Procesamiento del lenguaje Natural, nº 41 (2008), pp. 281-288 recibido 7-05-2008; aceptado 16-06-2008

Bilingual Terminology Extraction based on Translation Patterns

Extracción de terminología bilingüe con base en reglas de traducción

Alberto Simões
Departamento de Informática
Universidade do Minho
Braga, Portugal
ambs@di.uminho.pt

José João Almeida
Departamento de Informática
Universidade do Minho
Braga, Portugal
jj@di.uminho.pt

Resumen: Los corpora paralelos son fuentes ricas en recursos de traducción. Este documento presenta una metodología para la extracción de sintagmas nominales bilingües (candidatos terminológicos) a partir de corpora paralelos, utilizando reglas de traducción.
Los modelos propuestos en este trabajo especifican las alteraciones en el orden de las palabras que se producen durante la traducción y que son intrínsecos a la sintaxis de las lenguas implicadas. Estas reglas se describen en un lenguaje de dominio específico llamado PDL (Pattern Description Language) y son sumamente eficientes para la detección de sintagmas nominales.
Palabras clave: corpora paralelos, extracción de información, recursos de traducción, traducción automática

Abstract: Parallel corpora are rich sources of translation resources. This document presents a methodology for the extraction of bilingual nominals (terminology candidates) from parallel corpora, using translation patterns.
The patterns proposed in this work specify the order changes that occur during translation and that are intrinsic to the involved languages syntaxes. These patterns are described in a domain specific language named PDL (Pattern Description Language), and are extremely efficient for the detection of nominal phrases.
Keywords: parallel corpora, information extraction, translation resources, machine translation

1 Introduction

Machine Translation (MT) resources are expensive: translation dictionaries require a lot of hand-work, and translation grammars are impossible to develop for real languages. The advances on computer processing power and methods for statistical data extraction from texts lead to a burst in development of MT systems (Tiedemann, 2003). These systems are data-driven using just statistical information (like Statistical Machine Translation) or previously done translations (like Example Based Machine Translation). Actually, data-driven and rule-based methods coupled on hybrid translation systems. Thus, automatic extraction of translation resources is relevant.

On this document we describe a simple methodology for the extraction of parallel terminology entries (candidates) from parallel corpora using translation patterns and probabilistic translation dictionaries.

Translation patterns describe how a sequence (pattern) of words change order during translation. The patterns are described on a Domain Specific Language (DSL) named Pattern Description Language (PDL), that is formalized on section 3.

These patterns are matched against an alignment matrix where translation probabilities between words were defined using a probabilistic translation dictionary (Simões, 2004). Section 2 explain what these dictionaries are and how we can obtain them.

Each time one of the defined patterns match on the alignment matrix, the pair of sequence of words is extracted. To this pair, we associate the rule identifier so we can infer further information from it. Section 4 shows some of the defined rules, some of the extracted pairs of sequences and an evaluation as terminology candidates.

At the end, we present some remarks about the method efficiency, the results obtained and future directions and uses for this bilingual terminology.

ISSN: 1135-5948

2 Probabilistic Translation Dictionaries

One of the most important resources for MT is translation dictionaries. They are indispensable, as they establish relationships between the language atoms: words. Unfortunately, freely available translation dictionaries have small coverage and, for minority languages, are quite rare. Thus, it is crucial to have an automated method for the extraction of word relationships.

(Simões and Almeida, 2003) explains how a probabilistic *word alignment* algorithm can be used for the automatic extraction of probabilistic translation dictionaries. This process relies on sentence-aligned parallel corpora.

The used algorithm is language independent, and thus, can be applied to any language pair. Experiments were done using diverse languages, from Portuguese, English, French, German, Greek, Hebrew and Latin (Simões, 2008). The algorithm is based on word co-occurrences and its analysis with statistical methods. The result is a probabilistic dictionary, associating words on two languages.

These dictionaries map words from a source language to a set of associated words (probable translations) in the target language. Given that the alignment matrix is not symmetric, the process extracts two dictionaries: from source to target language and vice-versa.

