

Example-based Translation without Parallel Corpora: First experiments on a prototype

Vincent Vandeghinste, Peter Dirix and Ineke Schuurman

Centre for Computational Linguistics

Katholieke Universiteit Leuven

Maria Theresiastraat 21

B-3000 Leuven

Belgium

firstname.lastname@ccl.kuleuven.be

Abstract

For the METIS-II project (IST, start: 10-2004 – end: 09-2007) we are working on an example-based machine translation system, making use of minimal resources and tools for both source and target language, i.e. making use of a target language corpus, but not of any parallel corpora.

In the current paper, we present the results of the first experiments with our approach (CCL) within the METIS consortium : the translation of noun phrases from Dutch to English, using the British National Corpus as a target language corpus.

Future research is planned along similar lines for the sentence as is presented here for the noun phrase.

1 Introduction: Background of METIS-II

The METIS approach differs from other known statistical or example-based approaches to machine translation in that it does not make use of parallel corpora (or bitexts) (Dologlou et al., 2003).

It is conceived as a system to be used in those circumstances in which other MT-systems that are around cannot be used, for example, because there are no sufficiently large parallel corpora available, at least not in the given domain (be it a specific sub-domain, such as the automotive domain, or the domain of free language) and/or for a given language pair. The latter will often be the case in the European context when smaller languages are involved.

Constructing a rule based system would take too much time (and therefore be too costly). An alternative solution would be to use a hybrid system, not relying on parallel corpora and with relatively few rules. METIS-II is meant to become such a system.

The rationale behind the METIS projects is that a monolingual corpus in the target language guiding the validation of translations (choice of translation alternatives, word order), together with a bilingual dictionary guiding the raw lemma-to-lemma translation, should in principle suffice to generate good translations using a combination of statistics

and linguistic rules, i.e. a hybrid approach. This monolingual target language corpus is likely to contain (parts of) sentences with the target words in them, serving as target-language examples. Finding and recombining these is in fact what METIS-II is about. The target language corpus helps disambiguating between different translation possibilities and it is used to retrieve the target language word order.

The development of such a machine translation system which uses simple tools and cheap resources for a rather complex task could give natural language processing in circumstances in which little resources are available a real boost: tasks for which parallel corpora and other expensive resources were conceived to be indispensable, can become feasible without them.

Although languages for which parallel corpora are not available in a large quantity tend to lack other resources like lemmatizers or taggers, it is much cheaper to create such resources than to create a large enough parallel corpus that links the source language with the target language.

METIS-I aimed at constructing free text translations by relying on pattern matching techniques and by retrieving the basic stock for translations from large monolingual corpora. METIS-II aims at further enhancing the system's performance and adaptability by:

- Breaking sentence-internal barriers: the system will retrieve pieces of sentences (chunks) and will recombine them to produce a final translation. This approach was also used by (Veale and Way, 1997), (Nirenburg et al. 1994), and (Brown, 1996).
- Extending the resources and integrating new languages using post-editing facilities.
- Adopting semi-automated techniques for adapting the system to different translation needs.
- Taking into account real user needs, especially

as far as the post-editing facilities are concerned.

This paper describes the approach of the Centre for Computational Linguistics within the METIS consortium. Other approaches can be found in (Markantonatou et al., 2005, this volume) and (Bardia et al., 2005, this volume).

The experiments in this article are part of the investigations in the breaking of the sentence-internal barriers. We use noun phrase (NP) translation as a test case.

Our NP translation system differs from the approach explained in (Sato, 1993), in that we do not use parallel corpora, but a bilingual dictionary, and that our system is not domain specific. We also use a different weighing mechanism (cf. section 2.3.2).

Dutch is used as a source language with the parts-of-speech tagset of (Van Eynde, 2004). English is used as a target language, the British National Corpus (BNC) as target-language corpus with the CLAWS5 tagset. A reason why not to use the world wide web as a resource like (Grefenstette, 1999) is that our corpus needs to be preprocessed (tagged, chunked, lemmatized) and our target language is English from native speakers.

For a more extensive description of the METIS system see (Dirix et al., 2005, this volume).

