but forces us to use a morphological generator.

Finally, this preliminary translation should be offered to a human translator for post-editing. This way preferred translations can be stored as well. A future version of the system should allow to use these preferences when scoring alternative translations.

8.2.2 Scoring mechanism

At all steps in the processing chain, every node in every parse tree receives a weight. The mechanism is designed as such that the joint weight is in principle one, except in the search engine, where the weight represents the matching accuracy with the corpus. If the number of possibilities is higher than a parametrisable number N_{max}, the *beam* is cut off before the first element with a lower weight than N_{max}. In cases of ambiguity where the module is not assigning any weight itself, the weight is divided proportionally over the different alternatives.

8.2.3 The source-language model (SLM)

For each of the relevant languages, the SLM is constructed using language-specific tools. The only condition is that the output format is compatible with the search engine. In this paper, we describe the tools used for MT from Dutch to English.

Basic analysing tools

The *tokeniser* is a module that identifies the separate words and punctuation marks. Every punctuation is considered a separate token. The input for this module is a source-language sentence. The PoS tagger requires the output format to be separate tokens on a different line.

The *tagger* assigns part of speech categories to the Dutch tokens, using the *CGN*-

tagset[35]. This tagset is based on the morphosyntactic forms of Dutch. We use the TnT tagger (Brants 2000) trained on the CGN[36], with the option that not only the best (most probable) tag, but also the alternative tags with a lower probability, are used. These are combined into several source-language analysis alternatives, with as their weights the products of the tag probabilities of the elements. These alternatives go through the rest of the translation process, as they can result in different lemmatisation and chunking, and, of course, in different translations.

The Dutch *lemmatiser* is based on the PoS tags assigned by the tagger. It uses the CGN lexicon with more than 300 000 forms (Piepenbrock 2002) to find the correct lemma for a token. For certain tokens, the lemmatisation process generates more than one token. Using the tags as extra information reduces the ambiguity substantially. The Dutch word *was* e.g. could be a noun meaning 'wax' or 'laundry'[37], but also the simple past singular tense of the verb meaning 'to be' or the simple present singular of the verb meaning 'to wash'. The PoS tag allows the lemmatiser to disambiguate the lemma.

Next, the tokenised, tagged and lemmatised sentence is *chunked* by ShaRPa 2.0[38]. ShaRPa is a rule-based shallow parser which uses a set of context-free non-recursive grammars to identify chunks. The NPs, PPs, and verb groups are identified. The heads of the phrases are marked. ShaRPa returns only one result per input, as it is a purely rule-based tool. The weight of the returned result is therefore the same as the weight of the input.

Other analysing tools

We implemented some tools to identify the subjects and different kinds of subclauses[39] in a sentence. Even if the detection process is not perfect, it can resolve some word order problems, since the number of possible permutations is limited.

Dutch has an SVO order in main clauses and an SOV order in subclauses. This means that the subject of a sentence (and if applicable, of the subclause) is usually the first NP in the sentence or clause, unless the first NP is a temporal or spatial constituent. In this case, the algorithm chooses the next normal NP as subject. The borders of the subclauses are identified using subordinating conjunctions and the position of the verb (which is in Dutch usually at the end of the subclause). Relative clauses are identified using the relative pronouns which introduce them and the verb at the end of the clause. Initially, we also identified *om te* + infinitive constructions in Dutch, but since in the case of English, it is very difficult to delimit the corresponding infinitival phrases (because the Dutch trigger word *om* is not translated), we are not using this for the time being. Since these modules are rule-based and only give one result per input, the weights do not change.

[35] Tag set developed for the Spoken Dutch Corpus (CGN) (Van Eynde 2004).

[36] When the data from the D-CoI project becomes available, we will use the D-CoI tag set and train the tagger on the D-CoI corpus (Van den Bosch, Schuurman and Vandeghinste 2006).

[37] Two homonymous nouns with a different gender.

