D5.1.2: TRANSLATION EQUIVALENT SELECTION MODULE (VER.2)

Grant Agreement number	ICT-248307
Project acronym	**PRESEMT**
Project title	**P**attern **RE**cognition-based **S**tatistically **E**nhanced **MT**
Funding Scheme	Small or medium-scale focused research project – STREP – CP-FP-INFSO
Deliverable title	**D5.1.2: Translation equivalent selection module (ver.2)**
Version	**4**
Responsible partner	GFAI
Dissemination level	Restricted
Due delivery date	31.12.2011 (+60 days)
Actual delivery date	16.1.2012

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	
ACO	Ant Colony Optimisation
ACP	Aligned corpus (TL) phrase
ACS-TL	Aligned corpus sentence / TL
AIS	Artificial Immune System
CHP	Chosen phrase
GA	Genetic Algorithm
ISP	Input sentence phrase
IST	Input Sentence / TL
MCP	Monolingual corpus phrase
MCS	Monolingual corpus sentence
MT	Machine Translation
PoS	Part of Speech
PSO	Particle Swarm Optimisation
SL	Source Language
SSM	Structure selection module
TL	Target Language

1. Executive summary

Deliverable D5.1.2 reports on the work carried out in **WP5: Translation equivalent selection.** WP5 "*relates to the second phase of the machine translation process. To this end, a translation equivalent selection module will be designed and implemented, which will retrieve for each phrase of the source sentence the best-matching translational equivalent included within the monolingual corpus.*"

WP5 is divided into two tasks, *T5.1: Design and implementation of the translation equivalent selection module* that "*focuses on designing and implementing the second part of the main machine translation process, which consists in defining the best-matching translation patterns within the target language monolingual corpus with the help of semantic relevance of words defined via novel computational intelligence methods.*" and *T5.2: Optimisation of module-specific parameters,* which "*involves a series of similarity weights, whose values need to be optimally set via the use of learning algorithms. Based on the experience of the project partners, metaheuristic optimisation techniques (such as genetic algorithms) are expected to be used, the appropriate one[s] to be defined at the start of the task via a comparative evaluation process*".

In the PRESEMT architecture, the translation process is split into two phases. The first phase, **Structure selection**, determines the overall structure of the target language (TL) sentence that is the translation of the input source language (SL) sentence with the help of the syntactic information contained in a small bilingual corpus. The second phase, **Translation equivalent selection**, handles more fine-grained properties of the target language. In particular it aims at performing the following tasks:

1. **Word translation disambiguation:** Out of the translation alternatives of the SL words, the best translation in the given context has to be chosen.
2. Resolution of micro-level **word order** issues
3. Insertion or deletion of **functional words** (such as articles or prepositions)
4. Specification of **additional morphological information** such as gender and declension type (strong vs. weak) plus case or number that may or may not be present in the source language
5. Generation of **TL tokens** for the lemmas based on the morphological information provided

The Translation equivalent selection module uses for these tasks the information contained in a huge monolingual target language corpus. No bilingual information is used at this stage, apart from the output of the first phase of the translation process. The output of the Translation equivalent selection module is the final translation of the source language sentence.

The search by the module in the monolingual corpus data is guided by several parameters. At first, these parameters are set manually to approximate values. Later, the parameters will be optimised in an offline optimisation process, which is provided for in the module itself. The optimised parameters are then used online by the Translation equivalent selection module.

The deliverable has the following structure: First, the role of the Translation equivalent module in the overall PRESEMT architecture is explained (Section 2). Section 3 outlines the aforementioned tasks of the module and Section 4 provides a detailed account of the status of the implementation of the Translation equivalent selection. The deliverable is concluded by an outlook regarding the module-related work planned for the next few months (Section 5).

2. Introduction: The PRESEMT approach in brief

One bottleneck for most statistical translation systems is the availability of bilingual corpora. These are hard to find, particularly if not so widely used languages, such as Greek and Norwegian, are involved. Nonetheless, the quality of the translation of such systems depends to a large extent on the quality and the size of the bilingual corpus. Even if such corpora exist, they are frequently restricted to a very specific domain (such as parliamentary debates). Using the system for translation tasks from a different domain can then yield suboptimal results.

In PRESEMT, a novel approach has been chosen to overcome this difficulty of providing an appropriate bilingual corpus. Although it is true that in PRESEMT a bilingual corpus is also needed, only a small corpus of a few hundred sentences is utilised. Since the bilingual corpus is used for determining structural properties of the translation of the incoming SL sentence, the restriction to a particular domain is much less problematic. The information contained in the bilingual corpus is supplemented by the monolingual information contained in a huge monolingual TL corpus. Such monolingual corpora are much easier to obtain, for example through the web. Both the bilingual and the monolingual corpora are tagged with part of speech and (if applicable) morphological information such as case, number, gender and tense. They are annotated with a flat syntactic structure, viz. noun and verb chunks and possibly with clause boundaries. In addition to the two corpora, PRESEMT makes use of a bilingual dictionary.

The translation process is split into two phases which correspond to the two types of corpora used. Phase 1, which is called Structure selection, makes use of the small bilingual corpus. It is used to determine the appropriate target language structure for the input SL sentence. The Structure selection module focuses on the macro-level that concerns phrase order and phrase type. The output of the Structure selection module is a set of target language structures that contain TL phrase and tag information and sets of TL lemmas or tokens that have been found in the bilingual dictionary.

The first task of the second translation phase, implemented by the Translation equivalent selection module, is to resolve the **lexical ambiguity** by picking one lemma from each set of possible translations (as, for instance, provided by a bilingual dictionary). In doing so, this module makes use of the semantic similarities between words which have been determined by the Corpus modelling module (cf. Deliverable D3.3.2) through a co-occurrence analysis on the monolingual TL corpus. That way, the best combination of lemmas from the sets of candidate translations is found for a given context.

