Demonstration of the Greek to English METIS-II MT System

Sofianopoulos Sokratis
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25
Paradissos Amaroussiou Athens, Greece
s_sofian@ilsp.gr

Spilioti Vassiliki
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25
Paradissos Amaroussiou Athens, Greece
vspiliot@ilsp.gr

Vassiliou Marina
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25 Paradissos Amaroussiou Athens, Greece
mvas@ilsp.gr

Yannoutsou Olga
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25 Paradissos Amaroussiou Athens, Greece
olga@ilsp.gr

Markantonatou Stella
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25 Paradissos Amaroussiou Athens, Greece
marks@ilsp.gr

Abstract

METIS-II, the MT system presented in this paper, does not view translation as a transfer process between a source language (SL) and a target one (TL), but rather as a matching procedure of patterns within a language pair. More specifically, translation is considered to be an assignment problem, i.e. a problem of discovering each time the best matching patterns between SL and TL, which the system is called to solve by employing pattern-matching techniques.

Most importantly, however, METIS-II is innovative because it does not need bilingual corpora for the translation process, but exclusively relies on monolingual corpora of the target language.

1 Introduction

The system presented here further elaborates on the original METIS approach (Dologlou et al., 2003) which did not view translation as a transfer process between a source language and a target one, but rather as a matching procedure of patterns within a language pair (Markantonatou et al., 2006). With this approach, only basic NLP resources (such as taggers, lemmatisers, chunkers and simple bilingual lexica) are needed, while new languages, especially low density ones, can be easily included in the system. Furthermore, bilingual corpora are no longer essential; monolingual corpora of the target language suffice for the translation process.

METIS-II extends the original idea by handling patterns (translation units) at sub-sentential level, thus facilitating the elicitation of linguistic information from the TL corpus such as syntactic and/or semantic preferences of words as well as word order.

Four language pairs have been developed as yet, namely Dutch, German, Greek and Spanish into English, all with satisfactory results in terms of BLEU (Papineni et al. 2002) and NIST (2002) evaluations (Tambouratzis et al., (2006) and METIS II – Deliverable 5.2 (2007)); however, the METIS-II system reported here concerns only the Greek into English language pair.

METIS-II system comprises roughly four (4) modules/phases, namely Pre-processing (transformation of the input into patterns), Core Engine (pattern matching), Token Generation (creation of word forms) and Synthesising (composition of the final translation).

The structure of the paper is as follows: in Section 2 the main system features are presented. Sections 3, 4, 5 and 6 describe the respective system modules. Section 7 reports on system testing and evaluation results; section 8 provides a brief description of the translation process, while the last section summarises the plans for the future development and optimisation of the system.

2 System Features

METIS-II is regarded to be of hybrid nature, since it joins pattern-matching techniques with statistical information, while employing algorithms for handling combinatorial optimisation problems (such as the assignment problem). In addition, a very limited number of linguistic rules is employed, thus avoiding the explosion of rules in rule-based grammars (Gaizauskas, 1995).

Moreover, within this system, what is crucial is the notion of patterns, that is, phrasal models (tokens, chunks, clauses, sentences), which form the basis for measuring the similarity between SL and TL. Patterns are generated by the tools used for both languages and differ from the patterns employed in the corpus-based MT paradigm mostly in the sense that they are viewed as models of TL strings, which receive their final form after corpus consultation.

Therefore, METIS II is different both at implementation level, given that it employs a variety of algorithms, and conceptually, since translation is viewed as a matching process of patterns between SL and TL, aiming each time at detecting the best match.

Nowadays investigation of hybrid systems combines easy-to-obtain resources from all MT paradigms and shows a very promising path in research (Thurmair, 2005).

3 Pre-Processing

For the translation process both the SL input and the TL corpus are transformed to sets of patterns, which are generated with standard NLP techniques.

3.1 TL pattern generation

The TL pattern generation involves the off-line pre-processing of the British National Corpus (BNC[1]), which has been selected as TL corpus. BNC pre-processing comprises the following steps:

- Lemmatisation with a reversible lemmatiser (Carl et al., 2005)
- Segmentation of text into finite clauses with a purpose-built tool
- Syntactic annotation at chunk level with ShaRPa 2.0 chunker (Vandeghinste, 2005)
- Corpus indexation to allow for an efficient and fast search for a best match: in particular, clauses are indexed according to their finite verb, while chunks are classified according to their labels.

3.2 SL pattern generation

The SL pattern generation involves the annotation of the SL input by a tokeniser, lemmatiser, tagger (Labropoulou et al., 1996) and a chunker (Boutsis et al., 2000), resulting in a sequence of labelled patterns[2] and their contained tokens. In addition, the respective heads are identified.