[38] An evaluation for Dutch can be found in Vandeghinste and Tjong Kim Sang (2004) and Vandeghinste (2005).

[39] An evaluation for Dutch can be found in Vandeghinste and Pan (2004).

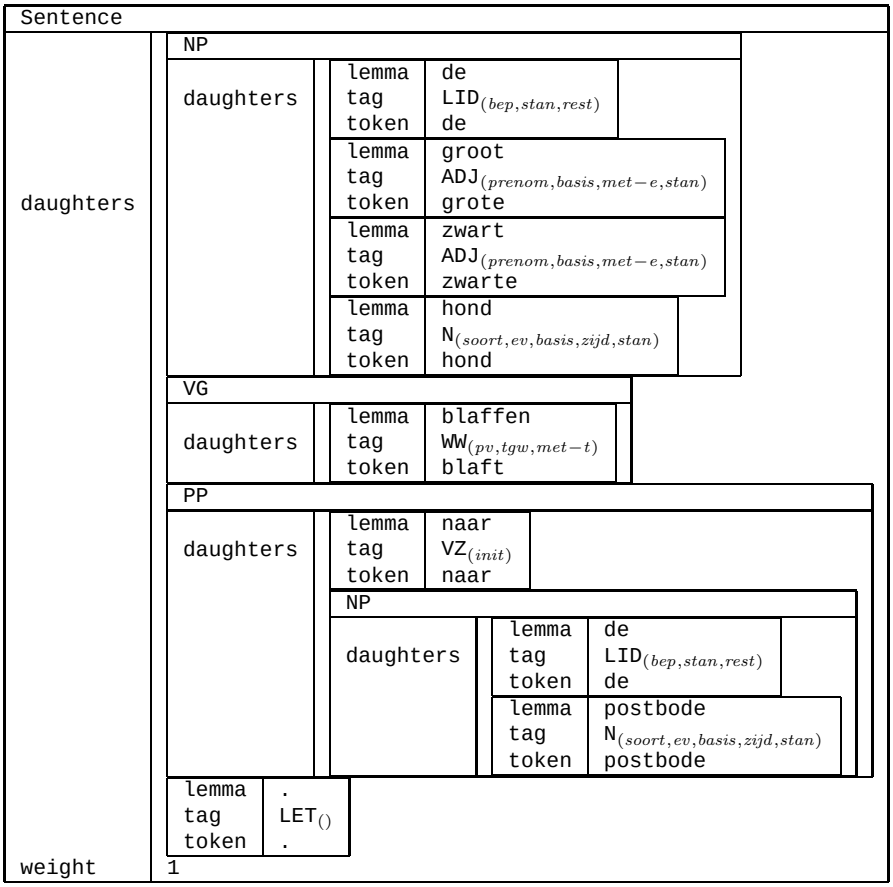
Example

de grote zwarte hond blaft naar de postbode.

⇓

SOURCE-LANGUAGE ANALYSIS

⇓



8.2.4 The translation model (TM)

Dictionary search and tag mapping

The Dutch-English dictionary was constructed using the free Internet dictionary Ergane and the Dutch EuroWordNet (Dirix 2002a). At this moment, there are about 110 000 lemma-to-lemma translations and a few hundred fixed expressions. The dictionary also contains a set of separable verbs, verbs with fixed prepositions and multiword expressions, as is shown in table 6. In these special cases, the right-hand side also contains the appro-

priate chunking of the English expressions. The dictionary format leaves the possibility to generalise categories and introduce extra words between the lemmas of the expression. We are currently correcting and extending the dictionary by hand.

