D4.1.2: STRUCTURE SELECTION MODULE (VER.2)

Grant Agreement number	ICT-248307
Project acronym	**PRESEMT**
Project title	Pattern **RE**cognition-based **S**tatistically **E**nhanced **MT**
Funding Scheme	Small or medium-scale focused research project – STREP – CP-FP-INFSO
Deliverable title	**D4.1.2: Structure selection module (ver.2)**
Version	**6**
Responsible partner	**ILSP**
Dissemination level	Restricted
Due delivery date	31.12.2011 (+ 60 days)
Actual delivery date	17.1.2012

Project coordinator name & title	**Dr. George Tambouratzis**
Project coordinator organisation	**Institute for Language and Speech Processing / RC 'Athena'**
Tel	+30 210 6875411
Fax	+30 210 6854270
E-mail	**giorg_t@ilsp.gr**
Project website address	**www.presemt.eu**

Contents

Figures

Tables

List of abbreviations	
ACS	Aligned corpus sentence
ISS	Input source sentence
MT	Machine Translation
PoS	Part of Speech
SL	Source Language
SSM	Structure selection module
TL	Target Language
GA	Genetic Algorithm
PSO	Particle Swarm Optimisation
SPEA2	Strength Pareto Evolutionary Algorithm

1. Executive summary

Deliverable D4.1.2 reports on the work carried out in **WP4: Structure selection.** The objective of WP4 is "*to design and implement the first phase of the machine translation process. This involves defining the basic pattern of each sentence (in terms of structure) in the target language, by making use of the limited-size bilingual corpus to define correspondences*".

WP4 is divided into two tasks, *T4.1: Design and implementation of the structure selection module*, which "*focuses on developing a module which uses pattern recognition principles to determine for a given source language sentence the best-matching structure, by utilising the information contained in a small parallel corpus, and correspondingly transform it*" and *T4.2: Optimisation of module-specific parameters.*

In the PRESEMT architecture, the Structure selection module is the only module that has direct access to bilingual resources, namely the aligned bilingual corpus and a bilingual dictionary. The key idea is to account for major structural differences between source (SL) and target (TL) languages based on the linguistic patterns found in the bilingual corpus. To this end, two methods are under investigation, based on dynamic programming and the use of a synchronous grammar respectively. Both aim at providing the appropriate TL structure for a given ISS in terms of phrase order.

A series of prototypes has been implemented that exemplify different variants of the Structure selection module.

The deliverable has the following structure: Section 2 provides a concise summary of the key design features of the PRESEMT approach with respect to the Structure selection module. Section 3 discusses the properties of the aligned bilingual corpus as the major linguistic resource for the module. Section 4 discusses dynamic programming as a method to exploit the contrastive linguistic knowledge encoded in the aligned bilingual corpus, while Section 5 presents synchronous grammars as an alternative methodology. Section 6 outlines the parameters of the Structure selection module that will undergo optimisation. The references (Section 7) conclude the deliverable.

2. Introduction: Summary of the PRESEMT approach

A major objective of the PRESEMT project is to develop a machine translation system, which can be easily extended to new language pairs in a simple manner, while allowing the user to extensively modify an existing language pair so that it better matches their requirements. To this end the PRESEMT project uses readily available linguistic resources to the greatest possible extent and avoids the time-consuming and cost-intensive development of specialised linguistic resources and tools.

Tools that are readily available for most languages are statistical taggers and (to some extent) chunkers that provide shallow linguistic structures. Large monolingual corpora are also readily available through the World Wide Web. More difficult to obtain are large bilingual corpora which are commonly used by statistical machine translation systems. The novel idea of PRESEMT is to develop a hybrid MT system that gets by with a small bilingual corpus supplemented by a large monolingual corpus for the target language.

The Main translation engine of PRESEMT consists of two phases, the **Structure selection module** (Phase 1), which makes use of a small bilingual corpus, and the **Translation equivalent selection module** (Phase 2) that relies on a large monolingual target language corpus.

The input to the Structure selection module is the output of the Phrasing model generator (see Deliverables D3.2.1 and D3.2.2), namely tagged-lemmatised and chunked SL sentences[1].

The resources used by the Structure selection module are an aligned bilingual corpus and a bilingual dictionary. An additional resource is a token generation table for target language lemmas.

The output of the Structure selection module is a (set of) TL structure(s) with tagged and chunked TL lemmas or tokens. The most important task of the Structure selection module is to determine the TL phrases and phrase order. Other micro-level phenomena such as word order within phrases can be handled later, within the Translation equivalent selection module.

The methods employed by the Structure selection module aim at exploiting the linguistic patterns exemplified in the aligned bilingual corpus. Currently two different methods have been developed, based on **dynamic programming** (Section 4) and **synchronous grammars** (Section 5) respectively:

(a) Dynamic Programming

1. A dynamic programming technique handles an input source sentence (ISS), which is annotated with information about tag, lemma, phrase and translation equivalents.
2. The algorithm determines for each ISS the most similar SL sentence (ACS) found in the aligned bilingual corpus in terms of phrases. Through the phrase similarity calculation the algorithm also returns alignments at phrasal level between the two sentences.
3. The phrases of the ISS are then reordered according to the TL structure attached to the most similar SL sentence of the bilingual corpus using the SL-TL alignment information stored in the bilingual corpus.

[1] The clause type of the target language (e.g. main versus subordinate clause) is needed if the structural conversion from source language to target language needs to make reference to the clause type or clause boundaries. The structural conversion from English to German is a case in point. For example, in order to account for the different positions of verbs in English and German reference has to be made to the clause type and clause boundaries.
If the source and target languages are structurally sufficiently similar or if the clause type does not play any role for the TL word order, then a generic clause type such as CL for "clause" can be assigned for both main and subordinate clauses. Similarly, if clause boundaries are not relevant for determining the TL word order, then clause boundaries need not be specified either.

(b) Synchronous grammars

1. **Pre-processing:** Bilingual productions that map SL structure onto TL structure are automatically extracted out of the aligned bilingual corpus.

2. **Runtime:** A parser for synchronous grammars converts the input SL structure into a TL structure based on the bilingual productions.

   The bilingual dictionary lookup is part of the synchronous parsing. The lexical productions that map SL lexemes onto TL lexemes are generated dynamically at runtime.

Dynamic programming search can be termed as a monolingual search algorithm, because it compares structures of the same language, namely SL structures to SL ones. The most similar SL structure of the bilingual corpus, that determines the TL structure of the sentence to be translated, is selected purely on SL properties.

Synchronous grammars can be considered a bilingual search algorithm, since in its search for the best TL structure both SL and TL properties are taken into account. So for example the bilingual lexicon lookup is part of the parsing and the type of TL lexical material associated with a given SL lexeme determines which TL structures are built up.

3. Aligned bilingual corpus

3.1 Linguistic information represented in the aligned bilingual corpus

The aligned bilingual corpus consists of a few hundred sentence pairs. The sentences are lemmatised, tagged, chunked and clause-chunked and aligned on word, phrase and clause level. The structure provided by the shallow parsers is a flat one.

The aligned bilingual corpus contains contrastive linguistic information on the following levels:

Table 1: Linguistic information residing in the aligned bilingual corpus

Tag mappings	Phrase mappings	Clause mappings	Structural identity	Structural differences
Mappings from SL tags onto TL tags	Mappings from SL phrases onto TL ones. The phrase types of the aligned SL and TL phrases are predicted to be predominantly the same, yet there are cases of category shift	SL-TL mappings at clause level	The aligned bilingual corpus can be used to read off which sequences (or subsequences) of phrases are identical in SL and TL.	The aligned corpus can also be used to read off which sequences (or subsequences) of phrases differ in terms of order, number or type in SL and TL. The observed structural differences can be termed phrase reordering, phrase splitting / merging, insertion / deletion of phrases and category shift (change of phrase type).

3.2 Linguistic information needed for the translation process

For the translation process the most important information needed from the aligned bilingual corpus is which phrase patterns undergo structural changes in the target language and which ones retain the phrase structure of the source language. Thus, the challenge is to determine the properties of (SL) language that trigger a structural change and to abstract away from the properties of language that do not trigger a structural change.

3.3 Productivity of language

The aligned bilingual corpus contains a fixed number of sentences and even if it exemplifies all structural differences between languages in some way, the challenge remains to account for the productivity of language, i.e. the fact that there is an infinite number of potential input sentences for which the appropriate TL structure has to be found. One way to do this is to combine information from different clauses to determine the best matching SL structure and the most suitable TL structure.

3.4 Extending the coverage of the bilingual corpus

3.4.1 Partitioning phrase sequences according to cross-linguistic criteria

In order to account for the productivity of language and to extend the patterns found in the bilingual corpus the basic idea is to partition the sentential phrase sequences into smaller phrase sequences. The partitioning criterion is which phrase sequences undergo a structural change and which ones do not.

To this end, both the SL side and the TL side of the aligned bilingual corpus are scanned for structural changes. Thus, the phrase sequences that run parallel to each other in SL and TL can be distinguished from phrase sequences that differ in terms of phrase order, number or type.

For the phrase sequences that exhibit identical structures in SL and TL, simple one-to-one phrase mappings are sufficient to determine the TL chunking. For the phrase sequences that exhibit structural differences more complex phrase mappings are necessary.

In the following examples A, B and C are phrase types[2], while the numbers represent phrase alignment.

Figure 1: Schematic example of structural identity between SL and TL

SL:	A1	B2	A3	C4
TL:	A1	B2	A3	C4

The phrase sequences in Figure 1 are identical in SL and TL, therefore simple, one-to-one mappings of phrases are sufficient to generate the TL structure (where the mapping from SL to TL is denoted by an arrow):

A1 -> A1

B2 -> B2

C4 -> C4

Figure 2: Schematic example of phrase reordering

SL:	A1	B2	C3
TL:	A1	C3	B2

In Figure 2, phrase B and phrase C are reordered in the TL. Phrase reordering is accounted for by the following many-to-many phrase mapping:

B2 C3 -> C3 B2

Figure 3: Schematic example of phrase splitting

SL:	A1	B2	A3	C4	D5	
TL:	A1	B2	A3	C4	B2	D5

In Figure 3 phrase B is split in the TL. Phrase splitting is accounted for by the following complex phrase mapping:

B2 A3 C4 -> B2 A4 C5 B2

Or more generally:

[2] Since the TL phrase structure is projected onto the SL sentences, the phrase names of equivalent phrases in SL and TL are the same.