This sequence is then enhanced by the Lexicon look-up, which provides all the possible translation equivalents together with PoS information, resembling thus a TL sequence.

It should be noted that the METIS-II system receives as input a sequence of sentences, but it handles each contained clause separately, synthesising in the end the translations of the various segments.

4 Core Engine

The core engine of METIS-II system is fed with a sequence of TL-like patterns (created as described in Section 3.2), which is handled by the pattern-matching algorithm in order to produce the final translation.

A characteristic feature of the pattern-matching algorithm, which mimics and exploits the recursive nature of language, is that it proceeds in stages: moving from wider patterns to narrower ones, it manages to discover the longest similar pattern in terms of overall structure and lexical head affiliations and then identify and correct any residual mismatches. Similarity is calculated on the basis of a series of weights, which mainly reflect grammatical information.

More specifically, the system searches the TL corpus for candidate patterns of clauses, which are similar to the given TL-like clause pattern in terms

[1] www.natcorp.ox.ac.uk/

[2] The pattern labels denote the categorical status of patterns.

of the main verb and the number of contained chunk patterns (Step 1).

In accordance to the above, the first comparison is performed at clause level, where similarity is calculated on the basis of the main verb, the chunk labels and the head lemmas, resulting in the establishment of chunk order within the TL-like clause (Step 2).

The subsequent comparison is narrower and confined within the boundaries of the chunk patterns. The pattern-matching algorithm calculates the similarity of contained tokens, fixing thus the correct order of tokens within each chunk (Step 3).

At the end of the comparison process a TL corpus clause is selected as the basis of translation, while chunk and token order has been established. Nevertheless, the final translation is derived from the specific corpus clause, only after the contained chunks have been processed, with the purpose of eliminating any mismatches. This processing entails either modification or substitution of given chunks, in order to include them in the final translation (Step 4).

The output of the pattern-matching algorithm is a sequence of translated lemmas and their respective tags, which is subsequently fed into the token generation module.

5 Token Generation

The token generation module receives as input a sequence of translated lemmas and their respective tags and is responsible for the production of word forms (tokens) out of lemmas and the handling of agreement phenomena, for instance subject-verb agreement, on the basis of morphological information.

For the generation task, METIS-II utilises resources produced and used in the reversible lemmatiser/token-generator for English (Carl et al., 2005).

The morphological features identified and used, which are essential for the specific TL, namely English, are tense, person, number, case and degrees of comparison (comparative and superlative degree). These features are integrated within the inflection rules employed for token generation.

Furthermore, morphological information is exploited for handling the syntactic phenomenon of subject-verb agreement, especially in cases of an empty subject. Given that Greek is a pro-drop language, subjectless clauses often occur. The generation module is based on the morphological features of the main verb of a given clause, in order to derive a suitable subject pronoun on every occasion.

6 Synthesising

As mentioned above, METIS-II receives as input a text, i.e. a sequence of sentences. Sentences consist of clauses, and very often a clause may be discontinued through the embedding of another clause. The METIS-II core engine creates separate translation processes for each clause, namely each clause process is a separate thread, running in parallel with the others. When a clause thread has finished translating, it reports back to the core engine.

When all SL clause processes have reported back, the corresponding target sentence is formed. Clauses are placed in the target sentence in the same order as they are found in the source sentence. However, in cases of discontinuous embedding, the translation output consists of clauses placed next to each other.

When the synthesising phase is concluded for a given sentence, then this sentence is added to the final text, following source text sentence order.

The entire translation process, from the input of the TL-like pattern to the core engine up to the synthesising phase, is presented in Figure 1.

7 System Testing and Evaluation

In the present section the results obtained for the Greek → English language pair are summarised. The experiment involved testing METIS-II in comparison to SYSTRAN, a commercial, widely-used MT system, which is mainly rule-based.

7.1 Experimental set-up

The corpus tested was extracted from real texts, mainly from newspapers, and consisted of fifty (50) sentences. The test sentences had an average length of 8,2 words, were of relative complexity, containing one to two clauses each and covered various syntactic phenomena such as word-order variation, NP structure, negation, modification etc.

There was no limitation defined regarding the possible translations of each source token, while the reference translations used for the evaluation have been restricted to three (3) and were produced by humans.

With respect to the evaluation of both MT systems, METIS-II and SYSTRAN, established metrics in the MT field were employed, namely BLEU (Papineni et al. 2002) and NIST (2002), which rely on calculating matching n-grams over words, as well as the Translation Error Rate (TER), which measures the amount of editing that a human would have to perform to change a system output, so that it exactly matches a reference translation (Snover et al., 2006: 1).

