Lexical token alignment: experiments, results and applications

Dan Tufiş, Ana-Maria Barbu

RACAI
13 Septembrie, 13, Bucharest 1, Romania
{tufis,abarbu}@racai.ro

Abstract

Lexical alignment is one of the most challenging tasks in processing and exploiting parallel texts. There are numerous applications that may benefit from an accurate multilingual lexical alignment of bi- and multi-language corpora. We describe in this paper a hypothesis-testing approach to the problem of automatic extraction of translation equivalents from sentence-aligned and tagged parallel corpora. The algorithm was used for automatic extraction of 6 bi-lingual lexicons with English as source language and Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene as the target one, as well as a 7-language lexicon with English as a hub and the other 6 CEE languages. For the experiments described here we used the 7-language aligned corpus based on Orwell's "1984" novel.

1. Introduction

A pair of texts that represent the translation of each other is called a parallel text or a *bitext*. Extracting bilingual dictionaries from a bitext is a process based on the notion of *translation equivalence*. In a given parallel text, the assumption is that the *same* meaning is linguistically expressed in two or more languages. Meaning identity between two or more representations of presumably the same thing is a notorious philosophical problem and even in more precise contexts than language (for instance in software engineering) it remains a fuzzy concept. Consequently the notion of translation equivalence relation, built on the meaning identity assumption, is inherently vague. In the area of machine translation, terminology, multilingual information retrieval and other related domains, one needs operational notions, defined in precise, quantifiable terms. One of the widely accepted interpretations (Melamed, 2001) of the translation equivalence defines it as a (symmetric) relation that holds between two different language texts such that expressions appearing in corresponding parts of the two texts are reciprocal translations. These expressions are called *translation equivalents*.

For bilingual dictionaries extraction from a bitext it is of interest the identification of translation equivalents at the lexical level (words or expressions).

In spite of the bi-directionality of the translation equivalence relation, the text in one language is usually called the *source* of the bitext and the text in the other language is called the *target* of the bitext.

One basic resource in translating a text (thus creating a bitext) is a bilingual dictionary (a set of lexical translation equivalents). Automatic extraction of lexical translation equivalents is the reverse process aiming at discovering the bilingual dictionary used in a bitext.

Most modern approaches to automatic extraction of translation equivalents (backed up by the power of nowadays computers) rely on statistical techniques and roughly fall into two categories. The *hypotheses-testing* methods such as (Gale and Church, 1991), (Smadja et all, 1996) etc. rely on a generative device that produces a list of translation equivalence candidates (TECs), each of them being subject to an independence statistical test. The TECs that show an association measure higher than expected under the independence assumption are assumed to be translation-equivalence pairs (TEPs). The TEPs are extracted independently one of another and therefore the process might be characterised as a local maximisation (greedy) one. The *estimating* approach (Brown et all, 1993), (Kupiec, 1993), (Hiemstra, 1997) etc. is based on building from data a statistical bitext model the parameters of which are to be estimated according to a given set of assumptions. The bitext model allows for global maximisation of the translation equivalence relation, considering not individual translation equivalents but sets of translation equivalents (sometimes called *assignments*).

There are pros and cons for each type of approach, some of them discussed in (Hiemstra, 1997). Essentially, the hypotheses testing is computationally cheaper since it works with a reasonable search space, proportional to N^2, where N is the maximum of the numbers of lexical items in the two parts of the bitext, but it has difficulties with finding rare translation equivalents of the bitext. The estimating approach is theoretically extremely expensive from the computational point of view, the search space being proportional to N! (N is the same as above), but in principle are expected to produce accurate bilingual dictionaries with broader coverage (better recall). Very efficient implementations, supported by reasonable assumptions, allow for fast convergence towards the interesting part of the huge search space (Brown et al., 1993).

Our method is a greedy one and makes decisions based on local contexts. It generates first a list of translation equivalent candidates and then successively extracts the most likely translation-equivalence pairs.

1.1. Words and multiword lexical tokens

In the previous section we defined a translation equivalent as a special pair of two lexical items, one in the source language of the bitext and the second in the other language of the bitext. In general, a lexical item is considered to be a space-delimited string of characters or what is usually called a word[1]. However, it is not necessary that a space in text be interpreted always as a lexical item delimiter. For various reasons, in many

[1] Obviously this comment applies for languages that use the space delimiter.

languages and even in monolingual studies, some sequences of traditional words are considered as making up a single lexical unit. For instance in English “in spite of”, “machine gun”, chestnut tree”, “take off” etc. or in Romanian “de la”(from), “gaura cheii” (keyhole), “sta în picioare”(to stand), (a)”-si aminti” (remember), etc. could be arguably considered as single meaningful lexical units even if one is not concerned with translation. For translation purposes considering multiword expressions as single lexical units is a must because of the differences that might appear in linguistic realisation of commonly referred concepts. One language might use concatenation (with or without a hyphen at the joint point), agglutination, derivational constructions or a simple word where other language might use a multiword expression (with compositional or non-compositional meaning).

In the following we will refer to words and multiword expressions as lexical tokens, or simply, tokens.