The formal specification for one probabilistic translation dictionary (PTD) can be defined as:

$$w_{\mathcal{A}} \mapsto (occs\,(w_{\mathcal{A}}) \times w_{\mathcal{B}} \mapsto \mathcal{P}(\mathcal{T}(w_{\mathcal{A}}) = w_{\mathcal{B}}))$$

Figure 1 shows two entries from the English:Portuguese dictionary extracted from the EuroParl(Koehn, 2002) corpus. Note that these dictionaries include the number of occurrences of the word on the source corpus, and a probability measure for each possible translation.

Regarding these dictionaries it should be noted that, although we use the term translation dictionaries, not all word relationships on the dictionary are real translations. This is mainly explained by the translation freedom, multi-word terms and a variety of linguistic phenomena.

Notwithstanding the probabilistic nature of these dictionaries, there is work on bootstrapping conventional translation dictionaries using probabilistic translation dictionaries (Guinovart and Fontenla, 2005). Also, (Santos and Simões, 2008) discusses the connection between dictionaries quality and corpora genre and languages.

$$europe \rightharpoonup 42583 \times \begin{cases} europa & 94.7\% \\ europeus & 3.4\% \\ europeu & 0.8\% \\ europeia & 0.1\% \end{cases}$$

$$stupid \rightharpoonup 180 \times \begin{cases} estúpido & 47.6\% \\ estúpida & 11.0\% \\ estúpidos & 7.4\% \\ avisada & 5.6\% \\ direita & 5.6\% \end{cases}$$

Figure 1: Probabilistic Translation Dictionary examples.

3 Pattern Description Language

This section presents the Pattern Description Language (PDL), a DSL used to describe translation patterns. It starts with a simple explanation of PDL syntax and how patterns are used to extract terminology candidates. Follows a section on pattern predicates, constraints on the applicability of the defined patterns.

3.1 PDL basics

The translation patterns defined with PDL are matched against a translation matrix. Each cell of this matrix contains the mutual translation probability between the words on that line and column. For instance, on the example of figure 2, words "discussion" and "discussão" have a mutual translation probability of 44%. This mutual translation probability is computed using a probabilistic translation dictionary[1].

Figure 2 includes some cells highlighted. These are anchor cells: cells which translation probability is 20% higher than the remaining probabilities in the same row and/or column.

As can be seen on the translation matrix shown, although it includes an optimal

[1]Note that there is no restriction on the corpus from which the PTD is created. It is possible and desirable to invest in a big and high quality PTD that is used to extract terminology candidates from diverse parallel corpora.

	discussion	about	alternative	sources	of	financing	for	the	european	radical	alliance	.
discussão	**44**	0	0	0	0	0	0	0	0	0	0	0
sobre	0	**11**	0	0	0	0	0	0	0	0	0	0
fontes	0	0	0	**74**	0	0	0	0	0	0	0	0
de	0	3	0	0	**27**	0	6	3	0	0	0	0
financiamento	0	0	0	0	0	**56**	0	0	0	0	0	0
alternativas	0	0	**23**	0	0	0	0	0	0	0	0	0
para	0	0	0	0	0	0	**28**	0	0	0	0	0
a	0	1	0	0	1	0	4	**33**	0	0	0	0
aliança	0	0	0	0	0	0	0	0	0	0	**65**	0
radical	0	0	0	0	0	0	0	0	0	**80**	0	0
europeia	0	0	0	0	0	0	0	0	**59**	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	**80**

Figure 2: Example of a Translation matrix.

	discussion	about	alternative	sources	of	financing	for	the	european	radical	alliance	.
discussão	**44**	0	0	0	0	0	0	0	0	0	0	0
sobre	0	**11**	0	0	0	0	0	0	0	0	0	0
fontes	0	0	0	**74**	0	0	0	0	0	0	0	0
de	0	3	0	0	**27**	0	6	3	0	0	0	0
financiamento	0	0	0	0	0	**56**	0	0	0	0	0	0
alternativas	0	0	**23**	0	0	0	0	0	0	0	0	0
para	0	0	0	0	0	0	**28**	0	0	0	0	0
a	0	1	0	0	1	0	4	**33**	0	0	0	0
aliança	0	0	0	0	0	0	0	0	0	0	**65**	0
radical	0	0	0	0	0	0	0	0	0	**80**	0	0
europeia	0	0	0	0	0	0	0	0	**59**	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	**80**

Figure 3: Translation matrix with detected patterns.

translation, the alignment includes word order changes. Also, these word changes are not related to the translator will. They are imperious given that involved languages syntaxes. As an example, consider the relative positioning changes between nouns and adjectives during a Portuguese to English translation. In Portuguese the noun appears before ("*gato gordo*"), while in English it is at the end ("*fat cat*").