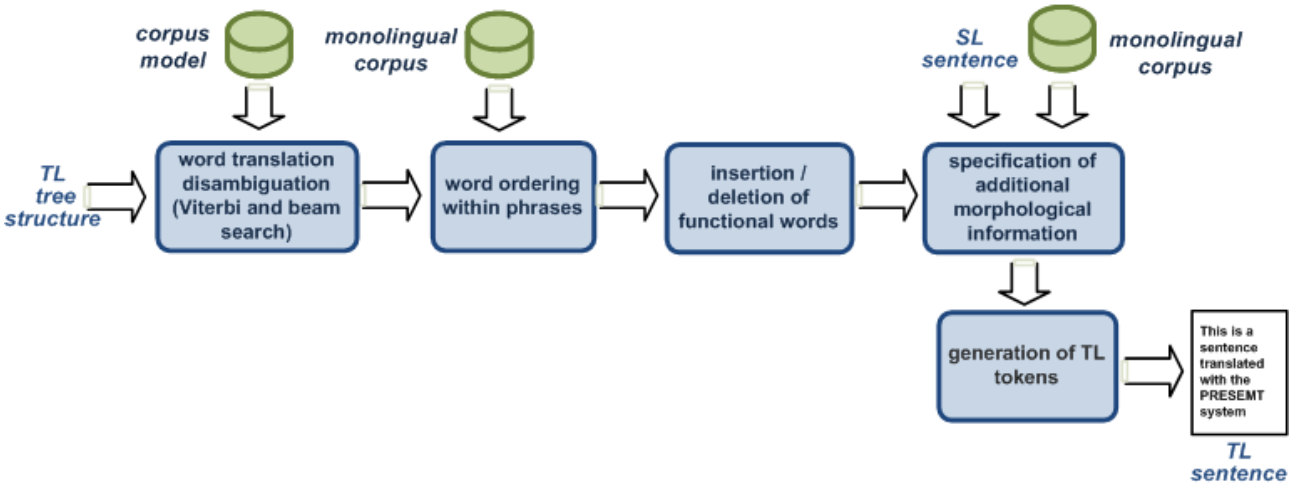
In the second task, the most similar phrases to the TL structure phrases are retrieved from the monolingual corpus. The phrases retrieved from the monolingual corpus serve to specify the **word order** within phrases and to **delete/insert function words** such as articles and prepositions. The notion of similarity used here will provide the foundation for the appropriate optimising strategies of the Optimisation module.

Another task of the Translation equivalent selection module is to add **morphological features** to the TL lemmas. Some morphological information can be transferred from the SL. However, particularly in the case of translating from a morphologically poor language into a morphologically rich one, some morphological information has to be extracted from the monolingual corpus. Whereas morphological phenomena such as NP agreement can be resolved on the basis of the most similar phrases retrieved, other phenomena such as the verb valency go beyond phrase level and therefore need another search step.

Once the lemmas and their morphological properties have been determined, the Translation equivalent selection module generates the **tokens** that correspond to this information. The information needed by the token generation component is also derived from the monolingual corpus.

The functionality of the Translation equivalent selection module is summarised in Figure 1. The output of the Structure selection module is fed into this module, which chooses one translation from the set of possible translations derived from the lexicon. Towards this direction a combined search (including both Viterbi search and beam search) is introduced. After lemma disambiguation, the monolingual TL corpus is searched to detect the most similar phrase to each TL sentence phrase in order to establish internal word order. The morphological features of the TL sentence (i.e. number, gender, case) are transferred either directly from the SL sentence or inferred from the morphological annotation of the large monolingual corpus. Finally, a token generator component is applied to the lemmas of the TL sentence along with their morphological features to map them into tokens.

Figure 1: Data flow in the ***Translation equivalent selection module***



It should be mentioned that the order of implementation of tasks within the Translation Equivalent selection is not fixed. More specifically, the word disambiguation step can precede or follow the search for phrasal equivalents that decides on the optimal word-ordering within phrases. This is discussed in more detail in the next section, but it should be pointed out that by re=ordering these modules a wider or narrower search is selected. Based on the flexible implementation of the PRESEMT platform, both candidates are still under evaluation to select the optimal trade-off in terms of both translation accuracy and processing speed.

3. Tasks of the Translation equivalent selection module

The Structure selection module generates one (or more generally a set of) TL sentence structure(s) that are forwarded for processing to the Phrase equivalent selection module. As noted earlier, the issues that need to be resolved in the second phase include:

1. Disambiguation of translation alternatives
2. Word ordering within phrases
3. Addition and/or deletion of auxiliary verbs, articles and prepositions

3.1 Picking the correct translation alternatives

As has been mentioned above, the Translation equivalent selection module utilises as input the output of the Structure selection module, which is a syntactic structure containing sets of lemmas instead of single lemmas. The present task consists in picking one lemma from each set and that way disambiguating multiple translations of single- or multi-word units of the SL. The disambiguation process uses the semantic similarities between words identified by the Corpus modelling module. In the Corpus modelling module (cf. PRESEMT Deliverables D3.3), three different approaches have been evaluated:

(i) an n-gram language model, extracted from the monolingual corpus for both German and English.

(ii) a model based on Vector Space Modelling of the word space.

(iii) a model based on the creation of a Self-Organising map.

These three approaches can be used alternatively to select the most appropriate translation.

As an example, the approach using an n-gram language model is described herewith. Since word order issues are to a large part resolved at this stage, a Viterbi search is used for this. A particular problem regarding the Viterbi search concerns discontinuous multi-word units such as separable prefix verbs in German have turned out to be problematic for this approach. In the sentence "*Das Kind geht an der Kirche vorbei*" ("*The child walks past the church*") the words "*geht*" and "*vorbei*" are elements of a multi-word unit which is separated by the prepositional phrase "*an der Kirche*". A standard Viterbi search cannot handle such discontinuous words in an effective manner, since it proceeds strictly from left to right. The solution to this problem was to combine the Viterbi search with a beam search: At each Viterbi state, instead of storing only the best result, the *n* best results are stored.

