Edited by
Marek Miłosz
Ualsher Tukeyev

Varia Informatica 2017

Varia Informatica 2017

EDITED BY
MAREK MIŁOSZ
UALSHER TUKEYEV

PIPS Polish Information Processing Society
Lublin 2017

Editors

Marek Miłosz
Ualsher Tukeyev

Reviewers

Sergio Lujan Mora (Alicante University, Alicante, Spain)
Stanisław Grzegórski (Lublin University of Technology, Lublin, Poland)

ISBN 978-83-936692-3-3

Table of Contents

2

The Structural Transformation of Sentences for the Kazakh-Russian and Kazakh-English Language Pairs in Machine Translation System

This work presents developed models, algorithms and programs for structural transformations of sentences for machine translation systems for the Kazakh-Russian and Kazakh-English language pairs. A model and a method for automatically generating structural rules for converting sentences have also been developed. Practical results applied to the free/open-source platform of Apertium machine translation system are presented.

2.1. INTRODUCTION

Active integration of Kazakhstan into the world community and the increasing volume of information flows between our country and its foreign partners, a real need for different segments of the population in the operational computer translation while working on the Internet determines the urgency of questions of machine (computer) translation of the Kazakh language into various leading world languages, such English, Russian, French, German, and most recently, Chinese, as well as reverse machine translation. The primary tasks for the information interaction of the population of Kazakhstan with foreign partners and within the country are defined by interactions in three languages: Kazakh, English and Russian. In this regard, highly effective instrumental support for machine translation of such a trilingual language interaction is highly relevant. In connection with what is relevant are the research and development of machine translation systems of industrial quality from Kazakh to English and Russian and vice versa.

Kazakh language, as one of Turkic languages, belongs to an agglutinative language, and it uses vowel harmony. Which means that translating text in Kazakh language into other languages with simpler morphology, such as English or Russian languages, with, for instance, statistical machine translation (SMT), will cause some quality loss because of morphological segmentation.

Statistical methods and rules-based methods are complementary approaches in machine translation (MT), which have different strengths and weaknesses. This complementarity emerged as a result of growing interest in hybrid systems combining statistical data analysis and linguistic approaches. A hybrid approach is a convenient solution in the case when the number of bilingual resources available for a particular pair of languages that are not big enough to use them for preparing a competitive statistical system of MT, and for creating a rule-based MT system, not enough money and time for its sustainable development.

Therefore, automatic generation of structural translation rules based on small parallel corpora with further integration into rule-based MT system is a method that helps to solve the above problem in less time and more efficiently. This method avoids the need to manually write these rules by a person.

The question of the sentences structural transfer in the rule-based machine translation (RBMT) can divided into two groups: the syntax structural transfer and the transfer of word's morphological structure into a phrase syntax structure. The second group of transfers usually occur when made machine translation of languages with complex morphology into languages with a simple morphology, for example, the Kazakh language into Russian or English. In this case, some source language word's morphological structures transformed into target language phrase's syntactic structure. This second group of structural transfers is called as "morphological chunk transfers".

The question of automatic inferring of the structural rules of machine translation from one language to another are rather actual for machine translation systems based

on grammatical rules (RBMT). This is due to the time-consuming process of drawing up the rules for RBMT.

Rule-based machine translation of natural language nearly always contains the following steps [1] morphological analysis, part-of-speech (POS) tagging, translating words into target language, execution of syntactic transformations and division into phrases (or chunks), generating new lexical forms (word's lemmas with lexical categories) of target language words. In rule-based MT systems, most of these stages are implemented by handwritten translation rules. The process of creating the handwritten rules is very laborious process. Therefore, very actual is automatic extracting of translation rules from bilingual corpora.

2.2. EXTRACTING CHUNKER TRANSLATION RULES FROM PARALLEL CORPORA

Word alignment for bilingual pairs and extracting structural transformation rules is one of the main steps of this method. The alignment pattern was first introduced in statistical machine translation [2] as one of the special functions of the maximum entropy model.

