FI MU

Faculty of Informatics
Masaryk University Brno

Phrase Aligner

by

Michalis Troullinos

FI MU Report Series **FIMU-RS-2013-2**

Copyright © 2013, FI MU **March 2013**

**Copyright © 2013, Faculty of Informatics, Masaryk University.
All rights reserved.**

**Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.**

Publications in the FI MU Report Series are in general accessible via WWW:

`http://www.fi.muni.cz/reports/`

Further information can be obtained by contacting:

**Faculty of Informatics
Masaryk University
Botanicka 68a
602 00 Brno
Czech Republic**

Phrase aligner module (PAM)

Michalis Troullinos

November 29, 2013

1 Aim of the Phrase aligner module

In the context of PRESEMT, Phrase aligner module (WP3: Corpus extraction & processing algorithms), which “aims at defining the algorithm required to define phrases in sentences in both the source and target languages of a given language pair, these phrases being aligned in the two sentences". The work described relates to processing a bilingual corpus with a twofold purpose: (a) corpus alignment and (b) elicitation of a phrasing model compatible with both languages of a given language pair.

To implement the aforementioned processing of a bilingual corpus, two distinct modules are being developed, the **Phrase aligner module (PAM)** and the **Phrasing model generator (PMG)**, both of which are envisaged to function as language-independent methods, thought their outputs will be specific to the given language pair. The Phrase aligner module is responsible for aligning a bilingual corpus at word level and mapping the TL phrasing model onto the source language. The Phrasing model generator, being dependent on the output of the first module, is assigned with the task of (a) eliciting the SL phrasing model, which is implicit in the aligned corpus, and (b) applying it to any new SL text.

In the present report, emphasis is placed on describing the Phrase aligner module.

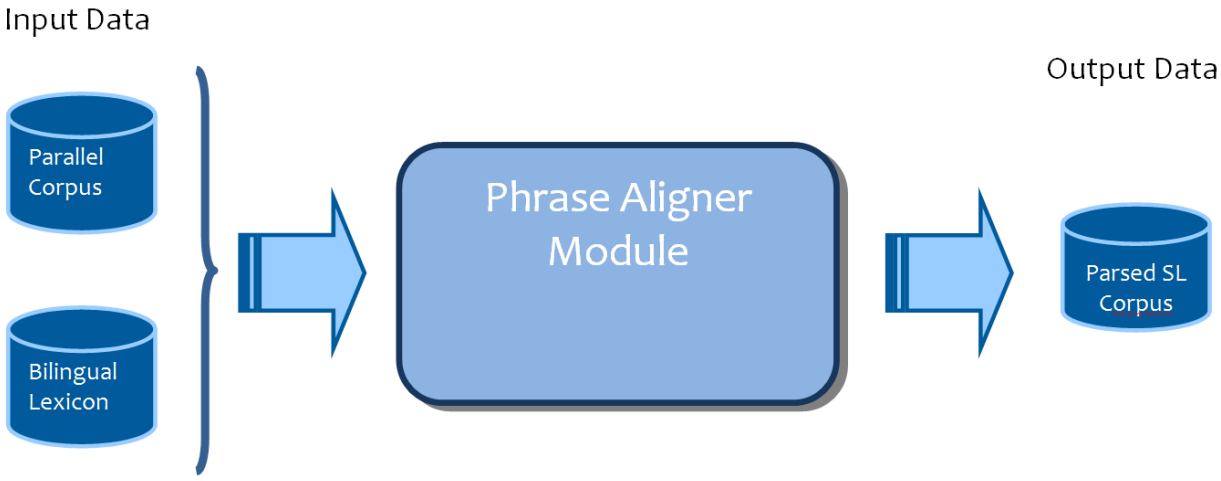


Figure 1: Overview of Phrase Aligner Module

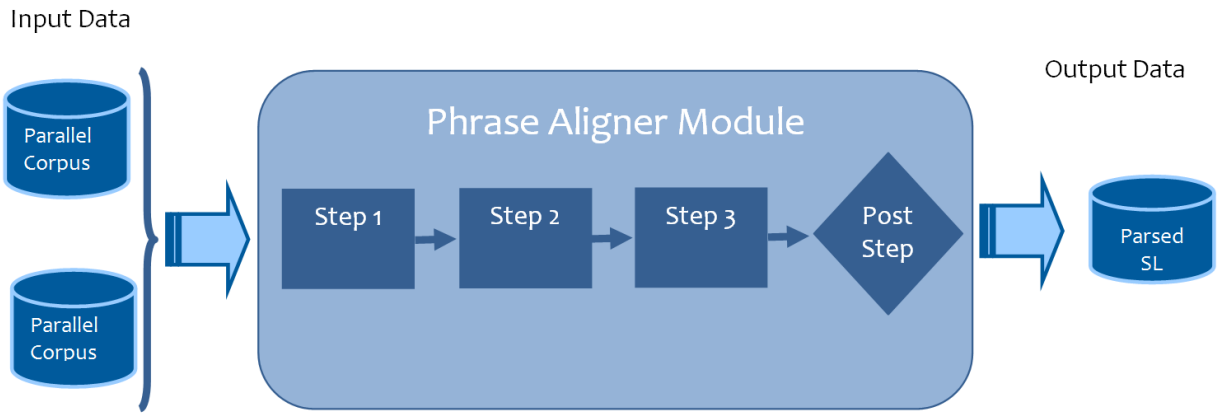


Figure 2: Overview of Phrase Aligner Module

2 Basic aspects

The approach to phrase alignment module (PAM) within a language pair is applied via a small bilingual corpus containing a few hundred sentences in the source and target languages. For each corpus sentence, the words are aligned to each other via the bilingual lexicon. In the present implementation, where the TL phrasing model is taken into account, the phrase alignment process essentially is the clustering of all words in an SL sentence into phrases, observing the condition that the given phrases in the two languages do not overlap. Figure 1 depicts the function of the Phrase aligner module. Figure 2 illustrates in more detail the PAM structure, which concerns three processing steps as well as one post-processing step. Finally, Figures 3 and 4 illustrate a simple conceptual example.

3 Design of the PAM algorithm

Based on the concept of the PRESEMT project, the Phrase aligner uses as input only the following resources and linguistic tools:

1. Bilingual lexicon from source to target language;
2. SL tagger and lemmatiser (the tagger may provide both basic PoS characterisation of words as well as detailed grammatical features such as case, number, person etc.);
3. TL tagger, lemmatiser and shallow parser[1];
4. TL clause boundary detection tool[2]
5. Transliteration rules from non-Latin to Latin characters. (optional)

Starting with these resources and tools, the following information is available:

- Information on possible **word** and **lemma correspondences between source and target languages**, extracted from the bilingual lexicon. This can be any of the following:
  - one-to-one correspondence (when an SL word has exactly one word translation in the target language), e.g. "τραπέζι" (Greek) → "table" (English)
  - one-to-many correspondence (when an SL word corresponds to a TL multi-word unit), e.g. "τραπεζομάντηλο" (Greek) → "table cloth" (English)
  - many-to-one correspondence (when an SL multi-word unit corresponds to a TL single one), e.g. "όχθη ποταμού" (Greek) → "riverbank" (English)
  - many-to-many correspondence (when an SL multi-word unit corresponds to a TL multi-word unit), e.g. "για να" (Greek) → "in order to" (English)

[1]Let us remind that a similar analysis could also be made in the case that the shallow parser was available for the source rather than the target language. Since in the present implementation the target language will serve as the basis for the extraction of the phrasing model, this case will not be further discussed.

[2]Note that there are parsers which identify the constituent clauses of sentences. Either way, in the current state of development, the information about clauses has not been utilised.

- **Tag** correspondence between the source and the target languages (in the case of languages that exhibit **rich morphology**, additional information is obtainable concerning the agreement on grammatical features apart from merely the part of speech)
- **Distance** between words[3]
- **Decomposition** of the sentence in the target language in sub-sentential segments based on the output of the parser available.

Based on this set of inputs, the Phrase aligner module needs to decide on the optimal segmentation of the source language into phrases. Thus a multi-criterion-type comparison is essential, where the different inputs are prioritised and combined accordingly. It is noteworthy that not all aforementioned inputs need be present for the Phrase aligner to work (for instance, in the case of morphologically poor languages, the input corresponding to the second bullet will not be available). Still the employment of all aforementioned inputs, if available, will result in a higher accuracy of alignment.

[3]In the absence of additional information this feature can facilitate the identification of optimal segments (phrases).

4 Implementation of the PAM algorithm

In the present section, the actual implementation of the PRESEMT Phrase aligner is described. This has been implemented in Java, using the definitions of the PRESEMT skeleton framework, in order to be seamlessly integrated with the other modules of the PRESEMT prototype.

The workflow of the Phrase aligner algorithm is presented in [4]. The module receives as input on the one hand the small bilingual corpus annotated with lemma, PoS and phrase information, and on the other a bilingual lexicon, which includes lemma and PoS info, for the given language pair.
PAM operates in a three-step mode:

Step 1 The words in the SL sentence are aligned to those of the TL sentence based on the correspondences provided by the bilingual lexicon. The word alignment thus guides the preliminary grouping of the SL words into phrases.

Step 2 The SL words which have remained unaligned are grouped into phrases by taking into consideration statistical-type information extracted from the bilingual lexicon and additional grammatical features (if these are available in the tagset).

Step 3 The residual unaligned SL words are each grouped into the phrase to which the majority of their immediate neighbours belongs to. Also the words in SL parsed sentence are reordered according to the order in the unparsed SL sentence.

Moreover there is a post-processing step that rejects the sentence pairs with low lexicon coverage. This step ensures that sentence pairs with low quality alignments will not form a part of the output.

A detailed description of these three steps and the post-processing step is provided in the subsequent sections, while Appendix 1 includes a pseudo-code implementation of the Phrase aligner mechanism.

4.1 Step 1: Alignments based on the bilingual lexicon

Alignments via the bilingual lexicon

The word aligner algorithm performs alignment of words in an SL sentence to words in the TL sentence via the bilingual lexicon. The alignment cases may be any of the following:

- **Single alignment**: The translation equivalent of an SL word appears only once in the TL sentence (single-aligned words).
- **Multiple alignment**: Three sub-cases are identified (multi-aligned words):
  - The translation equivalent of an SL word appears more than once in the TL sentence.
  - Two (or more) discrete SL words have the same translation equivalent appearing only once in the TL sentence.
  - The translation equivalent of two (or more) SL words appears more than once in the TL sentence.

The algorithm allows the one-to-one alignment between SL words and TL ones, while rejecting any multiple alignments, unless the lexicon explicitly provides such information. Those cases for which the lexicon explicitly provides such information are specially treated by creating entities with two or more words, which are called **multiple words**. This allows the constraint of one-to-one alignment between SL elements (words or multiple-words) and TL ones to be used without any exception. The multiple words make possible the use of "legal" multiple alignments but do not themselves solve the cases of actual multiple alignments (such as the three cases listed above).

As an example of single alignments, consider the following pair of sentences for Greek-English and the available translations in figure 3. Figure 4 illustrates the single alignments.