TABLE 6 Examples of different types of dictionary entries

SL-lemma	SL-tag	TL-lemma	TL-tag
eten	WW	eat	VV?[40]
weggaan	WW	go#away[41]	VV?#AV0
wachten∼op[42]	WW∼VZ	wait#for	VV?#PRP
de#morgen	$\text{LID}_{(gen)}$#$\text{N}_{(gen)}$[43]	in#the#morning	PP[in!#the#morning][44]
graag	BW	like∼to# <VVI>[45]	VV?!∼InP[TO0 #VVI]

The CGN tag set is based on morphosyntactic properties of the Dutch language. It has to be mapped to the CLAWS5 tag set, which is constructed more functionally (Dirix 2002b), and which is used to tag the BNC. Over 300 CGN tags have to be mapped to about 70 CLAWS5 tags. In general, there is a many-to-one relation between the Dutch and English tags, but there are some cases where one Dutch tag has to be mapped to more than one English tag.

Example (continued)

The dictionary entries for the words in our example sentence can be found in table 7, whereas the tag mapping rules can be found in table 8. The example sentence was introduced in section 8.2.3.

Expansion

There are often differences in word order between two languages. Various words are inserted or deleted in translation. These differences could force the MT system to introduce additional or modified translations into the generated list of possible translations. This is the role of the *expander*.

[40]The VV? tag is the tag we use for a lemma. The question mark is an underspecification of more specific features which contain tense and number.

[41]The # sign is used to indicate consecutive separate tokens.

[42]The ∼ sign is used to indicate separate tokens which are not necessarily consecutive.

[43]The use of features to restrict the translation of a lemma to certain circumstances is allowed.

[44]When the TL-lemma is a chunk of a different type than the SL, its type needs to be indicated, as well as its head (using the '!')

[45]The usage of <VVI> indicates that, together with the information in the TL-tag column, an expander rule needs to be triggered, that places the original main verb in the <VVI> slot, and that transfers the feature information from that main verb to the feature information of *like*.

[46]In this case, the translation `grown up` is considered as one token, which contains a space. What we consider as one token depends on the decisions taken in the target-language corpus, in our case the BNC.

TABLE 7 Dictionary entries for the example sentence

SL lemma	SL tag	TL lemma	TL tag	SL lemma	SL tag	TL lemma	TL tag
de	LID	the	AT0	blaffen	WW	bark	VV?
groot	ADJ	big	AJ?	naar	VZ	according_to	PRP
		great	AJ?			at	PRP
		grown_up[46]	AJ?			to	PRP
		large	AJ?			toward	PRP
		major	AJ?			towards	PRP
		tall	AJ?				
		in#size	PRP#NN?				
zwart	ADJ	black	AJ?	postbode	N	postman	NN?
		gloomy	AJ?			mailman	NN?
hond	N	dog	NN?				

TABLE 8 Tag mapping for the tags of the tokens in the example

SL-tag	TL-tag
LID()	AT0
ADJ(prenom,basis)	AJ0
N(soort,ev,stan)	NN0|NN1
WW(pv,tgw,met-t)	VBB|VDB|VDZ|VHB|VHZ|VM0|VVB|VVZ|VDB+VVI
VZ()	PRF|PRP|TO0

The list of possible translations can be expanded in two different ways. The first expansion is based on the target-language corpus in order to cover the word order transitions between source and target language. The fact that the normal word order in English is adjective-noun (as opposed to noun-adjective in most Romance languages) could be derived from an English text corpus. In this case, the source-language word order has no importance for the target language.

There are also a number of issues that are source-language-dependent and hence difficult to correct when only using a target-language corpus. These modifications can be modelled with a limited set of mapping rules. An example for this case is the *do*-insertion. In English, the verb *to do* has to be inserted in almost all interrogative sentences and other cases with inversion or emphasis. Such an approach is not feasible for constructions like *ik zwem graag*, where the whole sentence structure is changed. In this case we opted for adding an entry in the bilingual lexicon with a complex lemma '*graag* + verb' in the lexicon, translated as '*like to* + verb(infinitive)' (cfr. table 6).

