An MT System Embedding Pattern Knowledge

Stella Markantonatou
Artemidos 6 & Epidavrou
151 25 Maroussi
Athens, Greece
marks@ilsp.gr

Sokratis Sofianopoulos
Artemidos 6 & Epidavrou
151 25 Maroussi
Athens, Greece
s_sofian@ilsp.gr

Vassiliki Spilioti
Artemidos 6 & Epidavrou
151 25 Maroussi
Athens, Greece
vspiliot@ilsp.gr

Marina Vassiliou
Artemidos 6 & Epidavrou
151 25 Maroussi
Athens, Greece
mvas@ilsp.gr

Olga Yannoutsou
Artemidos 6 & Epidavrou
151 25 Maroussi
Athens, Greece
olga@ilsp.gr

Abstract

In this paper, we explain why we have adopted pattern matching for MT purposes and why we have embedded it into a hybrid approach. "Patterns" here are understood as independent meaningful sub-sentential segments received in a systematic way. We describe the nature and size of the patterns used as well as the comparison algorithm developed. We discuss results obtained by matching patterns of different types and complexity in four different language pairs. Our experiments indicate that better results are obtained when matching the longest possible patterns.

1 Introduction

Corpus driven MT, based on bilingual corpora, whether aligned or not, currently dominates research in the field. On the other hand, MT more and more adopts hybrid techniques to establish correspondences across language pairs and for generation purposes. The term 'hybrid techniques' refers to a range of techniques (statistical algorithms, rules etc) which are combined with linguistic resources of varied complexity to establish the required correspondences across language pairs and generate TL (target language) strings.

(Nagao, 1984) put forward the idea that a large amount of knowledge about translating between two languages is encoded in parallel texts of the particular language pair. Parallel corpora were aligned at sentence level. Input sentences were compared with the sentences on the SL (source language) part in order to retrieve the most similar one in the parallel corpus. This is how translation memories (TMs) and Example-Based MT (EBMT) came to stage.

Very soon, approaches based on bilingual corpora became the norm in the field. To this situation contributed the seminal work of Brown et al. (1990), who first introduced a promising MT approach that relied on (a) a word-to-word alignment of parallel corpora and (b) statistically obtained language models. It was expected that, by using the appropriate techniques, knowledge in the language strings would be extracted and reused and that human intervention would be rendered unnecessary, human intervention being limited to the employment of linguistic knowledge.

However, what actually happened is that work on TMs and MT started to converge with conclusions reached by researchers working on the cognitive aspects of translation. Human translators were already known to work with sub-sentential units (Gerloff, 1987). Work within TMs demonstrated that sub-sentential units increase similarity scores in TMs (Cranias et al., 1997), when comparing the input sentence with the SL corpus. In particular, it was shown that best similarity scores were obtained, when the sub-sentential units were syntactically 'complete' (and, by the same token, independent and meaningful) and comparison took into account the contained content words. In order to obtain meaningful units without resorting to parsing, Cranias et al. (1997) relied on functional words –

with the exception of articles – to define borders. The use of sub-sentential units was also advocated by members of the SMT community. Yamada and Knight (2001) for instance have used statistical models for structurally analysed sentences (adopting a "syntax-based translation model") in their effort to handle word order issues.

The corpus-based approach presented here draws on the tradition described above, but it is innovative in certain important ways:

a. It uses solely monolingual TL corpora, while no SL corpus is required; this is a radical answer to the problem of bilingual corpora sparseness, for the vast majority of language pairs, as well as the dubious quality of the existing ones for translation purposes, which constitute one of the major drawbacks of corpus-based approaches to MT.

b. It relies on a general pattern matching algorithm, for handling sub-sentential corpus material, together with statistical techniques in order to select certain TL word affiliations (Tambouratzis et al., 2006).

c. Patterns are generated by using basic NLP tools (lemmatiser, tagger, chunker) and resources (flat bilingual lexica). This technology is often available and, in any case, relatively easily obtained; therefore the application of the method is feasible for a large number of language pairs.