Although language dependent, these changes can be predicted, and thus, it is possible to describe them mathematically:

$$\mathcal{T}(N \cdot A) = \mathcal{T}(A) \cdot \mathcal{T}(N)$$

PDL is a domain specific language designed for the formal description of these rules (and their applicability constraints). The pattern for the simple rule shown above is schematized on figure 1.

	Olympic	Games
Jogos		X
Olímpicos	X	

```
[ABBA] A B = B A
```

Table 1: $\mathcal{T}(A\ B) = \mathcal{T}(B)\ \mathcal{T}(A)$ Pattern.

The PDL syntax is interpreted as follows:

- between rectangular braces is the identifier of the rule. It can be any valid identifier. We normally use identifiers that helps us remembering a specific case where the rule matches.
- follows a sequence of variables (placeholders for words) or specific words (as on table 4). This sequence is matched against words on the source language;
- at the right hand of the equal sign there is another sequence, with the same variables, but in the order of the translation.

Tables 2 to 5 shows some (other) typical Portuguese/English patterns. Each table includes the rule in PDL notation, and a graphical representation of it. To understand this matrix representation, consider the following:

- an X in a cell means that it will match against an anchor cell in the translation matrix;
- empty cells will be matched against cells with low values (non anchor cells);
- cells marked with a *Delta* symbol (as the one used on table 3) will match with any probability at all (being it an anchor cell or not). This type of relation is quite important because it is difficult to predict probabilities between some type of word classes, like articles or prepositions.

These patterns are applied directly in the alignment matrix, layering the pattern through it until it matches. Figure 3 shows the previous translation matrix from figure 2 with the detected patterns marked.

The word sequences that are related to the marked patterns are extracted, and result in the following nominals:

```
S: fontes de financiamento alternativas
T: alternative sources of financing

S: aliança radical europeia
T: european radical alliance
```

	Human	Rights
Direitos		X
do		
Homem	X	

```
[HR] A "de" B = B A
```

Table 2: HR Pattern

	ponto	de	vista	neutro
neutral				X
point	X			
of		Δ		
view			X	

```
[POV] P "de" V N = N P "of" V
```

Table 3: POV Pattern.

	protocolo	de	transferência	de	ficheiros
file					X
transfer			X		
protocol	X				

```
[FTP] P "de" T "de" F = F T P
```

Table 4: FTP Pattern.

	índice	de	desenvolvimento	humano
human				X
development			X	
index	X			

```
[HDI] I "de" D H = H D I
```

Table 5: HDI Pattern.

As described earlier, patterns are built not just from variables. Examples on tables 2 to 5 show patterns with specific words. The semantic of these words is exactly the expected: the pattern matches if the column or row includes that word.

Some languages, like Portuguese, have a rich gender and number flexion. Rules with specific words should take care of all possible flexions. To make this task easier, not forcing the repetition of rules for different flexions, it is possible to specify a sequence of alternating words ("*or'ed*" words), as in the following example:

```
[HDI] I "do"|"da"|"dos"|"das" D H = H D I
```

This small language makes it possible to define quickly and in an easy to read syntax almost any kind of translation rule.

3.2 Conditional Patterns

Patterns might be applied to word order changes that are not really noun phrases (and thus, not terminology candidates). For instance,

- in Portuguese the conjunction is used after the comma ("*, e*") while in English it is used before ("*and ,*"). Without conditional patterns, ABBA pattern would be applied;
- another example for the Portuguese/English pair is the verb negation: "*não é*" instead of "*is not.*"

To solve these problems, PDL supports pattern predicates to restrict their applicability. PDL supports two types of restrictions:

- generic predicates, that are written in a programming language (Perl) and can do almost anything;
- morphological conditions, that are written in PDL and use a morphological analyzer to test their applicability.