B1 X2 -> B1 X2 B1

where X is a variable ranging over a sequence of phrases. It is left open here whether and how the sequence of phrases that variable X covers is restricted.

The approach outlined so far ignores some fine points about the conditions under which structural changes between SL and TL apply.

3.4.2 Conditions for structural differences

Structural differences between phrase structures in SL and TL often depend on more than just the sequence and type of the phrases involved. Other properties that might be relevant are the head of the phrase, the type of clause (main versus subordinate clause) and the number and type of phrases preceding or following the phrase sequence. Some structural differences even depend on semantic information and should be restricted to certain lexical items such as negative polarity expressions. To provide a few examples, consider the following translation pairs taken from English and German.

The translation of an English verbal chunk (VC) depends on many factors; sometimes it is split into two VCs in German, sometimes it is moved to the end of the German clause, and sometimes it remains in the same position in German. The conditions are quite complex and involve several criteria.

The English VC is split in German if it contains a finite auxiliary verb plus some other verb form and if the VC is part of a main clause:

(1) They **have accepted** the offer.

Sie **haben** das Angebot **akzeptiert**.

The English VC is reordered to the end of the clause if it is part of a subordinate clause (and the elements of the VC are reordered):

(2) because they **have accepted** the offer.

weil sie das Angebot **akzeptiert haben**.

English VCs consisting of one verb form are not split in German unless the German translation equivalent is a separable prefix verb:

(3) They **accepted** the offer.

Sie **akzeptierten** das Angebot. ("*akzeptieren*" is a simple verb)

Sie **nahmen** das Angebot **an**. ("*annehmen*" is a separable prefix verb)

Structural differences between languages can also depend on the number and type of phrases preceding them. For example German imposes the restriction that only one constituent can precede the finite verb form in main clauses, whereas English allows two constituents to precede the finite verb form:

(4) **Then they** accepted the offer.

Dann akzeptierten **sie** das Angebot.

The condition for reordering the subject is further complicated by the fact that shallow parsing, which is used in PRESEMT, does not distinguish modifiers of arguments from sentential modifiers, or, more precisely, the notion of constituency is not well-reflected in shallow parsing. This becomes a problem if constituency plays a role in structural changes or conditions for structural changes.

In the following sentence pair there is no reordering because the expression "*The representatives in Sweden*" is one constituent although this fact is not reflected in the phrase sequence:

(5) The representatives in Sweden **accepted** the offer.

Phrase sequence: **NC PC** *VC NC*

Die Repräsentanten in Schweden **akzeptierte** das Angebot.

In (6) however the reordering is obligatory, since the expression "*In Sweden the representatives*" comprises two constituents:

(6) In Sweden **the representatives accepted** the offer.

Phrase sequence: **PC NC** *VC NC*

In Schweden **akzeptierten die Repräsentanten** das Angebot.

Different types of structural differences can also be combined in intricate ways. For example, phrase splitting and phrase reordering are combined in (7). If the tense is a simple one, then German separable prefix verbs and simple verbs behave in the same way:

(7) Then **they have accepted** the offer.

Dann **haben sie** das Angebot **akzeptiert**.

Dann **haben sie** das Angebot **angenommen**.

If the English finite verb is simple, then the splitting and reordering behaviour depends on the translation equivalent, since simple German finite verbs do not split, while separable prefix verbs do:

(8) Then **they accepted** the offer.

Dann **akzeptierten sie** das Angebot.

Dann **nahmen sie** das Angebot **an**.

By now it should be obvious that the conditions for structural differences can be quite complicated. The bilingual corpus is probably not large enough to capture all the conditions for structural differences within a given language pair. Thus the following avenues are left for dealing with the conditions for structural differences:

1. Ignore conditions on structural differences: One could produce all possible TL structures and leave it to the Translation equivalent selection module to determine the grammatical ones.

2. Use a maximum set of conditions for all structural differences.

3. View conditions on structural differences as a parameter that is optimised by the optimisation functionality.

At the time being option 2 is chosen, namely all patterns that account for a structural change are accompanied by the same set of conditions, namely clause type, phrase type of the elements before and after the phrases that undergo a change and the head of VC. These conditions are too strict in many cases and lead to a reduction of the number of cases covered.

It is envisaged to also explore option 3, namely treating the conditions as a parameter that can be optimised. However, option 1 is also an interesting one, for which at least an experiment should be set up.

3.5 Design of the bilingual corpus

The bilingual corpus should fulfil the following criteria, when employed in a synchronous grammar-based approach:

1. The corpus size should amount to a few hundred sentences.
2. The bilingual corpus should be preferably collected from a bilingual or multilingual website. It should be representative, authentic text. The purpose of this type of corpus is to have a representative coverage of the linguistic phenomena to be accounted for.
3. Structural differences between languages should be illustrated, preferably in relatively short sentences that contain the characteristic properties of the linguistic phenomena.

For the purposes of the Structure selection module it is not necessary that the alignment or the tagging and chunking of the bilingual corpus are manually corrected. Minor misalignments or errors in the tagging are expected to be filtered out by frequency counts.

4. Dynamic programming

In the first approach for structure selection, using an algorithm from the dynamic programming paradigm, the structure selection process is treated as a sequence alignment, aligning the input source sentence (**ISS**) to an SL sentence from the aligned corpus (**ACS**) and assigning a **similarity score**. The similarity score is calculated by taking into account the edit operations (replacement, insertion or removal) needed to be performed to the ISS elements in order to transform it to the ACS. Each of these operations has an associated cost, considered as a system parameter. The aligned corpus sentence that achieves the highest similarity score is the most similar one to the input source sentence.

The implemented algorithm is based on the Smith-Waterman algorithm (Smith and Waterman, 1981), initially proposed for performing local sequence alignment for determining similar regions between two protein or DNA sequences, structural alignment and RNA structure prediction. The algorithm is guaranteed to find the optimal local alignment between the two input sequences at clause level.

In the Structure selection module, the similarity of two clauses is calculated using intra-clause information, i.e. number and type of phrases. If the ISS contains more than one clause and a similar ACS does not exist in the aligned corpus, then the nearest structure will be retrieved and modified, by replacing one or more constituent clauses as required by sequences of clauses from another sentence of the aligned corpus. In this case, the aim is to choose the best matching sentence structure from the aligned corpus and make as few replacements as possible.

A perfect result cannot be realistically expected in such a case where parts from two distinct aligned corpus sentences are combined (two distinct ACSs); yet the system should attempt to generate a translation nevertheless. It is expected that the variability of sentence structures in the bilingual corpus will provide a good starting point for this process. A detailed description of the algorithm is described next.

4.1 Algorithm input

The algorithm tries to find the best alignment between an input source sentence and a retrieved SL sentence from the aligned corpus, working on the input of Table 2:

Table 2: Input of the dynamic programming algorithm

	Input source sentence (ISS)		**Aligned corpus sentence (ACS)**	
Description	Sequence of tokens annotated with lemma, tag, phrase and clause info		Sequence of tokens annotated with lemma, tag, phrase and clause info	
Elements	**Token**	**Not used**	**Token**	**Not used**
	Lemma	**Not used**	**Lemma**	**Not used**
	Tag	PoS tag & case of each phrase head	**Tag**	PoS tag & case of each phrase head
	Phrase	Number, type, sequence, phrase head and phrase functional head	**Phrase**	Number, type, sequence, phrase head and phrase functional head
	Clause	Number & sequence	**Clause**	Number & sequence

4.2 Calculating similarity

In the beginning, a two-dimensional table is built with the ISS[3] along the top and the ACS along the left side. A cell (i, j) represents the similarity of the subsequence of elements up to the mapping of the elements E_i of the ACS and E'_j of the ISS. Elements refer to phrases, taking into account the phrase type and the Part-of-speech (PoS) tag and case (where available) of each phrase head.

The value of the cell (i,j) is filled by taking into account the cell directly to the left (i, j-1), directly above (i-1, j) and directly above-left (i-1, j-1), each one containing values V1, V2 and V3 respectively, and is calculated as the maximum of the three numbers {V1, V2, V3+ElementSimilarity(E_i, E'_j)}. While calculating the value of each cell, the algorithm also keeps tracking information so as to construct the actual alignment vector.

The similarity of two phrases (**PhrSim**) is calculated as the weighted sum of the phrase type similarity (**PhrTypSim**), the phrase head PoS tag similarity (denoted as **PhrHPosSim**), the phrase head case similarity (**PhrHCasSim**) and the functional phrase head PoS tag similarity (**PhrfHPosSim**):

$$\mathrm{PhrSim}(E_i, E'_j)= W_{\mathrm{phraseType}}*\mathrm{PhrTypSim}(E_i, E'_j)+W_{\mathrm{headPoS}}*\mathrm{PhrHPosSim}(E_i, E'_j)+W_{\mathrm{headCase}}*\mathrm{PhrHCasSim}(E_i, E'_j) +W_{\mathrm{fheadPoS}}*\mathrm{PhrfHPosSim}(E_i, E'_j)$$

In the current implementation of the algorithm, the weights have been given the following initial values, yet the optimal values are to be determined during the optimisation phase:

- $\mathbf{W}_{\mathrm{phraseType}} = 0.6$
- $\mathbf{W}_{\mathrm{headPoS}} = 0.1$
- $\mathbf{W}_{\mathrm{fheadPoS}} = 0.1$
- $\mathbf{W}_{\mathrm{headCase}} = 0.2$

For normalisation purposes, the sum of the three aforementioned weights is equal to **1**.