As a further enhancement on this subject, lemmas are disambiguated in a hierarchical manner, from top to bottom, where the hierarchy is established by identifying sentence tokens which serve as phrase-heads. First, the heads of the phrases are disambiguated and then non-heads inside of phrases are disambiguated, taking into account the previous disambiguation results. It is expected that this approach will yield better results because content words are resolved first and depending on other content words. Moreover, to cover the need for a language model on which the disambiguation process will be based, the output of the Corpus modelling module (cf. D3.3.2) on the large monolingual corpus is used.

At present, it is still a research question being studied how to disambiguate German auxiliary verbs. In German, the past tense can be built either with the auxiliary verb "*sein*" (*to be*) or "*haben*" (*to have*). In some cases, both variants are possible (usually with a difference in meaning). It is expected that such cases can be handled with a special syntactic language model based on the monolingual corpus. However, this represents a language-specific phenomenon.

3.2 Searching the corpus for phrasal equivalents

3.2.1 Principles

After lemma disambiguation, the monolingual TL corpus is searched for detecting the most similar phrase to those TL sentence phrases, whose internal word order has not been established. The similarity measure used in this comparison reflects the phrase type, the tokens contained in the phrase in terms of lemma and PoS tag and its morphological features. These factors enter the comparison with different weights, whose relative magnitudes are subject to the optimisation process performed by the Optimisation module.

3.2.2 Selecting equivalent phrases from the monolingual TL corpus

To achieve that, the algorithm accesses the monolingual corpus to determine the most suitable structure for each phrase. Thus it works on the input illustrated in Table 1:

Table 1: Input of the phrase equivalent selection algorithm

	Input sentence / TL (IST)		Monolingual corpus phrase / TL (MCP-TL)	
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	Used	**Tag**	Used
	Phrase	Number, type & sequence	**Phrase**	Number, type & sequence
	Clause	**Not** used	**Clause**	**Not** used

The main issue at this stage is to be able to reorder appropriately any items within each phrase. This entails that tokens between a given phrase of the input sentence, call it **ISP (Input sentence phrase)**, and an aligned TL phrase denoted as **MCP** (**Monolingual corpus (TL) phrase**), are close to each other in terms of number and type. More specifically, the number and identity of items in a given MCP being used as a template is at least equal to (or larger than) the number of elements in the ISP (since it is required to be in a position to handle all tokens of the ISP, it is safer to delete existing MCP elements from their existing locations rather than introduce new ones). In principle, the number of ISP tokens should be equal to or very close to that of the MCP. This means that a search needs to be performed, algorithmically described by the following steps:

Step 1: Iteratively scan all phrases (ISP) of IST, and for each ISP retrieve the corresponding set of phrases of MCP from the TL monolingual corpus.

> The currently processed phrase is denoted by index *i* in the IST. Since there are likely to be several potentially good matches for the current ISP in the set of MCPs, the corresponding phrase from MCP/TL is indexed by index *j*. The corresponding phrases are based on the type of the phrase, the head token lemma and PoS tag as well as the number of words in the ISP.

Step 2: Apply an algorithm[1] for aligning the tokens of the ISP to those of the retrieved MCPs. Specifically the algorithm selected for the alignment of tokens is the Gale-Shapley algorithm and besides alignment information

Step 3: Based MPCs are ordered according to the similarity score and the best scoring phrase is set to be the Chosen Phrase (CHP) to that ISP. If there exists more than one MPC achieving the same similarity score, and each one provides a different word order for the ISP, then we select the MPC with the highest frequency of occurrence in the TL monolingual corpus.

[1] In the second translation phase, the order of tokens in the phrase may be changed to cover the change from the SL-order towards the TL-order, and thus here an assignment-solving algorithm is used instead of the dynamic programming algorithm used in phase 1.

Step 4: Having decided upon **CHP**, perform the following:

1. Order words within ISP directly based on the optimal match defined by the alignment algorithm for CHP.
2. Decide upon articles and possibly other functional words based on the results of the alignment algorithm for the CHP and the actual tokens in the CHP.

More details on the chosen implementation of this method are provided in the next section. More specifically, in section 4.1 the appropriate processing of the large monolingual corpus is discussed with the aim of generating a structured set of phrases which can be readily efficiently searched for to identify the best-matching candidates as quickly as possible. Afterwards, in section 4.2, the methodology for determining the best matching phrase from the set of monolingual phrases is discussed.

3.2.3 Optimising the selection process of phrasal equivalents

The search of the most similar phrase in the monolingual corpus makes use of a set of parameters which will be optimised by the Optimisation module. Within this set of parameters, different types of weights are included. First there are similarity weights which determine the similarity of the PoS tags, lemmas, phrase types and morphological features involved. Secondly, the output of each of these functions contributes to the overall similarity with a certain weight. All these function weights are fine-tuned by the Optimisation module, which handles in a unified manner both the parameters of the Structure Selection module and the Translation Equivalent Selection module. It should be noted that this represents a small change to the original PRESEMT concept, where two different optimisation modules had been envisaged, one for each phase of the translation process. However, the consortium, early in the 2nd year of the project decided to combine the two optimisation phases, mainly to avoid the need to create separate training data to achieve the optimisation in terms of sentence structure, which would increase the effort of optimisation while introducing the risk of not achieving complete decoupling between the results of the first and the second optimisation phases.