The recognition of multiword expressions as single lexical tokens, but also the splitting of single words into multiple lexical tokens (when it is the case) is generically called text segmentation and the program that performs this task is called segmenter or tokenizer. The simplest method for text segmentation is based on (monolingual) lists of most frequent compound expressions (collocations, compound nouns, phrasal verbs, idioms, etc) and some regular expression patterns for dealing with too many instantiations of similar constructions (numbers, dates, abbreviations, etc). This linguistic knowledge is referred to as tokenizer’s resources. In this approach the tokenizer would check if the input text contains string sequences that match any of the stored patterns and in such a case the matching input sequences are replaced as prescribed by the tokenizer’s resources. In spite of being very simple, the main criticism against this text segmentation method is that the tokenizer’s resources are never exhaustive.

For our experiments we used Philippe di Cristo’s multilingual segmenter MtSeg (http://www.lpl.univ-aix.fr/projects/multext/MtSeg/) developed for the MULTEXT project. The segmenter comes with tokenization resources for many Western European languages, further enhanced in the MULTEXT-EAST project (Erjavec& Ide, 1998), (Dimitrova et al., 1998), (Tufiş et al, 1998) with corresponding resources for Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene. The segmenter is able to recognise dates, numbers, various fix phrases, to split clitics or contractions etc.

To cope with the inherent incompleteness of the segmenter resources, besides using a collocation extractor, we experimented with a complementary method that takes advantage of the word alignment process by trying to identify partially correct translation equivalents. This procedure is briefly reviewed in section on partial translations.

In general one word in one part of a bitext is translated also by one words in the other part of the bitext. If this statement, called the “*word to word mapping hypothesis*” would be always true, the lexical alignment problem would become significantly easier to solve. But we all know that the “word to word mapping hypothesis” is not true. By introducing the notion of lexical token, we tried to alleviate this difficulty assuming that proper segmentations of the two parts of a bitext would make the “*token to token mapping hypothesis*” a valid working assumption. We will generically refer to this mapping hypothesis the “*1:1 mapping hypothesis*” in order to cover both word-based and token-based mappings.

With the “1:1 mapping hypothesis” considered, the translation equivalence pairs are certainly included in the Cartesian product computed over the sets of words or tokens in the two parts of the bitext: TEP={<seg$_{Si}$ seg$_{Tj}$ >}⊂ T$_S$⊗T$_T$. The search space contains $\mathbf{K}^2$ possible translation equivalence pairs (for a hypotheses testing approach) or **K!** possible assignments (for an estimating approach). If the 1:1 mapping hypothesis is not the underlying one, then the search space is much larger, namely $\wp(T_S)\otimes\wp(T_T)$, where $\wp(X)$ is the power-set of X with card($\wp(X)$)= $2^{card(X)}$ (for an hypotheses testing approach) or $\wp(X)!$ assignments (for an estimating approach).

Using token-based segmentation as described in the previous section most of the limitations of the”1:1 mapping” hypothesis are eliminated and the problem to be solved (dictionary extraction) becomes computationally much cheaper.

Melamed (1996) observed that most of the translation pairs are conserving their part of speech, that is most of the time a verb translates as a verb, a noun as a noun and so on. He called such translation pairs V-type, to distinguish form those translation pairs where the part of speech of one token in not the same as the one for the other token. This type was called P-type translation pairs. The third category of translation pairs is represented by the incomplete translation (I-type), incompleteness resulting from his 1:1 mapping underlying approach.

Melamed’s findings concerning the translation types distribution are quite similar to ours although our text was a literary one while his was an extract from the Canadian Parliament debates (a text containing more literal translations). What is worth mentioning is that the P-type pairs do not contain arbitrary paired parts of speech and one might consider regular patterns in part-of speech alternations (participle-adjective, gerund-noun, gerund-adjective) in order to assimilate most of the P-type pairs with the V-type ones.

Before proper extraction of translation equivalents, the parallel texts are sentence-aligned (using a slightly modified version of Gale and Church CharAlign program) and morpho-syntactically tagged (using a tiered-tagging approach as described in (Tufiş, 2000) build on TnT tagger (Brants, 2000)).

In the following we will get into the details of our method for bilingual dictionary extraction and its implementation.

2. The baseline algorithm (BASE)

There are several underlying assumptions one can consider in keeping the computational complexity of a word alignment algorithm as low as possible. None of them is true in general, but the situations where they are not true are rare enough so that ignoring the exceptions would not produce a significant number of errors and would not loose too many useful translations. Moreover, the assumptions we used do not prevent additional processing units for recovering some of the correct translations missed because they did not observe the assumptions. The assumptions we used in our basic algorithm are the following:

- a lexical token in one half of the TU corresponds to at most one non-empty lexical unit in the other half of the TU; this is the 1:1 mapping assumption which underlines the work of many other researchers (Kay & Röscheisen, 1993), (Melamed, 2001), (Brew & McKelvie, 1996), (Hiemstra, 1997), (Tiedemann, 1998), (Ahrenberg et al., 2000) etc. However, remember that a lexical token could be a multiple word expression previously found and segmented as such by an adequate tokenizer;
- a polysemous lexical token, if used several times in the same TU, is used with the same meaning; this assumption is explicitly used also by (Melamed, 2001) and implicitly by all the previously mentioned authors.
- a lexical token in one part of a TU can be aligned to a lexical token in the other part of the TU only if the two tokens have compatible types (part-of-speech); in most cases, compatibility reduces to the same POS, but it is also possible to define compatibility mappings (e.g. participles or gerunds in English are quite often translated as adjectives or nouns in Romanian and vice versa). This is essentially one very efficient way to cut off the combinatorial complexity and postpone dealing with irregular ways of POS alternations.
- although the word order is not an invariant of translation, it is not random either; when two or more candidate translation pairs are equally scored, the one containing tokens which are closer in relative position are preferred. This preference is also used in (Ahrenberg et al., 2000).