The similarity score range is from **100** to **0**, denoting exact match and total dissimilarity between two elements E_i and E'_j respectively. In case of a zero similarity score, a penalty weight (-50) is employed.

When the algorithm has reached the j^{th} element of the ISS, the similarity score between the two SL clauses is calculated as the value of the maximum j^{th} cell. The ACS that achieves the highest similarity score is the closest to the input SL clause in terms of phrase structure information.

Apart from the final similarity score, the comparison table of the algorithm is used for finding the phrase alignment between the two SL clauses. By combining the SL clause alignment from the algorithm with the alignment information between the ACS and the attached TL sentence, the ISS phrases are reordered according to the TL structure.

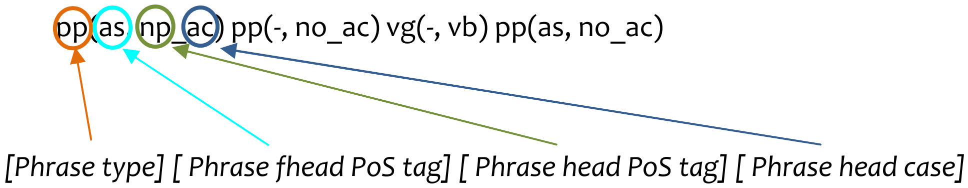
[3] Here a 1:1 correspondence is supposed, i.e. that a sentence is made up of only one clause.

4.3 Structural similarity example

For a better understanding of this approach an example is provided with Greek as the source language. The input source sentence is given in (1):

(1) Με τον όρο Μηχανική Μετάφραση αναφερόμαστε σε μια αυτοματοποιημένη διαδικασία.
with the term Machine Translation refer $_{\text{(1st pl)}}$ *to an automated procedure*
"The term Machine Translation denotes an automated procedure"

The ISS phrase structure information, after applying the Phrase aligner module, is the following:



pp(as, np_ac) pp(-, no_ac) vg(-, vb) pp(as, no_ac)

[Phrase type] [Phrase fhead PoS tag] [Phrase head PoS tag] [Phrase head case]

One of the retrieved ACS from the aligned corpus is given in (2):

(2) Οι ιστορικές ρίζες της Ευρωπαϊκής Ένωσης ανάγονται στο Δεύτερο Παγκόσμιο Πόλεμο .
the historical roots the$_{gen}$ *European*$_{gen}$ *Union*$_{gen}$ *lie* $_{\text{(3rd pl)}}$ *in-the Second World War*
"The historical roots of the European Union lie in the Second World War"

its structural information being:

pp(no_nm) pp(no_ge) vg(vb) pp(no_ac)

After calculating the similarity scores for each phrase pair of the above sentences the dynamic programming table is filled out (the arrows denoting the longest aligned subsequence):

Table 3: Example of a dynamic programming table

			Input source sentence (ISS)			
			pp(as, np_ac)	pp(-, no_ac)	vg(-, vb)	pp(-, no_ac)
		0	0	0	0	0
Aligned corpus sentence (ACS)	pp(-, no_nm)	0	60	80	-20	60
	pp(-, no_ge)	0	60	140	40	40
	vg(vb)	0	-50	10	240	140
	pp(as, no_ac)	0	100	30	-40	**340**

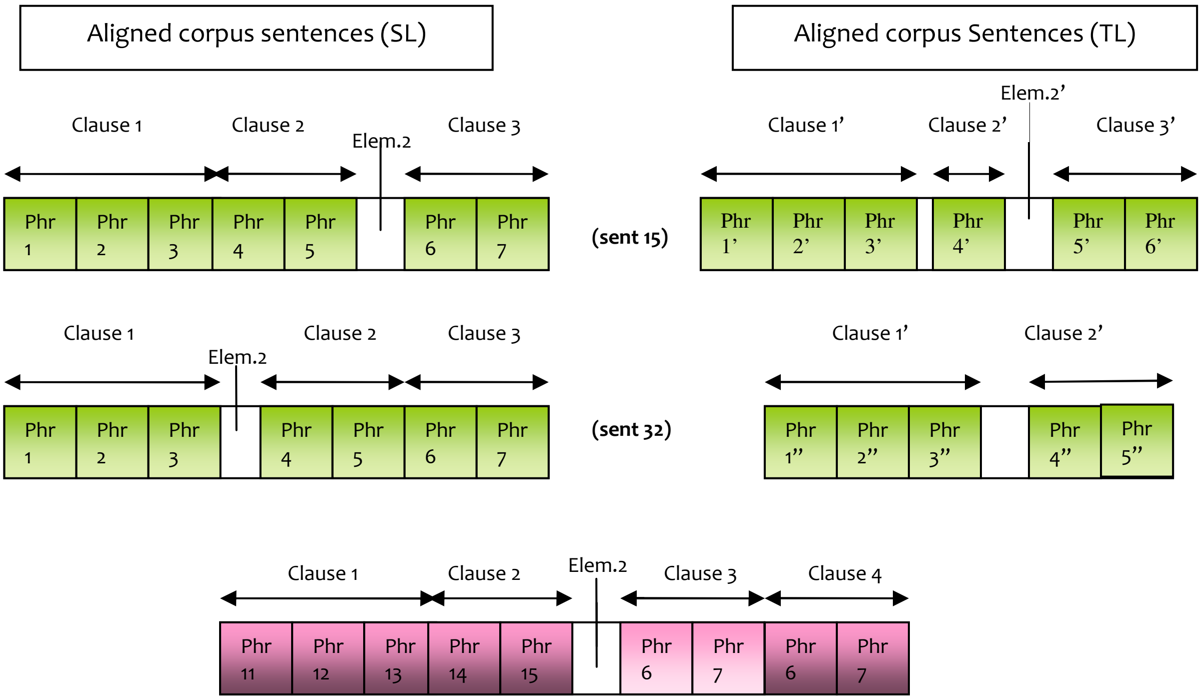
When an arrow moves diagonally from cell A to cell B, this denotes that the phrases mapped at cell A are aligned. When an arrow moves horizontally, the ISS phrase is aligned with a space, and when an arrow moves vertically the ACS phrase is aligned with a space.

Table 3 forms then the base for calculating the transformation cost (being 340 in this case), on the basis of which the ISS is modified in accordance to the attached TL structure.

4.4 Handling multi-clause sentences

If the number of clauses in the ISS is larger than that of the ACS for all sentences of the parallel corpus, it is necessary to create a new structure, by combining multiple sentences from the parallel corpus. Of course, in an MT application this needs to be done algorithmically.

Figure 4: Schematic representation of the aligned bilingual corpus



To combine the two sentence structures, a joining point needs to be determined, where two units (isolated elements or phrases/clauses) exist in both SL structures (and similarly in the TL sentences). This process would necessitate the matching of units (on the basis of (a) unit-type e.g. element or phrase/clause and (b) tag of head-of-unit) both immediately prior to and after the join point. One prerequisite is for these elements not to belong to the same clause, since then the joining operation between two TL sentences would be made without retaining the integrity of specific clauses in a given language.

Example

For the present example, the bilingual corpus of 200 sentences from Greek-to-English is used. For instance, all sentences including the CJC "όπως" (translated as "as" or "such as") are listed below (a total of 4 language pairs), denoted by their id. numbers in the bilingual corpus.

Sentence pair 15

[Με [αυτό τον τρόπο $_{NP}$] $_{PP}$],[κανένας $_{NP}$] [δεν μπορεί $_{VP}$] [πια $_{ADVP}$] [να φτιάχνει $_{VP}$] [όπλα $_{NP}$] [από [μόνος του $_{NP}$] $_{PP}$] [για να στραφεί $_{VP}$] [**εναντίον** [**των άλλων** $_{NP}$] $_{PP}$], [**όπως** $_{ADVP}$] [στο [παρελθόν $_{NP}$] $_{PP}$].

In this way, none can on its own make the weapons of war to turn **against the other**, **as** in the past.

PP NP VP PP(NP) VP NP PP(NP) VP PP(NP) PP(NP)

Sentence pair 32

[Τη δεκαετία $_{NP}$] [του 1960 $_{NP}$] [αναδύεται $_{VP}$]] [μία «κουλτούρα $_{NP}$]] [των νέων» $_{NP}$] [**με** [**συγκροτήματα** $_{NP}$] $_{PP}$]] [**όπως** $_{ADVP}$]] [οι Beatles $_{NP}$] [να προσελκύουν $_{VP}$], [όπου $_{ADVP}$] [εμφανίζονται $_{VP}$], [τεράστια πλήθη $_{NP}$] [εφήβων θαυμαστών $_{NP}$], [δίνοντας $_{VP}$] [το έναυσμα $_{NP}$] [σε [μία πολιτιστική επανάσταση $_{NP}$] $_{PP}$] και [διευρύνοντας $_{VP}$] [το χάσμα $_{NP}$] [των γενεών $_{NP}$].

The 1960s sees the emergence of 'youth culture', **with groups such as** The Beatles attracting huge crowds of teenage fans wherever they appear, helping to stimulate a cultural revolution and widening the generation gap.

NP VP NP PP(NP) PP(NP) ADJP PP(NP) VP NP PP(NP) ADVP NP VP VP VP NP VP NP

Sentence pair 74

[Ιδρύονται $_{VP}$] [**ομάδες** $_{NP}$] [**πίεσης** $_{NP}$], [**όπως** $_{ADVP}$] [η Greenpeace $_{NP}$].

Pressure groups such as Greenpeace are founded.

NP ADJP PP(NP) VP

Sentence pair 189

[Έχει σχεδιαστεί $_{VP}$] [για να κάνει $_{VP}$] [την ΕΕ $_{NP}$] [πιο δημοκρατική $_{ADJP}$], [αποτελεσματική $_{ADJP}$] και [διαφανή $_{ADJP}$], και [συνεπώς $_{ADVP}$] [ικανή $_{ADJP}$] [να αντιμετωπίσει $_{VP}$] [**παγκόσμιες προκλήσεις** $_{NP}$] [**όπως** $_{ADVP}$] [η κλιματική αλλαγή $_{NP}$], [η ασφάλεια $_{NP}$] και [η αειφόρος ανάπτυξη $_{NP}$].