2 System Description

In this section we describe our prototype system, which is used in the experiments in section 3, and which is implemented in perl 5.8.5 (Wall, 2004).

In figure 1, we present the general system flow (at the sentence level). The prototype we use is part of this general system as it translates noun phrase chunks.

First we describe how the source language analysis is performed (section 2.1), then we describe how we map the source language to the target language (section 2.2), and finally we describe the target language generation (section 2.3).

2.1 Source Language Analysis

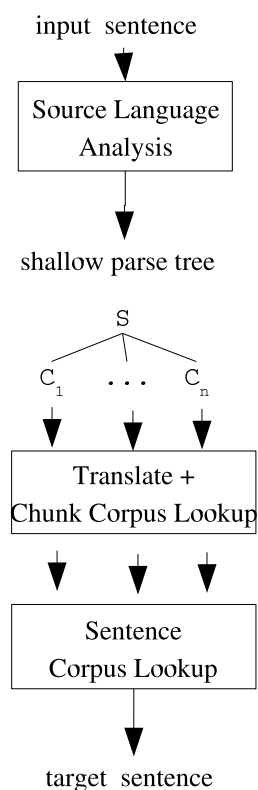
The source language (Dutch) text is analysed in a number of steps: tokenization (section 2.1.1), part-of-speech tagging (section 2.1.2), lemmatization (section 2.1.3) and chunking (section 2.1.4).

For the experiments in section 3, we used a test set of already analysed source language noun phrases.

Nevertheless, the prototype system is capable of doing its own source language analysis.

Let's take the following Dutch NP as an example: *een jonge champignon* [a young mushroom]

Figure 1: General System Flow



2.1.1 Tokenization

The first processing step in the source language analysis is the tokenization of the input sentence. The input sentence is converted into a series of tokens, representing separate words. All punctuation is considered as separate tokens.

Example

<i>“een jonge champignon”</i>	tokenized into	<i>“een”</i>
		<i>“jonge”</i>
		<i>“champignon”</i>

2.1.2 Part-of-Speech Tagging

The part-of-speech (PoS) tagger we use is TnT (Brants, 2001), which was trained on the spoken Dutch corpus (CGN) internal release 6. It is reported to have an accuracy of 96.2% (Oostdijk et al., 2002). The tagset which was used is the CGN-tagset (Van Eynde, 2004).

Example			
<i>een</i>	gets	the	<i>LID(onbep,stan,agr)</i> (indefinite article)
<i>jonge</i>			<i>ADJ(prenom,basis,met-e,stan)</i> (prenominal adjective)
<i>champignon</i>			<i>N(soort,ev,basis,zijd,stan)</i> (non-neutre singular common noun)

2.1.3 Lemmatization

Each token is lemmatized, by looking up the token and its PoS-tag in the CGN-lexicon (Piepenbrock, 2004), and retrieving the words lemma. For some tokens, the lemmatization process results in more than one lemma. By using the PoS-tag as additional input for the lemmatizer, the amount of ambiguity can be strongly reduced. For instance, the Dutch word *was* can be a noun meaning *wax* or *laundry* or the past tense singular of a verb meaning *to be*. It can thus be lemmatized as *was* (noun) or as *zijn* (verb). By using the PoS-tag as additional input, we can disambiguate between these two lemmas¹.

Example		
<i>een</i>	lemmatized into	<i>een</i>
<i>jonge</i>		<i>jong</i>
<i>champignon</i>		<i>champignon</i>

In future versions of our system we plan to implement a rule-based lemmatizer for Dutch, which would only use the lexicon for the exceptions to the rules and would have a larger coverage as it would also return lemmas for previously unseen words.

2.1.4 Chunking

The sentence is sent to the ShaRPa chunker, which was adapted for the METIS-II project and already used in (Vandeghinste and Pan, 2004) and (Vandeghinste and Tjong Kim Sang, 2004). The updated

¹As far as *was* as a noun is concerned, this is a homonym meaning either *laundry* or *wax*. The tag associated with both meanings is not identical: they differ in gender. *Was* (laundry) is non-neuter, whereas *was* (wax) can be used both as neuter and non-neuter. Whenever the word is used in a neuter context (determiner, neuter form of adjective), we know for sure that it is to be translated as *wax*. In the other cases we are to derive the proper translation via the BNC (searching for adjectival and verbal contexts in which *laundry*, resp. *wax* are used).