Based on the sentence alignment, tagging and lemmatisation, the first step is to compute a list of translation equivalence candidates (TECL). This list contains several sub-lists, one for each POS considered in the extraction procedure. Each POS-specific sub-list contains several pairs of tokens <token$_S$ token$_T$> of the corresponding POS that appeared in the same TUs. Let TU^j be the jth translation unit. By collecting all the tokens of the same POS$_k$ (in the order they appear in the text and removing duplicates) in each part of TU^j one builds the ordered sets L^{Sj}_{POSk} and L^{Tj}_{POSk}. For each POS$_i$ let TU^j_{POSi} be defined as $L^{Sj}_{POSi} \otimes L^{Tj}_{POSi}$. Then, CTU^j (mappings in the jth translation unit) is defined as follows:

$$CTU^j = \bigcup_{i=1}^{no.of.pos} TU^j_{POSi}$$

With these notations, and considering that there are *n* alignment units in the whole bitext, TECL is defined as:

$$TECL = \bigcup_{j=1}^{n} CTU^j$$

TECL contains a lot of noise and many TECs are very improbable. In order to eliminate much of this noise, TECL is filtered out of the very unlikely candidate pairs. For the ranking of the TECs and their filtering we experimented 4 scoring functions: MI (*pointwise* mutual information), DICE, LL (loglikelihood) and χ^2 (chi-square). After various empirical tests we decided to use loglikelihood test with the threshold value set to 9.

One baseline algorithm is not very different from the filtering discussed above. However, for improving the precision, the thresholds of whatever statistical test used is higher. Some additional restrictions such as a minimal number of occurrences for <T$_S$ T$_T$> (usually this is 3) are also used. This baseline algorithm may be enhanced in many ways (using a dictionary of already extracted TEPs for eliminating generation of spurious TECs, stop-word lists, considering token string similarity etc.). An algorithm with such extensions (plus a few more) is described in (Gale and Church, 1991). Although extremely simple, this algorithm, applied on a sample of 800 sentences from Canadian Hansard, was reported to provide impressive precision (about 98%). However, the algorithm managed to find only the most frequent words (4.5%) that cover more than half (61%) of the word occurrences in the corpus. Its recall is modest if judged in terms of word types (cf. Melamed, 2001).

Our baseline algorithm is an improvement over the one described before. It is a very simple iterative algorithm, significantly faster than the previous one, with much better recall even when the precision is required to be as high as 98%. It can be enhanced in many ways (including those discussed above). It has some similarities to the iterative algorithm presented in (Ahrenberg et all. 1998) but unlike it, our algorithm avoids computing various probabilities (or better said probability estimates) and scores (t-score). At each iteration step, the pairs that pass the selection (see below) will be removed from TECL so that this list is shortened after each step and eventually may be emptied. Based on TECL, for each POS a S_m* T_n contingency table (TBLk) is constructed, with S_m the number of token types in the first part of the bitext and T_n the number of token types in the other part of the bitext. The selection condition is expressed by the equation:

$$\text{(EQ1)}\quad TP^k = \{< T_{Si}\ T_{Tj} > | \forall p,q\,(n_{ij} \geq n_{iq}) \wedge (n_{ij} \geq n_{pj})\}$$

This is the key idea of the iterative extraction algorithm and it expresses the requirement that in order to select a TEC <T$_{Si}$, T$_{Tj}$> as a translation equivalence pair, the number of associations of T$_{Si}$ with T$_{Tj}$ must be higher than (or at least equal to) any other T$_{Tp}$ (p≠j). The same holds for the other way around. All the pairs selected in TP^k are removed (the respective counts are zeroed). If T$_{Si}$ is translated in more than one way (either because of having multiple meanings that are lexicalised in the second language by different words, or because of use in the target language of various synonyms for T$_{Tj}$) the rest of translations will be found in subsequent steps (if frequent enough). The most used translation of a token T$_{Si}$ will be found first. The TECL is implemented as a hash table.

3. A better extraction algorithm (BETA)

One of the main deficiencies of the BASE algorithm is that it is quite sensitive to what (Melamed, 2001) calls indirect associations. If <T$_{Si}$, T$_{Tj}$> has a high association score and T$_{Tj}$ collocates with T$_{Tk}$, it might very well happen that <T$_{Si}$, T$_{Tk}$> gets also a high association score. Although, as observed by Melamed, in general, the indirect associations have lower scores than the direct (correct) associations, they could receive higher scores than many correct pairs and this will not only generate wrong translation equivalents, but will eliminate from further considerations several correct pairs, deteriorating the procedure's recall. To weaken this sensitivity, the BASE algorithm had to impose that the number of occurrences of a TEC be at least 3, thus filtering out more than 50% of all the possible TECs. Still, because of the indirect association effect, in spite of a very good precision (more than 98%) out of the considered pairs another approximately 50% correct pairs were missed. The BASE algorithm has this deficiency because it looks on the association scores globally, and does not check within the TUs if the tokens making the indirect association are still there.