SL: "Πεθαίνουν σχεδόν 3.000 άνθρωποι."

TL: "ADVC2(Nearly) PC4(- 3.000 people) VC8(die)."

πεθαίνουν->[dead, die, pass away]
σχεδόν->[almost, nearly]
3.000->[]
άνθρωποι->[human, man, people, person]

Figure 3: Available translation from Greek-English bilingual lexicon

The cases of multiple alignments are very common and, apart from cases generated directly by the lexicon, multiple alignments may be created in other steps of the algorithm as well. Therefore, an efficient way must be found for the Phrase aligner to

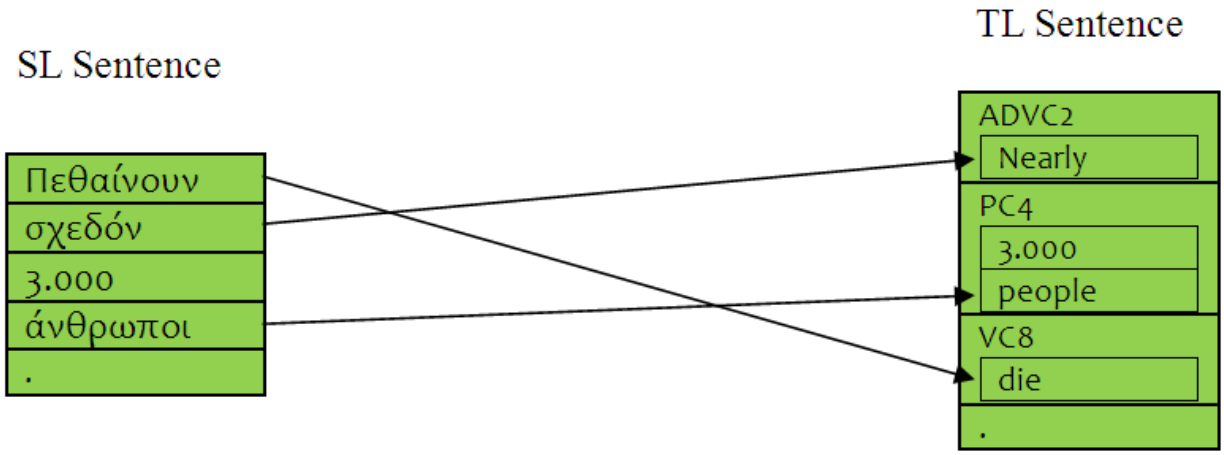


Figure 4: Example of sentence pair with only single alignments

successfully resolve them in order to achieve an accurate phrasing. To this end, PAM exploits additional information such as, for example, the information residing in the neighbourhood of the words involved in multiple alignments.

As an example of multiple alignments, consider the following pair of sentences, where the Greek SL word "Οι", is potentially aligned with the English word "the" at position 1, position 5 and position 10 in the TL sentence:

> SL: "Οι ιστορικές ρίζες της Ευρωπαϊκής Ένωσης ανάγονται στο Δεύτερο Παγκόσμιο Πόλεμο."
>
> TL: "PC2(The historical roots) PC7(of the European Union) VC12(lie) PC14(in the Second World War)."

Figure 5 illustrates the single and multiple alignments as suggested by the bilingual lexicon. The multiple alignments cannot be solved without additional information provided by the single alignments. The algorithm uses a distance-based principle which employs the distances provided by the single alignments in order to choose between the various candidate translations. The main idea besides the distance-based principle is to choose from within the multiple alignments the one with the minimum distance. Hence, since the distance between different sentences (one in SL and one in TL) cannot be calculated, the algorithm uses a technique to calculate the distances on the SL sentence. This technique locates the SL words that are single aligned with TL words that belong to the same phrase with the candidate translation. At this point there are candidate SL words but unlike the TL words, now the distance can be calculated because

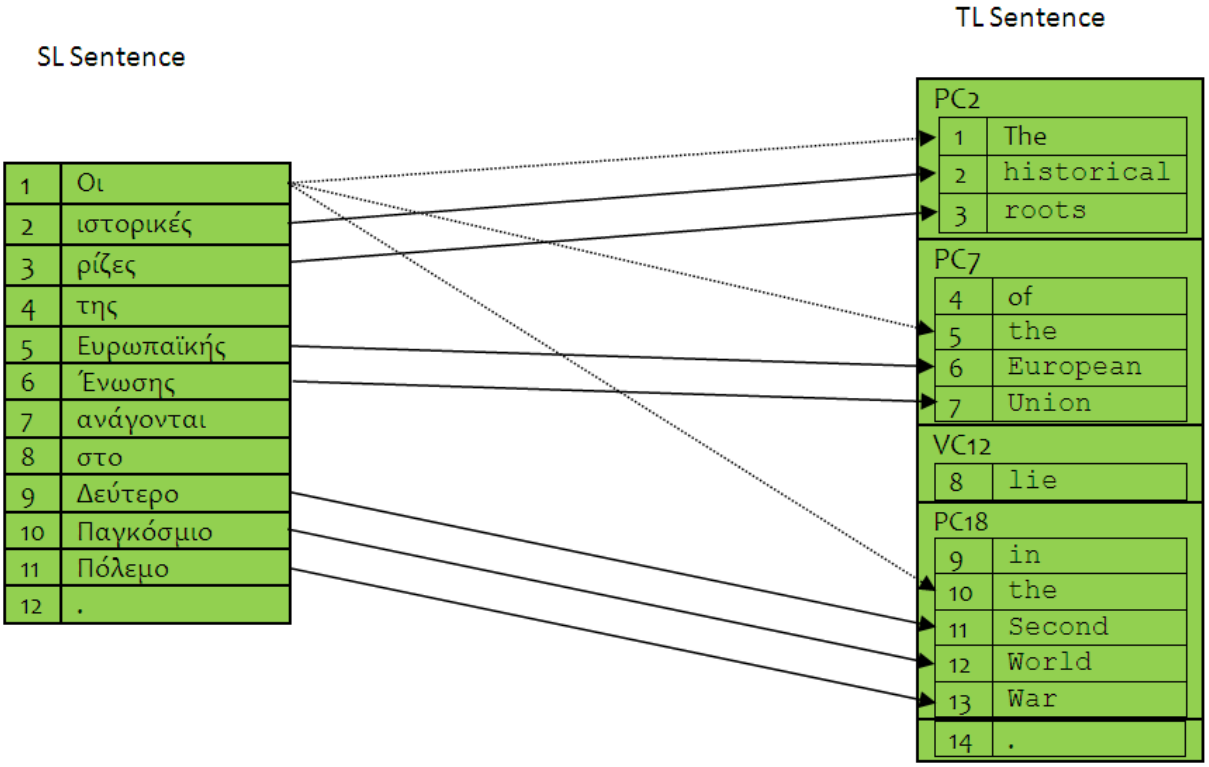


Figure 5: Example of sentence pair with single and multiple alignments

all words belong to the same sentence. Among the candidate SL words, the algorithm chooses the one with the minimum distance, and this distance is considered as the distance between the SL word and the TL word. This process is repeated for all TL words that belong to the multiple alignments. Finally the alignment with the minimum distance is selected and the rest of the alignments are rejected.

In the example above, the distance (d) between "Οι[1]" and "The[1]" is calculated by using the single alignments between "ιστορικές" $\rightarrow$ "historical" and "ρίζες" $\rightarrow$ "roots" because the TL words "historical" and "roots" belong to same phrase as the "The" at index 1.

$$d_{SL-TL}(\text{"Οι[1]"}, \text{"The[1]"}) =$$
$$= \min\{d_{SL-SL}(\text{"Οι[1]"}, \text{"ιστορικές[2]"}), d_{SL-SL}(\text{"Οι[1]"}, \text{"ρίζες[3]"})\} =$$
$$= \min\{1, 3\} = 1$$

The above procedure is repeated again for the two other TL words which are involved in the multiple alignments of token "Οι[1]".

$$d_{SL-TL}(\text{“Οι[1]"}, \text{“the[5]"}) =$$
$$= \min\{d_{SL-SL}(\text{“Οι[1]"}, \text{“Ευρωπαικής[5]"}), d_{SL-SL}(\text{“Οι[1]"}, \text{“ένωσης[6]"})\} =$$
$$= \min\{4, 5\} = 4$$

$$d_{SL-TL}(\text{“Οι[1]"}, \text{“the[10]"}) =$$
$$\min\{d_{SL-SL}(\text{“Οι[1]"}, \text{“Δεύτερο[9]"}), d_{SL-SL}(\text{“Οι[1]"}, \text{“Παγκόσμιο[10]"}),$$
$$d_{SL-SL}(\text{“Οι[1]"}, \text{“Πόλεμο[11]"})\} =$$
$$= \min\{8, 9, 10\} = 8$$

Finally the alignment is chosen with the minimum distance; in the above example this alignment is between the SL word “Οι" and the TL word “the" at index 0.

Similarity

When an SL word remains unaligned, this is usually due to gaps in the dictionary coverage (missing translation pairings). To overcome this problem, the algorithm makes use of three techniques. The first technique tries to create correspondences between unaligned SL words with its identical or similar word in the TL sentence.

Two words, for which no correspondence is established by the lexicon, are considered similar when their degree of character-wise similarity is above a given threshold. More specifically, the degree of similarity is defined as the measure of the longest common subsequent ratio. A similar approach has been suggested by Melamed (1995) and Mulloni & Pekar (2006) for identification of cognates in NLP.

In the following pair of sentences, where SL and TL share the same alphabet, the proper names “Robert Schuman", “Konrad Adenauer", “Alcide de Gasperi" and “Winston Churchill" are identical words, that can be aligned without lexical information:

> “Zwischen 1945 und 1950 machten sich eine Handvoll mutiger Staatsmänner darunter Robert Schuman, Konrad Adenauer, Alcide de Gasperi und Winston Churchill daran ihre Völker zu überzeugen in eine neue Ära einzutreten."

"Between 1945 and 1950 a handful of courageous statesmen including Robert Schuman, Konrad Adenauer, Alcide de Gasperi and Winston Churchill set about persuading their peoples to enter a new era."

For example, in the following pair of sentences the SL word "totalitarismus" is similar to the TL word "totalitarianism":

"Menschen die sich während des Krieges dem Totalitarismus widersetzt hatten waren entschlossen internationalem Hass und Rivalität in Europa ein Ende zu setzen und die Voraussetzungen für einen dauerhaften Frieden zu schaffen."

"People who had resisted totalitarianism during the war were determined to put an end to international hatred and rivalry in Europe and create the conditions for lasting peace."

Translation Similarity

The second technique tries to locate unaligned SL words which correspond to a compound TL word. More specifically the algorithm attempts to locate the translations of the unaligned SL words with exactly one compound TL word.

For example, in the following pair of sentences the SL word "κατανάλωση" is translated to "Verbrauch" and the SL word "πόρων" is translated to "Ressource" so those SL words can be located with the TL word "Ressourcenverbrauch":

"Η οικονομία της Ευρώπης βασίζεται σε μία υψηλή κατανάλωση πόρων."