In this paper, we explain why we have adopted a hybrid approach, employing pattern matching techniques over independent meaningful phrasal units received in a systematic way. We describe the types of the patterns used, as well as the comparison algorithm developed. Then we present results received by matching patterns of different complexity. Our experiments indicate that results are improved when matching the longest possible patterns that function as syntactic constituents.

2 Why Pattern Matching

In parallel corpus-based MT, patterns are very often viewed as strings of fixed length with or without slots for variables ranging over words (see, among others, Lepage, 1997; McTait, 2003; Brown, 2003; Kitamura, 2004). Variables are instantiated on the basis of corpus information. This notion of a pattern makes sense in a parallel corpus-based approach, where the exact length of a (pattern) string and the exact place of variables can be defined.

In our monolingual TL corpus-based approach, the problem to be solved corresponds to the similarity problem in TMs. However, in our case, comparisons are not performed on the SL side, but on the TL one: the system is fed with flexible 'models' of TL strings, which receive their final form (i.e. translated) only after the corpus has been consulted.

More formally, translation is viewed as an assignment problem (i.e. finding a maximum weight matching), which the pattern matching algorithm is called to solve. Translation units (chunk patterns in the current application) are assigned weights, which are compared with the aim of detecting the highest similarity scores (i.e. the best matching patterns). Pattern matching checks whether patterns have the desired structure. It computes the pattern similarity and substitutes, when necessary, the non-matching parts.

Like statistical techniques, pattern matching has been successfully used in other domains such as Voice Recognition (VR), Optical Character Recognition (OCR), Biometrics (face recognition, finger prints recognition, speech recognition) and Image Segmentation and Analysis (object recognition, medical diagnosis using X-Rays, EKG analysis).

We considered pattern matching appropriate for MT purposes because it handles unit (word or phrase) selection and order simultaneously. Statistics-based techniques do not perform both tasks in one step (Yamada and Knight, 2001). The core idea was that, if a lexicon was available, providing us with word translations, as well as a large corpus, where all grammatical word combinations occurred, we would only need to select the proper words in the right order. If the right words co-occurred in a TL sentence, they would obviously be in the right order, excluding the possibility of finding ungrammatical strings in the corpus. Thus, with a flat bilingual lexicon, a large TL corpus and a pattern matching methodology we would have a complete system. No rules or other extravagant linguistic information would be necessary.

The validity of this idea was demonstrated with METIS-I (Dologlou et al., 2003), where patterns corresponded to TL sentences rather than to TL sentence segments (phrases). Ideally an MT system would translate sentences as, in this way, both meaning and relations among words are minimally disturbed. Indeed, METIS-I

selected the best sentence among very similar ones.

Obviously, METIS-I only provided proof-of-concept, as the problem lies with the corpus: it can never be big enough to contain all the possible grammatical word combinations. The natural way of dealing with the data sparseness problem was to take advantage of the recursive nature of language: sentences can be broken to phrases and phrases can be broken to a limited number of types of smaller phrasal units. Smaller phrasal units consist of few words, therefore, the chance of achieving good (partial, probably) matches is greater, rendering thus the problem of data sparseness manageable.

The choice of working at sub-sentential level was supported by two more facts: (a) as said before, it was only common wisdom given the results of existing experience in SMT, TMs and cognitive studies; (b) pattern matching could be applied iteratively throughout the procedure (somehow simulating the recursive nature of language). The same core algorithm would be used to perform comparisons both at clause level (when comparing the constituent phrases) and lexical level (when comparing the 'content' of phrases). As with sentences, phrases presented us with a similarity problem.

3 Why a hybrid approach

A hybrid approach was adopted for the reasons explained below.

First of all, we used flat bilingual lexica (as already explained). We had to do so because no bilingual corpus was available to introduce mapping knowledge to the system. However, the use of lexica in corpus-based MT is a rather common requirement; similar practices have been reported for approaches using bilingual corpora, for instance in SMT (Ney, 2005).