To diminish the influence of the indirect associations and consequently removing the occurrence threshold, we modified the BASE algorithm so that the maximum score is not considered globally but within each of the TUs. This brings BETA closer to the competitive linking algorithm described in (Melamed, 1996, 2001). The competing pairs are only the TECs generated from the current TU and the one with the best score is the first selected. Based on the 1:1 mapping hypothesis, any TEC containing the tokens in the winning pair are discarded. Then, the next best scored TEC in the current TU is selected and again the remaining pairs that include one of the two tokens in the selected pair are discarded. The multiple-step control in BASE, where each TU was scanned several times (equal to the number of iteration steps) is not necessary anymore. The BETA algorithm will see each TU unit only once but the TU is processed until no further TEPs can be reliably extracted or TU is emptied. This modification improves both the precision and recall in comparison with the BASE algorithm. In accordance with the 1:1 mapping hypothesis, when two or more TEC pairs of the same TU share the same token and they are equally scored, the algorithm has to make a decision and choose only one of them. We used two heuristics: string similarity scoring and relative distance.

The similarity measure we used, COGN(T_S, T_T), is very similar to the **XXDICE** score described in (Brew&McKelvie, 1996). If T_S is a string of *k* characters $\alpha_1\alpha_2 \ldots \alpha_k$ and T_T is a string of *m* characters $\beta_1\beta_2 \ldots \beta_m$ then we construct two new strings T'_S and T'_T by inserting where necessary special displacement characters into T_S and T_T. The displacement characters will cause both T'_S and T'_T have the same length p (max (k, m)≤p<k+m) and the maximum number of positional matches. Let $\delta(\alpha_i)$ be the number of displacement characters that immediately precedes the character α_i which matches the character β_i and $\delta(\beta_i)$ be the number of displacement characters that immediately precedes the character β_i which matches the character α_i. Let *q* be the number of matching characters. With these notations, the COGN(T_S, T_T) similarity measure is defined as follows:

$$\mathrm{COGN}(\mathrm{T_S}, \mathrm{T_T}) = \begin{cases} \dfrac{\displaystyle\sum_{i=1}^{q} \frac{2}{1+|\,\delta(\alpha_i)-\delta(\beta_i)\,|}}{k+m} & \text{if q > 2} \\ 0 & \text{if q} \leq 2 \end{cases}$$

The threshold for the COGN(T_S, T_T) was empirically set to 0.42. This value depends on the pair of languages in the considered bitext. The actual implementation of the COGN test considers a language dependent normalisation step, which strips some suffixes, discards the diacritics and reduces some consonant doubling etc. This normalisation step was hand written, but, based on available lists of cognates, it could be automatically induced.

The second filtering condition, DIST(T_S, T_T) is defined as follows:

if ((<T_S, T_T>∈ $L^{Sj}{}_{posk}$⊗$L^{Tj}{}_{posk}$)&(T_S is the *n*-th in $L^{Sj}{}_{posk}$) & (T_T is the *m*-th in $L^{Tj}{}_{posk}$)) then DIST(T_S, T_T)=|n-m|

The COGN(T_S, T_T) filter stronger than DIST(T_S, T_T), so that the TEC with the highest similarity score is the preferred one. If the similarity score is irrelevant, the weaker filter DIST(T_S, T_T) gives priority to the pairs with the smallest relative distance between the constituent tokens.

4. Experiments and results

We conducted experiments on one of the few publicly available multilingual aligned corpora, namely the "1984" multilingual corpus (Dimitrova et al, 1998) containing 6 translations of the English original. This corpus was developed within the Multext-East project, published on a CD-ROM (Erjavec et all. 1998) and recently improved within the CONCEDE project. The newer version is distributed by TRACTOR-*TELRI Research Archive of Computational Tools and Resources* (www.tractor.de). Each monolingual part of the corpus (Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene) was tokenised, lemmatised, tagged and sentence aligned to the English hub.

In the context of this paper, we distinguish between hapax token and hapax translation pairs. The notion of hapax token (a token that appeared only once) is defined monolingually while the notion of hapax translation pair (a translation pair that appeared only once) is defined in a bilingual context. If one or either of the constituents of a translation pair is a hapax token than the translation pair is also a hapax one. But the other way around is not necessary true. A recurrent token might be used with different senses and these senses might be lexicalized by different tokens in the other part of the bitext. Also, a recurrent token, although used with the same meaning, might be translated by different synonyms. We relax the definition of "translation hapax" as being a pair of translation equivalents, which appears in a single TU. Therefore, even if a translation pair occurs twice or more in the same TU and in no other TU it will still be considered a translation hapax.