Making use of this information still needs to be implemented.

version of ShaRPa is using the same rules as before, but is now able to detect the heads of phrases, which was necessary for the approach described in this paper.

In the experiment described in this paper, it is used only to detect NPs and their heads. As described in Vandeghinste and Tjong Kim Sang, the chunking accuracy for noun phrases has an F-value of 94.7%.

Example	
<i>een jonge champignon</i>	
chunk type	NP
head	<i>champignon</i>

2.2 Source to Target Language Mapping

Source to target language mapping contains two stages: the translation of the source language lemmas into target language lemmas, using a bilingual dictionary (section 2.2.1) with a treatment for missing entries (section 2.2.2), and the conversion of the source language tags into the target language tags (section 2.2.3).

2.2.1 Bilingual Dictionary

For the mapping of the analysed source language NP to the target language, we use a bilingual dictionary, taking a lemma and a PoS-tag (without features) as input and returning a target language lemma and a partial target language tag.

The initial bilingual dictionary was compiled from various sources, like the Ergane Internet Dictionaries (Travlang Inc., <http://www.travlang.com/Ergane>) and the Dutch WordNet (Vossen et al., 1999) and manually edited and improved (Dirixa, 2002).

After some more editing and correcting the resulting dictionary contains about 37000 different source language lemmas. The average source language lemma has more or less three translations.

Note that one source language lemma can be translated into several consecutive target language lemmas.

Example		
<i>een</i>	is translated into	<i>a / an / one</i> <i>anybody</i> <i>some</i> <i>somebody</i> <i>someone</i>
<i>jonge</i>		<i>young</i>
<i>champignon</i>		<i>mushroom</i>

Together with the target-language lemmas, we retrieve target-language lemma tags from the dictionary. These tags contain only partial information,

compared with the CLAWS5 target-language tagset. Because the tag contains information about a lemma and not about a token it cannot contain certain feature values (e.g. number), but it can contain others (e.g. gender). In our current system it only contains the PoS, and no feature-information.

In some cases, one word in the source language is translated into several consecutive words in the target language. The dictionary should contain the PoS information for each of those words, which is not yet the case in the current version, where we use underspecification in those cases where that information is missing.

There is certainly room for other improvements to the dictionary, as it still contains mistakes and some high-frequency words are still missing (especially Belgian Dutch items). Future versions of our system will use updates of this dictionary.

As Dutch is a language with productive word formation processes (amongst others, Booij and van Santen, 1995), it is impossible to include all words in the dictionary.

As a weight for the different translation alternatives we use the frequency of that lemma and tag combination in the target-language corpus, divided by the total frequency of all the translation alternatives for that entry. If the translation alternative contains two words, we look up the frequency of that bi-gram in the target-language corpus instead of the frequencies of the separate words. When there are more than two words in the translation of the word, for now we use a back-off procedure of giving them the frequency of 1.

2.2.2 Out-of-Vocabulary Treatment

When translating NPs, there are always words missing from our lexicon. In these cases we apply the following approach:

- If tokens are tagged as *proper nouns* in the source language, keep them as they are. If there are no translation alternatives, set the weight for the translated entry to 1.
- Check if the tokens are *compounds*. If this is the case, then translate the compounds' modifier and head instead of the token as a whole. Here we use the same hybrid decompounding/compounding module as in (Vandeghinste, 2002), which is used in its decompounding mode. It takes a word (lemma or token) as its input and generates the word parts plus a confidence value. The modifier and the head are considered as separate tokens for the rest of the processing, and they are treated like dictionary entries which contain one word on the source

language side and two on the target language side.

It is clear from our experiments that this approach works only in a number of cases but fails in others. Nevertheless it improves translation accuracy.

For instance, the word *maffiakenner* is not present in our lexicon. The word is split up into two parts: *maffia* and *kenner*, which are both in our lexicon. This results in the translation *Maffia expert*, which is a correct translation.