"Die Wirtschaft Europas basiert auf einem hohen Ressourcenverbrauch."

Transliteration

The last technique that is used by the algorithm to overcome the low lexicon coverage transliterates the SL word (in case of difference in alphabets between SL and TL) and consequently tries to locate its identical or similar word in the TL sentence. Identical or similar words usually involve proper names, dates, non-translated abbreviations or tokens that are highly similar in the SL and TL (e.g. medical or chemical terminology).

The example in (4) illustrates a transliteration case, where the SL word "Σπούτνικ", after transliteration has been applied, can be matched to its TL counterpart "Sputnik":

"Το 1957 η Σοβιετική Ένωση νικά τις Ηνωμένες Πολιτείες στην κούρσα του διαστήματος εκτοξεύοντας τον πρώτο τεχνητό διαστημικό δορυφόρο το Σπούτνικ 1."

"The Soviet Union beats the United States in the space race by launching the first man-made space satellite Sputnik 1 in 1957."

At the end of Step 1, all possible alignments using single-word information are achieved. The remaining unaligned SL words are handled by the algorithms at the next step.

4.2 Step 2: Alignments based on tag correspondence and extended PoS tags

The input for Step 2 is based on the output of Step 1 with the aim of increasing the number of words aligned between the SL and TL sentences. At this stage, the alignment of so far unassigned SL words is based on (a) statistical-type information extracted from the bilingual lexicon and (b) additional grammatical features. The present section describes the methods, applied in their order of appearance below, which exploit such information.

Alignment based on unique Part-of-speech

This method utilises the fact that some parts of speech, such as verbs, nouns, adjectives and pronouns, have a one-to-one correspondence between language pairs (for instance, a SL verb almost always corresponds to a verb in the TL). In this vein, the algorithm maps an SL word not aligned in the previous step to a TL word, only if they share the same PoS.

Similarity of extended PoS tags (at the level of Words)

This method is based on identifying those SL words that are similar with the unaligned SL words (not included in a phrase) in terms of their similarity of grammatical features, as this is reflected in the extended PoS tag. The similarity score of extended PoS is defined based on the agreement of the extended PoS tag of the candidate SL word with the extended PoS tags of the aligned SL words. For each SL word, this similarity is

normalised with a Gaussian function taking into account the physical distance in the sentence between the aligned SL word and the candidate SL word.

So, for every word not yet assigned to a phrase, the algorithm calculates the similarity of extended tags, with all the words in the SL sentence that: (1) have been assigned to a phrase and (2) for which the number of PoS tags in the lexicon exceeds a threshold (the second criterion is introduced to prevent the decision regarding an alignment to be based on words that are very frequently used and thus have a high probability of generating wrong alignments, e.g. articles). The extended PoS similarity score is then defined by normalising the tag similarity by multiplication with a Gaussian function that models the distance of words on the sentence in terms of tokens, in order to give a lower matching to distant words. By adopting this score, closely-positioned words tend to be chosen over long-distanced ones, although the similarity score of the long-distanced words might be higher. This normalisation enables the Phrase aligner algorithm to select the clustering of words that match to an acceptable extent in terms of extended tag but are close in the sentence. The variance of the Gaussian is tuneable to the application requirements, and is thus a system parameter whose effect is studied in the experimental section. Amongst these matches, the one with the highest similarity is selected. Then it is possible to retrieve the single aligned TL word's phrase.

The next step involves extracting all unaligned TL words which belong to a TL phrase. As a final stage the algorithm makes alignments between the unaligned SL word and the TL word with the highest tag correspondence to that word. In the aforementioned example (2), let us assume that the highest similarity of "politische" is with the single-aligned German word "Ziel" and that the TL phrase to which the TL word "Ziel" belongs has exactly one unaligned English word "goal". Then the algorithm makes a new alignment between the unaligned German word "Ziel" and the previously unaligned English word "goal".

This method can be applied only to words whose Part-of-speech tag contains morphological information such as verbs, nouns, adjectives, pronouns. It should also be noted that this method is effective only in the case of morphologically rich languages (e.g. Greek). However, the same method without any modifications can still be applied in morphologically poor languages without loss of generality, though naturally the number of words that will be aligned in this stage will be unavoidably reduced.

Similarity of extended PoS tags (at the level of Phrases)

This method is a variation of the above method “Similarity of extended PoS tags" with the difference that at the present stage, the unaligned SL words are grouped to TL phrases instead of making alignments with TL words. This variation covers more cases because it does not require the existence of an unaligned TL word in the TL phrase. The drawback of this method is that the rate of wrong alignments is higher, and for this reason it is applied after the similarity of PoS tags is examined at the level of words.

SL-TL Tag Correspondence

In this method words can also be aligned based on the SL-TL tag correspondence. This correspondence is based on statistical-type information extracted from the bilingual lexicon by studying macroscopically the average frequency with which a SL word of PoS type ‘X’ is translated to a TL word of PoS type ‘Y’. Assuming that the majority (exceeding a chosen threshold) of words of type ‘X’ do translate into words of type ‘Y’, then if an SL word of type ‘X’ could be assigned to a TL word of type ‘Y’, this would probably be correct, thus improving the Phrase aligner accuracy.

If more than one TL word of type ‘Y’ exist, then the most likely candidate for assigning the SL word can be determined by applying the minimum distance principle, as described in the previous subsection.

4.3 Step 3: Alignments based on neighbours

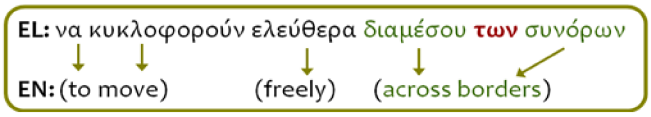
The third step of the Phrase aligner completes the alignment process, operating on the output of the second step. The aim here is to increase the number of SL words that are included to TL phrases based on the existing alignments of their neighbours. More specifically, an unaligned SL word will be grouped to a certain TL phrase, to which the majority of its immediate neighbours belong to. This approach may seem to be drastic. However, it should be noted that in the case of the Phrase aligner module, the aim is to align the largest possible number of SL words, and using this criterion increases the number of alignments via the most intuitive manner.

At this stage, the alignment of so far unassigned SL words is based on (a) alignments of neighbour, (b) alignments with unused TL phrases, and (c) alignments based on existing alignments from other sentence pairs. The present section describes the methods

which exploit such information. The sequence they are applied in is reflected by their order of presentation below.

Alignment of words flanked by words from the same phrase

This method is based on identifying those SL words for which both of its immediate neighbour words belong to the same TL phrase. This provides a strong indication that the SL word belongs to this common phrase. For example, in the following pair of sentences, the SL word "των" has both of its neighbors belonging to the same TL phrase. Hence, this word will (in the absence of other evidence) be assigned to this phrase.

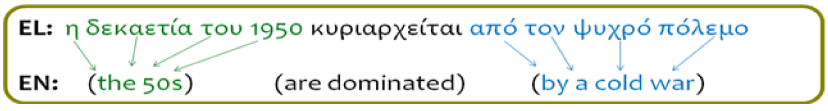


Assignment of word to neighbouring phrase

This method is based on identifying those unaligned SL words whose immediate neighbour words belong to consecutive TL phrases. In this case, the TL phrase with the higher similarity score of the extended tag of its head word is selected.

Assignment of word to an intermediate unused phrase

In this method, assignments can also be created between unaligned SL words and unused TL phrases. But it is required that the unused phrase be between the consecutive phrases that are created with the alignments of the neighbour SL words. For example, in the following pair of sentences the SL word "κυριαρχείται" and the TL phrase "(are dominated)" are both unaligned. Also the TL phrases, which are aligned with the words "1950" and "από" actually bracket the unaligned TL phrase:



Alignment with tag model pattern

This method utilises alignment patterns extracted from other sentence pairs (if they are available). More specifically the sequence of tags contained in the unaligned word

and its neighbours are identified in other sentence pairs, which contain already aligned words for the specified tags. The identified patterns are grouped together and then the tag model selects the type of alignment with the most appearances. Hence, this type of alignment suggests the appropriate TL phrase through the alignment of its neighbours.

Reordering of parsed sentence

The segmentation of SL words into phrases might cause the parsed SL sentence to contain its words in a different order in comparison to the unparsed sentence. The different order of SL words is unacceptable since the system uses this method for reordering the SL words. In the following example the existing alignments cause the change of the initial order of the SL words:

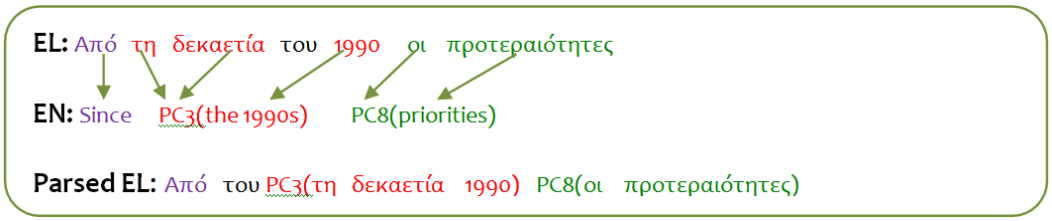


Figure 6: Example with alignments that cause disordering of SL words

In this example, the order of the parsed SL sentence has been changed. More specifically the Greek word "του" has a different position in the parsed sentence. As a result of applying this method, the reordering of is reverted so that "του" is placed in the appropriate position. The following example illustrates the appropriate reordering.

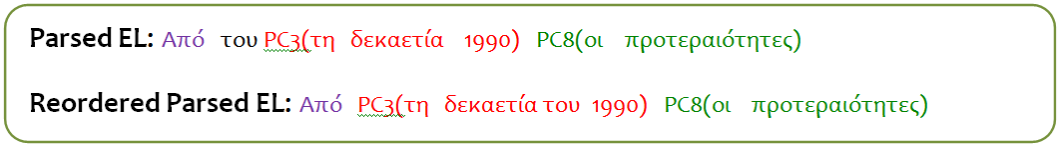


Figure 7: Example of reordering for SL words

Post-Processing – Rejection of Sentence Pairs

The main information that is worth extracting from the previous steps is the fraction of alignments performed in the first step over the number of SL tokens. The alignments that take place in the first step are the most trustworthy because they are based on the

information of the bilingual lexicon. So, if the fraction is below a given threshold, it may be deduced that the SL and TL sides of the bilingual corpus for the given sentence are not consistent with each other or that a free translation is used, in which case the specific sentence could be marked as being of “limited dependability" and be removed from the training input for the Phrasing model generator. The reasoning behind such a decision is that if a sentence pair has too few dependable alignments, the structural transformations it represents and which may be used in the translation process have a high likelihood of being erroneous.

5 Phrase aligner module: Experimental setup & results

5.1 Datasets

The Phrase aligner module has been tested on eight different language pairs as described below. For each language pair a bilingual corpus has been compiled, extracted manually from the web. More specifically the corpora used have the following characteristics:

- German → English corpus: Extracted from a multilingual website[4], it comprises 164 sentences. The SL side (German) of the corpus has been tagged and lemmatised by the TreeTagger[5] and the RFTagger[6], while the TL side (English) has been processed with the TreeTagger, yielding tag, lemma and phrase annotations.