The evaluation protocol specified that all the translation pairs are to be judged in context, so that if one pair is found to be correct in at least one context, then it should be judged as correct. The evaluation was done for both BASE and BETA algorithms but on different scales. The BASE algorithm was run on all the 6 bitexts with the English hub and native speakers of the second language in the bitexts (with good command of English) validated 4 of the 6 bilingual lexicons.

The lexicons contained all parts of speech defined in the MULTEXT-EAST lexicon specifications (Erjavec & Monachini, 1997) except for interjections, particles and residuals. Each monolingual part of the corpus (Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene) was tokenised, lemmatised, tagged and sentence aligned to the English hub. In Table 1 there are shown the numbers of lemmas in each monolingual part of the multilingual corpus as well as the number of lemmas that occurred more than twice.

Language	Bulgarian	Czech	English	Estonian	Hungarian	Romanian	Slovene
No. of wordforms*	15093	17659	9192	16811	19250	14023	16402
No. of lemmas*	8225	8677	6871	8403	9729	6987	7157
No.of >2-occ lemmas*	3350	3329	2916	2729	3294	2999	3189

Table 1:The lemmatised monolingual "1984" overview

4.1. The evaluation of BASE algorithm

For validation purposes we limited the number of iteration steps to 4. The extracted dictionaries contain adjectives (A), conjunctions (C), determiners (D), numerals (M), nouns (N), pronouns (P), adverbs (R), prepositions (S) and verbs (V). Table 2 shows the evaluation results for those languages, where we found voluntary native speaker evaluators. The precision (**Prec**) was computed as the number of correct TEPs divided by the total number of extracted TEPs. The recall (considered for the non-English language in the bitext) was computed two ways: the first one, ***Rec****, which took into account only the tokens processed by the algorithm (those that appeared at least three times). The second one, **Rec**, took into account all the tokens irrespective of their frequency counts. ***Rec**** is defined as the number of source lemma types in the correct TEPs divided by the number of lemma types in the source language with at least 3 occurrences. **Rec** is defined as the number of source lemma types in the correct TEPs divided by the number of lemma types in the source language.

The accuracy of the extraction process varies with respect to different parts of speech. Table 3 displays extraction precision differentiated per part of speech.

Bitext	Bg-En Prec/*Rec**/*Rec*	Cz-En Prec/*Rec**/*Rec*	Et-En Prec/*Rec**/*Rec*	Hu-En Prec/*Rec**/*Rec*	Ro-En Prec/*Rec**/*Rec*	Sl-En Prec/*Rec**/*Rec*
Extracted pairs (4 Steps)	1986 NA/NA/*NA*	2188 NA/NA/*NA*	1911 96.2/57.9/*18.8*	1935 96.9/56.9/*19.3*	2227 98.4/58.8/*25.2*	1646 98.7/57.9/*22.7*

Table 2: Partial evaluation of the BASE algorithm after 4 iteration steps

POS	A	C	D	M	N	P	R	S	V	Total
Extracted pairs	299	29	30	25	1243	39	170	21	371	2227
Wrong pairs	5	0	0	0	21	0	8	0	2	36
POS precision	98.3	100	100	100	98.3	100	95.3	100	99.7	98.38

Table 3: Romanian-English dictionary; POS precision of the BASE algorithm after 4 iteration steps

The rationale for showing ***Rec**** is to estimate the proportion of the missed considered tokens. This might be of interest when precision is of utmost importance. When the threshold of minimal 3 occurrences is considered, the algorithm provides a high precision and a good recall (***Rec****). The evaluation was fully done for Estonian, Hungarian and Romanian and partially for Slovene (the first step was fully evaluated while the rest were evaluated from randomly selected pairs).

As one can see from the Table 2, after four iterations, the precision is higher than 98% for Romanian and Slovene, almost 97% for Hungarian and more than 96% for Estonian. ***Rec**** ranges from 50.92% (Slovene) to 63.90% (Estonian). The standard recall **Rec** varies between 19.27% and 32.46% (quite modest, since on average, the BASE algorithm did not consider 60% of the lemmas).

To facilitate the comparison with the evaluation of the BETA algorithm we ran the BASE algorithm for extracting the noun translation pairs from the Romanian-English bitext. The noun extraction had the second worst accuracy (the worst was the adverb), and therefore we considered that an in-depth evaluation of this case would be more informative than a global evaluation. We set no limit for the number of steps and lowered the occurrence threshold to 2. The program stopped after 10 steps with a number of 1900 extracted translation pairs, out of which 126 were wrong (see Table 4). Compared with the 4 steps run the precision decreased to 93.36%, but both ***Rec**** (70.12%) and **Rec** (39.76%) improved.

Noun types in text	No. entries	Correct entries	Types in correct entries	Prec/*Rec**/Rec
3435 (1948 occ>1)	1900	1774	1366	93.36/70.12/39.76

Table 4: BASE evaluation on the noun dictionary extracted from the Romanian-English bitext (non-hapax)

Another way of evaluating the recall is by showing what percentage of the tokens in the text the types in the dictionary cover. This measure is better called *coverage* and we denote it by **Coverage**. However, the coverage is not very informative since few most frequent tokens ensure usually a large coverage. So, if an extraction algorithm would find translations only for these token types, its **Coverage** score will be pretty good. As we mentioned previously, the 4.5% token types (this is the recall in our evaluation) for which the algorithm described in (Gale, Church, 91) found a translation, covered more than 61% of the text.