The word *fractieleider* (leader of a parliamentary party) is also missing from our lexicon. We could also split it up into two parts: *fractie* and *leider*, which could both be in our lexicon. This would result in the translation *fractie leader* which is an inaccurate translation.

- If none of the above apply², keep the word as it is, as we do not have a clue on how to translate it. In the experiment, we do not produce a translation in this case as it is definitely incorrect.

2.2.3 Tag Mapping Rules

Apart from what is described in the previous sections, tag mapping rules are used (Dirix, 2002b). For each source language PoS tag, the equivalent target language tags were identified and put in a database. Some of the morpho-syntactic features are 'translated' from source to target language. The source language tagset is described in (Van Eynde, 2004) and the target language tagset CLAWS5 is described on the UCREL website (University Centre for Computer Corpus Research on Language), <http://www.comp.lancs.ac.uk/ucrel/claws5tags.html>.

Example

<i>LID()</i>	into	<i>ATO</i>
<i>ADJ(prenom,basis)</i>		<i>AJO</i>
<i>N(soort,ev,stan)</i>		<i>NNO</i> or <i>NNI</i>

By combining the partial tag from the dictionary and the tag mapping rules, we can reduce a number of ambiguities which would otherwise arise.

2.3 Target Language Generation

Generating the target language by using the BNC as a data-set of examples is a rather complex task.

The target language generation uses the head of the NP, plus the bag of the other lemmas in the NP,

²Some other regularities in the translation of compounds will be implemented at a latter stage (e.g. *parlementslid* into *member of parliament* instead of *parliament member*).

together with their target language tags. In order to find out the exact word order, and disambiguate the different translation possibilities coming from the bilingual dictionary we use the BNC, which is preprocessed as described in the following section.

First, we describe how the target language corpus was preprocessed (section 2.3.1), and then we describe how we match the bag with the corpus (sections 2.3.2, 2.3.3 and 2.3.4).

2.3.1 Preprocessing of the Target Language Corpus

We lemmatized the BNC, using the lemmatizer described in (Carl et al., 2005). Then, we chunked the BNC, using ShaRPa2.0 with a rule-set for English. This was done only up to the lowest NP level.

This results in a huge number of NPs, for which we have their head and the structure of the chunk (containing the tags of the leaf nodes and possible intermediate levels between the NP and the leaf nodes).

We put this in a database, indexed on the head, allowing fast retrieval of NPs based on their head.

If an NP is found for which the lemmas exactly match the lemmas in the bag of lemmas, we use this NP as a possible translation. The frequency with which this NP occurs in the BNC, divided by the total frequency of all the possible translations found this way is used as the weight for that translation.

If there is no exact match with the bag of lemmas, we try to find an NP with the same head, but for which the tags of the tokens in the NP match the tags in the bag of lemmas, and replace the words which are not occurring in the retrieved NP from the BNC, hence producing a translated NP.

2.3.2 NP Retrieval from BNC

When having the bag of lemmas and the head as input, we retrieve all noun phrases from BNC with this head. From these noun phrases, we extract the noun phrases in which each lemma of each word corresponds with the lemmas from the words in the bag.

When such a noun phrase is found, it is considered a translation alternative, with weight w_k which is calculated as follows:

$$w_k = \frac{freq(a_k)}{\sum_k freq(a_k)}$$

The frequency with which the alternative occurs in the BNC, divided by the total frequency of all matching NPs is used as the weight for that translation, ignoring the information about the frequency of the separate tokens in the BNC. When we cannot

find such a noun phrase, we switch to NP Template Retrieval, which is described in the next section.

Example

In the BNC we find 273 different NPs with *mushroom* as the lemma of their head. Of these, there is only one which contains all the words from the bag, but it contains also a number of other tokens, which are not present in the bag, and therefore we switch from NP Retrieval to Head-based NP Template Retrieval.

2.3.3 Head-based NP Template Retrieval from BNC

When no noun phrase can be retrieved from the BNC in which all the lemmas in the bag correspond to the target language, we try to retrieve a noun phrase template, with the same head. In order to do so, we retrieve all the noun phrases from the BNC with the current head, and try matching the tag structure of these noun phrases with the tags of the translations coming from the dictionary.