3.2.1 Generic predicates

The most powerful way to add restrictions to translation patterns is by the definition of a generic function, written in a specific programming language, that validates the applicability of the pattern for those specific words.

Given that PDL is implemented in Perl, and Perl is a reflexive language, generic predicates are written in Perl. These predicates predicates receive the word or words that should be validated, and return a boolean value, on wether the pattern should or not be applied.

One of the main advantages of writing predicates in Perl is the ability to perform external actions. It is easy to apply a regular expression, to query a morphological analyzer tool, perform concordancies or queries on a relational database, or yet a query on an Internet search engine.

These predicates are defined as Perl functions in the same file as the translation patterns. There is a Perl code zone where the user can write their own functions (that can be used as predicates or auxiliary code for other functions). These functions are called each time a pattern might match against the translation pattern.

As a simples example, for illustration purposes only, consider the following code to validate if we are not matching the `A B = B A` pattern with commas:

```
[ABBA] A B.notComma = B.notComma A

%%

sub notComma {
    my $word = shift;
    return $word ne ","
}
```

Note that the variables on the rule have attached the name of the predicate (as a method on an OO language), and that the code section is separated from the main rules section with a sequence of special characters. During processing time, these functions are parsed and it is constructed a symbol table that is used later when evaluating patterns applicability.

3.2.2 Morphologic restrictions

The most usual predicate code is the morphological analysis of a word, checking for a specific morphological category, genre or number. To help on the definition of this kind of predicates, PDL supports morphological restrictions directly on its syntax:

```
[ABBA]  A B[CAT<-adj] = B[CAT<-adj] A
```

This example means that B, on both languages, should be analyzed morphologically and that the rule will be applied only if the words can[2] be analyzed as adjectives.

To perform this validation the algorithm uses the Jspell[3] morphological analyzer (Almeida and Pinto, 1994).

3.2.3 Inference rules

These are not restriction rules, but infer rules: if the pattern is applied, then we can infer something about the words that matched. The syntax is very close to morphologic restrictions, just changing the direction of the arrow. Suppose we do not have a morphological analyzer for the Portuguese language, but we have for the English language. We can write down a rule to infer a rough morphological analyzer for the Portuguese language:

```
[ABBA] A[CAT->noun] B[CAT->adj] =
       B[CAT<-adj]  A[CAT<-noun]
```

Although the result will include some false positives, this is a fast way to help inferring properties from languages.

4 Bilingual Terminology Extraction and Evaluation

The entries extracted using translation patterns are mostly nominal phrases. Although a lot of the extracted nominal phrases are not terminology in its common sense definition, they can be easily filtered. At the moment we are not interested in this filtering task, as we are not using them to the creation of glossaries, but to use them directly on translation systems.

These nominal phrases are counted, and multi-sets are created. These multi-sets elements include the identifier of the rule, and the nominal phrase on both languages. Figure 6 shows some examples (the top occurring, and the least occurring) of the extracted nominal phrases, together with their occurrence counter. A quick look on the examples show that the overall quality of these multi-sets is quite good.

[2] In case of ambiguity we check if there is at least one categories accordingly with the restriction.

[3] Jspell is actually being released as an hybrid Perl module, and is available on CPAN as Lingua::Jspell.

occs	patt	Portuguese	English
39214	ABBA	comunidades europeias	european communities
32850	ABBA	jornal oficial	official journal
32832	ABBA	parlamento europeu	european parliament
32730	ABBA	união europeia	european union
15602	ABBA	países terceiros	third countries
1	ABBA	órgãos orçamentais	budgetary organs
1	ABBA	órgãos relevantes	relevant bodies
1	HR	óvulos de equino	equine ova
1	HR	óxido de cádmio	cadmium oxide
1	HR	óxido de estireno	styrene oxide

Table 6: Nominals multisets.

We performed a quick evaluation for six different patterns, evaluating their fertility and translation quality. Were used 700 000 translation units from EuroParl. These translation units were processed, and extracted 578 103 different occurrences. After consolidation, it were created 139 781 different multi-sets. These multi-sets were filtered and removed entries with punctuation, stop words and random noise. These filtering resulted in 103 617 multi-sets.