It is designed to make the EU more democratic, efficient and transparent, and thereby able to tackle **global challenges such as** climate change, security and sustainable development.

NP VP VP NP ADJP ADJP ADJP ADVP ADJP VP NP ADJP PP(NP) NP NP

In red, the positions of possible joins are indicated. It can be seen that, for instance, by joining together sentences 15 and 32, much larger structures may be handled.

Let us assume now that a new sentence to be translated is provided, as shown below, together with its ideal translation (denoted as sentence pair X1).

Sentence pair X1

[Με [αυτό τον τρόπο $_{NP}$] $_{PP}$], [κανένας $_{NP}$] [δεν μπορεί $_{VP}$] [πια $_{ADVP}$] [να φτιάχνει $_{ADVP}$] [όπλα $_{NP}$] [από [μόνος του $_{NP}$] $_{PP}$] [για να στραφεί $_{VP}$] [**εναντίον** [**των άλλων** $_{NP}$] $_{PP}$], [**όπως** $_{ADVP}$] [οι Beatles $_{NP}$] [να προσελκύουν $_{VP}$], [όπου $_{ADVP}$] [εμφανίζονται $_{VP}$], [τεράστια πλήθη $_{NP}$] [εφήβων θαυμαστών $_{NP}$], [δίνοντας $_{VP}$] [το έναυσμα $_{NP}$] [σε [μία πολιτιστική επανάσταση $_{NP}$] $_{PP}$] και [διευρύνοντας $_{VP}$] [το χάσμα $_{NP}$] [των γενεών $_{NP}$].

In this way, none can on its own make the weapons of war to turn **against the other**, **such as** The Beatles attracting huge crowds of teenage fans wherever they appear, helping to stimulate a cultural revolution and widening the generation gap.

PP(NP) NP VP PP(NP) VP NP PP(NP) VP PP(NP) ADJP PP(NP) VP NP PP(NP) ADVP NP VP VP VP NP VP NP

In this case, the sentence X1 (and its desired translation equivalent in English) has a much larger (in terms of phrases) structure than that of the two original sentences. The structure of X1 cannot be directly covered by the bilingual corpus, with a single sentence. However, by combining the two sentences, for instance around the CJC token "όπως", the structure can be extended.

5. Synchronous grammars

The policy of the PRESEMT project is to use shallow parsers for the annotation of the corpora. Thus, the structures of the aligned sentences in the bilingual corpus are flat. This has advantages and disadvantages. The advantage is that phrase reordering can be represented as simple reordering of constituents that are on the same level. The disadvantage is that there is no immediate account for the productivity of language, which is usually dealt with by recursive (hierarchical) structures.

The novel idea advocated here is to combine shallow parsing strategies and synchronous grammars. Synchronous grammars are used to model the transition from SL-structure to TL-structure. Synchronous grammars allow partitioning the flat structures provided by the shallow parser into smaller units which account for the productivity of language.

The output of the statistical shallow parser is fed into the synchronous-grammar parser. In order to account for the productivity of language, the phrase sequences that do not undergo structural changes receive a hierarchical structure. The phrase sequences undergoing structural changes keep their flat structure.

The synchronous-grammar parser is not used to change the chunking and tagging information provided by the shallow parsers. Its main purpose is to model the transition from the SL structure to the TL structure; in particular, its purpose is to account for the possible combinations of mappings of sub-sentential phrase sequences.

5.1 Motivation for using a bilingual formalism such as synchronous grammars

The use of a bilingual formalism such as synchronous grammars in the Structure selection module can be motivated in different ways:

(a) In order to account for the productivity of language sub-sentential structures of different sentences of the aligned bilingual corpus have to be combined. In order to **partition phrase sequences into smaller phrase sequences a bilingual criterion** lends itself to be used as guiding line, namely the criterion which phrase sequences exhibit the same chunking in the SL and the TL and which exhibit a different structure. A bilingual formalism such as synchronous grammars provides an adequate formal framework for combining the sub-sentential phrase sequences found and generating the respective TL structure.

(b) A bilingual formalism such as synchronous grammars takes into account **TL information of the input source sentence (ISS)** when determining the TL structure. For example if an SL lexeme translates into a complex expression in the TL then the complex structure is generated, if an SL expression translates into a simple TL lexeme then the simple structure is generated.

(c) A bilingual formalism such as synchronous grammars offers **a unified account for the three steps** that are needed to generate TL structures for ISSs, namely:

1. Generating a TL phrase structure for the ISS phrase structure
2. Generating TL lemmas (and/or tokens) for the ISS lemmas (or tokens) (bilingual dictionary lookup)
3. Generating TL tags for TL lemmas (and/or tokens)

The synchronous grammars used in the Structure selection module consist of productions which map SL structure onto TL structure. Thus they account for the mapping from SL tags onto TL tags and the mapping from SL phrases onto TL phrases. The bilingual dictionary lookup is also integrated into the formalism. The lexical productions needed for the mapping of SL lemmas/tokens onto TL lemmas/tokens are generated dynamically at runtime.

5.2 TL-structure generation

The TL-structure generation proceeds in the following steps:

1. The input source sentence is parsed with shallow parsers.
2. The synchronous grammar parser reads in the shallow parse of the input source sentence and transforms it into a tree structure.
3. The synchronous grammar parser builds up a TL tree structure for the SL tree structure.

5.3 Productions as the rule base for synchronous grammars

The major function of the synchronous grammar is to generate a TL structure for an ISS structure. Synchronous grammars relate tree structures; the structural mappings from the SL structure to the TL structure are hence encoded in context-free rules. The version of synchronous grammars adopted here consists of bilingual productions, which relate SL context-free rules to TL context-free rules:

Bilingual production:

'context-free rule(s) for SL' ⬄ 'context-free rule(s) for TL'

The constituents on either side of the production are linked by indexes:

A1 –> B2 ⬄ A1 -> B2

As it has been mentioned above, there is a distinction between structure preserving and structure changing mappings from SL to TL. Structure preserving mappings are transformed into pairs of recursive binary rules. That way, the flat structures of the shallow parsers are converted into more hierarchical ones, which results in a much better coverage when parsing new incoming sentences. The transformation into binary recursive rules therefore accounts for the productivity of language. A multitude of SL phrase and tag combinations can thus be mapped onto the respective TL phrases and tags.

Structure preserving mappings are transformed into sets of binary recursive productions:

A1 –> B2 A3 ⬄ A1 -> B2 A3

A1 –> C2 A3 ⬄ A1 -> C2 A3

A1 –> D2 A3 ⬄ A1 -> D2 A3

The context-free rules on either side of the production are recursive insofar as the left hand side occurs as last constituent on the right hand side. To stop the recursion at some point we additionally generate productions with unary rules on either side:

A1 –> B2 ⬄ A1 -> B2

A1 –> C2 ⬄ A1 -> C2

A1 –> D2 ⬄ A1 -> D2

Structure changing mappings, on the other hand, retain their flat structure. They are converted into context free rules that have the constituents that undergo a structural change as siblings on the right hand side. The rules may or may not be recursive. The following production accounts for phrase reordering, assuming that B and C are phrases:

A1 –> B2 C3 ⬄ A1 -> C3 B2

The next production is an example for phrase splitting; phrase B is split into two here:

A1 –> B2 A1 ⇔ A1 -> B2 A1 B2

When such split phrases are expanded further, productions with more than one context-free rule on one side of the production are used:

B1 -> X2 Y3 ⇔ B1 -> X2 B1 -> Y3

Phrase merging works quite analogously; for an example just interchange the SL and the TL side in the above productions.

5.4 Automatic generation of productions

As a pre-processing step, a synchronous grammar is automatically extracted out of the aligned bilingual corpus. The SL and TL sentences in the bilingual corpus are tokenised, lemmatised, tagged, and annotated with a flat syntactic structure, namely phrases and clause-chunks. The sentences are aligned on word, phrase and clause level.

The pre-processing step of the generation of synchronous grammars comprises the following steps:

1. Determine tag mappings
2. Determine phrase mappings
3. Create productions that account for structural identity between SL and TL
4. Create productions that account for structural differences between SL and TL

Tag mappings

The aligned bilingual corpus exemplifies a range of mappings from SL tags to TL tags. The tags comprise part-of-speech information and morphological information. Closer inspection of aligned bilingual corpora drawn from the web has shown that the majority of tag mappings are one-to-many mappings rather than one-to-one mappings.

The reasons are at least fourfold:

1. The partitioning of part-of-speech (PoS) may differ cross-linguistically. This trend is fortified by statistical taggers which use tags that tend to ignore differences in terms of part-of-speech if the differences are hard to disambiguate.

   Example: The English TreeTagger does not distinguish prepositions and subordinate conjunctions. Both are tagged as IN. The German TreeTagger tagset distinguishes them. Prepositions are APPR or APPRPART (for contractions of preposition and determiner, as "zum") and subordinate conjunctions are KOUS.

2. The tagsets have different granularity, some tagsets distinguish further subclasses, others do not.

   Example: The English TreeTagger provides one tagset that does not distinguish between main and auxiliary verbs. Both are tagged as VB. The German TreeTagger tagset distinguishes main verbs and auxiliary verbs. VV stands for main verb and VA stands for auxiliary verb.

3. Incorrect taggings lead to additional tag mappings. These cases can either be viewed as unwanted cases of category shift or they can be filtered out by frequency counts (if they are not systematic and rampant).

   Example: In the sentence "The cat drinks the milk." the TreeTagger incorrectly tags the verb "drinks" as noun. If "drinks" is aligned to the correct German translation "trinkt" then the tag mapping NNS -> VVFIN is produced.

4. Incorrect alignments can also lead to unwanted tag mappings. Again, as long as the incorrect alignments are not systematic and rampant, they can be filtered out by frequency counts.