- English → German corpus: It is the same corpus as the previous one, but used in the opposite direction. The SL side (now English) of the corpus has been tagged and lemmatised by the TreeTagger, while the TL side (German) has been processed with the TreeTagger and the RFTagger, yielding tag, lemma and phrase annotations.

- Greek → English corpus: Extracted from a multilingual website[7], it comprises 200 sentences. The SL side (Greek) of the corpus has been tagged and lemmatised by the ILSP FBT Tagger & Lemmatiser (Papageorgiou et al., 2000), while the TL side (English) has been processed via the TreeTagger, yielding tag, lemma and phrase annotations.

- Greek → German corpus: Also extracted from a multilingual website[8], it comprises 200 sentences. The SL side (Greek) of the corpus has been tagged and lemmatised by the ILSP FBT Tagger & Lemmatiser, while the TL side (German) has been processed with the TreeTagger and the RFTagger, yielding tag, lemma and phrase annotations.

- Norwegian → English: Extracted from a multilingual website[9], it comprises 200 sentences. The SL side (Norwegian) of the corpus has been tagged and lemmatised

[4] http://europa.eu/abc/12lessons/index_en.htm
[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
[6] http://www.ims.uni-stuttgart.de/projekte/corplex/RFTagger/
[7] http://europa.eu/abc/history/index_en.htm
[8] http://www.eea.europa.eu/
[9] http://europa.eu/abc/history/index_en.htm

with the Oslo-Bergen tagger[10], while the TL side (English) has been processed with the TreeTagger, yielding tag, lemma and phrase annotations.

- Norwegian → German: Extracted from the same multilingual website as the previous one, it comprises 197 sentences. The SL side (Norwegian) of the corpus has been tagged and lemmatised by the Oslo-Bergen tagger, while the TL side (German) has been processed with the TreeTagger and the RFTagger, yielding tag, lemma and phrase annotations.

- Czech → English: It is a selection of diverse sentences from the "Parallel Corpus of English and Czech Texts"[11], which contains classic English books and their Czech translations. The selection consists of 200 sentences in both languages. The SL side was annotated, using the Brno tagset, with the Desamb tagger (Šmerk 2004), providing tag and lemma information. The TL side was processed by TreeTagger, yielding tag, lemma and phrase annotations.

- Czech → German: It is a selection of diverse sentences from the "Czech-German Parallel Corpus"[12], which contains mostly newspaper articles in Czech or German and translations to the other languages. The selection consists of 202 sentences in both languages. The Czech part was annotated, using Brno tagset, with Desamb tagger, providing tag and lemma information, while the TL side was processed with TreeTagger and RFTagger, yielding tag, lemma and phrase annotations.

- Czech → Italian: The collected data for all bilingual corpora have been manually modified so that the SL side of the corpus is as “close" as possible to the TL side. The main modifications are aimed at removing metaphors or elliptical constructions and smoothing out divergences between the two languages, to remove free translations. Moreover, the corpus NLP annotations have been manually corrected, so as to be able to focus on testing the Phrase aligner module’s performance on data devoid of errors.

[10] http://tekstlab.uio.no/obt-ny/english/index.html

[11] http://www.phil.muni.cz/angl/kacenka/kachna.html

[12] http://www.ped.muni.cz/katedry-a-instituty/nemecky-jazyk-literatura/aktivity/cesko-nemecky-paralelni-korpus/

5.2 Experimental results

In order to evaluate the performance of the Phrase aligner module, the results of the module were compared with the ground truth, which was created manually by segmenting the SL side sequence of words into the corresponding phrases, as a reference set. The completed gold-standard sets cover the language pairs (i) Czech-to-German, (ii) Czech-to-English, (iii) German-to-English, (v) Greek-to-English, and (vi) Norwegian-to-German. The rest of the language pairs were not delivered (at least for all 200 sentences in the relevant corpus) and therefore the PAM output for those language pairs has not been compared.

For comparison purposes at this point the evaluation results for earlier versions of PAM are presented. Table 1 illustrates the evaluation results as reported in Deliverable D3.3.1 (dated December 2010). Due to changes in the evaluation process these results are not fully comparable (for instance, the alignment of punctuation marks is no longer included in the calculation of accuracy, resulting in a stricter test).

LANGUAGE PAIR	PAM Accuracy
Greek-English	89.99%
German-English	83.33%

Table 1: Evaluation results (Version 2 of PAM, as reported in D3.2.1)

LANGUAGE PAIR	PAM Accuracy	
	Sentences 1-50	Sentences 51-100
Greek-English	94.00%	90.60%
German-English	84.94%	88.76%

Table 2: Evaluation results as they reported in Deliverable D3.3.2 (December 2011)

For the language pairs with available gold-standard sets, different configurations have been examined by using different values for system parameters. The configurations reported here vary only in terms of certain parameters to which the system is more sensitive, as listed below:

- the number of lexicon entries that distinguish between low-frequency and high-frequency SL tags

- the distance threshold for a single alignment to be made,
- the threshold for the minimum extended tag similarity of words normalised by the word distance,
- the threshold for the similarity of 2 words based on the SL-to-TL correspondences of the lexicon, in the case when the SL tag frequency is below the value of a specified frequency threshold (as set in the relevant configuration file)
- the threshold for the similarity of 2 words based on the SL-to-TL correspondences of the lexicon, in the case when the SL tag frequency is higher than a specified frequency threshold (this case complements case (iv) above)

For the purpose of integrating into the translation system, the optimum configuration for each language pair is sought for the values of the above parameters, see Table 3.

Configuration Parameter	Testing Values
Distinction between Low- and High-frequency SL tag entries	10,20,50,100,200,500
Distance threshold	2.0, 2.5, 3.0, 3.5,4.0
Normalised extended tag similarity	0.10, 0.20, 0.30, 0.40, 0.50
Minimum required percentage of low-entries tag pair	0.05, 0.10, 0.20, 0.30, 0.40, 0.50
Minimum required percentage of high-entries tag pair	0.05, 0.10, 0.20, 0.30, 0.40, 0.50

Table 3: Configuration parameters and testing values

The Table 4 contains the accuracy of word alignment for each language pair for various configuration parameters.[13] The setting denoted "config 0" is the default config-

[13]The results reported concern version 4. The earlier version of PAM (version 3) included most of the techniques described here, but it was designed in such a way so it could not take advantage of the object-oriented approach, Moreover, it didn't use the core structures of PRESEMT and could not support multiple words. As a result, this version could not be easily integrated into the main system platform. Consequently, given also that debugging was very difficult, it was necessary to design a new version from scratch. The new version 4 supports all these features and in addition makes a more efficient use of memory. Last but not least the evaluation results of the current version are slightly improved.

uration and it is used by the language pairs for which no golden set is available. It is generated by using the average parameters' value from the language pairs with the golden set. The rest configurations are the optimum ones for each one of the language pair. Those values are extracted by testing a wide range of parameters' value.

As a general comment, it can be stated that the various language pairs differ considerably in terms of the achieved accuracy. According to this table the most accurate alignment results are obtained for the Greek-to-English language pair, while the less accurate results are obtained for the pair Czech to English.

LANGUAGE PAIR	PHRASE ALIGNER ACCURACY					
	Config 0	Config 1	Config 2	Config 3	Config 4	Config 5
Czech-German	72.83%	**75.40%**	73.15%	72.54	73.23%	73.19%
Czech-English	71.58%	73.21%	**73.60%**	73.00%	71.44%	72.25%
German-English(1-100)	89.15%	88.67%	89.81%	**89.85%**	88.15%	88.34%
Greek-English	96.63%	95.89%	96,13%	96.13%	**97.52%**	95.98%
Norwegian-German	77.16%	79.64%	79.52%	79.52%	77.75%	**80.35%**

Table 4: Evaluation results (Version 4 of PAM – Oct 2012)

	OPTIMAL CONFIGURATION PARAMETERS				
Config Name	Distinction Low-High SL tag entries	Distance threshold	Normalised extended tag similarity	Minimum required percentage of low-entries tag pair	Minimum required percentage of high-entries tag pair
Config 0	100	3.5	0.15	0.20	0.10
Config 1	200	4.0	0.30	0.20	0.05
Config 2	100	4.0	0.10	0.15	0.10
Config 3	100	4.0	0.10	0.15	0.15
Config 4	200	3.0	0.20	0.10	0.05
Config 5	10	3.0	0.20	0.30	0.10

Table 5: Configuration Parameters

5.3 Phrase Aligner evaluation in terms of each step

For the language pairs with available gold-standard sets, the three steps of the Phrase aligner were adapted for evaluation. Due to the fact that each language pair has its

own configuration parameters we examine independently each language pair. For each language pair, the optimum configuration is selected.

Step 1 achieves results with accuracy ranging between 87% and 99%. The rather large range depends on the coverage provided by the lexicon used. This is expected, since Step 1 creates alignments based on the bilingual lexicon. Word correspondences extracted via the translations contained in bilingual lexica are considered the most trustworthy resources. Step 2 achieves results with accuracy ranging between 59% and 99%. Those differences are not attributable to a single cause, and warrant further study. Finally in the third step the accuracy ranges between 45% and 93%. As a reminder, Step 3 creates alignments that based on the neighbours of the unaligned words, so it is expected that it propagates the errors from the previous steps. This explains the very large variation of accuracy over the various language pairs.

So the question which is still under investigation is the degree to which the reduced alignment accuracy of the Phrase aligner as a whole is due to the nature of the language pair or whether it can be attributed to properties of the linguistic resources used (e.g. low coverage of the bilingual lexicon or a bilingual corpus with longer sentences or freer translations).

LANGUAGE PAIR	OPTIMUM CONFIGURATION	STEP1 ACCURACY	STEP2 ACCURACY	STEP3 ACCURACY
Czech-German	"Config 1"	91.21%	69.64%	49.86%
Czech-English	"Config 2"	86.92%	59.52%	44.91%
German-English	"Config 3"	95.38%	88.24%	63.71%
Greek-English	"Config 4"	99.08%	99.28%	93.06%
Norwegian-German	"Config 5"	93.29%	73.44%	72.65%

Table 6: Step accuracy

5.4 Phrase Aligner alignment accuracy per step

As a further insight into the operation of the PAM module, Table 7 comparatively presents the percentage of SL words which are grouped into phrases (regardless of whether they are correctly aligned or misaligned).