We use these two types of expansion in order to extend the list of possible translations that will be ranked by the search engine. We consider the input of the expander as a structured bag of bags, representing the structure of the sentence after all the source-to-target-language mapping has been applied. We want to convert this structured bag into a sentence, by resolving each subbag by searching for it in the target-language corpus (depth-first). In fact, we try to find a matching phrase that consists of all the elements of the bag. Depending on how well the corpus phrases match the bag elements, a score is

calculated, resulted in a ranking of permutations, which get a final score from the search engine.

In the CCL system, the expander is currently dealing with the following list of phenomena:

1. The different parts of verb clusters are put together in one bag. In Dutch, the different parts of compound tenses can be separated by direct and indirect object, preposional phrases and even whole subclauses. The past participles and their auxiliaries are put into one bag in order to retrieve the corresponding BNC bags from the target-language corpus.
2. The literal translation of *om* in the *om te* + infinitive construction is deleted, since it remains untranslated. Again, the word *om* could be separated from the remainder of the infinitival phrase by several constituents.
3. In Dutch, the usual form of the active compound tenses is formed with the appropriate tenses of the verb *hebben* and the past participle. However, some intransitive verbs (esp. verbs of motion) are using the verb *zijn* as auxiliary in these tenses. For transitive verbs, *zijn* is used to form the passive voice of the aforesaid compound tenses. Since in English the combination *to be* and past participle is used for the translation of the Dutch '*worden* + past participle', we rewrite the literal translations '*to be* + past participle' to '*to have* + past participle' and '*to have been* + past participle'. In order not to confuse these with the passive of the non-compound tenses, we only introduce *get* and *become* as translations of *worden*. After the former rule fired, we substitute these verbs, if they are followed by a past participle, for the appropriate form of *to be*.
4. The expander is assigning the correct tags in order to translate properly the combination of a verb followed by the adverb *graag* into *to like to*, followed by a verb. We do this, using the dictionary information[47] and the fact that the tense of the original Dutch verb has to be mapped on the tense of *to like*, while the translation of the original verb gets an infinitive tag. The word order is also switched to get correct English.

8.2.5 The target-language model (TLM)

The consortium chose the British National Corpus (BNC) as target-language corpus. The BNC is processed analogous to the source-language input sentences: it is tokenised, PoS-tagged with the CLAWS5 tag set, lemmatised and chunked. The lemmatiser used is described in Carl, Schmidt and Schütz (2005). The corpus was chunked using ShaRPa 2.0 with an English rule set. The NPs, PPs and verb groups are identified. The head of each phrase, the sentence subject, and if applicable, the subclauses are also marked.

[47] See table 6.

The search engine

The search engine is the nucleus of the METIS-II system. The four project partners have experimented with different types of engines. The CCL chose a bottum-up approach, as described in Dirix, Vandeghinste and Schuurman (2005) and Vandeghinste, Dirix and Schuurman (2005), and which is explained in detail in this section. The ILSP group has applied the same method in a top-down approach (Markantonatou, Sofianopoulos, Spilioti, Tambouratzis, Vassiliou, Yannoutsou and Ioannou 2005). The IAI tried the *Shake & Bake* method to select BNC constituents (Carl et al. 2005). The Spanish group finally used an *n*-gram approach (Badia, Boleda, Malero and Oliver 2005).

The search engine takes a *bag* as input. This bag can represent a chunk, a clause, or a sentence. The *elements* of a bag can be considered to be the daughters of the chunk, clause, or sentence the bag represents, but the order in which these elements have to appear in the target language has to be determined by matching the bag with the corpus.

For a given bag, we look in the corpus for a chunk, clause, or sentence (dependent on the bag level) that matches as many of the bag elements as possible. A bag element is matching a corpus element when the lemma (or lemma of the head of the constituent) matches. The accuracy of matching is quantified as follows:

$$a_i = \frac{m_i}{n_i + p_i},$$

where m_i is the number of matching bag elements, n_i is the total number of bag elements, and p_i is the number of elements in the corpus chunk which are not in the bag (i.e. the number of insertions). When $m_i < n_i - 4$, the bag is not retained as a possible solution, because the number of insertions is too big to trust the outcome.