A hidden Markov model alignment (HMM). The alignment $\Pr(f_1^J, a_1^J | e_1^I)$ can be described by the following model:

$$\Pr(f_1^J, a_1^J | e_1^I) = \Pr(J|e_1^I) \cdot \prod_{j=1}^{J} \Pr(f_j, a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) =$$
$$\Pr(J|e_1^I) \cdot \prod_{j=1}^{J} \Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \cdot \Pr(f_j | f_1^{j-1}, a_1^{j}, e_1^I) \quad (2.1)$$

Using the decomposition, there can be obtained three different probabilities: the probability of the length $\Pr(J|e_1^I)$, the probability of aligning $\Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I)$, and the probability of lexicon (words) $\Pr(f_j | f_1^{j-1}, a_1^{j}, e_1^I)$. In a hidden Markov model alignment is assumed that the dependence of the first order to align a_j, and the probability of the lexicon(words) depends only on the words in the position of a_j:

$$\Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) = p(a_j | a_{j-1}, I) \quad (2.2)$$

$$\Pr(f_j|f_1^{j-1}, a_1^j, e_1^I) = p\left(f_j|e_{a_j}\right) \tag{2.3}$$

Now, if everything will be put together, there can be got a simple model of length: $\Pr(J|e_1^I) = p(J|I)$. The following basic decomposition $p(f_1^J|e_1^I)$ is based on HMM:

$$p(f_1^J|e_1^I) = p(J|I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} [p(a_j|a_{j-1}, I) \cdot p\left(f_j|e_{a_j}\right)] \tag{2.4}$$

with a probability of alignment $p(i|i', I)$ and translation probability $p(f|e)$. In order to make the alignment parameters independent of absolute word positions, it is assumed that the probability of alignment $p(i|i', I)$ depends only on the width of the shift $(i - i')$. Using non-negative parameters $\{c(i - i')\}$, there can be got the alignment probabilities from the form:

$$p(i|i', I) = \frac{c(i-i')}{\sum_{i''=1}^{I} c(i''-i')} \tag{2.5}$$

This form ensures that the alignment probabilities satisfy the normalization of constraints for each conditional position of the word i', $i' = 1, \ldots, I$.

The original formalization of the alignment model based on the hidden Markov models does not create an empty word that allows to generate the word of the source language that does not have a straight-aligned word of the target language. The model was extended by adding the I empty word e_{I+1}^{2I}. The target word e_i has the corresponding empty word e_{i+I} (the position of the empty word is encoded by the previous target word). The following restrictions on transitions in the HMM network $(i \leq I, i' \leq I)$ with the participation of empty words e_0^I are implemented:

$$p(i + I|i', I) = p_0 \cdot \delta(i, i') \tag{2.6}$$

$$p(i + I|i' + I, I) = p_0 \cdot \delta(i, i') \tag{2.7}$$

$$p(i|i' + I, I) = p(i|i', I) \tag{2.8}$$

The parameter p_0 expresses the probability of an empty word transition, which should be optimized by the information from the data.

The HMM is based on the first-order dependence $p(i = a_j|a_{j-1}, I)$ for the alignment distribution. IBM models 1 and 2 use zero-order relations $p(i = a_j|j, I, J)$.

Model 1 uses the uniform distribution $p(i|j, I, J) = 1/(I + 1)$:

$$\Pr(f_1^J, a_1^J|e_1^I) = \frac{p(J|I)}{(I+1)^J} \cdot \prod_{j=1}^{J} p\left(f_j|e_{a_j}\right) \tag{2.9}$$

Therefore, the order of words will not affect the probability of alignment.

From **Model 2** it is got:

$$\Pr(f_1^J, a_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} [p(a_j|j, I, J) \cdot p\left(f_j|e_{a_j}\right)] \tag{2.10}$$

To reduce the number of alignment parameters, the dependence on J in the alignment model is ignored and the distribution $p(a_j|j, I)$ is used instead of $p(a_j|j, I, J)$.

In the first approach of inferring transfer rules will be used the method described by Sánchez-Cartagena et al. (2015) [3], which was inspired by the work of Sánchez-Martínez and Forcada (2009) [4] where alignment templates were also considered for structural transfer rule inference.

As a result, the calculation of word alignment consists of the following steps:

1. Training IBM 1 [5] in 5 iterations. In this model, the word order does not affect the alignment probabilities.
2. Training model alignment HMM [6] in 5 iterations. This alignment model has the property of calculating the probabilities of alignment, depending on the alignment position of the previous word.
3. Training the IBM 3 model for 5 iterations. In this model, the probability of alignment depends on the position of the aligned words and on the length of sentences in the source and target language. In addition, IBM model 3 also takes into account fertility levels. The productivity of a word is determined by the number of words aligned with it in another language.
4. Training the IBM 4 model in 5 iterations. This model is identical to IBM 3 model, except that the model reorders phrases that can be moved as separate units.