Certainly, acquisition of patterns was the major issue. What kind of patterns we would use such that could be obtained automatically? In line with our argument about data sparseness, we have decided to use conventional taggers to acquire patterns at PoS level, lemmatisers to generalise over morphological paradigms (and reduce data sparseness) and chunkers to split up sentences into noun, verb, adjectival and prepositional chunks in both SL and TL. The chunkers used are rule-based (but of course, one can think of approaches where machine learning techniques are applied to annotated corpora).

Our decision to use the patterns that were generated by the stepwise application of a tagger, a lemmatiser and a chunker to both the TL corpus and the input SL sentence was motivated by the following facts:

a. Patterns would be of a very general type which would allow for a powerful algorithm applying to a very wide range of data. Sentences would consist of a central unit (the verbal 'head' as a rule) and its satellite units. Those, in turn, would consist of a central word (again, the 'head') and its satellite words. The study of languages has shown that this is a wide spread pattern for putting words together into sentences.
b. Sub-sentential units would be meaningful and this was advantageous according to Cranias et al. (1997). Actually, the units produced by the chunker were just a finer version of the units produced by Cranias et al. (1997).
c. We could take advantage of selection phenomena among chunk heads.
d. It would help to reduce to a minimum the friction phenomena between selected TL chunks.

As regards point (a) above, the discrepancies between the SL and the corresponding TL ones, even after lemmatisation, may vary between language pairs, for instance, the presence or absence of the constituent VP in fixed word order languages, such as English, as opposed to relatively free word order ones, such as Modern Greek, pro-drop phenomena etc. Such phenomena are among the ones that pave the way to rule-based transformations. However, although we did not abstain from using transformation rules, we wanted to keep their number to a minimum and resort to them only when nothing else (simple enough, in line with the modest resource requirement) could be done.

The following questions were put forward in order to develop an approach general enough to handle more than one language pair:

1. How much detailed should the optimum sentence segmentation be?
2. Which type of information should be introduced with transformation rules in order to provide the pattern matching algorithm with TL-like patterns (rather than SL-like ones)?
3. Which other techniques could be used?

In the remainder of this paper, we will take up issue (1) above and use material from our ex-

periments to explain that, for the range of language pairs we have treated, better results are obtained when nominal patterns are maximally defined.

As regards point (2), the results obtained so far show that only discrepancies non-reducible to a word-order problem or to a local selection one (e.g. the syntax of "like" verbs) should be treated with transformation rules in order to facilitate the pattern matching algorithm. The results discussed below have been obtained by employing only a limited number of transformation rules (e.g. the formation of infinitival chunks for Greek, a language lacking infinitives). However, a wider range of data is required in order to present a typology and an estimation of the number of the necessary transformation rules.

As regards point (3), we have decided to employ statistical, frequency-of-occurrence-based techniques to optimise lexical selection for those cases where the pattern matching algorithm may not be able to disambiguate between the various translations.

4 The chunk patterns

As mentioned above, both SL and TL models comprise chunks and their respective constituents, generated by equivalent chunkers. Originally (Tambouratzis et al,. 2005), for modelling both languages we used only a very small number of chunk patterns. Thus, for both SL and TL the chunk patterns used are: the Clause pattern, the VG Pattern, the PP pattern, the ADJ pattern and the INP pattern.

Clause pattern

(ADJ* PP* token*)* VG (ADJ* PP* INP* token*)* [where 'token' refers to adverbials or punctuation]

The **Clause pattern** describes the overall structure of a clause: the verbal group head and the number, labels and heads of the chunks (if any chunks exist other than the verb group).

The **VG pattern** describes the verb group. Other tokens, such as adverbs for example, if found within the verb phrase are considered as part of it, while if found in isolation, do not form a chunk.

The **INP pattern** describes the infinitival chunks. In Modern Greek, a language that lacks infinitives, an INP pattern is formed by merging a *na* subordinate clause with the matrix verb.

The **ADJ pattern** describes the adjectival chunks headed by an adjective.