Not every bag alternative matches with the same accuracy, so some alternatives are preferred over other alternatives, leading to translation candidate selection when a certain combination of words occurs in the corpus.

Apart from this matching accuracy, we also take into account the relative frequency of the corpus chunk with respect to the total frequency of all corpus chunks in which the same number of elements match, as in this formula:

$$g_j = a_j . \sqrt{\frac{f_j}{\sum_{k=1}^{q} f_k}},$$

where $\frac{f_j}{\sum_{k=1}^{q} f_k}$ is the relative frequency of the corpus chunk with respect to the total frequency of all corpus chunks in which n_i elements match, with k iterating over these elements. We take the square root of the relative frequency to make this factor less strong. The new weight for the bag i matching a specific chunk j is

$$w_{new,i} = w_{previous,i} . g_j .$$

Once a lower level bag is solved and results in a number of translation candidates for that chunk, the head of that chunk is used at the next level, when looking for matching bag elements, and so on, until we reach the sentence level.

The corpus is indexed on the heads of chunks, so when we want to translate a chunk with a given head that is not in the corpus, we switch to *template* matching, where the same procedure is applied, but without looking at specific lemmas. Only the PoS-tags are used for matching in this case. This enables us to determine the correct word order, but is insufficient for solving the problem of different translation candidates.

Example

TABLE 9 An example of bag matching at the NP level

Bag Elements				n_i	f_i	a_i	w_{new}	result
the	large	black	dog	4	1	1.00	0.71	the large black dog
the	big	black	dog	4	1	0.67	0.47	the big black dog
the	big	gloomy	dog	3	5	0.75	0.37	the big gloomy dog
the	great	black	dog	3	2	0.75	0.23	the great black dog
					2	0.43	0.13	the black great dog
					1	0.27	0.06	black dog the great
the	great	gloomy	dog	3	1	0.75	0.16	the great gloomy dog
					1	0.43	0.09	the gloomy great dog
the	large	gloomy	dog	3	1	0.75	0.16	the large gloomy dog
					1	0.43	0.09	the large dog gloomy
...								

As shown in table 9, the bag with the four elements *the, large, black, dog* matches perfectly with a chunk from the corpus: all four elements from the bag match with the corpus (n_i) and all elements from the corpus chunk are matched with bag elements. This results in $a_i = 1$. There is another bag for which four elements match with the corpus, but here, the corpus chunk contains more information than the bag elements, resulting in an $a_i = 0.67$. Both these chunks occur once in the corpus, so we multiply their matching accuracy with the square root of the relative frequency with respect to all bags that match with the same n_i ($f_{rel,i} = \sqrt{\frac{1}{2}} = 0.71$), resulting in the values in column w_{new}.

The morphological generator

Up to now, the translated sentence consists of lemmata. This means that the correct morphological forms still have to be generated. The algorithm of the English lemmatiser used for the BNC is reversible and hence, could be used as a morphological generator (Carl et al. 2005). The tag coming from the tag-mapping rules allows us to resolve the specific features (like number, degree of comparison) of the tokens to be generated. The morphological generator also deals with capitalisation. The generation information is provided by

a simple rule-based module that keeps a capital when it is in the target-language side of the dictionary (or equivalently, when the token has an `NP0` tag) and furthermore introduces a capital when a token is at the beginning of a sentence.

8.3 Evaluation

A lot of discussion is currently going on in the MT community about evaluation. Automated scores have been presented, each with their pros and cons, and with different purposes. Amongst the most famous are BLEU (Papineni, Roukos, Ward and Zhu 2001), NIST (Doddington 2002), WNM (Babych 2004), Test Point Method (Yu 1993), X and D-score (Rajman and Hartley 2001), and the Entropy Method (Liu, Hou, Lin, Qian, Zhang and Isahara 2005). We will present BLEU scores, as they have become a kind of standard in MT, and are easy to calculate, but they only correlate moderately (this holds for all automated scores) with human judgements about fluency and adequacy, and should be taken with a grain of salt.