To increase the level of alignment [6] and use alignments for the structural transfer phase in the machine translation, before alignment the text is transformed into a transient representation using morphological analysis and determining the part of speech. Unlike the tagging method [7], the Apertium machine translation system was used for the morphological analysis and the tagger of parts of speech.

2.3. "Chunking" Rules For The Apertium Platform

As can be seen in Fig. 2.1, the Apertium modules consist of different stages, and the translation from the source language to the target language. And each stage performs certain work.

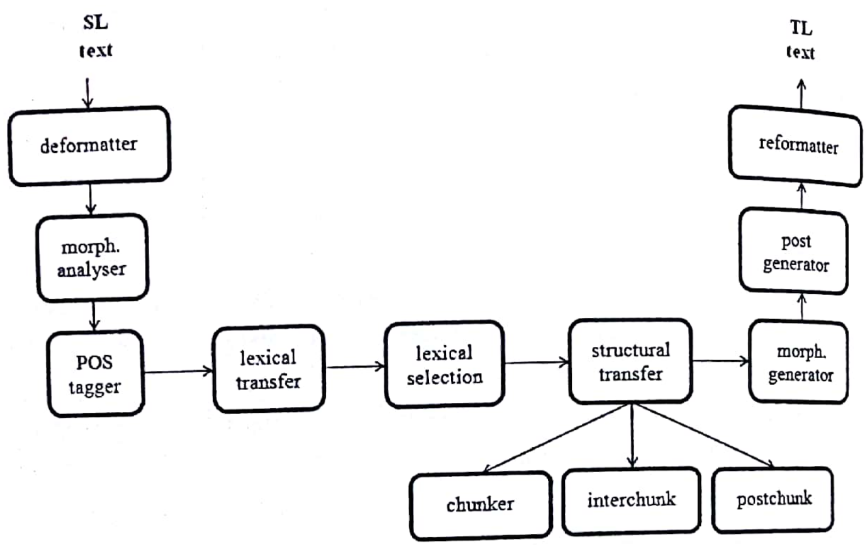

Fig. 2.1. The modular architecture of the Apertium MT platform

Source: own elaboration

A module of *structural transfer* consisting of three sub-modules:

- The chunker module rules match words by gender and number in phrases. By the rules of chunker different types of phrases are translated, for instance, from Kazakh into Russian (and vice versa): NP (noun phrases: книга ученика - оқушының кітабы), VP (verb phrases: я играю – мен ойнаймын), AdjP (очень умный – өте ақылды), AdvP (апта сайын - еженедельно), NumP (двадцать один – жиырма бір). This module takes into account endings of words in phrases (Fig. 2.2).
- Interchunk rules are used to connect chunks. Interchunk module rules translate phrases that consist of more than 4 words. This module takes into account the order of words when translating phrases and sentences. For example: «мен балалармен аулада ойнаймын – я с детьми в дворе играю».
- Postchunk is used for internal fix after using interchunk rules.

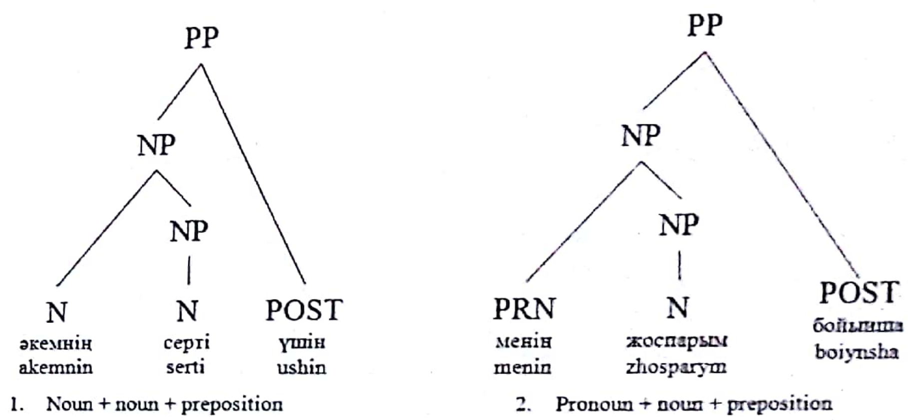

Fig. 2.2. Example of chunk model consisting of three words for prepositional phrases

Source: own elaboration

Based on the structure of the Kazakh and Russian language proposals, the following types of phrases were implemented:

1. A noun phrase – is chunk that is created from one part of speech together with one or more nouns.
2. A prepositional phrase – is chunk that is created from one part of speech together with a preposition.
3. An adjective phrase – is chunk that is created from the adjective and the degree of the adjective.