In the 10-step run of the BASE algorithm, the extracted noun pairs covered 85.83% of the nouns in the Romanian part of the bitext.

We should mention that in spite of the general practice in computing recall for bilingual dictionary extraction task (be it ***Rec****, **Rec** or **Coverage**) this is only an approximation of the real recall. The reason for this approximation is that in order to compute the real recall one should have a gold standard with all the words aligned by human evaluators. In general such a gold standard bitext is not available and the recall is either approximated as above, or is evaluated on a small sample and the result is taken to be more or less true for all the bitext.

In the initial version of the BASE algorithm we used a chi-square test to check the selected TEPs. However, as the selection condition (EQ1) is highly restrictive, the vast majority of the selected TEPs passed the chi-square test while many pairs that used to pass the chi-square threshold did not pass the condition (EQ1). Therefore we eliminated this unnecessary statistical test, which resulted in a very small decrease in recall but is compensated by a better precision and a significant improvement in response time.

From the 6 bilingual lexicons we also derived a 7-language lexicon (2862 entries), with English as a search hub (see Table 5). As more than half of the English words had equivalents only in 2 or three languages, we considered only those entries for which our algorithm found translations in all but at most one of the other 6 languages.

This filtered multilingual lexicon contains 1237 entries and can be found at the same site as the bilingual lexicons. A typical entry in this multilingual lexicon is given below (in Figure 15 the multiword dictionary entry is shown by using each language character set; in the actual file there are used SGML entities).

En	Bg	Cs	Et	Hu	Ro	Sl
cold	студен	studený/chladný	külm	Hideg	rece/friguros	mrzel/hladen

Table 5: An entry from the extracted multilingual lexicon

4.2. The evaluation of the BETA algorithm

The BETA algorithm preserves the simplicity of the BASE algorithm but it significantly improves its recall (**Rec**) at the expense of some loss in precision (**Prec**).

As said before, at the time of this writing, the evaluation for the BETA algorithm was done only for the Romanian-English bitext and only with respect to the dictionary of nouns. The filtering condition in case of ties was the following:

$(\max(COGN(T^j_S, T^j_T)\geq 0.42)) \vee (\min(DIST(T^j_S, T^j_T)\leq 2))$.

The figures in the tables below summarise the results for this case. The results show that the **Rec** (72.66%) almost doubled compared with the best **Rec** obtained by the BASE algorithm for nouns (39.85%, see Table 4). The **Coverage** also improved up to 93.06%.

Noun types in text	No. entries	Correct entries	Types in correct entries	Prec/Rec
3435	4023	3149	2496	78.27/72.66

Table 6: BETA evaluation, TECs filtered with the condition $(\max(COGN(T^j_S, T^j_T)\geq 0.42))\vee (\min(DIST(T^j_S, T^j_T)\leq 2))$.

Noun types in text	No. entries	Correct entries	Types in correct entries	Prec/Rec
3435	3713	3007	2371	80.98/69.02

Table 7: BETA evaluation, hapax TECs filtered with the condition $\max(COGN(T^j_S, T^j_T)\geq 0.42)$.

However, the price for these significant improvements was a serious deterioration of the **Prec** (78.27% versus 93.36%).

The analysis of the wrong translation pairs revealed that most of them were hapax pairs and they were selected because the DIST measure enabled them, so we considered that this filter is not discriminative enough for hapaxes. On the other hand for the non-hapax pairs the DIST condition was successful in more than 85% of the cases. Therefore, we decided that the additional DIST filtering condition be preserved for non-hapax competitors only. The results in Table 7 show that 166 erroneous TEPs were removed but also 144 good TEP were lost. **Prec** improved (80.93% versus 78.28%) but **Rec** depreciated (69.02% versus 72.65%). The **Coverage** score for this modified version of BETA slightly decreased to 92.36%.

The BASE algorithm allows for trading off between **Prec** and ***Rec**** by means of the number of iteration steps. The BETA algorithm allows for similar trading off between **Prec** and **Rec** by means of the COGN and DIST thresholds and obviously by means of an occurrence threshold. For instance when BETA was set to ignore the hapax pairs, its **Prec** was 96.11% (better then the BASE precision 93.36%) ***Rec**** was 96.41% (BASE with 10 iterations had a ***Rec**** of 70.12%) and **Rec** was 59.66% (BASE with 10 iterations had a **Rec** of 39.76%),

5. Partial translations

As the alignment model used by the translation equivalence extraction is based on the 1:1 mapping hypothesis, inherently it will find partial translations for those cases where one or more words in one language must be translated by two or more words in the other language. Although we used a tokenizer aware of compounds in the two languages, its resources were obviously partial. In the extracted noun lexicon, the evaluators found 116 partial translations (3.86%). In this section we will discuss one way to recover the correct translations for the partial ones, discovered by our 1:1 mapping-based extraction program.