LANGUAGE PAIR	CONFIGURATION	STEP1	STEP2	STEP3	NO-ALIGNMENTS
Czech-German	"Config 1"	60.43%	15.78%	14.21%	9.58%
Czech-English	"Config 2"	71.55%	7.41%	9.39%	6.20%
German-English	"Config 3"	76.83%	10.43%	5.85%	3.73%
Greek-English	"Config 4"	78.42%	12.34%	5.16%	1.28%
Norwegian-German	"Config 5"	63.37%	19.57%	9.22%	7.84%

Table 7: Alignment percentage (Version 4 of PAM)

5.5 Factors that affect accuracy

As mentioned before, the ranges here are smaller, in particular for steps 2 and 3. The accuracy of PAM alignment varies drastically between the language pairs so additional investigation of the PAM results is warranted. The ranges here are smaller, in particular for steps 2 and 3. A high percentage of failed alignments can be correlated to a low alignment in Step 1. The range in step 1 is equal to 18 percentage points (from 78.4 to 60.4%), while the range of alignment percentages for step 2 is 12 points and for step 3 it is 9 percentage points. The question that is still under investigation is the degree to which the reduced accuracy in alignment performance is due to the nature of the language pair or whether it can be attributed to properties of the linguistic resources used. The current section provides a preliminary study of the PAM performance against the following factors: a) the length of tokens in a given sentence, b) the number of phrases contained in the TL sentence of a given sentence pair, c) the coverage of bilingual lexicon. This section is followed by a detailed analysis for each one of the above factors.

a) Length in tokens of the given SL sentence

It is expected that the length of a given sentence in terms of tokens affects the PAM alignment' accuracy. However the following experiments do not confirm this expectation. In the following figures the average accuracy is shown over the length of tokens in the SL sentence. The sentence lengths in tokens have been organized in groups so the average accuracy for each one of the groups is shown.

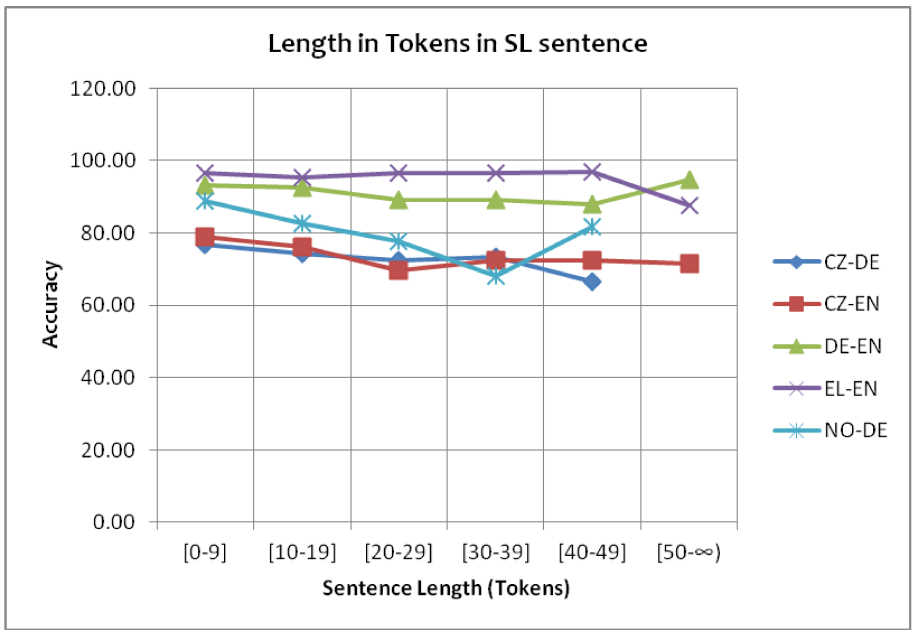


Figure 8: Alignment Accuracy per sentence's length

b) Number of Phrases in a given TL sentence

It is expected that the number of phrases in a given sentence affects the PAM alignment' accuracy. However the experiments performed do not confirm this intuitive expectation. Figure 9 shows that there are various trends for the different language pairs and as a result no global trend can be extracted. Only for a few language pairs is a downward trend encountered when larger sentences are translated (namely for NO→DE and to a lesser extent CZ→DE and EL→EN).

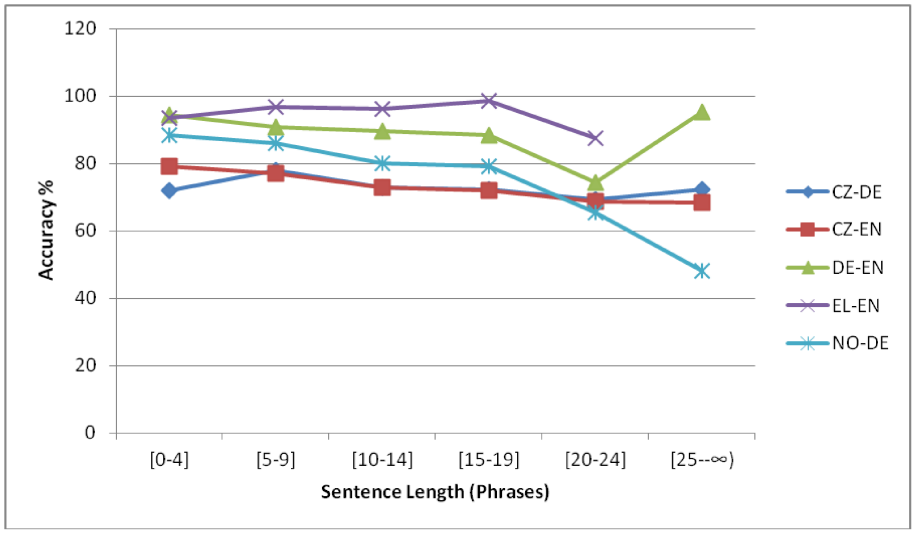


Figure 9: Alignment Accuracy over the number of phrases per sentence

c) Coverage of Bilingual Lexicon

It is expected that the lexicon's coverage affects the alignment accuracy of PAM. The following experiments confirm this expectation. The lexicon's coverage is the number of SL words with equivalent translation in TL sentence to the total SL words.

Figures 8-12 illustrates the correlation between the lexicon's coverage and the segmentation accuracy for the five language pairs fro which the available golden set comprises 200 sentences.

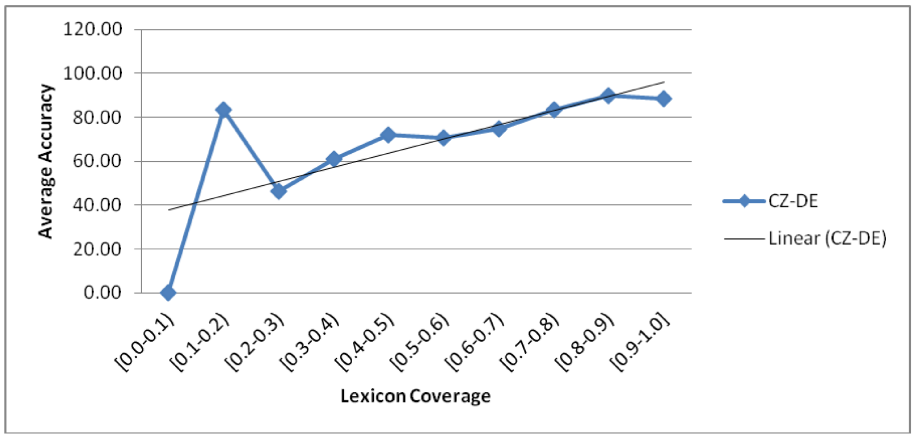


Figure 10: Correlation between Lexicon's Coverage and Accuracy for CZ-DE

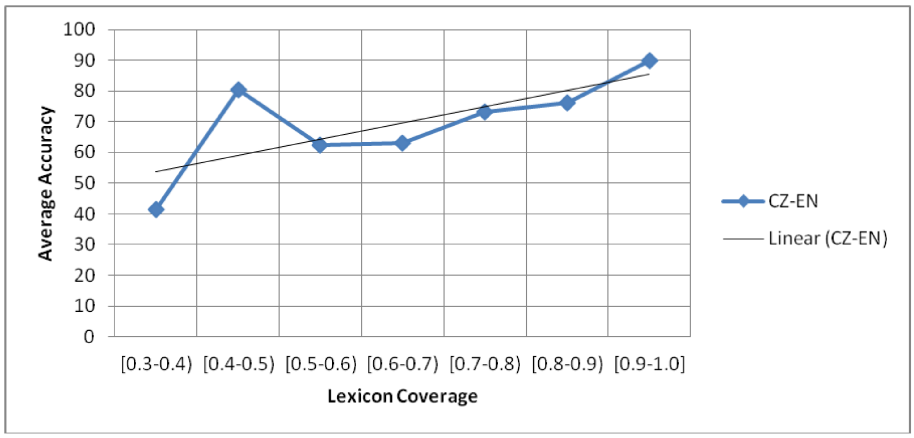


Figure 11: Correlation between Lexicon's Coverage and Accuracy for CZ-EN

From Figures 10 to 14, it can be seen that there is a correlation between the lexicon coverage and the accuracy of the alignment obtained. In general, there appears to be a trend of diminishing PAM accuracy as the lexicon coverage is reduced, this being more marked as the coverage falls below 70%. However, in some language pairs, this

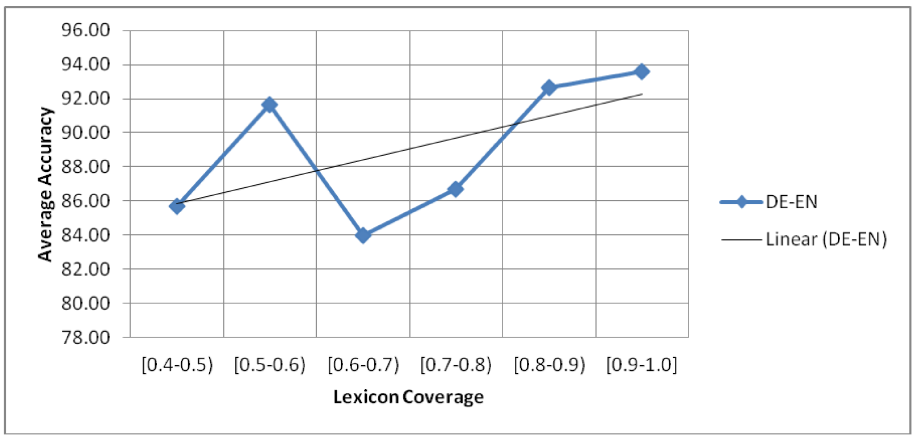


Figure 12: Correlation between Lexicon's Coverage and Accuracy for DE-EN

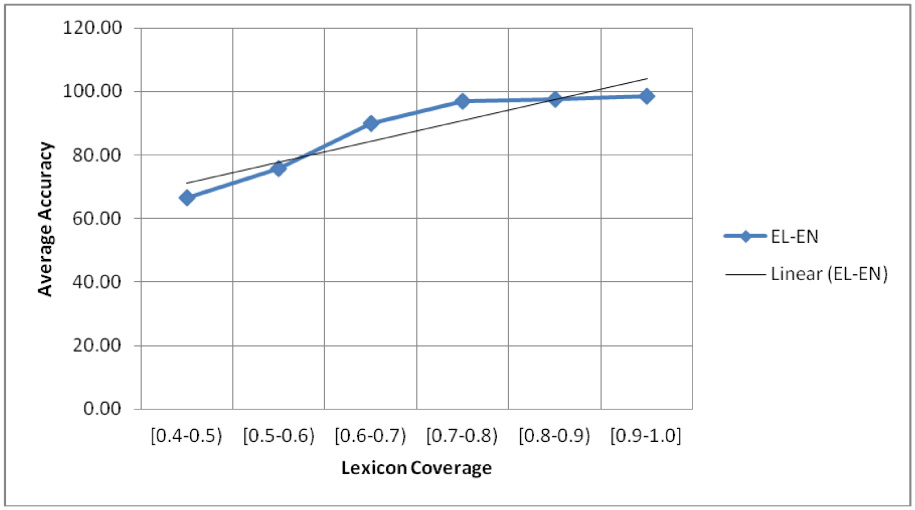


Figure 13: Correlation between Lexicon's Coverage and Accuracy for EL-EN

coverage is reduced monotonically (e.g. EL→EN or NO→DE) while for other pairs there exist oscillations.