The tag mappings found in the bilingual aligned corpus are extracted relative to the phrase type and their frequencies are counted. The result is a listing of SL-phrases, SL-tags, TL-phrases, sequence of TL-tags and their frequencies.

The listing of tag mappings sorted by phrase type disambiguates the two uses of IN as tag for prepositions and as tag for subordinate conjunctions. If IN is part of an NC (noun phrase) then it is mapped onto the German prepositional tags APPR and APPRART whereas if IN is part of the phrase – (which means no phrase type has been assigned) then IN maps onto the subordinate conjunction tag KOUS amongst others. [4] The pipe (|) is used as a delimiter that separates different tags and their frequencies.

SL-phrase	SL-tag	TL-phrase	TL-tags and frequencies
NC	IN	NC	APPR.191|APPRART.41|ART.11|NN.Sg.6|ADV.Comp.3|PROAV.2|NN.Pl.1|PRF.1|
--	IN	--	KOUS.19|KOKOM.14|KON.4|

Converting tag mappings into productions

Tag mappings are encoded in bilingual productions in order to be able to generate a tree structure for input source sentences and the corresponding TL translation.

The tag mappings that are used as the basis for bilingual productions should contain only the morphological information that can be transferred from SL to TL, namely person and number of nouns and verbs, and tense and mood of verbs. Other types of morphological information such as number of adjectives or case should not be included in the tag mappings for several reasons:

1. They cannot be reliably assigned based on the information drawn from the limited-size bilingual corpus.

2. They may lead to a data sparseness problem if there are too many combinations of morphological features.

The one-to-many tag mappings and the frequencies found in the bilingual corpus are maintained in the productions, because they help accounting for different distributions of tags in SL and TL.

Thus a tag mapping such as:

SL-tag -> TL-tag.frequency or TL-tag.frequency …

is converted into a recursive and a non-recursive production as follows:[5]

[4] Some familiarity with the TreeTagger tagsets is assumed here, because there is no space to explain all tags.

[5] The actual disambiguation of phrase borders is not done by the productions but read off the shallow structures provided by the statistical chunkers.

Format of a recursive production relating tags:

SL-phrase+link -> SL-tag+link SL-phrase+link ⇔ TL-phrase+link -> TL-tag.frequency|TL-tag.frequency|…|+link TL-phrase+link

Format of a non-recursive production:

SL-phrase+link -> SL-tag+link ⇔ TL-phrase+link -> TL-tag.frequency|TL-tag.frequency|…|+link

Example for a recursive production:

NC1 -> IN2 NC1 ⇔ NC1 -> APPR.191|APPRART.41|ART.11|NN.Sg.6|ADV.Comp.3|PROAV.2|2 NC1

Example for a non-recursive production:

NC1 -> IN2 ⇔ NC1 -> APPR.191|APPRART.41|ART.11|NN.Sg.6|ADV.Comp.3|PROAV.2|2

The synchronous grammar formalism also needs special productions if tags undergo structural changes such as deletions or insertions.

Pluralia tantum

One-to-many tag mappings also account for special cases such as the translation into pluralia tantum without further ado. The translation of English "furniture" into German 'Möbel' is a point in case, "Möbel" is a plurale tantum whereas "furniture" is singular in English. The English tag for singular noun NN maps most often onto German singular nouns, namely 204 times. But there are also cases where it maps on a German plural noun, namely 11 times, (and there are cases where it maps onto a German adjective).

NC1 -> NN2 ⇔ NC1 ->NN.Sg.204 NN.Pl.11|ADJA.2|

Now if the ISS is

(1) The furniture is here.

Die Möbel sind da.

then the bilingual dictionary provides the translation equivalent "Möbel". Since there is no translation that fits the most frequent tag NN.Sg, the second most frequent tag is taken. If "furniture" also has a singular translation such as "Möbelstück" then both translations would be selected by the Structure selection module and it would be left to the Translation equivalent selection module to disambiguate them. Then, if "Möbelstück" is selected, the singular would be generated, because it is the most frequent form that the lexical item can realise. (For "Möbel" the most frequent form that can be realised is the plural because the singular does not exist.)

The distribution of tag information

The distribution of tag information within the different linguistic resources has been guided by considerations about the User adaptation module.

One linguistic resource that end users often want to modify or extend is the bilingual dictionary. End users want to enter their own terminology and add expressions and phrases that are relevant to their translation domains. Extending the bilingual dictionary becomes easier for the end user if the entries in the bilingual dictionary need not be linguistically annotated with part-of-speech tags or other linguistic information. Therefore the guiding line of the current approach is to get the tag information that is no doubt needed from resources other than the bilingual dictionary. The current approach makes the following assumption:

∗ No tag information (or any other linguistic annotation) from the bilingual dictionary is utilised.

The tag information needed for the translation steps is gained from other linguistic resources, namely:

1. The statistical taggers provide the tagging information for the input source sentence.
2. The tag mappings found in the aligned bilingual corpus provide information about which SL-tags can be mapped onto which TL-tags.
3. The TL token generation table stores information about which TL lemmas and tokens can be associated with which TL-tags.

In order to illustrate the distribution of tag information let us take the translation of an ambiguous word such as "access". English "access" can be a noun or a verb. The bilingual dictionary contains entries for "access" as a noun and as a verb without specifying part of speech in the SL or TL. "Zugang" is a German noun and "aufrufen" is a German separable prefix verb with two entries, one in split form and one in contiguous form. Both are possible translations of "access":

(i) bilingual dictionary:

access ⇔ Zugang

access ⇔ aufrufen

access ⇔ rufen_auf

The tags of the input source sentence are determined by the statistical tagger:

1. The access was difficult.

   DT NN VBD JJ

2. They access the system.

   PP VBZ DT NN

   NN stands for singular Noun, VBZ stands for verb, 3rd person singular present.

The tag mappings found in the bilingual aligned corpus contain the following mappings for VBZ and NN:

VBZ -> VVFIN.3.Sg.Pres.25|VAFIN.3.Sg.Pres.21|VMFIN.3.Sg.Pres.4|VVPP.4|NN.Sg.1|

VBZ -> VVFIN.3.Sg.Pres PTKVZ

NN ->NN.Sg.204 NN.Pl.11|ADJA.2|

The TL token generation table contains the following entries for "Zugang", "aufrufen", "rufen" and "auf":

lemma	tag	token
aufrufen	VVFIN.3.Sg	aufruft
rufen	VVFIN.3.Sg	ruft
auf	PTKVZ	auf
Zugang	NN.Sg	Zugang

The parser generates the adequate lexical productions online. Four resources are used by the parser to generate the adequate lexical productions.

1. The SL-tagging provided by the statistical tagger.
2. The bilingual dictionary entries.
3. The tag mappings found in the aligned corpus.
4. The TL token generation table.

The lexical productions generated for "access – VBZ" are the following:

VBZ1 -> access ⬄ VVFIN.3.Sg.Pres.25|VAFIN.3.Sg.Pres.21|VMFIN.3.Sg.Pres.4|VVPP.4|NN.Sg.1|2 -> aufrufen

VBZ1 -> access2 ⬄ VVFIN.3.Sg.Pres1 -> rufen2 PTKVZ1 -> auf2

The lexical productions generated for "access – NN" are the following:

NN1 -> access2 ⬄ NN.Sg.204 NN.Pl.11|ADJA.2|1 -> Zugang2

The lexical productions have the effect that the noun "access" is translated into the noun "Zugang" and the verb "access" is translated into the verb "aufrufen" or "rufen_auf".[6]

Phrase mappings

The phrase mappings needed as a basis for the bilingual productions are one-to-one mappings. For each SL phrase type the equivalent TL phrase type has to be determined.

If the phrasing model generator is used, the phrase names of the equivalent SL and TL phrases are identical since the TL phrase types are projected onto the SL structure.

However, the production generation algorithm can also handle bilingual corpora that have been chunked by a pre-existing chunker. In this case the phrase names are not necessarily identical. The equivalent phrase types are determined via a frequency count. For each SL phrase type, the most frequent alignment to a TL phrase is considered the equivalent phrase type.

Structure-preserving productions

In order to account for the productivity of language, the sequences of phrases that are mapped onto TL phrases are kept as small as possible. For phrases that do not exhibit structural changes in the TL this means that they are mapped one by one onto TL phrases. In terms of productions this means that they are accounted for by productions that relate unary context-free rules or binary recursive context-free rules.

Format of the recursive production:

SL-clause-type+link -> SL-phrase-type+link SL-clause-type+link ⬄ TL-clause-type+link -> TL-phrase-type+link TL-clause-type+link

Format of the non-recursive production:

SL-clause-type+link -> SL-phrase-type+link ⬄ TL-clause-type+link -> TL-phrase-type+link

[6] If the aligned bilingual corpus contains alignments of SL nouns to both TL nouns and TL verbs, then the frequency count should indicate that the noun translation is the more likely one.

Structure changing productions

Structure-changing productions contain all the phrases that are involved in the structural change. Therefore they often contain more than one phrase that is to be mapped. The resulting structures are flatter than the ones produced by structure-preserving productions.

Example for a translation with and without structural changes

Consider the following sentence which has a structure preserving and a structure changing translation:

(1) The commission **accepted** the offer.

Die Kommission **akzeptierte** das Angebot.

Die Kommission **nahm** das Angebot **an**.