As already described in Annex I of the PRESEMT Grant Agreement, the optimisation algorithms to be used include a number of candidates such as Genetic Algorithms (GAs), Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO) and Artificial Immune Systems (AIS). In the initial experiments, Genetic Algorithms have been used. ACO is the second algorithm in line to be used. Finally, work has been completed on an implementation of the PSO, which is expected to be studied in the later stages of the project.

Within research in earlier MT systems, partners from the consortium have worked mainly with GAs and multi-objective functions such as SPEA2 (Sofianopoulos et al., 2010) for the optimisation of parameters. It is intended to continue this work within the current project, employing GAs, while also investigating some of the aforementioned alternatives. The results of the optimisation activities will be reported upon in detail in the next reporting period.

3.3 Adding morphological features

Some of the morphological features of the TL sentence like number can be transferred from the source language. Others must be inferred from the large monolingual corpus, if they do not exist in the SL (for instance case, that exists only rudimentarily in the English language) or if they are of no help because their respective values are too different in the source and target languages. But even in such cases it is expected to be able to transfer more abstract properties of the SL to the TL such as grammatical functions like subject, object and complement.

In the implementation for the 1st PRESEMT prototype, lexical morphological information such as gender is collected from the morphological annotation of the monolingual corpus. The morphological information for each lemma – token pair is condensed into a token generation table in order to save memory space. In morphologically poor languages without any case marking such as English some rudimentary grammatical function information is inferred. When translating into a morphologically rich language with case marking such as German it has proven sufficient to mark the subject in the non-case marking language. This corresponds in the morphologically rich language to a distinction of the case used for marking the subject versus the cases used for marking the objects (ignoring propositions for the time being). English subjects translate into a nominative NP in German in most instances. All other English noun phrases translate into dative or accusative noun phrases in German in most instances. In order to account for agreement, the information from the different sources is then unified via a simple unification mechanism. That way agreement within the NP is also accounted for.

Gender of nouns and agreement within the NP could also be handled with appropriate language models that model the head lemmas of noun phrases. Because adjectives in German agree with determiners in the feature declension, determiners should be modelled that way as well. Alternatively, there is the possibility to assign features within the NP by unifying the tokens with grammar fragments that were extracted from the monolingual corpus.

One problematic phenomenon that has scarcely been addressed in statistical machine translation is the valency of verbs. Experiments have been conducted for extracting verb case frames from a large German corpus. The initial results were encouraging insofar as the frames with the top frequencies were all typical for the corresponding verbs. So it seems promising to build special language models that incorporate syntactical information depending on verb lemmas and to use them for assigning case to the verb arguments.

However, it is unclear yet how the case frames found are projected onto the TL structure since no information is available about grammatical functions. Another possibility is to view the assignment of a verbal case frame as a classification problem that is handled by a statistical learning algorithm. Features used in this task could be the position and the head lemmas of the NPs and the VP as well as morphological features and grammatical functions (if available). The classification problem could then be handled by one of the several statistical learning algorithms as maximum entropy, support vector machines, decision trees and so on. Implementations of these algorithms are freely available through software packages such as MALLET[2].

3.4 Generating word forms

A token generator component is applied to the lemmas of the TL sentence together with their morphological features. That way the final word tokens are generated. At the moment, this token generator is simply a mapping from lemmas and morphological features to word tokens. This mapping has been extracted from morphological and lemma information contained in the monolingual corpus.

The advantage of this simple method of setting up tables is that it can be easily applied to new target languages. Automatically constructing a rule-based token generator for a new TL would be much harder. The disadvantage is that the token generation table might contain inflectional gaps and that it takes a lot of memory space. In order to lower the size of the token generation table, this has been restricted to the lemmas found in the bilingual dictionaries.

Due to data sparseness, such a mapping will always contain gaps particularly in case of rather infrequent words. A more sophisticated approach would therefore try to close the gaps in the inflectional paradigms of the lemmas. This could for instance be done by inferring inflectional paradigms of infrequent words from those of more frequent words.

[2] http://mallet.cs.umass.edu/classifier-devel.php

4. Translation equivalent selection implementation

The objective of the Translation equivalent selection module is to handle more fine-grained properties of the TL sentence which has been generated from the first phase of the PRESEMT translation process and produce the final translation of the input source sentence. During this phase, only information extracted from the TL monolingual corpus is utilised to resolve issues such as word translation disambiguation, micro-level word order and specification of additional morphological information.

The Translation equivalent selection module utilises the output of several PRESEMT modules as a prerequisite.

* The Corpus creation and annotation module provides large monolingual corpora.
* The Corpus modelling module provides appropriate language models based on the monolingual corpora.
* The Structure selection module (depending in its turn on the Phrase aligner module) provides the input structure for the Translation equivalent module.

A range of disambiguation tasks have been identified that need to be resolved by the Translation equivalent selection module.

The extraction of large monolingual corpora has been implemented, as reported in Deliverable D3.1.1., while language models based on these corpora have been created (for detailed information cf. Deliverable D3.3.1). For the word reordering task of the Translation equivalent selection module, a phrase-based indexing of the monolingual TL corpus needs to be performed during pre-processing. This phrase-based organisation of the corpus will be described in detail, followed by details regarding the implementation of the two major tasks of the module.

4.1 Organising the monolingual corpus

The language models that have been produced by the Corpus modelling module can only be used for the translation disambiguation step, and thus another form of interfacing with the monolingual corpus needed to be established for the word reordering step. The TL corpus needed to be indexed in terms of the phrases it contains, based on the following information: (i) type of the phrase, (ii) lemma of the phrase head, (iii) PoS tag of the phrase head and (iv) number of tokens in the phrase.