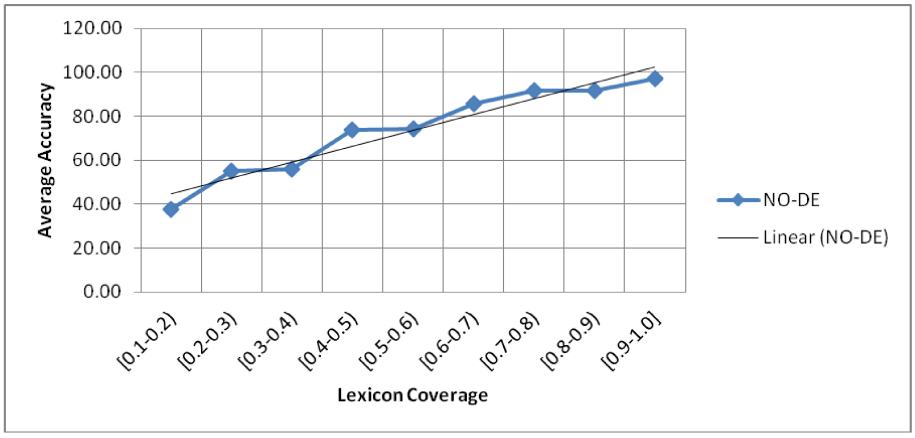


Figure 14: Correlation between Lexicon's Coverage and Accuracy for NO-DE

6 References

[1] Melamed, I. D. (1995). Automatic Evaluation and Uniform Filter Cascades for Inducing N-best Translation Lexicons, Third Workshop on Very Large Corpora (WVLC3), Boston, MA.

[2] Mulloni, A. and Pekar, V. (2006). Automatic Detection of Orthographic Cues for Cognate Recognition. Proceedings of LREC 2006, Genoa, Italy.

[3] Tambouratzis, G., Simistira, F., Sofianopoulos, S., N. Tsimboukakis & Vassiliou, M. (2011). A resource-light phrase scheme for language-portable MT, Proceedings of the 15th International Conference of the European Association for Machine Translation, (eds. M. L. Forcada, H. Depraetere & V. Vandeghinste) 30-31 May 2011, Leuven, Belgium, pp. 185-192.

[4] G. Tambouratzis, M. Troullinos, S. Sofianopoulos, M. Vassiliou: Accurate phrase alignment in a bilingual corpus for EBMT systems. Proceedings of the 5th BUCC Workshop, held within the LREC-2012 Conference, May 26, Istanbul, Turkey, pp. 104-111.

7 PAM User Manual

7.1 Introduction

This is a brief guide on how to use the Phrase aligner module (PAM). PAM processes the bilingual corpora by performing text alignment at word and phrase level within a language pair. It operates in offline manner, processing the set of parallel sentences so as to determine how phrases are transformed from SL to TL.

a. How to run PAM

The software implementation of PAM uses as input a number of arguments denoting the sentence(s) and the language pair to be processed. If the required resources for the given sentences or the specific language pair are not available, the program terminates. The command line arguments for invoking PAM have the following form:

`phraseAligner [-lang <srcLang>-<tgtLang>] [-sent <minSent>-<maxSent>]`

where:

- `<srcLang>` is the source language denoted by the first two characters of the language
- `<tgtLang>` is the target language denoted by the first two characters of the language
- `<minSent>` is the minimum id of a set of consecutive sentence pairs to be retrieved
- `<maxSent>` is the maximum id of a set of consecutive of sentence pairs to be retrieved

Example: `phraseAligner -lang DE-EN -sent 5-5`

This is used for retrieving the sentence pair with id: 5 for the language pair German → English.

Example: `phraseAligner -lang EL-DE -sent 3-10`

This is used for retrieving the eight consecutive sentence pairs with id: 3-10 for the language pair Greek → German

b. How to handle a language pair in PAM

PAM requires the following resources: (a) a bilingual corpus and (b) a bilingual lexicon, and (c) special attributes of the languages, (d) language pair parameters and (e) the transliteration rules. The language pairs' parameters require default values to be chosen but the parameters for language pairs are optional. Also the transliteration rules are optional resources. Each one of those resources is described below.

The Bilingual Corpus: To be used by PAM the corpus must be processed as follows:

1. Tag and lemmatise the SL-side of the corpus
2. Tag, lemmatise and chunk the TL-side of the corpus
3. Convert the tagged-lemmatised SL corpus to .xml format using the class AnyFormatToXML.java (Path: `/src/ilsp/linguisticTools`) from the PRESEMT project
4. Convert the tagged-lemmatised-parsed TL corpus to .xml using the class TreeTagger.java (Path: `/src/ilsp/chunker`) from the PRESEMT project

The bilingual corpus that is created with the above process must be stored under the path `/data/Corpora/<SL_LANG>-<TL_LANG>/` in two individual XML files, one corresponding to SL and the other to TL. Each of these files is named after the two characters that are used by the system to determine the corresponding language.

Example: The bilingual corpus for the language pair Norwegian-English is stored so that the NO-side is stored as `/data/Corpora/NO-EN/NO.xml` and the EN-side is stored as `/data/Corpora/NO-EN/EN.xml`

An extract of the `/data/Corpora/NO-EN/NO.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<text>
 <sent id="1">
  <clause id="1" type="">
   <word id="2" head="n" fhead="n" token="Den" tag="det_dem_mask_ent" lemma="den"/>
   <word id="3" head="n" fhead="n" token="Europeiske" tag="subst_prop" lemma="Europeiske"/>
   <word id="4" head="n" fhead="n" token="Union" tag="subst_prop" lemma="Union"/>
   <word id="5" head="n" fhead="n" token="opprettes" tag="verb_pres_inf_pass" lemma="opprette"/>
   <word id="6" head="n" fhead="n" token="med" tag="prep" lemma="med"/>
   <word id="7" head="n" fhead="n" token="det" tag="pron_nøyt_ent_pers_3" lemma="det"/>
   <word id="8" head="n" fhead="n" token="formål" tag="subst_appell_nøyt_ub_ent" lemma="formål"/>
   <word id="9" head="n" fhead="n" token="å" tag="inf-merke" lemma="å"/>
   <word id="10" head="n" fhead="n" token="gjøre" tag="verb_inf" lemma="gjøre"/>
   <word id="11" head="n" fhead="n" token="slutt" tag="subst_appell_mask_ub_ent" lemma="slutt"/>
```

```
   <word id="12" head="n" fhead="n" token="på" tag="prep" lemma="på"/>
   <word id="13" head="n" fhead="n" token="de" tag="det_dem_fl" lemma="de"/>
   <word id="14" head="n" fhead="n" token="mange" tag="adj_fl_pos" lemma="mange"/>
   <word id="15" head="n" fhead="n" token="blodige" tag="adj_fl_pos" lemma="blodig"/>
   <word id="16" head="n" fhead="n" token="nabokrigene" tag="subst_appell_mask_be_fl_samset"
     lemma="nabokrig"/>
   <word id="17" head="n" fhead="n" token="som" tag="sbu" lemma="som"/>
   <word id="18" head="n" fhead="n" token="kulminerte" tag="verb_pret" lemma="kulminere"/>
   <word id="19" head="n" fhead="n" token="i" tag="prep" lemma="i"/>
   <word id="20" head="n" fhead="n" token="andre" tag="adj_be_ent_pos_&lt;ordenstall&gt;"
     lemma="andre"/>
   <word id="21" head="n" fhead="n" token="verdenskrig" tag="subst_appell_mask_ub_ent"
     lemma="verdenskrig"/>
   <word id="22" head="n" fhead="n" token="." tag="&lt;punkt&gt;" lemma="$."/>
  </clause>
 </sent>
 <sent id="2">
  <clause id="1" type="">
   <word id="2" head="n" fhead="n" token="Fra" tag="prep" lemma="fra"/>
   <word id="3" head="n" fhead="n" token="og" tag="konj" lemma="og"/>
   <word id="4" head="n" fhead="n" token="med" tag="prep" lemma="med"/>
   <word id="5" head="n" fhead="n" token="1950" tag="det_fl_kvant" lemma="1950"/>
   <word id="6" head="n" fhead="n" token="begynner" tag="verb_pres" lemma="begynne"/>
   <word id="7" head="n" fhead="n" token="de" tag="det_dem_fl" lemma="de"/>
   <word id="8" head="n" fhead="n" token="europeiske" tag="adj_fl_pos" lemma="europeisk"/>
   <word id="9" head="n" fhead="n" token="landene" tag="subst_appell_nøyt_be_fl" lemma="land"/>
   <word id="10" head="n" fhead="n" token="å" tag="inf-merke" lemma="å"/>
...
```

An extract of the `/data/Corpora/NO-EN/EN.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<text>
 <sent id="1">
  <clause id="1" type="">
   <phrase id="2" type="PC">
    <word id="3" head="n" fhead="y" token="-" tag="-" lemma="-"/>
    <word id="4" head="n" fhead="n" token="The" tag="DT" lemma="the"/>
    <word id="5" head="n" fhead="n" token="European" tag="NP" lemma="European"/>
    <word id="6" head="y" fhead="n" token="Union" tag="NP" lemma="Union"/>
   </phrase>
   <phrase id="7" type="VC">
     <word id="8" head="n" fhead="y" token="is" tag="VBZ" lemma="be"/>
     <word id="9" head="y" fhead="n" token="set" tag="VVN" lemma="set"/>
   </phrase>
   <phrase id="10" type="PRT">
     <word id="11" head="y" fhead="n" token="up" tag="RP" lemma="up"/>
   </phrase>
   <phrase id="12" type="PC">
    <word id="13" head="n" fhead="y" token="with" tag="IN" lemma="with"/>
    <word id="14" head="n" fhead="n" token="the" tag="DT" lemma="the"/>
    <word id="15" head="y" fhead="n" token="aim" tag="NN" lemma="aim"/>
   </phrase>
   <phrase id="16" type="PC">
```

```
    <word id="17" head="n" fhead="y" token="of" tag="IN" lemma="of"/>
   </phrase>
   <phrase id="18" type="VC">
    <word id="19" head="y" fhead="n" token="ending" tag="VVG" lemma="end"/>
   </phrase>
   <phrase id="20" type="PC">
    <word id="21" head="n" fhead="y" token="-" tag="-" lemma="-"/>
    <word id="22" head="n" fhead="n" token="the" tag="DT" lemma="the"/>
    <word id="23" head="n" fhead="n" token="frequent" tag="JJ" lemma="frequent"/>
    <word id="24" head="n" fhead="n" token="and" tag="CC" lemma="and"/>
    <word id="25" head="n" fhead="n" token="bloody" tag="JJ" lemma="bloody"/>
    <word id="26" head="y" fhead="n" token="wars" tag="NNS" lemma="war"/>
   </phrase>
   <phrase id="27" type="PC">
    <word id="28" head="n" fhead="y" token="between" tag="IN" lemma="between"/>
    <word id="29" head="y" fhead="n" token="neighbours" tag="NNS" lemma="neighbour"/>
   </phrase>
    <word id="30" head="n" fhead="n" token="," tag="," lemma=","/>
   <phrase id="31" type="PC">
    <word id="32" head="n" fhead="y" token="-" tag="-" lemma="-"/>
    <word id="33" head="y" fhead="n" token="which" tag="WDT" lemma="which"/>
   </phrase>
   <phrase id="34" type="VC">
    <word id="35" head="y" fhead="y" token="culminated" tag="VVD" lemma="culminate"/>
   </phrase>
   <phrase id="36" type="PC">
    <word id="37" head="n" fhead="y" token="in" tag="IN" lemma="in"/>
    <word id="38" head="n" fhead="n" token="the" tag="DT" lemma="the"/>
    <word id="39" head="n" fhead="n" token="Second" tag="NP" lemma="Second"/>
    <word id="40" head="n" fhead="n" token="World" tag="NP" lemma="World"/>
    <word id="41" head="y" fhead="n" token="War" tag="NP" lemma="War"/>
   </phrase>
    <word id="42" head="n" fhead="n" token="." tag="SENT" lemma="."/>
  </clause>
 </sent>
 <sent id="2">
  <clause id="1" type="">
   <phrase id="2" type="ADVC">
    <word id="3" head="y" fhead="n" token="As" tag="RB" lemma="as"/>
   </phrase>
   <phrase id="4" type="PC">
    <word id="5" head="n" fhead="y" token="of" tag="IN" lemma="of"/>
    <word id="6" head="y" fhead="n" token="1950" tag="CD" lemma="card"/>
   </phrase>
    <word id="7" head="n" fhead="n" token="," tag="," lemma=","/>
   <phrase id="8" type="PC">
    <word id="9" head="n" fhead="y" token="-" tag="-" lemma="-"/>
    <word id="10" head="n" fhead="n" token="the" tag="DT" lemma="the"/>
    <word id="11" head="n" fhead="n" token="European" tag="NP" lemma="European"/>
    <word id="12" head="n" fhead="n" token="Coal" tag="NP" lemma="Coal"/>
    <word id="13" head="n" fhead="n" token="and" tag="CC" lemma="and"/>
    <word id="14" head="n" fhead="n" token="Steel" tag="NP" lemma="Steel"/>
    <word id="15" head="y" fhead="n" token="Community" tag="NP" lemma="Community"/>
...
```