First, from each part of the bitext a set of possible collocations was extracted by a simple method called “repeated segments” analysis. Any sequence of two or more tokens that appears more than once is retained. Additionally, the tags attached to the words occurring in a repeated segment must observe the syntactic patterns characterizing most of the real collocations. For the noun dictionary we considered only forms of <head-noun (functional_word) modifier> as Romanian patterns and <modifier (functional_word) head-noun> as English patterns. If all the content contained in a repeated segment have translation equivalents, then the repeated segment is discarded as not being relevant for a partial translation. Otherwise, the repeated segment is stored in the lexicon as a translation for the translation of its head-noun. For instance, “machine gun” was found as a repeated segment with the translation for “gun” as “mitralieră”. Since “machine” was not translated in the corresponding TUs, the new entry (mitralieră machine_gun) was added to the dictionary. Similarly, “muşuroi de cârtiță” has been found as a repeated segment in the Romanian part of the bitext. Since “muşuroi” was translated in the corresponding TU as “molehill” and “cârtiță” had no translation in the dictionary, the new entry (muşuroi_de_cârtiță molehill) was added to the dictionary. This simple procedure managed to recover 62 partial translations and improve other 12 (still partial, but better). An example of improved partial translation is “poziție de drepți” = “attention” which started with “drept”=”attention” and should have finished with “poziție de drepți” = “stand to attention”| “spring to attention”| “call to attention”.

6. Failures analysis

Any statistical word alignment method is confronted with two principled problems: some tokens are wrongly associated and some valid ones are missed. There are various reasons for both of them and in this section we will discuss our findings with respect to our bitext.

The BETA extraction algorithm did not find translations for 892 Romanian noun lemmas. Out of these, 47 occurred 3 or more times, 102 exactly 2 times and 743 occurred only once. As we have shown in the presentation of our algorithm, the initial phase is to build a search space for the translation equivalence pair. As this space is in general very large, one has to filter it out one way or another. We used the loglikelihood (Dunning, 1993), (Melamed, 2001) removing all the pairs with a score below 9. However, besides throwing away a large number of noisy candidates, some correct pairs were lost as well. This was responsible for about 60% of the correct missed translation pairs (the vast majority of them were hapax pairs, translating secondary meanings of quite frequent words). Working with a much larger corpus might decrease the influence of this factor.

We found 20 English sentences (192 lemmas) not translated in Romanian, out of which 85 lemmas appeared in no other part of the novel. We noticed that many erroneous translation pairs were extracted from very long TUs. The explanation is that the long TUs produce a high level of noise for the way we computed the list of candidates. Because of the alignment problems (errors, missing translations and long TU) the recall was affected by about 6% (1% direct influence and about 5% indirect influence due to the noise).

Tagging errors (about 1% in the Romanian part and about 2.8% in the English part) were responsible for about 22% of the missed correct translations.

Many missing translations got an explanation by virtue of the human translation idiosyncrasies as well as by the different nature of the language pairs considered. Being a literary translation, several words in the original were paraphrased and some words were translated differently (by synonyms). In many cases words in one language were translated in the other by words of different part of speech (from the algorithm point of view this is identical to a tagging error). A few words were wrongly translated and some others were simply ignored. For instance out of the 47 Romanian lemmas occurring more than twice in the text, and missed by the extraction algorithm, 43 are due to one of these causes. Altogether, the translator was deemed responsible for 12% of the missed translations.

7. Implementation

The BASE and BETA programs are written in Perl and run under practically any platform (Perl implementations exists not only for UNIX/LINUX but also for Windows, and MACOS). Table 8 shows the BASE running time for each bitext in the "1984" parallel corpus (all POS considered).

Bitext	Bg-En	Cz-En	Et-En	Hu-En	Ro-En		Si-En
					4 steps	28 steps	
Extraction time (sec)	181	148	139	220	183	415	157

Table 8: BASE extraction time for each of the bilingual lexicons (all POS)

Algorithm	BASE (10 steps)	BETA
Extraction time (s)	105	232

Table 9: BASE and BETA extraction time for the Romanian-English noun dictionary

Table 9 shows the running time for extraction of the noun Romanian-English dictionary (LINUX on a Pentium III/600Mhz with 96 MB RAM) for BASE and BETA.

A quite similar approach to our BASE algorithm (also implemented in Perl) is presented in (Ahrenberg *et al*, 2000) and for a novel of about half the length of Orwell's "1984" their algorithm needed 55 minutes on a Ultrasparc1 Workstation with 320 MB RAM. They used a frequency threshold of 3 and the best results reported are

92.5% precision and 54.6% recall (our ***Rec****). For a computer manual containing about 45% more tokens than our corpus, their algorithm needed 4.5 hours with the best results being 74.94% precision and 67,3% recall (***Rec****).

The BETA algorithm is closer to Melamed's extractor, although our program is greedier and never returns to a visited translation unit. In (Melamed, 2001) information is not provided on any of the extraction times, which we suspect it to be higher than in our case.

8. Conclusions and further work

We presented two simple but very effective algorithms for extracting bilingual lexicons, based on a 1:1 mapping hypothesis. We showed that in case a language specific tokenizer is responsible for pre-processing the input to the extractor, the 1:1 mapping approach is not an important limitation anymore. Incompleteness of the segmenter's resources may be accounted for by using a post-processing phase that recovers the partial translations by taking advantage of the already extracted entries.