Two evaluations have currently been performed: an evaluation on 150 sentences in which the source language independent parts were tested, and a second evaluation in which 50 sentences went through the whole processing chain from Dutch to English.

8.3.1 Source-language-independent evaluation

The search engine was tested on sentences coming from Dutch, Greek, and Spanish, on which source-language analysis was performed and manually corrected. This resulted in 150 bags of bags which we used as input for the search engine. The average BLEU score was 0.2117.

A detailed error analysis led to the introduction of the expander. The expander was taken into account in the full chain evaluation of the next section.

8.3.2 Full chain evaluation

We also tested our system on the full chain of processes which has to be performed in our translation system. This resulted in a BLEU score of 0.2354.

Note that not all phenomena which occur in the test set have been implemented, and that there is still a lot of room for improvement. The sentences in the test set were not selected randomly, but they are selected from newspaper material and are made sure to cover a number of different known difficulties in automated translation.

A detailed error analysis showed that our source-language analysis returned the correct result as best result in 54% of the cases. In an additional 16% of the cases the correct result was the second best. Tagging was correct for 76% of the test sentences. Tagging errors almost always lead to chunking errors. A weak point in the chunker is the scope of coordination, which is very hard to determine using context-free techniques, and which often leads to inaccurate chunking. In some cases the system finds the most plausible

translation using the second-best tag path, instead of the best tag path.

Nevertheless, there is room for improvement both in source language analysis and in the translation engine. In the near future, we intend to switch to the D-CoI tagger for Dutch (Van den Bosch et al. 2006), improve our chunking grammars, and add some more rules to our expander so that more MT phenomena can be solved.

8.4 Conclusion and future

As said before, the actual goal of the METIS-II project is not to construct a better MT system than the currently existing ones, but to find a methodology to simplify the construction of new MT systems and language pairs, especially for lesser-used languages and domains where no parallel corpora are available. After the success of METIS-I, we have started to improve the quality of the translations. The first step was introducing chunking in order to increase the probability of finding an exact match in the target-language corpus.

Basically, translation is done by the bilingual dictionary and the tag-mapping rules. However, in order to provide the search engine with better translation candidates to rank, an expander was introduced. The expander uses a very limited rule set in order to rewrite or expand the candidates provided by the dictionary and the tag mapping.

The results generated by the system up to now, can be seen as a baseline for future improvements of the system. The BLEU score of 0.2354 can be augmented in a lot of ways and currently, we are working on correcting generic errors that happen to occur in our test set.

The Dutch-English dictionary is being revised at this time. The chunking rules of ShaRPa 2.0 can be refined, both for Dutch and English. The subject position is still not used in the target-language model but is in the process of being integrated. Postprocessing modules can be constructed to correct generic errors introduced by the search engine.

Finally, we need to develop some post-editing modules. The proposed translation(s) will be presented to a human editor, who can choose the best option and correct mistakes still there. We can use these corrections as an extension to the target-language model.

A more elaborate test set needs to be created, so more extensive evaluations can be done, using automated metrics like BLEU, NIST, and Levenshtein, and human judgment scores.

References

Babych, B.(2004), Weighted N-gram model for Evaluating Machine Translation Output, *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*, Birmingham.

Badia, T., Boleda, G., Malero, M. and Oliver, A.(2005), An *n*-gram approach to exploiting a monolingual corpus for Machine Translation, *Proceedings of MT Summit X, Workshop on EBMT*, Phuket, pp. 1–7.

Brants, T.(2000), TnT – A Statistical Part-of-Speech Tagger, *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.

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

Carl, M., Schmidt, P. and Schütz, J.(2005), Reversible Template-based Shake & Bake Generation, *Proceedings of MT Summit X, Workshop on EBMT*, Phuket, pp. 17–25.

Daelemans, W. and Strik, H.(2002), Het Nederlands in taal- en spraaktechnologie: prioriteiten voor basisvoorzieningen, Report by order of the Dutch Language Union.