7.2 Experimental results

The experimental results obtained are summarised in Tables 1-3, where the mean of the 50 sentence scores obtained for each system are indicated, together with the median, the standard deviation, as well as the maximum and minimum scores.

As can be seen from Table 1, where the evaluation results based on the BLEU metric are presented, both systems exhibit the same maximum and minimum accuracy; however, METIS-II has a significantly higher mean accuracy. More specifically, METIS-II achieves perfect scores for 16% of the test sentences, while the respective SYSTRAN percentage is 4%.

Nevertheless, SYSTRAN gets slightly better scores at the middle score range, which explains why this system has a higher median accuracy. Moreover, SYSTRAN seems to be more stable, given that its scores are characterised by a lower standard deviation.

With respect to the NIST metric, the picture seems more straightforward. METIS-II consistently generates more accurate translations, while SYSTRAN continues behaving in a more stable manner, since its standard deviation is lower.

The opposite conclusions are obtained, as regards the TER metric, according to which the lowest scores are equated to a smaller number of edits. Therefore, apart from its high maximum accuracy, SYSTRAN consistently exhibits a better mean and median accuracy, while once more is proved to be a more stable system than METIS-II, since its scores are characterised again by a lower standard deviation. It should be noted, though, that METIS-II achieved a perfect translation for 9 out of the 50 sentences, while SYSTRAN translated perfectly only 3.

In order to investigate whether the differences in the accuracy populations (where each sentence corresponds to one element of the population) of the two systems, METIS-II and SYSTRAN, are significant, a set of t-tests were performed on the metric (BLEU, NIST, TER) results per system. More specifically, 3 paired t-tests were performed, in order to determine whether the means of the translation scores for the two systems differed significantly.

The output of the t-tests indicated that the differences in the mean accuracy of the two systems were not statistically significant for any of the three metrics at a confidence level of 95%.

	METIS-II	**SYSTRAN**
Mean accuracy	0,3841	0,3214
Median accuracy	0,3537	0,3715
Standard Deviation	0,3718	0,2960
Maximum accuracy	1,0000	1,0000
Minimum accuracy	0,0000	0,0000

Table 1. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **BLEU** metric

	METIS-II	**SYSTRAN**
Mean accuracy	6,8088	6,3128
Median accuracy	7,4175	6,6791
Standard Deviation	2,5878	2,2869
Maximum accuracy	10,9051	10,8134
Minimum accuracy	1,2651	0,4828

Table 2. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **NIST** metric

	METIS-II	**SYSTRAN**
Mean accuracy	33,7873	33,3587
Median accuracy	34,7700	29,2855
Standard Deviation	23,9438	21,1764
Maximum accuracy	90,9090	105,8820
Minimum accuracy	0,0000	0,0000

Table 3. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **TER** metric

8 Web Application

METIS-II has been implemented as a web application, providing a common interface (Figure 2) for all four language pairs. The whole process is pretty simple, with the end user selecting the preferred source language and entering a sentence for translation. When the “Translate” button is pressed, the corresponding web service is initiated and the given sentence is handled by the various system modules.

When the translation process is terminated, the result appears on the web page, while the intermediate system outputs are available to the end user in .html form (Figure 3).

9 Future plans

In METIS-II we have succeeded in restricting the use of structure-modifying rules by using adjustable weights in various phases of the translation process. The employment of adjustable weights makes it possible for the system to move within, i.e. to choose from, a range of potential decisions, thus leading to a different translation output.

Apart from delimiting the use of rules, weights also render METIS-II user-customisable, as the system can be tuned to the end user needs via appropriate weight selection. In this way, the system adapts to a specific operational environment and the output gradually improves, leaving intact the processes of the core engine.

At this point of development, however, all the aforementioned weights have been initialised manually, based on intuitive knowledge. What is, thus, essential is an automated process for defining and calculating the optimal weight values. To achieve that, exploration of appropriate machine learning methods has been planned.

References

Boutsis, S., Prokopidis, P., Giouli, V. & Piperidis, S. (2000). A Robust Parser for Unrestricted Greek Text. In Proceedings of the Second International Conference on Language Resources and Evaluation, Vol. 1 (pp. 467—482). Athens, Greece.

Carl, M., Schmidt, P. & Schütz, J. (2005). Reversible Template-based Shake & Bake Generation. In Proceedings of the Example-Based Machine Translation Workshop held in conjunction with the 10th Machine Translation Summit (pp. 17--26). Phuket, Thailand.