There were 578 103 pattern occurrences that were consolidated in 139 781 multisets (different patterns). We performed some filtering removing entries with punctuation, stop words and noise, resulting in 102 151 patterns. Table 7 shows the distribution for the six different patterns: number of occurrences and precision of the obtained nominal phrases.

pattern	occs.	prec.
A B = B A	77497	86
A *de* B = B A	12694	95
A B C = C B A	7700	93
I *de* D H = H D I	3336	100
P *de* V N = N P *of* V	564	98
P *de* T *de* F = F T P	360	96

Table 7: Evaluation of nominal phrases.

For the evaluation we took 20 nominal phrases from the top occurring one, 20 from the less occurring, and 20 from the middle of the list (median). As most of the entries have just one occurrence, the 20 nominal phrases from the middle of the list have a low occurrence count (normally 1 or 2 occurrences). Thus, this was a really unfavourable test (2/3 of the nominal phrases have a low occurrence count).

5 Terminology Generalization

Although terminology extraction is an important task, if we want to use these resources for machine translation, they need to be the more generic possible, so that they can be applied to different situations.

Generalization (Brown, 2001) is the common approach to make translation examples (where bilingual nominal phrases are a specific case) have a wider range of application on translation. This section shows two ways to generalize examples:

- a simple approach based on non-word classes;
- the creation of word classes based on alignment patterns;

5.1 Numeric classes

The easiest way to generalize translation examples is to detect non-words, and associate a class to them. This class can be anything, from numbers, years, emails or urls: any non-textual object that is easy to detect.

The nominal phrases are parsed and these non-textual objects are detected, being replaced by the class name. For instance, if we define a class named *year* for numbers $x : 1900 \leq x \leq 2200$, and another class named *int* for any other integer value, it is possible to extract the following generalized terminology entries:

```
187 orçamento de {year}   {year} budget
136 {int} euros           eur {int}
127 directiva de {year}   {year} directive
 51 orçamento {year}      {year} budget
 46 {int} de setembro     september {int}
```

At the moment we are using about ten non-word classes including numbers, years,

hours, time periods, cardinals and currency values.

5.2 Word Classes

Another type of generalization is the substitution of a word by a member of a specific set. For instance, we can define a set with words for gentilics, like:

$$G = \{Nigerian, Mexican, Norwegian, \ldots\}$$

and then have a dictionary map each one of the words from this set to the respective translation. After the creation of these classes, it is possible to use the class identifier in the nominal phrases, with:

$$X \text{ People} \Rightarrow \text{Povo } \mathcal{T}(X) \quad X \in G$$

To construct these word classes we used our translation patterns.

Consider the pattern "A B = B A" from Portuguese to English. We are expecting that words matching the variable B are adjectives. If we chose a specific noun on A, we will get all the adjectives that are likely to be applied to it (as can be seen on figure 4). This kind of approach is a quick and efficient way to construct bilingual word classes and generalize terminology entries.

```
ácido => clorídrico (hydrochloric acid)
         sulfúrico  (sulphuric acid)
         acético    (acetic acid)
         fólico     (folic acid)
         cítrico    (citric acid)
         ...
livro => verde      (green paper)
         branco     (white paper)
         azul       (blue paper)
         laranja    (orange book)
         vermelho   (red book)
         azul       (blue book)
         ...
```

Figure 4: Automatic word class creation.

Unfortunately these classes should be analyzed manually before being used, specially in the case where the translation varies not only on the word being cycled but also on the fixed word. Check for instance the second example of figure 4, where the word "*livro*" can be translated by "*paper*" or "*book*." This would be pacific if there wasn't two different translations for the same noun phrase: "*livro azul*." Given this ambiguity, some care should be taken on extracting word using this approach.

6 Conclusions and Future Work

Using statistical methods to obtain bilingual resources is possible. If we attach scalable tools to the statistical methods, resources quality can raise.

Translation patterns shown to be an interesting and efficient method for noun phrases extraction. Further evaluation should be done on counting how many of the extracted noun phrases can be considered real terminology. For the time being that evaluation is not really important as the noun phrases usage for machine translation is equally effective for both terminology and non-terminology entries.