When we find a matching template, we have to replace the original words in the retrieved noun phrase with the actual translations of the input words, where the tags of the original words match the tags of our dictionary translations. In this process, we replace as minimal as possible, maximizing the influence of the target language corpus.

This greatly enhances the coverage of using the noun phrases of the BNC.

Example

Of the 273 different NPs with *mushroom* as its head, there are 9 NPs which only differ one word with the bag of TL lemmas derived from the dictionary. They all contain three tokens, of which two are present in the lists of translation alternatives from the dictionary. Only the adjective differs. So we replace the adjective in these NPs by the translations of the adjective coming from our dictionary, which leads to the desired result, being *a young mushroom*.

Again, the relative frequency of occurrence of the NP Template is used as a weight for the different translation alternatives.

2.3.4 Other Cases

It still happens that no matching NP Template can be found in the BNC with the same head. When this is the case, we want to apply an even more general template approach, in which the head word does

not play any role anymore, but all the noun phrase structures we find in the BNC are taken into account (with their frequencies), so we match the words and target tags coming from the dictionary with the different tag-structures we find in the BNC, giving the most frequent tag-structure the highest translation priority.

As this is not yet implemented, when no solution is found following the procedure described in the previous sections, we generate a word-by-word translation, using the word frequency based weights to rank different translation alternatives.

3 Experiments

In these experiments we wanted to validate our approach by testing it on noun phrase translations. Different teams in the METIS2 consortium are investigating different approaches.

First we describe the methodology of our experiments (cf. section 3.1), and then an overview of the results is given (cf. section 3.2).

3.1 Methodology

For our experiments, we used a test set of 685 NPs, of which 467 come out of the Spoken Dutch Corpus³, and 218 noun phrases out of recent newspaper texts.

All the input NPs are correctly tagged and chunked. When they were not correctly tagged or chunked, they were left out of the test set. This concerns a small number (about 1%) of mainly complex NPs⁴.

We did only take NPs into account which contain at least one noun. NPs containing only a personal pronoun are not taken into account.

For the rest, the tool as it is currently implemented for these experiments follows what is described in section 2.

As the results described in this paper are only first prototype results, we did not apply any of the automated evaluation approaches for machine translation, like BLEU (Papineni et al., 2002), but evaluated our results manually by judging the translation quality.

3.2 Results

Table 1 and figure 2 show the results of our evaluation. Each NP translation resulted in a number of translation alternatives, ranked by their weight. For each NP translation, we judged whether the first

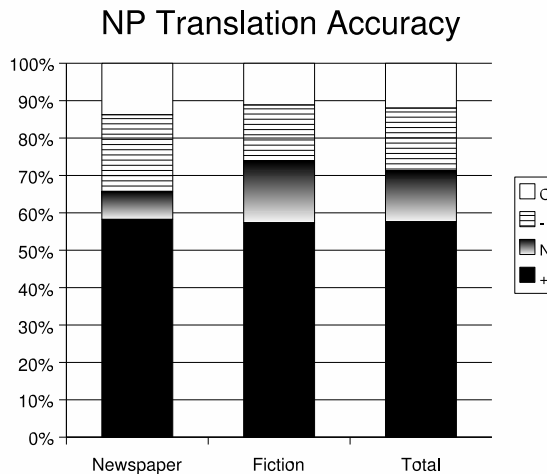
³They were extracted from the section of the Spoken Dutch Corpus (CGN) which contains read-aloud fiction.

⁴With complex NPs, we mean NPs which consist of a number of elements amongst which a lower level NP

	Newspaper	Fiction	All
Correct	58.26%	57.39%	57.66%
N-best correct	7.34%	16.49%	13.58%
Incorrect	20.64%	14.99%	16.79%
No output	13.76%	11.13%	11.97%

Table 1: NP translation accuracy

Figure 2: NP translation accuracy



translation alternative was *correct* (+). When this was not correct we looked among the other translation alternatives. When a correct translation was present this response was classified as *N-best correct* (N). We did not limit N, because we wanted to see whether our system was capable of generating a correct translation. When only incorrect output was generated, the response was classified as *incorrect* (-). In some cases the system did return *no output* (0).