Dologlou, I., Markantonatou, S., Tambouratzis, G., Yannoutsou, O., Fourla, A. & Ioannou, N. (2003). Using Monolingual Corpora for Statistical Machine Translation. In Proceedings of EAMT/CLAW 2003 (pp. 61--68). Dublin, Ireland.

Gaizauskas, R. (1995). Investigations into the Grammar Underlying the Penn Treebank II. Research Memorandum CS-95-25, Department of Computer Science, University of Sheffield.

Labropoulou, P., Mantzari, E. & Gavrilidou, M. (1996). Lexicon-Morphosyntactic Specifications: Language-Specific Instantiation (Greek), PP-PAROLE, MLAP 63-386 report.

Markantonatou, S., Sofianopoulos, S., Spilioti, V., Tambouratzis, G., Vassiliou, M. & Yannoutsou, O. (2006). Using Patterns for Machine Translation (MT). In Proceedings of the European Association for Machine Translation (pp. 239--246). Oslo, Norway.

METIS II – Deliverable 5.2 (2007). Validation & Fine-Tuning Results for the First Prototype.

NIST (2002). Automatic Evaluation of Machine Translation Quality Using n-gram Co-occurrences Statistics (Available at www.nist.gov/speech/tests/mt/)

Papineni, K.A., Roukos, S., Ward, T. & Zhu, W.J. (2002). BLEU: A method for automatic evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 311--318). Philadelphia, USA.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas.

Tambouratzis, G., Sofianopoulos, S., Spilioti, V., Vassiliou, M., Yannoutsou, O. & Markantonatou S. (2006). Pattern matching-based system for Machine Translation (MT). In Proceedings of Advances in Artificial Intelligence: 4th Hellenic Conference on AI, SETN 2006, Vol. 3955 (pp. 345—355). Heraklion, Crete, Greece. Lecture Notes in Computer Science, Springer Verlag.

Thurmair, G. (2005). Improving MT Quality: Towards a Hybrid MT Architecture in the Linguatec ‘Personal Translator’. Talk given at the 10th MT Summit. Phuket, Thailand

Vandeghinste, V. (2005). Manual for ShaRPa 2.0. Internal Report. Centre for Computational Linguistics, K.U.Leuven.