The indexing is performed by processing the XML representation of the monolingual corpus and extracting only the phrases. Each phrase is then transformed from XML to the java object instance used within the PRESEMT system. The phrases are then organised in a hash map that allows multiple values for each key, using as a key the 4 criteria mentioned above. Along with each phrase we also keep the number of occurrences of the phrase in the corpus, so as to produce a frequency of occurrence. Finally, each map is serialized and stored in the appropriate file in the PRESEMT path, with each file given an appropriate name, for easy retrieval. For example, for the English monolingual corpus, all verb phrases with the lemma of the head token being "read" (verb) and the PoS tag "VV" that contain 2 tokens in total, are stored in the file "Corpora\EN\Phrases\VC\read_VV". A separate file will contain all verb phrases with the head "read" (verb) and 3 tokens in total, while a further file will be created for all noun phrases with the head "read" (noun) and 3 tokens in total.

For example, let us assume a very small TL-side monolingual corpus consisting only of the following sentence: "A typical scheme would have eight electrodes penetrating human brain tissue; wireless electrodes would be much more practical and could be conformal to several different areas of the brain." Then, the processed monolingual corpus, emanating from only this single sentence, following the operation of a parser is shown in Table 2.

Table 2: General parser output (TL side)

Phrase id	Phrase type	Phrase content
1	PC	A typical scheme
2	VC	Would have
3	PC	Eight electrodes
4	VC	penetrating
5	PC	Brain tissue
6	PC	Wireless electrodes
7	VC	Would be
8	PC	Much more practical
9	VC	Could be
10	PC	conformal
11	PC	To several different areas
12	PC	Of the brain

In Table 3, the resulting set of phrases is depicted, processed to generate the selected indexes.

Table 3: Set of phrases generated from the monolingual corpus of Table 2

Phrase id	Phrase type	Token number	Phrase head	Phrase content
1	PC	3	Scheme/NN	A typical scheme
2	VC	2	Have/VH	Would have
3	PC	2	Electrode/NN	Eight electrodes
4	VC	1	Penetrate/VV	penetrating
5	PC	2	Tissue/NN	Human brain tissue
6	PC	2	Electrode/NN	Wireless electrodes
7	VC	2	Is/VB	Would be
8	PC	3	Practical/JJ	Much more practical
9	VC	2	Is/VB	Could be
10	PC	1	Conformal/JJ	conformal
11	PC	4	Area/NN	To several different areas
12	PC	3	Brain/NN	Of the brain

Figure 2: Organised set of files for phrasal equivalents according to the indexing scheme selected

File 1	VC/Have_VH_2
Id	**content**
2	Would have

File 2	VC/Is_VB_2
Id	**content**
7	Would be
9	Could be

File 3	VC/penetrate_VV_1
Id	**content**
2	penetrating

File 4	PC/scheme_NN_3
Id	**content**
1	A typical scheme

File 5	PC/electrode_NN_2
Id	**content**
3	Eight electrodes
6	Wireless electrodes

File 6	PC/Tissue_NN_3
Id	**content**
5	Human brain tissue

File 7	PC/Practical_JJ_3
Id	**content**
8	Much more practical

File 8	PC/conformal_JJ_1
Id	**content**
10	conformal

File 9	PC/areas_NN_4
Id	**content**
11	To several different areas

File 10	PC/brain_NN_3
id	**content**
12	Of the brain

Based on this example, a number of observations may be made:

* The number of files to be created is large.
* the existence of a token in a phrase does not necessitate that this specific phrase will be grouped together with other phrases. For instance, the phrase "human brain tissue" is organised according to the head, i.e. the token "tissue".
* in the current organization the functional head (as defined for the purposes of the PRESEMT translation process) is not involved in the phrase comparison.

Finally it should be noted that for large corpora, in order to reduce the number of files created, if a subgroup file remains very small (based on the definition of a small threshold value, it does not remain independent but is grouped with all other phrases from very small files. This allows (i) the reduction of files, in order to prevent the creation of an excessive number of groups but also allows the system to process phrases with heads for which no groups have been created from the monolingual corpus. Another step towards reducing the number of produced files is to altogether skip the creation of files for phrases that only contain a single word, as these would not be useful for word reordering.

One issue that has been studied during the implementation of the Translation equivalent selection is the sheer size of the monolingual corpus, which necessitates special techniques to organise and process it, so that during run-time the required intermediate results are readily available, to minimise the computational load. To obtain a more precise understanding of the task, it is essential to have a quantitative view of the corpora involved. As described in Deliverable D3.1.2, the monolingual corpora for the three PRESEMT target languages are summarised in Table 4.

Table 4: Characteristics of monolingual corpora

	English	German	Italian
Size in tokens	3,658,726,327	3,076,812,674	2,874,779,294
Number of files (in Mwords blocks)	87,000	96,000	Pending
Number of sentences[3]	$1,0*10^8$	$9,5*10^7$	Pending
Number of phrases	$8,0*10^8$	$6,0*10^8$	Pending

Of course, one can expect that the number of unique phrases for a given corpus will be much lower than that quoted in Table 4. As an indication, for the first experiments, a small subset of the corpus for English was processed to support the first version of the Translation Equivalent selection. The actual characteristics of this subset are summarised in Table 5. Clearly, though this is a small subset, the number of phrases is substantial. In addition, it becomes evident that as larger portions of the corpus are processed to provide a more complete (in terms of both frequencies of both patterns of tokens as well as their frequencies of occurrence) models are created, the appropriate organisation of such large sets of phrases becomes of paramount importance to speed up the translation process.

Table 5: Characteristics of sub-corpus used initially

	English	German	Italian
Size in tokens	3,658,726,327	3,076,812,674	2,874,779,294
Number of files (in Mwords blocks)	87,000	96,000	Pending
Number of sentences[4]	$1,0*10^8$	$9,5*10^7$	Pending
Number of phrases	$8,0*10^8$	$6,0 *10^8$	pending

In the current version, the TL monolingual corpus is indexed in terms of (a) the phrase type (i.e. whether it is a noun phrase or a verb phrase), (b) the lemma and part-of-speech tag of the phrase head and (c) the number of tokens. However, it is likely that a different indexing scheme may prove more effective. For instance, the environment of the phrase may also be required to be stored (i.e. the type of the previous and next phrases within the sentence may be of use in the translation equivalent selection, and in this case the phrase organisation may be modified). In addition, the size of a phrase may be calculated on the basis of all tokens contained or alternatively on the basis of tokens excluding function words. These modifications may result in a different modelling of the corpus.