The bilingual Lexicon: It should be processed via the following procedure:

Convert the lexicon to .xml format using the class LexVertToXML.java (Path: `/src/ilsp/linguisticTools`)

The bilingual lexicon corpus that is created with the above process must be stored under the path `/data/Lexica/` in an XML file. The XML file should be named as `lex_<SL_LANG>-<TL_LANG>`. It should be noted at this point that a single lexicon can be used by two different language pairs with reversed languages. So if the system cannot find the corresponding lexicon under the default path it will try to find it under the path of the reversed languages.

Example: The bilingual lexicon for the language pair German-English is stored under the path `/data/Lexica/lex_DE-EN.xml`

Example: The bilingual lexicon for the language pair English-German does not exist so the system will attempt to access the lexical information from the file named `/data/Lexica/lex_DE-EN.xml`

The phrase aligner module creates automatically a specific file (see an indicative example in Figure 15) under the path `data/PhraseAligner/TagCorrespondence/` that contains PoS tag correspondences within a given language pair, drawing on statistical/probabilistic information extracted from the bilingual lexicon.

```
S-S|aj|aj|1 13451
S-S|aj|ADJ VVP|1 13451
S-S|aj|n.name|1 13451
S-S|aj|PROADV|1 13451
S-S|vbmn|VV|11287 11334
S-S|vbmn|NN|19 11334
S-S|vbmn|VA|11 11334
S-S|vbmn|ADJ|10 11334
S-S|vbmn|VVP|3 11334
S-S|vbmn|VM|2 11334
S-S|vbmn|PRF|2 11334
S-M|vbmn|ADJ NN VV|88 7651
S-M|vbmn|ADJ NN PRF VV|9 7651
S-M|vbmn|NN VV|559 7651
S-M|vbmn|ADJ VV|816 7651
S-M|vbmn|ADJ PRF VV|105 7651
S-M|vbmn|APPR NN PRF VA|2 7651
S-M|vbmn|PRF VV|895 7651
S-M|vbmn|APPR NN VV|460 7651
```

Figure 15: A sample of the tag correspondence file for the EL-DE language pair

ELEMENT	DESCRIPTION
lang	The language code; this must be unique for each language.
tagSeparator	The regular expression denoting the form of the tag
verb*	The PoS tag for verbs
noun*	The PoS tag for nouns
adjective*	The PoS tag for adjectives
adverb*	The PoS tag for adverbs
tagWildCard	(Optional) The wild card character that is used in the tags
unsplittedTag*	(Optional) The PoS tag that must not be splitted with element "tagSeparator"
skipTagSimilarity*	(Optional) The parts of tag that must be omitted by the extended tag similarity
exclusiveConsecutiveMWs	(Optional) Determines whether the Multiple-Word units have to be consecutive

* (More than one element can be used)

Table 8: Elements in the language attributes file

Based on the example on page 30, entries with tag `vbmn` correspond to `VV` (11287 occurrences), `NN` (19 occurrences), `VA` (11 occurrences), `ADJ` (10 occurrences), `VVP` (3 occurrences), "VM" (2 occurrences), `PRF` (2 occurrences), out of 11334 appearances in total.

Language attributes file: A file needs to be created for each language of the language pair (e.g. DE.xml and EN.xml) under the folder `/data/PhraseAligner/Attributes`, which will contain information about the resources per language (e.g. alphabet, tagset etc.). The essential elements of this file are listed in Table 8. Figure 16 illustrates the contents of such a file for the English language (the order of appearance of the elements being free).

The most common types of regular expressions that are used by element `tagSeparator` are listed below together with a brief explanation:

`(?<=\G.{2})` This regular expression splits the tags into sub-tags, each of a length of two characters.

Example: `ATDFMA` → `AT, DF, MA`

. This regular expression splits the tags at the point of occurrence of the given character (in the given example, the full-stop character).

Example: `APPR.APPR.Auf` → `APPR, APPR, Auf`

`(?<!^)(?=[A-Z])` This regular expression splits tags into sub-tags that start with a capital letter.

Example: `PnnReuMaq` → `Pnn, Reu, Maq`

Configuration file: A configuration file (config.xml) needs to be created under the folder `/data/PhraseAligner`, for stating the special characteristics for all language pair. The

```
<?xml version="1.0" encoding="UTF-8"?>
<Attributes>
 <Language lang="EN">
  <!-- Specific PoS -->
  <verb>V</verb>
  <noun>N</noun>
  <adjective>ADJ</adjective>
  <adverb>ADV</adverb>

  <!-- Tag separator expression -->
  <tagSeparator>.</tagSeparator>

  <!-- (Optional) Wild card character of tags -->
  <tagWildCard>*</tagWildCard>

  <!-- (Optional) Exclusive consecutive MultiWords -->
  <exclusiveConsecutiveMWs>false</exclusiveConsecutiveMWs>
 </Language>
</Attributes>
```

Figure 16: The `DE.xml` file for the German language

essential elements of this file are listed in Table 9, while Figure 17 illustrates the contents of a sample configuration file for the German → English language pair (the order of appearance of the elements being free).

The minimum required correspondence between SL and TL tags, for SL tags with high-frequency of lexicon entries

Transliteration file: An xml-format file must be created named with the two characters that used by the system for each language under the folder `data/PhraseAligner/Transliteration/` containing all required transliterations from non-Latin to Latin languages. The transliteration file is optional and required only by languages with non-Latin alphabets. Figure 18 illustrates the contents of the transliteration file for the Greek language.

7.2 c. How to add a new language pair in PAM

Adding a new language pair in PAM involves (1) repeating all the steps mentioned above, i.e. processing a bilingual corpus, using a bilingual corpus, and creating language attribute files and (2) updating the configuration file with information about the new resources. Figure 19 illustrates the contents of the configuration file updated with information concerning a new language pair, in this case the pair Greek → English (denoted as EL-EN).

ELEMENT	DESCRIPTION
lexiconCoverageThreshold	The coverage threshold below which sentence pairs are rejected
lexiconTagSeparator	The string or character for separating multiple tags in the bilingual lexicon
lexiconWithPartialTags	Determines whether the lexicon contains partial tags
lexiconCheckTokens	Determines whether to check tokens
lexiconLowSlEntries	Lower threshold for ignoring SL tags
similarityRequiredLength	The required length of tokens for character-wise similarity to be taken into account during token-matching
similarityMatchPerc	The required percentage match of tokens
distanceThreshold	Distance threshold for a single alignment to be made
extagGaussSigma	The sigma of the Gaussian distribution for incorporating the distance between tokens
extagSLSLNormSimThreshold	The lower threshold determining the minimum normalized extended tag similarity for a match to be allowed
SlSlExtagSimilarityThreshold	The minimum extended tag similarity threshold
SlTlTagCorrLowEntriesThreshold	The minimum required correspondence between SL and TL tags, for SL tags with low-frequency of lexicon entries
SlTlTagCorrHighEntriesThreshold	The minimum required correspondence between SL and TL tags, for SL tags with high-frequency of lexicon entries

Table 9: PAM config.xml elements

```
<?xml version="1.0" encoding="UTF-8"?>
<LanguagePair>
 <DE-EN>
  <!--Lexicon-->
  <lexiconCoverageThreshold>0.40</lexiconCoverageThreshold>
  <lexiconTagSeparators>-</lexiconTagSeparators>
  <lexiconWithPartialTags>false</lexiconWithPartialTags>
  <lexiconCheckTokens>true</lexiconCheckTokens>

  <!--Lexicon low SL Entries-->
  <lexiconLowSlEntries>100</lexiconLowSlEntries>

  <!--Similarity Thresholds-->
  <similarityRequiredLength>3</similarityRequiredLength>
  <similarityMatchPerc>50</similarityMatchPerc>

  <!--Distance-->
  <distanceThreshold>4.0</distanceThreshold>

  <!--Extended Tag-->
  <extagGaussSigma>1.4</extagGaussSigma>
  <extagSLSLNormSimThershold>0.10</extagSLSLNormSimThershold>

  <!--SL-SL extended tag similarity-->
  <SlSlExtagSimilarityThreshold>0.10</SlSlExtagSimilarityThreshold>

  <!--Tag correspondence(exported by Lexicon)-->
  <SlTlTagCorrLowEntriesThreshold>0.15</SlTlTagCorrLowEntriesThreshold>
  <SlTlTagCorrHighEntriesThreshold>0.10</SlTlTagCorrHighEntriesThreshold>
 </DE-EN>
</LanguagePair>
```