The synchronous grammar parser takes as input the lemmatised, tagged, chunked and clause chunked version of the SL sentence. In the following the TreeTagger is used for English tagging and chunking. The identifiers (ids) are unique within a sentence.

word id	token	tag	lemma	phrase + id	clause + id
0	The	DT	the	NC6	MC10
1	commission	NN	commission	NC6	MC10
2	accepted	VBD	accept	VC7	MC10
3	the	DT	the	NC8	MC10
4	offer	NN	offer	NC8	MC10
5	.	SENT	.	--9	MC10

The phrase structure of the SL sentence is thus: NC1 VC2 NC3 --4. The structure preserving translation has the same phrase structure, i.e. there is structural identity for all phrases. Then phrase mappings of the following form account for the correct TL phrase structure:

NC1 -> NC1

VC1 -> VC1

--1 -> --1

The above chunk mappings are converted into the following productions:

MC1 -> NC2 MC3 ⇔ MC1 -> NC2 MC3

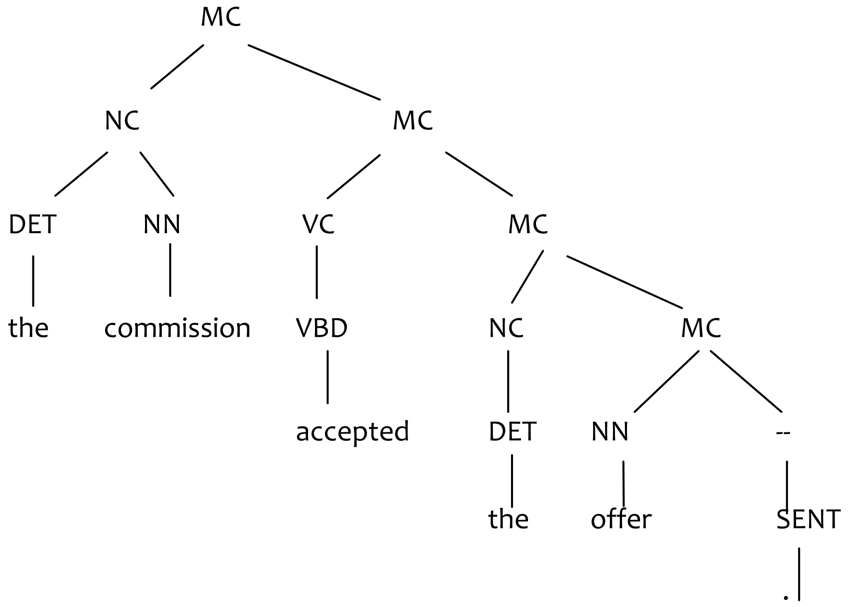
MC1 -> NC2 ⇔ MC1 -> NC2

MC1 -> VC2 MC3 ⇔ MC1 -> VC2 MC3

MC1 -> VC2 ⇔ MC1 -> VC2

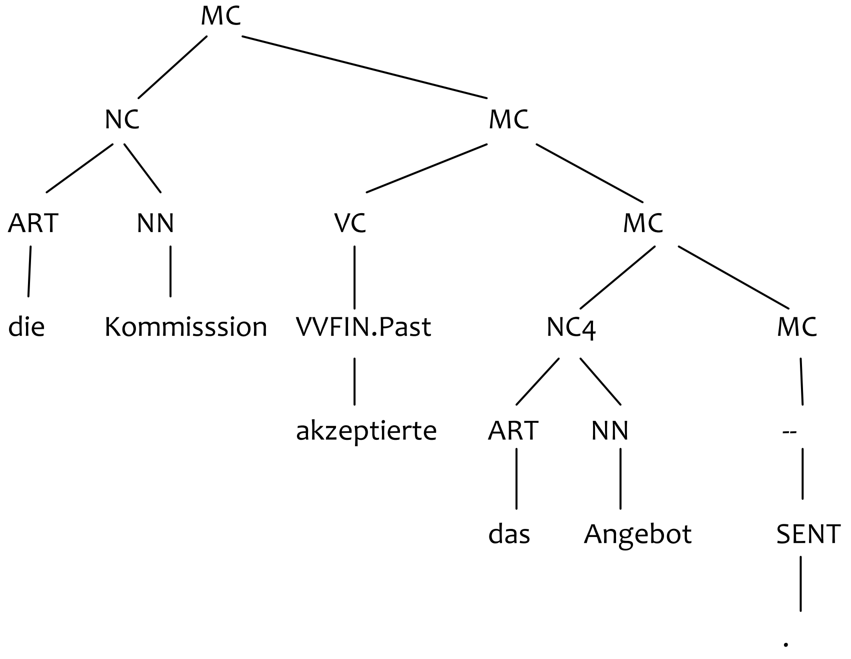
MC1 -> --2 MC3 ⇔ MC1 -> --2 MC3

MC1 -> --2 ⇔ MC1 -> --2



The SL structure also stores which productions have applied. The TL rules associated with each production are used to build up the TL structure. It is identical to the SL structure modulo the tag names if there is no structural change.

Die Kommission akzeptierte das Angebot.



Now consider the translation that involves a structural change:

Die Kommission nahm das Angebot an.

This translation involves phrase splitting, i.e. the SL sequence NC1 VC2 NC3 –4 translates into the TL sequence NC1 VC2 NC3 VC2 --4.

VC split is accounted for by the following phrase mapping:

VC1 NC2 --3-> VC1 NC2 VC1 --3

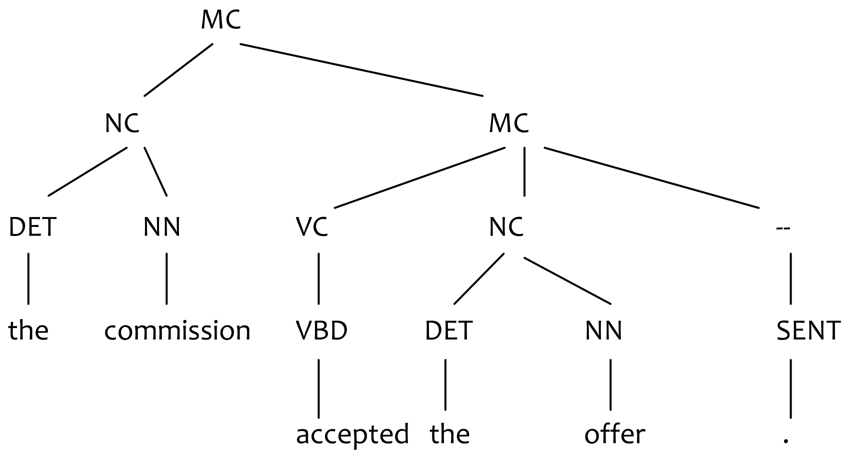
This mapping translates into the following production:

MC1 -> VC2 NC3 --4 ⬄ MC1 -> VC2 NC3 VC2 --4

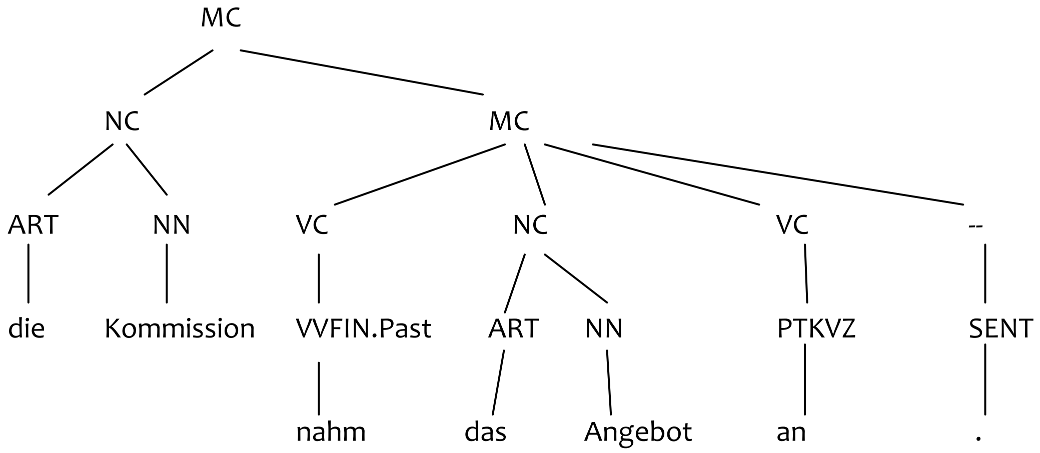
The split VC is expanded by the following production:

VC1 -> VBD2 ⬄ VC1 -> VVFIN2 VC1 -> PTKVZ2

The SL-structure that generates the structure-changing translation is slightly different from the SL-structure that generates the structure-preserving translation. The sequence of phrases that undergoes a structural change is represented as a flat structure rather than a hierarchical one. This type of flattening is motivated by the needs of the translation process and not by any monolingual SL properties.



The associated TL structure is:



Features that encode conditions for structural changes

In order to account for the conditions on structural changes a set of features has been introduced that encodes clause type, the type of head, the type of f-head and the category preceding a specific category. These features are passed up and down the tree so that they are available at the places needed.

Production weights

Productions can be weighted. Weights can be used to rank the TL structures. However, for the 1st system prototype, weights are not used to rank the TL structures. Simple weights that distinguish structure-preserving from structure-changing productions have not proven useful so far. In the future, a different concept of weights based on frequencies or some other sort of possibly optimisable parameter might be introduced to rank the TL structures and to select the best ones.

Language-independent algorithm for automatic generation of productions

A general procedure for automatically extracting productions out of an aligned bilingual corpus has been developed. The procedure itself is language-independent. However, while working on DE-EN and EN-DE it has proven useful to set certain language-specific parameters such as which lemmas can be deleted in the translation from one language to another or what the conditions for certain structural divergences are. These parameters are set in configuration files that are language-pair specific. The parameter settings have been specified for EN-DE and DE-EN.

The core algorithm for generating productions

In the core algorithm, the sentential chunk and tag alignments are broken down into the **smallest self-contained alignments** and then converted into productions. In the case of isomorphic SL and TL structures the resulting productions produce hierarchical trees, in the case of structural divergences between SL and TL flat trees are generated. The combinatory potential of the productions thus derived already extends the structures covered by the system beyond the structures found in the bilingual corpus. The algorithm accounts for all kinds of structural divergences including:

- Reordering of phrases and tags
- Phrase and tag splitting
- Phrase and tag merging
- Deletion and insertion of tags
- Category shift in as much as it is represented in the alignment
- Any combination of the above

The core algorithm makes use of two additional devices that allow expressing linguistic generalisations:

- Tag reduction
- Specification of lemmas that can be deleted or inserted

Tag reduction

A potential problem for the PRESEMT approach arises with morphologically rich languages such as German, Greek and Czech. The morphological richness is mirrored in rich tag sets with large numbers of tags that are possibly not all covered in all potential translation patterns in the small bilingual corpus. However, the experience with German both as SL and TL has shown that it is appropriate to reduce the information expressed in the tags to part of speech and those morphological features that are (1) represented in the TL and (2) cannot be inferred from monolingual TL information.