We showed elsewhere (Erjavec et al., 2001) how we used the translation dictionaries presented here in sense discrimination of English target words based on their translations. In (Tufiş, Cristea 2002) we describe the use of the automatically extracted Ro-En dictionary for building from scratch a Romanian wordnet and how the synset mapping onto the Inter Lingual Index (ILI) can be checked for consistency in a EuroWordNet-like semantic network.

In cooperation with the Birmingham University we recently started experiments on a 5 million-word Chinese-English parallel corpus. Some preliminary evaluations showed for the first 5000 noun candidate pairs (no filtering other than a very high (250) LL-score) an estimated precision higher than 94% (the precision dramatically dropped to about 10% for the candidates with a LL-score lower than 35).

Acknowledgements.

The research reported here started as an AUPELF/UREF co-operation project with LIMSI/CNRS (CADBFR) and used multilingual corpus and multilingual lexical resources developed within MULTEXT-EAST, TELRI and CONCEDE European projects. Special thanks are due to Wolfgang Teubert for cooperation on the En-Cn experiment and Chang BaoBao for results evaluation.

9. References

Ahrenberg, L., M. Andersson, M. Merkel. (2000).A knowledge-lite approach to word alignment, in Véronis, J. (ed), (2000). Parallel Text Processing. *Text, Speech and Language Technology Series*, Kluwer Academic Publishers Vol. 13, 2000

Ahrenberg, L., Andersson, M. & Merkel, M. (1998) "A Simple Hybrid Aligner for Generating Lexical Correspondences in Parallel Texts". In Proceedings of COLING'98, Montreal, Canada

Brants, T.(2000) "TnT – A Statistical Part-of-Speech Tagger", in *Proceedings of the Sitth Applied Natural Language Processing Conference, ANLP-2000,* April 29 – May 3, 2000, Seattle, WA

Brew, C., McKelvie, D., (1996). Word-pair extraction for lexicography,*http:///www.ltg.ed.ac.uk/~chrisbr/papers*

Brown, P., et al (1993), The mathematics of statistical machine translation: parameter estimation in *Computational Linguistics 19(2)*: 263-311

Dimitrova, L, T. Erjavec, N. Ide, H. Kaalep, V. Petkevic and D. Tufiş, (1998). " Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and East European Languages" in *Proceedings of the 36th Annual Meeting of the ACL and 17th COLING International Conference*, Montreal, Canada, 315-319.

Dunning, T. (1993), Accurate Methods for the Statistics of Surprise and Coincidence in *Computational Linguistics*19(1):61-74

Gale, W.A. and K.W. Church, (1991), Identifying word correspondences in parallel texts. In *Fourth DARPA Workshop on Speech and Natural Language*, pp. 152-157

Gale, W.A. and K.W. Church, (1993). "A Program for Aligning Sentences in Bilingual Corpora". In *Computational Linguistics*, 19(1), pp. 75-102

Erjavec, T., Monachini, M., (eds.), (1997). Specifications and Notation for Lexicon Encoding, *MULTEXT-East Final Report, D1.1*, December 1997.

Erjavec, T., Lawson A., Romary L.. (1998). *East Meet West: A Compendium of Multilingual Resources.* TELRI-MULTEXT EAST CD-ROM, 1998, ISBN: 3-922641-46-6.

Erjavec T., Ide N., Tufiş, "Automatic Sense Tagging Using Parallel Corpora", in Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, Tokyo, Japan, 27-29 November, pp. 212-219, 2001

Hiemstra, D., (1997). Deriving a bilingual lexicon for cross language information retrieval". In *Proceedings of Gronics* 21-26

Kay, M., Röscheisen, (1993). Text-Translation Alignment. In *Computational Linguistics*, 19(1), 121:142

Kupiec, J., (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Meeting of the ACL*, 17:22

Melamed, D., (2001). *Empirical methods for exploiting parallel texts.* MIT Press.

Melamed, D., (1996). "Automatic Construction of Clean Broad-Coverage Translation Lexicons". In *Proceedings of AMTA*

Smadja, F., K.R. McKeown, and V. Hatzivassiloglou, (1996). Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1), 1:38

Tiedemann, J., (1998). Extraction of Translation Equivalents from Parallel Corpora, In *Proceedings of the 11th Nordic Conference on Computational Linguistics*, Center for Sprogteknologi, Copenhagen, 1998, http://stp.ling.uu.se/~joerg/

Tufiş, D., Barbu, A.M., (2001). Extracting multilingual lexicons from parallel corpora in *Proceedings of the ACH/ALLC 2001*, New York University, June 2001.

Tufiş, D., Barbu, A.M. (2001a). Computational bilingual lexicography: automatic extraction of translation lexicons. In *Intl. Journal on Science and Technology of Information*, Vol.4, No.3, pp. 126-148, Nov. 2001

Tufiş, D., Cristea, D. (2002). Methodological issues in building the Romanian Wordnet and consistency checks in Balkanet. In *Proceedings of the Workshop Wordnet Structures and Standardization,* LREC'2002, Las Palmas, Spain, 2002