4. A verb phrase – is chunk that is created from the verb and the types of verb.
5. An adverb phrase – is chunk that is created from the verb together with the adverb.

2.4. EXTRACTING "CHUNKER" RULES FROM CORPORA

The previous method used by S'anchez-Mart'ınez and Forcada (2009) needs a human-made set of lexical units [4]. This set consists of two lexical forms from the target and the initial language (usually the corresponding closed lexical categories) participating in lexical changes and which should not be generalized.

However, this new approach overcomes the main limitations of that by Sánchez-Martínez and Forcada (2009). This method consists of following steps:

1. Obtaining lexical forms by converting the two sides of the parallel corpus into an intermediate representation using the Apertium machine translation system. Intermediate representation consists of lexical forms of words from the corpus. For example, the lexical form w, for example, garden N-gen: ε.num: sg.case: nom, consists of:

- lemmas $\lambda(w)$, that is $\lambda(w)$=garden,
- Lexical category $\rho(w)$, $\rho(w) = \mathrm{N(noun)}$
- Set of attributes of morphological inflexions $a(w)$, $a(w) = \{\mathrm{gender, num, case}\}(\mathrm{gender, number\ and\ case})$
- Attribute value $v(w,a), v(w,\mathrm{num}) = sg(\mathrm{singular})$.

Some morphological attributes may not be assigned, and denoted by an empty symbol ε, for example, there are no nouns in the English language, so the value will be as follows: v (w, gen) = ε.

The lexical form of the source language is translated into the target language using the Apertium's bilingual dictionary. One word can have several translations, in this case the rules of the constraint grammar (Constraint Grammar - CG) [8] or the speech element tagger based on the Hidden Markov Models (HMM), for solving the morphological polysemy, and the lexical selection rule for solving lexical ambiguity [9].

2. For alignment, IBM models 1, 3 and 4 [6] and the HMM equalization model [7] are used for the 5-iteration implemented in the Giza++ program for two translation directions [10].

3. Calculation of the Viterbi alignment, according to the models for the two directions of translation.

4. Synchronization of two sets of Viterbi alignments by finding the intersections by the method of Och and Ney (2003) [10] to obtain word-aligned pairs of sentences.

5. Extract bilingual phrases corresponding to these alignments [11, 12] (Fig. 2.3).

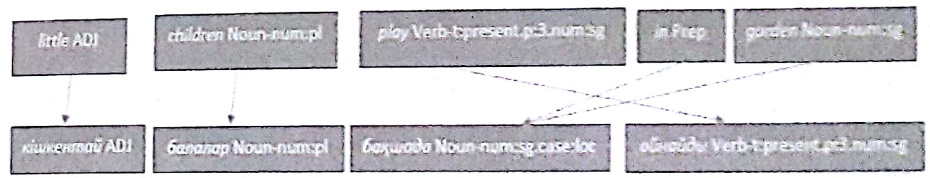

Fig. 2.3. English-Kazakh bilingual phrase pairs

Source: own elaboration

6. Extract generalized aligned templates (GAT). Summarizing the morphological information and constraints from each class of words of the source language and applying them for the target language classes, GAT z = β (p) is generated from each bilingual phrase p. Also information about alignment of the bilingual pair A 'is copied into the alignment of GAT A: A ← A'. Fig. 2.4 illustrates how the GAT z is extracted from the bilingual phrase p.

Limitations of r are added by checking each lexical form of the source language from the bilingual dictionary as follows:

$$\forall w_i \in W, \alpha(r_i) \leftarrow \alpha(w_i), \tag{2.11}$$

$$\forall r_i, \forall a \in \alpha(r_i), v(r_i, a) \leftarrow v(\tau(w_i), a). \tag{2.12}$$