Our system produces several translation alternatives, ranked according to their weight. In 57.66% of the cases, the system provides a correct translation. In another 13.58% of the cases, the correct translation is among the translation alternatives, but did not receive rank 1. This implies that, by only changing the weighing mechanism, we could get a maximum of 71.24% correct NP translations.

There are slight differences between newspaper texts and fiction texts. Fiction seems a little easier to translate (at least when we include the N-best solutions)

The fact that these results are not higher is due to the coverage of the lexicon, as illustrated in table 2. Although some of these uncovered cases are solved by the decomposing module, most of them re-

main unsolved and hence result in an incomplete translation or no translation at all. One of the test texts contained a high number of exclusively Belgian Dutch words, which are missing from our lexicon and which explains the low translation accuracy of that text (50.59% correct + 2.35% N-best).

Also, a number of cases where no output was generated can be explained due to bugs in our prototype system, which we expect to solve in future versions.

	Coverage
Newspaper	80.28%
Fiction	80.51%
Total	80.44%

Table 2: Coverage of the dictionary by token

4 Conclusions

Looking at these results and some of the reasons why the results are not better than they are, we can conclude that the approach adopted in our system works reasonably well for the translation of noun phrases.

As this is work in progress (initial version of the code, the dictionary and the weighing system), we expect our system to perform better in future versions.

NP translation is a substantial part of full sentence translation, but it is not safe to assume that because our approach works for noun phrase translation, it will work for full sentence translation.

In NP translation from Dutch to English, there are not many word order issues to solve. Translating VPs is already much more difficult (Way and Gough, 2003), and we want to translate full sentences. There are also no agreement issues to solve, which certainly would be the case when translating full sentences (like the agreement between the subject and the verb).

But still, as the approach seems promising, we plan to use the same strategy when implementing our full sentence translation system, although many issues will have to be solved during the process.

5 The Near and Not Too Distant Future

In the near future, we plan to implement a full sentence translation system. In order to do so, there are a number of tasks which need to be executed.

Amongst others, we need to ameliorate the Dutch language analysis tools, because when mistakes are made in the SL-analysis, this will most certainly lead to incorrect translations.

We also need to improve the English language analysis tools, with which we preprocess the TL-corpus, because the better the TL-corpus is preprocessed, the higher the probability is to retrieve useful information from the corpus.

Work on the bilingual dictionary is also not finished. We need to extend and ameliorate it, because when dictionary information is incorrect or missing, it becomes almost impossible to generate a correct translation. We also need to add some words which are typical for Belgian Dutch, as they tend to be left out of the dictionary.

As mentioned in section 2.1.3 we are using the PoS-tag to assign the correct lemma to a word. We may also make use of the further features of the PoS-tag to distinguish between the various meanings (plus associated translations) of a lemma.

For the experiments described in this paper we used a lexicon with very underspecified PoS (only main PoS (N,ADJ etc.), cf. section 2.2.1, without further features), we are in the process of adding some features in those cases where it might help translation (like the noun *was*). Further experiments will have to prove of this.

The TL-corpus needs to be preprocessed at the sentence level, analogous to the way it is preprocessed now at the NP level.

The Head-based Template Retrieval mechanism needs to be enhanced to get more information out of the corpus, and we need to implement the general Template Retrieval mechanism, which does not make use of heads.

We need to implement some extra language analysis tools (e.g. a subject detector) to enable us to enhance translation quality.

A number of frequency tables need to be created, derived from the TL-corpus, which will allow for a more accurate weighing system

We need to come up with a solution concerning prepositional phrase attachment and the translation of light verbs.

In all, there are numerous tasks still to be performed to get to a “good” translation system, but the general system outline is emerging in the process.

6 References

- T. Badia, G. Boleda, M. Melero, A. Oliver. 2005. An n-gram approach to exploiting a monolingual corpus for Machine Translation. In *Proceedings EBMT Workshop 2005 - this volume*.
- G. Booij and A. van Santen. 1995. *Morfologie. De woordstructuur van het Nederlands*. Amsterdam University Press, Amsterdam.