The PDL language makes it possible to define translation patterns in a concise way, without discarding the readability. Also, the possibility of adding constraints to the patterns applicability make them even more effective, raising the quality of the extracted nominal phrases.

At the moment we are applying these resources to two different systems:

- using `Text::Translator`, a Perl system to prototype translation systems. This Perl module is quite versatile as we can use all translation approaches on it, from statistical to example based or even rule based translation;
- using the extracted noun phrases to prepare translation resources for Apertium (Armentano-Oller et al., 2006; Loinaz et al., 2006), the well known machine translation system for close languages.

The extracted noun phrases are being useful not just as translation examples (Way, 2001), but also as the source for other resources extraction, like the construction of bilingual translation dictionaries.

While there is work on pattern extraction from corpora (Och and Ney, 2004), they are not used for bilingual resources extraction. Instead, Och and Ney use parallel corpora to infer translation templates that are used later on machine translation. This approach can be used to bootstrap our translation patterns. In any case, they must be manually reviewed before being applied for the real extraction of nominal phrases.

The tools used are available as Open-Source and can be easily downloaded from the Internet at `http://natools.sf.net`. They rely on the NatServer translation resources server (Simões and Almeida, 2006) for querying efficiency.

Acknowledgments

Alberto Simões has a scholarship from Fundação para a Computação Científica Nacional and the work reported here has been partially funded by Fundação para a Ciência e Tecnologia through project POSI/PLP-/43931/2001, co-financed by POSI, and by POSC project POSC/339/1.3/C/NAC.

References

Almeida, José João and Ulisses Pinto. 1994. Jspell – um módulo para análise léxica genérica de linguagem natural. In *Actas do X Encontro da Associação Portuguesa de Linguística*, pages 1–15, Évora.

Armentano-Oller, Carme, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A. Scalco. 2006. Open-source portuguese-spanish machine translation. In *7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006*, pages 50–59, Itatiaia, Rio de Janeiro, Brazil, May.

Brown, Ralf D. 2001. Transfer-rule induction for example-based translation. In Michael Carl and Andy Way, editors, *Workshop on Example-Based Machine Translation*, pages 1–11, September.

Guinovart, Xavier Gómez and Elena Sacau Fontenla. 2005. Técnicas para o desenvolvemento de dicionarios de tradución a partir de córpora aplicadas na xeración do Dicionario CLUVI Inglés-Galego. *Viceversa: Revista Galega de Traducción*, 11:159–171.

Koehn, Philipp. 2002. EuroParl: a multilingual corpus for evaluation of machine translation. Draft.

Loinaz, Iñaki Alegría, Iñaki Arantzabal, Mikel L. Forcada, Xavier Gómez Guinovart, Lluis Padró, José Ramom Pichel Campos, and Josu Waliño. 2006. OpenTrad: Traducción automática de código abierto para las lenguas del Estado español. *Procesamiento del Lenguaje Natural*, 37:357–358.

Och, Franz Josef and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.

Santos, Diana and Alberto Simões. 2008. Portuguese-English word alignment: some experiments. In *LREC 2008 — The 6th edition of the Language Resources and Evaluation Conference*, Marrakech, 28–30, May.

Simões, Alberto and J. João Almeida. 2006. NatServer: a client-server architecture for building parallel corpora applications. *Procesamiento del Lenguaje Natural*, 37:91–97, September.

Simões, Alberto M. and J. João Almeida. 2003. NATools – a statistical word aligner workbench. *Procesamiento del Lenguaje Natural*, 31:217–224, September.

Simões, Alberto Manuel Brandão. 2004. Parallel corpora word alignment and applications. Master's thesis, Escola de Engenharia - Universidade do Minho.

Simões, Alberto Manuel Brandão. 2008. *Extracção de Recursos de Tradução com base em Dicionários Probabilísticos de Tradução.* Ph.D. thesis, Escola de Engenharia, Universidade do Minho, Braga, May.

Tiedemann, Jörg. 2003. *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing.* Ph.D. thesis, Studia Linguistica Upsaliensia 1.

Way, Andy. 2001. Translating with examples. In Michael Carl and Andy Way, editors, *Workshop on Example-Based Machine Translation*, pages 66–80, September.