Thus in DE-EN agreement information on determiners and adjectives is deleted, because it is not reflected in EN as TL. Likewise, gender information and case of non-subjects is deleted because it is retrieved from TL resources. The DE agreement information is also deleted on the DE tags for DE as TL since agreement within the noun phrase can be accounted for by a monolingual TL mechanism. Guidelines have been developed regarding the tag information that can be reduced so that the setup of a new language pair is alleviated.

Specification of lemmas that can be deleted or inserted

There are productions that delete or insert lexical items. They are based on non-aligned lexical items found in the bilingual corpus. These items are typically grammatical function words such as degree particles („more“, „most“) that are inserted/deleted if SL and TL differ with respect to the formation of the comparative. Auxiliary verbs are also inserted/deleted if SL and TL differ with respect to complex and simple tenses. In order to distinguish these wanted cases of non-alignment from unwanted cases of non-alignment that result from accidental gaps in the bilingual dictionary it has been proven useful to define the set of lemmas that can be inserted or deleted. Again, guidelines have been developed how to set up these sets of elements that can be inserted or deleted.

Further devices for expressing linguistic generalisations

In order to further extend the coverage of linguistic phenomena derived from the small bilingual corpus, two additional devices have been implemented:

- Equivalence classes of tags and chunks
- Conditions on structural changes

Equivalence classes of tags

Since it is very unlikely that the small bilingual corpus contains all tag variants of certain structure-changing patterns, equivalence classes of tags have been introduced. They are used to generate additional productions that cover these predictable additional patterns. Thus for example, if the bilingual corpus EN-DE contains one example of a split verb phrase the pattern is generalised to all possible tag variants by defining an equivalence class that contains all finite verb tags of the particular finite verb type. Another typical equivalence class consists of singular and plural noun tags, since structure-changing patterns such as compounding commonly hold for both singular and plural nouns. Again, guidelines have been developed for specifying these equivalence classes.

Conditions on structural changes

Using the smallest self-contained alignment as the basis for productions leads to the problem of overgenerating TL structures. Many structure-changing patterns are limited to structural conditions that lie outside the structure-changing constituents themselves. For example, the necessity of inversion of subjects in EN-DE depends on whether there is a phrase to the left: “Yesterday she was here.” requires subject inversion as in: “Gestern war sie hier. ”. In contrast, a simple SVO sentence does not require subject inversion: “She was here.” translates into the structurally isomorphic “Sie war hier.” whereas inverting the subject would not lead to an appropriate translation: “War sie hier.”. The latter exhibits the word order of a direct question.

The conditions on structural changes are translated into feature values such as leftSiCat!=START which means there must be a category to the left. Other features that can get values are head, and leftDauCat which corresponds to functional head. Yet another feature is moCat which is used to restrict verb splitting in German to main clauses.

A device has been implemented for specifying which productions get which additional conditions. The device uses regular expressions to select the productions. The conditions can be selected from a preconfigured set of conditions. These conditions have been developed based on the experience gained from DE-EN and EN-DE. They are probably not sufficient for all phenomena in all possible languages but they are a start. There are guidelines for setting up the conditions; however, setting them up takes some sophisticated linguistic skills and insight into the form of the productions.

Optionality of the additional devices

The above mentioned additional devices help improving the linguistic coverage and performance of the MT system. However, they are optional. If the linguistic skills that are needed are missing, it is possible to set up and initialise a new language pair without them. The general procedure for generating productions works without any human interference.

Grammatical functions

In order to account for case in the translation from fixed word order languages without case marking into free word order languages with case marking it has proven useful to mark the subject in the fixed word order language. A rule-based heuristics for marking subjects in SVO languages has been developed. It is used for English. For Norwegian it can be used insofar as Norwegian is SVO. The exceptions to SVO in Norwegian are not accounted for yet.

Portability to other languages

The system has been developed using the language pairs English → German and German → English. Even though English and German are closely related they are typologically quite different and exhibit a fair number of translation problems as is also noted by Koehn et al. (2008). Typologically, German is a free word order language with rich morphology and case marking whereas English is a fixed word order language with little morphology and no case marking. Thus translation from English into German is paradigmatic for a translation from a fixed word order language with no case marking into a free word order language with rich morphology and case marking. The translation from German into English is paradigmatic for a translation from a free word order language with case marking into a fixed word order language with no case marking.

It is expected that the strategies that have been developed for English → German and German → English also account for other language pairs with typologically similar languages. In particular, it is expected that Czech and Norwegian do not pose any problems since Czech is a morphologically rich, free word order language like German, and Norwegian is a fixed word order language without case marking like English.

Status of the implementation

A general procedure for extracting productions out of the aligned bilingual corpus has been implemented. It accounts for all sorts of structural divergences between languages. The procedure itself is language-independent. Additional devices have been implemented in order to better account for linguistic generalisations and in order to extend the coverage of the MT system beyond the cases found in the bilingual corpus. These devices take language-pair-specific parameter settings as input. General guidelines for setting these parameters have been developed so that the introduction of new language pairs is alleviated. The generation of productions is implemented in PERL. Since the generation of productions is also be used online in the user adaptation module, a JAVA wrapper has been implemented.

The language pairs covered so far are English → German and German → English. Norwegian into English and German and Czech into English and German are on the way. The language-pair specific parameter settings have already been defined.

Further extensions

Further algorithmic extensions are not planned unless the additional language pairs pose some unforeseen difficulties that require an algorithmic extension.

According to the current timetable of the project, the application of the approach to language pairs involving Czech and Norwegian as source language has to be completed and tested.

SL	TL
Czech	English
Czech	German
Norwegian	English
Norwegian	German

Then, as an additional test case, the approach is planned to be applied to Italian as target language, i.e. for the language pairs listed in the next table.

SL	TL
Czech	Italian
English	Italian
German	Italian
Greek	Italian
Norwegian	Italian

5.5 Parser for synchronous grammars

Related work: literature survey

There is a rich literature on synchronous grammars (see Chiang (2006) and Ahmed et al. (2005) for an overview). The types of synchronous grammars differ with respect to the complexity of the structural changes they can account for but also with respect to the efficiency of the parsing algorithm.

The simplest and most efficient synchronous grammar is a synchronous context-free grammar, which is also known as 2-multitext grammar. Structural changes have to be represented in flat structures. This is the approach adopted for the Structure selection module.

Multitext grammars (Melamed, 2004 and Melamed et al., 2004) are formally equivalent to context-free synchronous grammars. The notations adopted are slightly different from the notations adopted in the present approach. Both multitext grammars and the approach advocated here have means to account for discontinuous constituents.

The more powerful types of synchronous grammars manage to represent structural changes in binary trees but this is not necessary for the approach chosen, since the aligned bilingual corpus allows to distinguish the parts of the structure that undergo structural changes from the ones that do not and hierarchical versus flat structures can be defined according to the cross-linguistic properties of SL and TL.

The most powerful type of synchronous grammar is the synchronous tree adjoining grammar (see Shieber & Schabes, 1990). At the same time it is also the one with the most inefficient parsing strategy.

Some (see e.g. Eisner, 2003) use learning algorithms to learn the mappings of tree pairs from linguistically annotated parallel text. This avenue is not open for the approach adopted here because the flat structures of the linguistically annotated bilingual corpus do not render themselves as a basis for tree mappings that account for the productivity of language. One would have to convert the flat structure of the bilingual corpus annotations into a mix of flat and hierarchical structures first before the tree mappings could be learned from the aligned bilingual corpus. Eisner adopts a synchronous tree substitution grammar, which is almost as powerful as a synchronous tree adjoining grammar.

The version of synchronous grammar proposed here deviates from all the synchronous grammars found in the literature in that the input to the parser is not simply an SL sentence but an SL sentence with a linguistic structure which is provided by statistical taggers and chunkers.

Thus, the novel idea is to combine shallow parsing strategies and syntax-based MT techniques. The synchronous grammar is not used in order to determine the ISS syntactic structure but only to change the hierarchical order of SL-phrases if necessary and to create the corresponding TL structure.

Earley chart parser

The core part of the Structure selection module (SSM) is an Earley chart parser (see Earley 1968) with a few (standard) additions.

Features and unification

The parser is able to handle flat feature structures ("flat" meaning that the value of a feature cannot be a feature itself). Feature values can be checked for identity and non-identity. Disjunctive feature values are possible as well. Unification of features is supported in the usual way and in combination with the negation operator.

Weighted grammars

The parser processes weighted grammars, i.e. grammars in which (not necessarily all) rules have a weight associated with it. It is not necessary that the weights of all rules with the same left hand side sum up to 1, i.e. it does not need to be a probabilistic grammar. It should be pointed out that weights are not used in the 1st system prototype.

Robustness

The parser adopts a robustness strategy. In case the SL-grammar cannot correctly analyze the SL sentence, partial trees are generated and joined to a complete parse tree. The best partial tree is determined by applying Dijkstra's shortest path algorithm. Robustness is needed for example if there is no production that accounts for a certain SL tag in the input source sentence.

Grammar pre-processing

Before the bilingual grammar is sent to the parser it is converted into a standard context-free grammar. Also, lexical rules for the input source sentence are generated. This has the advantage that not all lexical rules have to be created beforehand what would make the grammar rather hard to handle.

Conversion of the bilingual grammar to a standard grammar

As it was mentioned above, the bilingual grammar consists of bilingual productions, i.e. SL rules paired with corresponding TL rules.