[3] The numbers of sentences and phrases are estimates rather than exact values.
[4] The numbers of sentences and phrases are estimates rather than exact values.

4.2 Translation Equivalent Selection tasks

The implementation for the Translation equivalent selection module, which is complementary to the dynamic programming approach for the Structure selection module, is detailed below.

The module input is the output of the Structure selection module, appended with the TL lexical translations of the SL words. In the simplest case, each sentence contained within the document to be translated is processed separately, so at this point, to simplify the description no reference will be made to the use of inter-sentential information. The Translation equivalent selection module undertakes two basic tasks: the disambiguation of translation alternatives and the word reordering within the phrases of each sentence.

4.2.1 Solving translation ambiguities

The first task is to select the correct TL translation of each word. Translation disambiguation is performed either based on either of the following 3 models, (i) the n-gram vector space models, (ii) the vector space model or the (iii) Self-Organising Maps extracted from the TL monolingual corpus by the Corpus modelling module as developed in Task T3.3 (cf. Deliverables D3.3). In this respect, the lexicon look-up has been performed in the end of the first translation phase (Structure Selection)to translate tokens from the SL to their candidate TL translations. Following that operation, the disambiguation algorithm is invoked to resolve multiple translations, and determine the optimal translation for each token (or group of tokens) for the given input sentence.

For lemma disambiguation a number of different lemma-based language models for German and English have been tested. In addition, tag-based models have been set up and tested. They were meant to improve the disambiguation of different syntactic structures. However, tag-based 5-gram models in combination with lemma-based 3-gram models did not lead to any improvements. The reason could be that lemma-based n-gram models also indirectly disambiguate syntactic structure since they are word-order sensitive. The tag-based models might still prove useful if lemma disambiguation strategies are used that are not word order sensitive (vector space models or SOM without Viterbi search).

In order to generate TL tokens with the appropriate morphology, mechanisms for the treatment of agreement phenomena and simple valency patterns have been established. And finally, token generation components for English and German have been set up. In order to develop a method that can be easily applied to new target languages the TL token generator is not rule-based but based on a large lookup table called token generation table that has been automatically extracted out of the huge, tagged monolingual corpora for DE and EN.

4.2.2 Establishing correct word order

The second task involves establishing the correct word order within each phrase. To simplify the description, at this point, it is assumed that the disambiguation step described in the previous sub-section precedes establishing the order of tokens. Therefore, at this point, the system has established the correct TL order of phrases within each sentence and has solved all translation ambiguities, but the order of words is still the same as in the initial ISS.

In the present implementation of the Translation equivalent selection, the TL monolingual corpus is indexed in terms of (a) the phrase type, (b) the lemma of the main verb of the sentence from which a given phrase was extracted and (c) the phrase head & functional head. The matching algorithm first retrieves similar phrases from the indexed TL phrase corpus and then selects the most similar one through a comparison process, which is viewed as an assignment problem, in order to enable word reordering.

This process can be solved via an algorithm such as the Gale-Shapley algorithm (Gale & Shapley, 1962 & Mairson, 1992) or the Kuhn–Munkres algorithm (Kuhn, 1955 & Munkres, 1957). The Kuhn-Munkres approach computes an exact solution of the assignment problem, to indicate the optimal matching between elements. During experimentation with EBMT approaches in the METIS-2 project, it has been found that the solution of the assignment problem is responsible for a large fraction of the computation.

On the contrary, the Gale-Shapley algorithm solves the assignment problem by separating the items into 2 distinct sets with different properties. In this approach, the two sets are termed (i) suitors and (ii) reviewers. In the present MT application, the aim is to create assignments between tokens of the SL (which are assigned the role of suitors) and tokens of the TL (which undertake the roles of reviewers). In the Gale-Shapley algorithm, the two groups have different roles. More specifically, the suitors have the responsibility of defining their order of preference of being assigned to a specific reviewer, giving an ordered list of their preferences. Based on these lists, the reviewers can select one of the suitors by evaluating them based on their ordered lists of preference, in subsequent steps revising their selection so that the resulting assignment is optimised. As a consequence, this process provides a solution which is suitor-optimal but potentially non-optimal from the reviewers' viewpoint. However, the complexity of the algorithm is substantially lower to that of Kuhn-Munkres and thus it is used in the translation equivalent selection process as the first choice, so as to reduce the computation time required.

In order to control the number of retrieved phrases that are handled in this comparison process and therefore keep under control the execution time of the algorithm, the TL phrase retrieval can be performed in more than one step, taking advantage of the multiple corpus indices. Through this comparison process the algorithm can also resolve other issues, i.e. the insertion or deletion of words such as articles and other auxiliary tokens.

As an indicative example, assume that the translation pair studied is French-to-English, and that one of the phrases to be translated reads: "electrodes neufs". In that case, by accessing the set of files of figure 3, the search algorithm will determine that the most similar indexed file is file 5. Therefore, the Gale-Shapley algorithm will be invoked to compare the phrase "electrodes new" with the two phrases "eight electrodes" and "wireless electrodes" based on the lemmas and tags of tokens (note that this comparison is performed in the target language, solely, but with the ISP keeping the order of the source-language side following the lexicon look-up. In this case, both MCSs share one identical word with the ISP phrase. Thus the actual similarity will be determined on the basis of the tag similarity between adjectives (PoS tag of token "neuf" from the ISP) and numerals ((PoS tag of token "eight" from MCS with id.3) or nouns ((PoS tag of token "wireless" from MCS with id.6). Assuming that the PoS tag similarity between numerals and adjectives is higher, the CHP will be MCS(id.3), which will result in a reordering of the tokens in the ISP from "electrodes new" to "new electrodes".