The **PP pattern** describes both prepositional and noun chunks. The generalisation here is that a noun chunk can be represented as a prepositional one with an empty prepositional head. This representation captures phrase category mismatches between SL and TL of the sort exemplified in the following example:

$[_{pp}$ ∅ $[_{np\_nm}$ ο διαρρήκτης]] $[_{vg}$ προσπάθησε]

[$_{pp}$ ∅ [$_{np1}$ the burglar]] [$_{vg}$ tried]

$[_{inp}$ να $[_{vg}$ μπει]] $[_{pp}$ **στο** $[_{np\_ac}$ σπίτι]]

[$_{inp}$ to [$_{vg}$ enter]] [$_{pp}$ ∅ [$_{np2}$ the house]]

The selection of the best matching TL clause is carried out in two steps. At the first step, comparison is performed at clause level using as clause pattern comparison features the chunk labels and chunk head tokens (lemma and PoS tag). This step establishes the order of chunks in the SL clause pattern by using the specific TL clause as a template, and each SL chunk is directly mapped to its corresponding TL chunk.

At the second step, the comparison is confined within the boundaries of the chunk patterns, in order to establish the order of the tokens within each chunk. The comparison features used are the chunk tokens in terms of their lemma and PoS tag.

For both steps, the same pattern matching algorithm is employed, each time evaluating different pattern features. In Table 1 (see Appendix A) we can see the first step of a clause comparison between an input Modern Greek (MG) clause (in lemmatised form after the lexicon lookup[1]) and an English one (lemmatised, as retrieved from the corpus):

Input (MG) clause: *the girl often describe several close member of the royal family as a gift by the god*

English clause: *The Pakistani village of Mohinuddinpur was described to me by Shamim, a woman*

As can been seen in Table 1, if this English clause was to be selected as the template for the final translation, then the chunk order of the MG clause would be changed to:

the girl of the royal family often describe several close member by the god as a gift

The assignment algorithm used for the comparison process allows all chunks to change position within the clause according to their mapping to the TL template. Certain chunks, however,

[1] For simplicity, only one translation has been assigned to each token

should never be split, for instance the prepositional ones *pp(- np_ac(several{DT0}close{AJ} member{NN}))* and *pp(of{PRF} np_ge(the{AT0} royal{AJ }family{NN}])).*

In order to tackle the issue raised, we introduced two new complex chunk patterns, the PPOF pattern and the PPOS pattern. The PPOF pattern is used in both SL and TL models, while the PPOS chunk pattern is language-specific, used only in English.

The **PPOF pattern** [PPOF (PP PP_gen)] describes the combination of a noun chunk with its Genitive post-modifier (np_ge).

The **PPOS pattern** [PPOS (PP 's PP)] describes a noun chunk pre-modified by a Saxon Genitive.

The chunk patterns above are considered to be equivalent during the matching process. For instance, the SL PPOF chunk *the map of the city* can be equally mapped either to the PPOF chunk *the hall of the city* or the PPOS chunk *the city 's hall*, both found in the TL corpus.

The introduction of the new complex chunk patterns prevents the splitting of nominal chunks and their modifiers, while reducing the number of chunks within the clause, having, thus, the advantage of comparing smaller matrices, even if we need to define new and more complex comparison processes for the new chunk patterns.

The results of the introduction of more complex chunks can be seen in the following example. Using the same SL clause, but having applied the new chunk patterns, we achieve a smoother transition from the SL model to the TL one.

Input (MG) clause: *the girl often describe several close member of the royal family as a gift by the god*

English clause: *Ann Messenger describes the condition of women writers in the seventeenth and eighteenth centuries in very different terms*

As can been seen in Table 2 (see Appendix A), a different English clause is selected as the template for the final translation, establishing hence the desired (right) word order.

In Tables 3 and 4 (see Appendix A), we present an equivalent example for the language pair Spanish→ English, the SL string being (after the lexicon lookup) *Certain part of the state united be as poor as the third world*. Without the complex patterns the nominal chunk is severed from its post-modifier (*Certain part be as poor as the third world **of the** united state*), while the introduction of complex chunks leads to a correct chunk order (*Certain part **of the** united state be as poor as the third world*).