Figure 17: Sample config.xml file for the German → English language pair

```
<?xml version="1.0" encoding="UTF-8"?>
<transliterations>
 <transliteration>
 <replacement>α</replacement>
 <replacement>ά</replacement>
 <replacement>Α</replacement>
  <replacement>Ά</replacement>
  <target>a</target>
 </transliteration>
 <transliteration>
 <replacement>β</replacement>
 <replacement>Β</replacement>
  <target>b</target>
 </transliteration>
 <transliteration>
 <replacement>γ</replacement>
 <replacement>Γ</replacement>
  <target>g</target>
 </transliteration>
</transliterations>
```

Figure 18: Part of `data/PhraseAligner/Transliteration/EL.xml` with transliterations rules for the Greek characters.

```
<?xml version="1.0" encoding="UTF-8"?>
<LanguagePair>
 <DE-EN>
    <!--Lexicon-->
  <lexiconCoverageThreshold>0.40</lexiconCoverageThreshold>
  <lexiconTagSeparators>-</lexiconTagSeparators>
  <lexiconWithPartialTags>false</lexiconWithPartialTags>
  <lexiconCheckTokens>true</lexiconCheckTokens>
    <!--Lexicon low SL Entries-->
  <lexiconLowSlEntries>100</lexiconLowSlEntries>
    <!--Similarity Thresholds-->
  <similarityRequiredLength>3</similarityRequiredLength>
  <similarityMatchPerc>50</similarityMatchPerc>
    <!--Distance-->
  <distanceThreshold>4.0</distanceThreshold>
    <!--Extended Tag-->
  <extagGaussSigma>1.4</extagGaussSigma>
  <extagSLSLNormSimThershold>0.10</extagSLSLNormSimThershold>
    <!--SL-SL extended tag similarity-->
  <SlSlExtagSimilarityThreshold>0.10</SlSlExtagSimilarityThreshold>
    <!--Tag correspondence(exported by Lexicon)-->
  <SlTlTagCorrLowEntriesThreshold>0.15</SlTlTagCorrLowEntriesThreshold>
  <SlTlTagCorrHighEntriesThreshold>0.10</SlTlTagCorrHighEntriesThreshold>
 </DE-EN>
 <EL-EN>
    <!--Lexicon-->
  <lexiconCoverageThreshold>0.40</lexiconCoverageThreshold>
  <lexiconTagSeparators>-</lexiconTagSeparators>
  <lexiconWithPartialTags>true</lexiconWithPartialTags>
  <lexiconCheckTokens>false</lexiconCheckTokens>
    <!--Lexicon low SL Entries-->
  <lexiconLowSlEntries>200</lexiconLowSlEntries>
    <!--Similarity Thresholds-->
  <similarityRequiredLength>4</similarityRequiredLength>
  <similarityMatchPerc>50</similarityMatchPerc>
    <!--Distance-->
  <distanceThreshold>3.0</distanceThreshold>
    <!--Extended Tag-->
  <extagGaussSigma>1.4</extagGaussSigma>
  <extagSLSLNormSimThershold>0.20</extagSLSLNormSimThershold>
    <!--SL-SL extended tag similarity-->
  <SlSlExtagSimilarityThreshold>0.10</SlSlExtagSimilarityThreshold>
    <!--Tag correspondence(exported by Lexicon)-->
  <SlTlTagCorrLowEntriesThreshold>0.10</SlTlTagCorrLowEntriesThreshold>
  <SlTlTagCorrHighEntriesThreshold>0.05</SlTlTagCorrHighEntriesThreshold>
 </EL-EN>
</LanguagePair>
```

Figure 19: The config.xml file updated with information about the new language pair Greek–English

Appendix 1

Pseudo-code for Step 1: Alignments based on bilingual Lexicon

Pure alignments based on lexicon

```
//COMMENT: Some translation from lexicon may include SL multiWords and/or TL multiWords
1. Translate all words of sentence
//COMMENT: The SL multiWords include more than one word
2. Order the translations by max number of SL translated words
3. For each translation {
  4. If at least one TL part of translation exists in TL sentence {
    5. If SL part translation ISA multiWord {
      6. Create the corresponding multiWord in SL sentence
    }
    7. For each TL_part of translation {
      8. If TL_part exists in TL sentence {
        9. If TL_part is a single Word {
          10. Make alignment of SL part with the TL_part
        }
        //COMMENT: a multiWord cannot contains parts of another multiWord
        11. Else if TL_part is an existent multiWord OR
        a MultiWord which does not contain any part of existent TL multiword {
          12. Create the corresponding multiWord in TL sentence
          13. Make alignment of SL part with the TL_part
        }
      }
    }
  }
}
```

Alignments based on Transliteration

```
1. If SL has different alphabet from TL {
  2. For each unaligned word or multiword of SL sentence {
    3. For each unaligned word or multiword of TL sentence {
      4. Transliterate the SL word or multiword
      5. If the translitared SL word or multiword matches with TL word or multiword {
        6. Make alignment
      }
    }
  }
}
```

Alignments based on similar or identical words

```
1. If SL has different alphabet from TL {
  2. For each unaligned word or multiword of SL sentence {
    3. For each unaligned word or multiword of TL sentence {
      4. If the SL word or multiword is similar or identical with TL word or multiword {
        5. Make alignment
      }
```

```
    }
  }
}
```

Alignments based on Translation Similarity

```
//COMMENT Alignment based on translation similarity:
1. For each unaligned word or multiword of SL sentence {
  2. For each unaligned word or multiword of TL sentence {
    3. If the translation of SL word or multiword is part of TL word or multiword {
      6. Make alignment
    }
  }
}
```

Pseudo-code for Step 2: Alignments based on tag correspondence and extended PoS tags

Alignment by unique part-of-speech

```
//COMMENT:  9.2.1 Alignment by unique part-of-speech
1. For each part-of-speech of List with part-of-speech {
  //COMMENT: The part-of-speech may have different form in different languages
  2. If part-of-speech is unique in SL sentence AND
    3. If part-of-speech is unique in TL sentence {
      4. If SL word and TL word with unique part-of-speech are both unaligned {
        5. Make alignment between SL word and TL word
        6. Continue with the part-of-speech
      }
  }
}
```

Similarity of extended PoS tags (Words)

```
//COMMENT:  9.2.2 Alignment by similarity of extended tags (Words)
1. While at least a new alignment take place {
  2. For each unaligned SL word or SL multiword {
    3. Take the similarity of all single aligned SL words or SL multiWords normalized by distance

    //COMMENT: To avoid an alignment to take place that based on a risky single alignment
    //as one with article
    4. Remove all single aligned words or multiWords which their tag has low number of references
    5. Remove all single aligned words or multiWords which have low tag similarity
    6. Collect all unaligned TL words which belong to the same phase as the one
       with the single aligned TL word
    7. Remove all unaligned TL words which low tag correspondence, based on
       lexicon, with unaligned SL word
    8. Make alignment between unaligned SL word and all unaligned TL words
    9. Solve multiple alignments
    10. Remove unsolved multiple alignments
  }
```

```
}
```

Tag Correspondence

```
//COMMENT:  9.2.3 Alignment by tag correspondence
1. Collect all unaligned SL words or multiWords to a list
2. Collect all unaligned TL words or multiWords to a list
3. For each unaligned SL word or SL multiword in the list {
  4. For each unaligned TL word or TL multiword in the list{
    5. If tags of unaligned SL and TL words exceeds the tag correspondence threshold {
      6. Make alignment between unaligned SL word and all unaligned TL words or multiWords
    }
  }
}
7. Solve multiple alignments
8. Remove unsolved multiple alignments
```

7.2.1 Similarity of extended PoS tags (Phrases)

```
//COMMENT:  9.2.4 Alignment by tag correspondence (Phrases)
1. While at least a new alignment take place {
  2. For each unaligned SL word or SL multiword
    3. Take the similarity of all single aligned SL words or SL multiWords normalized by distance

    //COMMENT: To avoid an alignment to take place that based on a risky single alignment
    //as one with article
    4. Remove all single aligned words or multiWords which their tag has low number of references
    5. Remove all single aligned words or multiWords which have low tag similarity
    6. Take the single aligned SL word or multiWord with the maximum extended tag similarity
    7. Get the TL word or multiWord which is aligned with the single aligned SL word or multiword
    8. If the TL word belongs to exactly one phrase {
      9. Make alignment with the phrase which belong the above TL word or multiWord
    }
  }
}
```

7.3 Pseudo-code for Step 3: Alignments based on neighbours

Alignment of word between same phrases

```
//COMMENT:  9.2.4 Alignment of word between same phrases
1. For each unaligned SL word {
  2. If neighbours of unaligned SL word are single aligned and belong to same TL phrase
    5. Make alignment between unaligned SL word and TL phrase
  }
}
```

7.4 Alignment with neighbour phrase

```
//COMMENT:  9.2.4 Alignment with neighbor phrase
1. For each unaligned SL word {
```

```
  2. If neighbours of unaligned SL word are single aligned and aligned with different TL phrase {
    3. Take the single aligned SL word of previous phrase with the maximum extended tag similarity
    4. Take the single aligned SL word of next phrase with the maximum extended tag similarity
    5. If the maximum extended tag similarity exceeds a threshold make alignment with the TL phrase
  }
}
```

8 Alignment with middle unused phrase

```
//COMMENT:  9.2.4 Alignment of middle unused phrase
1. For each unaligned SL word {
  2. If neighbours of unaligned SL word belong to difference TL phrases {
    3. If the TL phrases are among an unused TL phrase {
      4. If the unused TL phrase support the PoS of unaligned SL word {
        5. Make alignment between the unaligned SL word and the unused TL phrase
      }
    }
  }
}
```

9 Alignment with tag model pattern

```
//COMMENT:  9.2.4 Alignment with the tag model pattern
1. For each unaligned SL word {
  2. If neighbours of unaligned SL word are single aligned
    3. If the patterns of the tags exported from the previous, the unaligned and
       the next SL word exists {
      4. Make alignment between the unaligned SL word and the TL phrase
         according to the tag model
    }
}
```

Appendix 2

The experiments that are described in this report are exported by using PAM revision 3589 and the resources that are represented in Table 10.

Language Pair	Revision of Resources					
	SL.xml	TL.xml	ParsedGolden.xml	SL Attributes	TL Attributes	Lexicon
Czech-German	3294	2895	2852	3508	3512	3583
Czech-English	3307	1797	2536	3508	3512	3587
German-English	3000	3027	3027	3512	3512	3583
Greek-English	3478	3229	3414	3508	3512	3505
Norwegian-German	3495	2092	3495	3508	3512	3284

Table 10: Evaluation Results Resources