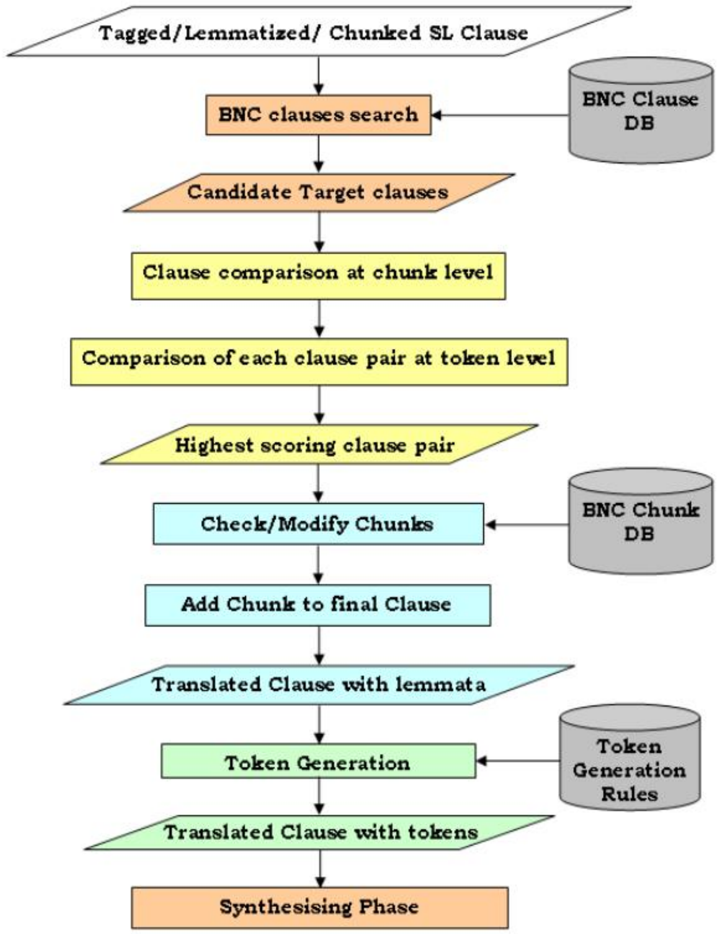

Figure 1: Clause Processing

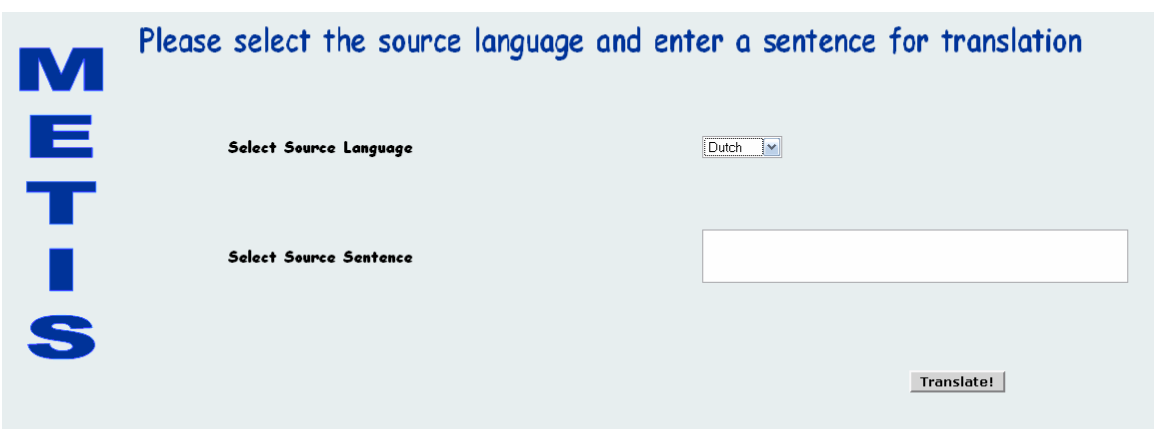

Figure 2: METIS-II home page

Source Clause: pp(np_nm(the greece**)) vg(**dismiss|fail|reject**) ppof(pp(np_ac(**any|anyone|each|every face|figure|form**)) pp(np_ge(**xenophobia**)))**

Final Clause: Greece rejects every form of xenophobia

Source Chunk:	pp([-{-}] np_nm(the{at} [greece{np0}]))
Corpus Chunk:	pp([-{-}] np_1([piatakov{NP0}]))
Final Chunk:	**pp([-{-}] np_1([greece{np0}]))**

Score=82.85715%	-{-}	the{at}	greece{np0}
-{-}	**-{-} 100.0%**	the{at} 0.0%	greece{np0} 0.0%
piatakov{NP0}	-{-} 0.0%	the{at} 0.0%	**greece{np0} 80.0%**
PAD	null 20.0%	**null 20.0%**	null 20.0%

- Processing Corpus chunk: pp([-{-}] np_1([Piatakov]))
- Replacing [piatakov{NP0}] with token:greece{np0}
- NOT Adding the{at}

Source Chunk:	vg([reject{vv}])
Corpus Chunk:	vg([reject{VVD}])
Final Chunk:	**vg([reject{vv}])**

Score=100.0%	reject{vv}
reject{VVD}	**reject{vv} 100.0%**

- Keeping chunk :vg([reject{vv}])

Source Chunk:	ppof(pp([-{-}] np_ac(any{dt0}|anyone{pn}|each{pn}|every{at} [face{nn}|figure{nn}|form{nn}])) pp([-{-}] np_ge([xenophobia{nn}])))
Corpus Chunk:	ppof(pp([as{PRP}] np_2(a{AT0} [form{NN1}])) pp([of{PRF}] np_2([dazzle{NN1}])))
Final Chunk:	**ppof(pp([-{-}] np_2(every{at} [form{nn}])) pp([of{PRF}] np_2([xenophobia{nn}])))**

Score=70.0%	-{-}	any{dt0}|anyone{pn}|each{pn}|every{at}	face{nn}|figure{nn}|form{nn}	-{-}	xenophobia{nn}
as{PRP}	**-{-} 70.0%**	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 0.0%	-{-} 70.0%	xenophobia{nn} 0.0%
a{AT0}	-{-} 0.0%	**every{at} 70.0%**	face{nn}!figure{nn}!form{nn} 0.0%	-{-} 0.0%	xenophobia{nn} 0.0%
form{NN1}	-{-} 0.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	**form{nn} 100.0%**	-{-} 0.0%	xenophobia{nn} 40.0%
of{PRF}	-{-} 70.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 0.0%	**-{-} 70.0%**	xenophobia{nn} 0.0%
dazzle{NN1}	-{-} 0.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 40.0%	-{-} 0.0%	**xenophobia{nn} 40.0%**

- Replacing [as{PRP}] with token:-{-}
- Replacing [a{AT0}] with token:every{at}
- Replacing [form{NN1}] with token:form{nn}
- Replacing [of{PRF}] with token:of{PRF}
- Replacing [dazzle{NN1}] with token:xenophobia{nn}

Figure 3: Step 4 output