This example also illustrates the establishment of the correct token order by the pattern matching algorithm (cf. state united vs. united state).

The use of broader nominal patterns has improved the performance of the system in three language pairs: Modern Greek, Dutch and German → English. The results that were derived with the BLEU and NIST metrics are reported in Table 5 refer to a set of 15 sentences for each language pair.

	No complex chunks		**With complex chunks**	
SL	**BLEU**	**NIST**	**BLEU**	**NIST**
Dutch	0.4022	5.7545	**0.4865**	**6.0209**
German	0.4737	5.7340	**0.4816**	**5.7875**
Greek	0.5432	6.6295	**0.6541**	**6.8646**
Spanish	**0.5676**	**6.6805**	0.5378	6.6213

Table 5: BLEU & NIST scores

5 Future Work

Presently, we are working on the optimisation of our system along the following lines:

- Improving the corpus indexing scheme and narrowing down the search space
- Accelerating the search process and improving its effectiveness
- Exploring further the issue of synthesising the final translation from multiple segments (clauses)
- Employing machine learning techniques

References

Peter Brown, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Fredrick Jelinek, John Lafferty, Robert Mercer and Paul Roosin. 1990. *A Statistical Approach to Machine Translation.* Computational Linguistics, 16(2):79-85.

Ralf Brown. 2003. *Clustered Transfer Rule Induction for Example-Based Translation.* In M. Carl & A. Way (eds.) Recent Advances in Example-Based Machine Translation, Kluwer Academic Publishers: 287-305.

Lambros Cranias, Harris Papageorgiou and Stelios Piperidis. 1997. *Example Retrieval from a Translation Memory.* Natural Language Engineering, 3:255-277.

Ioannis Dologlou, Stella Markantonatou, George Tambouratzis, Olga Yannoutsou, Athanassia

Fourla and Nikos Ioannou. 2003. *Using Monolingual Corpora for Statistical Machine Translation*. In "Proceedings of EAMT/CLAW 2003", Dublin, Ireland, 15-17 May.

Pamela, Gerloff. 1987. *Identifying the Unit of Analysis in Translation.* In G. Faerch and G. Kasper (eds): Introspection in Second Language Research, Clevedon: Multilingual Matters: 135-158.

Mihoko Kitamura. 2004. *Translation Knowledge Acquisition for Pattern-Based Machine Translation*. PhD, Nara Institute of Science and Technology, Japan.

Yves Lepage. 1997. *String approximate pattern-matching.* In Proceedings of the 55th Meeting of the Information Processing Society of Japan, Fukuoka, August 1997: 139-140.

Kevin McTait. 2003. *Translation Patterns, Linguistic Knowledge and Complexity in EBMT*. In M. Carl & A. Way (eds.): Recent Advances in Example-Based Machine Translation, Kluwer Academic Publishers 307-338.

Makoto Nagao. 1984. *A Framework of a Mechanical Translation between Japanese and English by Analogy Principle.* In "Artificial and Human Intelligence", A. Elithorn and R. Banerji (eds). North-Holland.

Herman Ney. 2005. *One Decade of Statistical Machine Translation*: 1996-2005. In Proceedings of the 10th MT Summit, September 12-16, Phuket, Thailand, i12-i17.

George Tambouratzis, Sokratis Sofianopoulos, Vassiliki Spilioti, Marina Vassiliou, Olga Yannoutsou, and Stella Markantonatou. 2006. *Pattern Matching-Based System for Machine Translation (MT)*. In G. Antoniou et al. (Eds.): *SETN 2006, LNAI 3955*, Springer-Verlag Berlin Heidelberg, pp. 345 – 355.

Kenji Yamada and Kevin Knight. 2001. *A syntax-based statistical translation model*. Proceedings of 39th Annual Meeting of the Association for Computational Linguistics ACL: 523-530.