5. Further work

A number of research directions are open with respect to the Translation Equivalent selection. These have to do mainly with improving the quality of the generated translation but also reducing the computational load of determining the translation of a given phrase. As described in the previous sections, excluding any improvements in other PRESEMT modules, to a large extent this involves the suitable preprocessing of the raw monolingual corpus into suitable groups of phrases, since it is this off-line preprocessing that reduces the computational load of the actual on-line translation.

Each group of phrases is stored into a single file. Hence, the optimal size of groups represents a trade-off between the creation of (i) small files with a limited number of phrases whose retrieval is fast and for which all member phrases can be evaluated to determine the optimal phrase and (ii) having a limited number of files containing phrase groups, so that the search process is not hindered by the fragmentation of the data.

An additional variation concerns the use of the Kuhn-Munkres algorithm instead of the Gale-Shapley one. In this case, the benefits expected would be in terms of translation accuracy, since the Kuhn-Munkres method implements an exhaustive search for the global optimum, potentially giving more accurate alignments. Such benefits would have to be achieved without an excessive increase in the processor requirements to complete the translation process.

Another open question concerns the optimal order of operations in the Translation Equivalent selection. More specifically, the disambiguation step via the language model created on the basis of the monolingual process, which can precede or follow the search for phrasal equivalents. According to the organisation of phrases introduced in section 4.1, the access to phrasal equivalents is based on the head lemma of the given phrase. If multiple translations for the head are returned by the lexicon, but these are disambiguated prior to searching for phrasal equivalents, a much smaller set of candidate phrases will be retrieved. On the contrary, a wider set of phrases including all phrases with all candidate head translations will be considered in the case that the search for phrasal equivalents precedes the disambiguation step. Based on the flexibility of the PRESEMT platform, it remains possible to choose any of these candidate approaches in order for them to be comparatively evaluated in order to determine the most appropriate one, though a trade-off exists in terms of both translation accuracy and processing speed. The merits and drawbacks of each approach will be presented in subsequent deliverables.

The next task is to start using the large monolingual corpora to resolve disambiguation problems that are left open by the output of the Structure selection module and to apply the optimisation strategies. This work will be reported upon in the next version of this deliverable.

6. References

Emami Ahmad, Jelinek, Frederick. 2004. Exact Training of a Neural Syntactic Language Model. ICASSP. Montreal, Quebec.

Brent, Michael R. 1997. Automatic acquisition of subcategorization frames from untagged text. ACL '91 Proceedings of the 29th annual meeting on Association for Computational Linguistics.

Bojar, Ondrej. 2003. Towards Automatic Extraction of Verb Frames. Prague Bulletin of Mathematical Linguistics, 79.

Bordag, Stephan. 2007. Unsupervised and knowledge-free morpheme segmentation and analysis. In Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum (CLEF).

Charniak, Eugene. 2001. Immediate-head parsing for language models. In: Meeting of the Association for Computational Linguistics.

Charniak, Eugene, Knight, Kevin, and Yamada, Kenji. 2003. Syntax-based Language Models for Statistical Machine Translation. In MT Summit IX, Intl. Assoc. for Machine Translation.

Chelba, Ciprian and Jelinek, Frederick. 1998. Exploiting syntactic structure for language modeling. In Christian Boitet and Pete Whitelock, editors, Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics, San Francisco, California. Morgan Kaufmann Publishers.

Chen, Yidong, Shi, Xiaodong, Zhou, Changle, Hong, Qingyang. 2008. Incorporating Syntax-Based Language Models in Phrase-Based SMT Models. In Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering. Xiamen, China.

Creutz, Mathias & Lagus, Krista. 2002 Unsupervised discovery of morphemes. In Proceedings of the ACL workshop on Morphological and phonological learning.

Gale, D. & Shapley, L. S., 1962. College Admissions and the Stability of Marriage. American Mathematical Monthly, Vol. 69, pp. 9-14.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. Computational Linguistics, Vol.27, No.2, pp. 153-193.

Goldsmith, John. 2006. An algorithm for the unsupervised learning of morphology. Natural Language Engineering, Vol. 12, pp.353-371.

Graham, Yvette and van Genabith, Josef. 2010. Deep syntax language models and statistical machine translation. In SSST-4 - 4th Workshop on Syntax and Structure in Statistical Translation at COLING 2010. Beijing, China.

Korhonen, Anna. 2002. Subcategorization Acquisition. Technical Report UCAM-CL-TR-530, University of Cambridge, Computer Laboratory, Cambridge, UK, February.

Kuhn, H.W. 1955. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, Vol. 2, pp.83-97.

Mairson, H., 1992. The Stable Marriage Problem. The Brandeis Review, 12:1. Available at: http://www.cs.columbia.edu/~evs/intro/stable/writeup.html

Munkres, J., 1957. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, Vol. 5, pp.32-38.

Schone, Patrick and Jurafsky, Daniel. 2000. Knowledge-free induction of morphology using latent semantic analysis. In Proceedings of CoNLL-2000 and LLL-2000.

Schulte im Walde, Sabine. 2009. The Induction of Verb Frames and Verb Classes from Corpora. In: Anke Lüdeling and Merja Kytö (eds.) Corpus Linguistics. An International Handbook. Mouton de Gruyter, Berlin.

Snover, Matthew G., Jarosz, Gaja E. and Brent, Michael R. 2002. Unsupervised learning of morphology. In Proceedings of the ACL workshop on Morphological and phonological learning.

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.