Dirix, P.(2002a), The METIS Project: Lexical Resources, Internship Report, K.U.Leuven.

Dirix, P.(2002b), The METIS Project: Tag-mapping Rules, Paper, K.U.Leuven.

Dirix, P., Vandeghinste, V. and Schuurman, I.(2005), METIS-II: Example-based translation using monolingual corpora – System description, *Proceedings of MT Summit X, Workshop on EBMT*, Phuket, pp. 43–50.

Doddington, G.(2002), Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics, *Proceedings of the 2th Human Language Technologies Conference*, San Diego, pp. 128–132.

Dologlou, Y., Markantonatou, S., Tambouratzis, G., Yannoutsou, O., Fourla, A. and Ioannou, N.(2003), Using Monolingual Corpora for Statistical Machine Translation: The METIS System, *Proceedings of EAMT-CLAW 2003: Controlled Language Translation*, Dublin City University, Dublin, pp. 61–68.

Liu, Q., Hou, H., Lin, S., Qian, Y., Zhang, Y. and Isahara, H.(2005), Introduction to China's HTRDP Machine Translation Evaluation, *Proceedings of Machine Translation Summit X*, Phuket.

Markantonatou, S., Sofianopoulos, S., Spilioti, V., Tambouratzis, Y., Vassiliou, M., Yannoutsou, O. and Ioannou, N.(2005), Monolingual Corpus-based MT using Chunks, *Proceedings of MT Summit X, Workshop on EBMT*, Phuket, pp. 91–98.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.(2001), BLEU: a method for automatic evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL, Philadelphia.

Piepenbrock, R.(2002), CGN Lexicon v. 9.3, Spoken Dutch Corpus, TST-centrale, Leiden/Antwerp.

Rajman, M. and Hartley, A.(2001), Automatically predicting MT systems rankings compatible with Fluency, Adequacy or Informativeness scores, *Proceedings of MT Summit VIII: 4th ISLE Workshop on MT Evaluation*, Santiago de Compostella.

Sadat, F., Déjean, H. and Gaussier, E.(2002), A Combination of Models for Bilingual Lexicon Extraction from Comparable Corpora, *Proceedings of the Séminaire Papillon 2002*, Tokyo.

Thurmair, G.(2005), Improving Machine Translation Quality, *Proceedings of MT Summit X*, Phuket.

Van den Bosch, A., Schuurman, I. and Vandeghinste, V.(2006), Transferring PoS-tagging and lemmatization tools from spoken to written Dutch corpus development, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, European Language Resources Evaluation, Paris.

Van Eynde, F.(2004), Pos-tagging en lemmatisering, TST-centrale, Leiden/Antwerp.

Vandeghinste, V.(2005), Manual for ShaRPa 2.0, Internal document, K.U.Leuven.

Vandeghinste, V. and Pan, Y.(2004), Sentence Compression for Automated Subtitling. A Hybrid Approach., *Proceedings of ACL-workshop on Text Summarization*, Barcelona.

Vandeghinste, V. and Tjong Kim Sang, E.(2004), Using a Parallel Transcript/Subtitle Corpus for Sentence Compression, *Proceedings of the 4th International Confererence on Language Resources and Evaluation (LREC)*, European Language Resources Evaluation, Paris.

Vandeghinste, V., Dirix, P. and Schuurman, I.(2005), Example-based Translation without Parallel Corpora: First experiments on a prototype, *Proceedings of MT Summit X, Workshop on EBMT*, Phuket, pp. 135–142.

Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S. and Badia, T.(2006), METIS-II: Machine Translation for Low Resource Languages, *Proceedings of the 5th International Confererence on Language Resources and Evaluation (LREC)*, European Language Resources Evaluation, Paris.

Yu, S.(1993), Automatic Evaluation of Output Quality for Machine Translation Systems, *Machine Translation* **8**, 117–126.