- T. Brants. 2001. *TnT - A Statistical Part-of-Speech Tagger*. Published online at <http://www.coli.uni-sb.de/thorsten/tnt>.
- R.D. Brown. 1996. Example-Based Machine Translation in the Pangloss System. *COLING 1996*, Copenhagen, Danmark. pp. 169-174.
- M. Carl, P. Schmidt and J. Schütz. 2005. Reversible Template-based Shake & Bake Generation. In *Proceedings EBMT Workshop 2005 - this volume*.
- P. Dirix. 2002a. *The METIS Project: Lexical Resources*. Internship Report, KULeuven.
- P. Dirix, 2002b. *The METIS Project: Tag-mapping rules*. Paper, KULeuven.
- P. Dirix, I. Schuurman, V. Vandeghinste. 2005. METIS: Example-Based Machine Translation Using Monolingual Corpora - System Description. In *Proceedings EBMT Workshop 2005 - this volume*.
- Y. Dologlou, S. Markantonatou, G. Tambouratzis, O. Yannoutsou, A. Fourla, and N. Ioannou. 2003. Using Monolingual Corpora for Statistical Machine Translation: The METIS System. In *Proceedings of EAMT - CLAW 2003*, Dublin, pp. 61-68.
- G. Grefenstette. 1999. The World Wide Web as a Resource for Example-Based Machine Translation Tasks. *ASLIB, Translating and the Computer 21*. London.
- S. Markantonatou, S. Sofianopoulos, V. Spilioti, Y. Tambouratzkis, M. Vassiliou, O. Yannoutsou, N. Ioannou. 2005. Monolingual Corpus-based MT using Chunks. In *Proceedings EBMT Workshop 2005 - this volume*.
- S. Nirenburg, S. Beale, and C. Domashnev. 1994. A Full-Text Experiment in Example-Based Machine Translation. In *Proceedings of the Int. Conf. on New Methods in Language Processing*. Manchester, UK. pp. 78-87.
- N. Oostdijk, W. Goedertier, F. Van Eynde, L. Boves, J.P. Martens, M. Moortgat, and H. Baayen. 2002. Experiences from the Spoken Dutch Corpus Project. In *Proceedings of LREC 2002*, vol. 1, pp. 340-347.
- K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40st Annual Meeting of the Association for Computational Linguistics*.
- R. Piepenbrock. 2004. *CGN Lexion v.9.3*. Spoken Dutch Corpus.
- S. Sato. 1993. Example-Based Translation of Technical Terms. *Proceedings of TMI 1993*. Kyoto, Japan.
- V. Vandeghinste. 2002. Maximizing Lexical Coverage in Speech Recognition through Automated Compounding. In *Proceedings of LREC2004*. ELRA. Paris.
- V. Vandeghinste and Y. Pan. 2004. Sentence Compression for Automated Subtitling. A Hybrid Approach. In *Proceedings of ACL-workshop on Text Summarization*. Barcelona.
- V. Vandeghinste and E. Tjong Kim Sang. 2004. Using a Parallel Transcript/Subtitle Corpus for Sentence Compression. In *Proceedings of LREC2004*. ELRA. Paris.
- F. Van Eynde. 2004. *Tagging and Lemmatisation for the Spoken Dutch Corpus*. Internal report.
- T. Veale and A. Way. 1997. Gaijin: A Bootstrapping, Template-Driven Approach to Example-Based MT. In *Proc. of the 2nd Int. Conf. on Recent Advances in NLP*. Tzigov Chark, Bulgaria. pp. 239-244.
- P. Vossen, L. Bloksma, and P. Boersma. 1999. *The Dutch WordNet*. University of Amsterdam.
- L. Wall. 2004. *Perl 5.8.5*. <http://www.perl.com>.
- A. Way and N. Gough. 2003. WEBMT: Developing and Validating an Example-Based Machine Translation System using the World Wide Web. *Computational Linguistics*, 29 (3), pp. 421-457.