7. Appendix I: Formulation of the similarity of phrases

The main point is to determine what can be expected to be useful in the search for the appropriate microstructure. By utilising the information in the monolingual corpus, the following information could be useful to search for when going through the set of Monolingual corpus sentences, MCS:

A. The phrase-type of the currently-studied phrase in the ACS-TL (denoted as *phr_typ(0)*) as well the phrase-types of its immediate neighbours (denoted as *phr_typ(-1)* and *phr_typ(+1)* for the preceding and following phrases, which would provide information about the immediate environment of the current phrase;

B. The lemma of the head of the currently-studied phrase in the ACS-TL (denoted as *phr_head(0)*) as well as possibly the corresponding heads of its immediate neighbours (denoted as *phr_head(-1)* and *phr_head(+1)* for the preceding and following phrases;

C. The extended tags of the heads of the currently studied ACS-TL as provided by the input sentence, in relation to the MCS;

D. The set of candidate heads in the current ACS-TL, in comparison to the number and type of heads contained in the current MCS.

Item A contribution: This expresses the similarity between types of phrases in the neighbourhood of the current phrase. Assuming that a neighbourhood of [-1, +1] is studied (this can be generalised in a rather straightforward manner), this is expressed for the p-th phrase as:

$$Simil\_A_p = \sum_{i=-1}^{+1} \left\{ w_A(i) \bullet sim\_typ(TYP\_phr(IST, i), TYP\_phr(MCS_j, i)) \right\} \quad \text{(B1)}$$

where $TYP\_phr(x, y)$ is a function returning the type of the y-th phrase of sentence x. The output of function sim_typ is the similarity of the types of the 2 phrases that are specified as the arguments of the function, each of which is the Tag of the respective phrase. Finally IST_j corresponds to the currently evaluated sentence of the monolingual large corpus. The weights $w_A(i)$ correspond to the tunable parameters of this type of similarity.

Item B contribution: This expresses the similarity between heads of phrases in the neighbourhood of the current phrase. Assuming that a neighbourhood of [-1, +1] will be studied, this is expressed as:

$$Simil\_B_p = \sum_{i=-1}^{+1} \left\{ w_B(i) \bullet sim\_head(HL\_phr(IST, i), HL\_phr(MCS_j, i)) \right\} \quad \text{(B2)}$$

where $HL\_phr(x, y)$ is a function returning the lemma of the y-th phrase of sentence x. It is expected that the output of function sim_head is the similarity of the lemmas of the heads of the 2 phrases that are specified as the arguments of the function. Finally MCS_j corresponds to the currently evaluated sentence of the monolingual large corpus. The weights $w_B(i)$ correspond to the tuneable parameters of this type of similarity.

Item C contribution: This expresses the similarity between the tags of phrase heads in the neighbourhood of the current phrase. Assuming that a neighbourhood of [-1, +1] will be studied, this is expressed as the sum of two similarities in terms of tags, one being expressed based on the base tags and the other on the extended tags, as expressed by (B3) and (B4) respectively:

$$Simil\_C1_p = \left\{ w_{C1}(i) \bullet sim\_basetag(HT\_phr(IST,i), HT\_phr(MCS_j,i)) \right\} \quad \text{(B3)}$$

$$Simil\_C2_p = \left\{ w_{C2}(i) \bullet sim\_exttag(HT\_phr(IST,i), HT\_phr(MCS_j,i)) \right\} \quad \text{(B4)}$$

where $HT\_phr(x, y)$ is a function returning the tag of the head of the y-th phrase of sentence x. It is expected that the output of function $sim\_basetag$ is the similarity in terms of PoS tags of the types of the two phrases that are specified as the arguments of the function. Similarly, the output of function sim_exttag is the similarity in terms of extended tags of the types of the two phrases that are specified as the arguments of the function. Finally IST_j corresponds to the currently evaluated sentence of the monolingual large corpus. The weights $w_{C1}(i)$ for the base tag and $w_{C2}(i)$ for the extended tag correspond to the tuneable parameters of this type of similarity.

Item D contribution: This expresses the similarity between the lemmas of candidate heads in the neighbourhood of the current phrase. Assuming that a neighbourhood of [-1, +1] will be studied, this is expressed as:

$$Simil\_D_p = \sum_{i=-1}^{+1} \left\{ w_E(i) \bullet \sum_{l=1}^{N} sim\_lemma(HL_l\_phr\ (IST,i), HL_l\_phr\ (MCS_j,i)) \right\} \quad \text{(B5)}$$

where $HL\_phr(x, y)$ is a function returning the lemma of the candidate heads of the y-th phrase of sentence x. It is expected that the output of function sim_lemma is the string-based similarity of the tokens that are candidate heads in the two phrases that are specified as the arguments of the function. The internal summation is performed by scanning through all candidate heads as provided in the current phrase of the IST sentence and trying to locate them in the corresponding phrase in the MCS_j sentence currently being evaluated. Finally MCS_j corresponds to the currently evaluated sentence of the monolingual large corpus. The weights $w_E(i)$ correspond to the tunable parameters of this type of similarity.

Then the final formulation is as follows:

$$Score\_Phase2(MCS_j) = \sum_p \left\{ Simil\_A_p + Simil\_B_p + Simil\_C1_p + Simil\_C2_p + Simil\_D_p \right\} \quad \text{(B6)}$$

where p sums over the phrases of the IST sentence.

where the process of phase 2 of the MT process is to locate the one phrase that maximises the corresponding score. During the optimisation process (Task T5.2), the parameters to be optimised are weights $w_A(i)$, $w_B(i)$, $w_{C1}(i)$, $w_{C2}(i)$ and $w_D(i)$, for $i \in \{-1,0,+1\}$.

Candidate weights (to be used e.g. in the optimisation phase of Task 5.2), would then relate to weighing factors for points A, B, C and D above, where one factor could be provided for each value of displacement (of -1, 0, +1).