Appendix A. Tables

	pp(- np_nm(the{AT0} girl{NN}))	vg(describe{VV})	pp(- np_ac(several{DT0}close{AJ} member{NN}))	pp(of{PRF} np_ge(the{AT0} royal{AJ}family{NN}]))	pp(as{PRP} np_ac(a{AT0} gift{NN}))	pp(by{PRP} np_ac(the{AT0} god{NN}))
PP(- NP1(the{AT0} pakistani{AJ0} village{NN1}))	**79%**	0%	68%	61%	68%	61%
PP(of{PRF} NP2(mohinuddinpur{NP0}))	40%	0%	78%	**78%**	78%	78%
VG(be{VBD} [describe{VVN}])	0%	**100%**	0%	0%	0%	0%
PP(to{PRP} NP2(me{PNP}))	31%	0%	**63%**	70%	63%	70%
PP(by{PRP}] NP2(shamim{NP0 }))	40%	0%	78%	78%	78%	**79%**
PP(- NP2(a{AT0} woman{NN1}))	47%	0%	79%	78%	**79%**	78%

Table 1. Greek →English clause comparison without complex chunks

Score=85.034485%	pp(- np_nm(the{AT0} girl{NN}))	vg(describe{VV})	ppof(pp(- np_ac(several{DT0} close{AJ} member{NN})) pp(- np_ge(the{AT0} royal{AJ} family{NN})))	pp(as{PRP} np_ac(a{AT0} gift{NN}))	pp(from{PRP} np_ac(the{AT0} god{NN}))
PP(- NP_1(ann{NP0} messenger{NN1}))	**84%**	0%	21%	46%	46%
VG(describe{VVZ})	0%	**100%**	0%	0%	0%
PPOF(PP(- NP_2(the{AT0} condition{NN1})) PP(of{PRF} NP_2(woman{NN2} writer{NN2})))	21%	0%	**88%**	45%	45%
PP(in{PRP} NP_2(the{AT0} seventeenth{ORD} and{CJC} eighteenth{ORD} century{NN2}))	46%	0%	45%	**83%**	83%
PP(in{PRP} NP_2(very{AV0} different{AJ0} term{NN2}))	46%	0%	45%	83%	**83%**

Table 2. Greek →English clause comparison with the employment of complex chunks

	pp(- np_nm(certain{AJ0} part{NN}))	pp(of{PRF} np_ac(the{AT0} State{NN} united{AJ0}))	vg(be{VB})	pp(as{PRP} np_ac(poor{AJ0}))	pp(as{PRP} np_ac(the{AT0} third{ORD} world{NN}))
PP(- NP_1(the{AT0} aircraft-carrier{AJ0-NN1} centaur{NN1}))	**84%**	46%	0%	27%	46%
VG(be{VBD})	0%	0%	**100%**	0%	0%
PP(in{PRP} NP_2(the{AT0} in-dian{NP0} ocean{NP0}))	40%	77%	0%	**64%**	77%
PP(on{PRP} NP_2(passage{NN1}))	46%	**83%**	0%	64%	83%
PP(to{PRP} NP_2(the{AT0} far{AJ0} east{NN1}))	46%	83%	0%	64%	**83%**

Table 3. Spanish →English clause comparison without complex chunks

	ppof(pp(- np_nm(certain{AJ0} part{NN})) pp(of{PRF} np_ge(the{AT0} State{NN} united{AJ0})))	vg(be{VB})	pp(as{PRP} np_nm(poor{AJ0}))	pp(as{PRP} np_ac(the{AT0} third{ORD} world{NN}))
PPOF(PP(- NP_1(this{DT0} kind{NN1})) PP(of{PRF} NP_1(fundamentalism{NN1})))	**69%**	0%	0%	20%
VG(be{VBZ})	0%	**100%**	0 %	0 %
ADJP_1(identifiable{AJ0})	0%	0%	**0%**	0 %
PP(in{PRP} NP_2(the{AT0} world{NN1}))	20%	0%	0%	**99%**

Table 4. Spanish →English clause comparison with the employment of complex chunks