SL rules ⇔ TL rules

Each constituent on either side has to be linked with at least one constituent on the other side. If two or more constituents on one side link to the same constituent(s) on the other side, we call them co-links.

The bilingual grammar is converted to a standard grammar by storing the information of the TL part in features of the SL constituents. After the parse this information is extracted from the resulting parse trees and is used to construct parallel TL trees that contain the translation of the corresponding SL tree.

Particularly the following pieces of information are stored:

- For each SL constituent the category/categories of the TL constituent(s) it is linked to is stored.
- For each SL constituent the TL rule and the position of the TL constituent(s) within that rule it is linked to are stored.

These two features are already sufficient to ensure that a parallel tree can be build. There are some other features to ensure that no trees that are not licensed by the parallel grammar are generated.

- The number and order of the SL rules is preserved by storing rule numbers in left hand side and right hand side constituents that have to unify with each other.
- The combination of two SL rules that were not parallel rules to each other is excluded by storing the structure of the TL rule(s) that the constituents of a SL rule link to.

On-line generation of lexical productions

The generation of lexical rules is done by dynamic lexical templates. They look similar to productions the main difference being that some left hand sides are not spelled out but left variable. So a dynamic template can look as follows:

VVD1 ->_2 ⇔ VAFIN1 -> haben2 VVPP1 -> _2

This template says that a past tense verb in English is translated into German by "haben" + the corresponding past participle. Every ISS structure is matched against all available dynamic templates and productions are created by substituting one or several words of the SL sentence for the variable on the SL side and the corresponding TL words on the TL side. The corresponding words are determined by a lexicon lookup.

A special routine has been implemented for the treatment of digits and for unknown words.

Status of the implementation

The Early chart parser and the generation component for lexical productions have been programmed in JAVA using common base classes of PRESEMT platform to ensure compatibility with future modules.

Further extensions

The scoring device might be re-assessed if it seems useful to rank TL-structures. For more details the reader may refer to the section on Optimisation. Other extensions are not planned unless some unforeseen problem arises with the remaining language pairs.

5.6 Further resources used by the Structure selection module

Bilingual dictionary

The bilingual dictionary contains lemma-based single-word and multi-word entries. In addition, it contains linguistic annotations. However, the synchronous grammar approach does not use the linguistic annotations in the bilingual dictionary. Therefore, the quality of the linguistic annotations is not decisive. This is advantageous if machine-readable dictionaries provided by publishers are converted into system dictionaries. Most dictionaries provided by publishers do not contain linguistic annotations in appropriate and systematic ways.

The German → English and English → German dictionaries are based on dictionaries used in METIS II and EUROTRA projects. The same holds for the Greek → English dictionary. The remaining dictionaries are based on machine-readable dictionaries provided by publishers.

The current English → German dictionary contains roughly 1,000,000 entries of mixed quality. (Some multi-word entries contain tokens.)

For German → English the same dictionary is used in reverse form.

The current Norwegian → English dictionary contains roughly 45,000 entries of mixed quality.

The current Norwegian → German dictionary contains roughly 37,000 entries of mixed quality.

The current Czech → English dictionary contains roughly 180,000 entries of mixed quality.

The current Czech → German dictionary contains roughly 70,000 entries of mixed quality.

The current Greek → English dictionary contains roughly 40,000 entries of mixed quality.

The current Greek → German dictionary contains roughly 80,000 entries of mixed quality.

TL token generation table

The TL token generation table contains information about the relation of lemmas, tags and tokens. Its main purpose is to facilitate word token generation for given lemmas if tag information is provided. However, it is also used in the online generation of lexical productions.

The TL token generation table is a monolingual resource which can be automatically extracted out of large monolingual corpora. In order to save memory space, only lemmas that can be found in the bilingual dictionaries are included in the token generation tables. The current token generation table for German contains roughly 1,000,000 entries, while for English it contains roughly 150,000 entries. The difference in size is caused by the fact that German has rich morphology as well as compounding, and these two phenomena combined lead to a larger number of tokens and lemmas.

6. Optimisation of the Structure selection module

The optimisation of the Structure selection module falls within Task T4.2. In addition to the already foreseen parameters for optimisation, the development of the 1st system prototype has brought to light some additional parameters that can be fed into an optimisation algorithm and subsequently be defined automatically.

Within the synchronous grammar framework it is planned to optimise the usage of memory space and processing time in dependency of a translation quality matrix such as BLEU scores. The Synchronous grammar approach used in 1st system prototype faces two bottlenecks:

Main memory bottleneck: If the generated TL-structures exhaust the main memory, the system crashes. The system can be optimised by limiting the number of TL structures that is generated. This is possible if the TL structures are ranked. The cut-off point for the number of TL structures is a parameter for optimisation.

Time bottleneck: The number of TL-structures causes a time bottleneck in the lemma disambiguation in TES. The more TL-structures there are, the longer it takes to disambiguate the lemmas. There are several ways to optimise the processing time. Limiting the number of TL structures is one of them. It is also planned to investigate whether parts of the lemma disambiguation can be parallelised. Another parameter that can be optimised is the beam size of the Viterbi decoder. The smaller the beam size is, the faster the system is.

The parameters for optimisation include the following:

1. Number of generated TL structures (and possibly weights of productions)
2. Beam size of the Viterbi decoder
3. Weights used for calculating phrase similarity, namely
   - Phrase type weight: $\mathbf{W_{phraseType}}$
   - Phrase head PoS tag weight: $\mathbf{W_{headPOS}}$
   - Functional phrase head PoS tag weight: $\mathbf{W_{fheadPOS}}$
   - Phrase head case weight: $\mathbf{W_{headCase}}$

The optimisation process for these parameters is based on evolutionary computation techniques. Such approaches include Genetic algorithms (GAs), Ant-Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO), which at the end of the first year were the prime candidates for experimentation. The optimisation experiments are expected to be broadly based on the process described in Sofianopoulos et al. (2010).

An off-line iterative process is performed during optimisation experiments, involving the translation via PRESEMT of a set of development sentences and the evaluation of these translations. The evaluation is performed using MT quality evaluation methods such as BLEU or METEOR, using a set of corresponding reference translations for the chosen source sentences. To validate these results, multiple sets of sentences (with their reference translations) have been defined and these are applied to the optimisation algorithms. The use of multiple optimisation sets per language pair allows for cross-checks regarding the convergence of the process itself and the stability of the optimal values for parameters, by performing multiple sets of runs that can be compared and contrasted.

So far, optimisation experiments have involved the METIS-II implementation for the Greek-to-English language pair, which possesses conceptual similarities to the PRESEMT approach, such as the two-phase translation process. Optimisation techniques used so far include GAs and the SPEA2 multi-objective evolutionary algorithm. SPEA2 was selected as a typical multi-objective algorithm used for optimisation tasks that include more than one evaluation criteria. For these two types of optimisation algorithms, parameters such as the population size, the number of iterations and the type of automated evaluation metric to assess the translation quality have been studied. Also, the behaviour of each algorithm was evaluated in terms of the translation quality as well as the convergence behaviour and the diversity of the population. On the whole, SPEA2 was found to provide a better optimisation than GA for a given set of iterations. Also, it was found that it is preferable to use a larger population for fewer training epochs rather than a smaller population for more iterations. Also, the values to which the SPEA2 population of solutions settled over the different parameters indicates the different behaviour of specific parameters which settle to a small set of possible values versus other parameters for which the diversity is much larger.

Within the next reporting period, other candidate optimisation techniques will also be evaluated for the optimisation task, working on the actual PRESEMT MT system. These are expected to involve ACO, for which an implementation exists, as well as GA and SPEA2. The relevant results will be reported in future PRESEMT deliverables.

7. References

Ahmed, A. & Hanneman, G. (2005) Syntax-based statistical machine translation: A review. Association for Computational Linguistics.

Chiang, David (2006) An introduction to synchronous grammars, notes for a tutorial given at ACL 2006 with Kevin Knight.

Damerau, F. J. (1964): A technique for computer detection and correction of spelling errors. Communications of the ACM.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik 1: 269–271.

Earley, J. (1968): An efficient context-free parsing algorithm. Ph.D. thesis, Carnegie Mellon University, Pittsburg, PA.

Eisner, J. (2003): Learning non-isomorphic tree mappings for machine translation. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.

Hamming, R. W. (1950): Error detecting and error correcting codes. Bell System Technical Journal 26 (2): 147–160.

Hirschberg, D. S. (1975): A linear space algorithm for computing maximal common subsequences. Communications of the ACM 18 (6): 341–343

Koehn, P., Och, F.J. & Marcu, D. (2003): Statistical phrase-based translation. In HLT-NAACL-03, pp.48-55.

Koehn, Philipp, Arun, Abhishek & Hoang, Hieu (2008): Towards better Machine Translation quality for the German – English Language pairs. Proceedings of the Third Workshop on Statistical Machine Translation, Association for Computational Linguistics, pp 139-142.

Levenshtein VI (1966): Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10: 707–10.

Melamed, I. Dan (2004): Statistical machine translation by parsing. Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL), Barcelona, Spain.

Melamed, I. Dan, G. Satta, B. Wellington (2004): Generalized multitext grammars, Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL), Barcelona, Spain.

Och F.J., Gildea, D., Khundanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V, Jin, Z. & Radev, D. (2003): Syntax for statistical machine translation. Final Report of Johns Hopkins 2003 Summer Workshop.

Shieber, S.M., & Schabes, Y. (1990) Synchronous tree-adjoining grammars. In Proceedings of the 13th International conference on Computational Linguistics (COLING), vol 3, pp 1-6.

Smith, T. F. & Waterman, M. S. (1981): Identification of Common Molecular Subsequences. Journal of Molecular Biology, 147: 195–197.

Sofianopoulos, S. & Tambouratzis, G. (2010) Multiobjective Optimisation of real-valued Parameters of a Hybrid MT System using Genetic Algorithms. Pattern Recognition Letters, Vol. 31, pp. 1672-1682.