Automatic Detection of the Type of “Chunks” in Extracting Chunker Translation Rules from Parallel Corpora

Aida Sundetova[1], Ualsher Tukeyev[1]

[1]Al-Farabi Kazakh National University, Research Institute of Mechanics and Mathematics,
Al-Farabi av., 71, 050040 Almaty, Kazakhstan
{sun27aida, ualsher.tukeyev}@gmail.com

Abstract. This paper describes the method of the automatic detection of the type of “chunks” which are generated in methodology presented by Sánchez-Cartagena et. al. (Computer Speech & Language 32:1(2015) 46–90). The proposed automatic detection method type of “chunks” improves above methodology of extracting grammatical translation rules from bilingual corpora. Proposed improvement of methodology of extracting grammatical translation rules from corpora allows to apply output phrases of extracted translation “chunk” rules for next “interchunk” stage in machine translation system and improve of machine translation quality. Experiments are done for the English–Kazakh[1] language pair using the free/open-source rule-based machine translation (MT) platform Apertium and bilingual English–Kazakh corpora.

Keywords: rules extraction**,** machine translation, Apertium, transfer rules, chunks.

1 Introduction

Rule-based machine translation (MT) 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.

This paper presents the automatic detection method of the type of “chunks” rules, obtained by using the methodology automatic extracting of translation rules from bilingual corpora by Sánchez-Cartagena et al. (2015) [2], which is described in the following section. Their method requires to create tag groups and tag sequences for new pair and

[1] https://svn.code.sf.net/p/apertium/svn/staging/apertium-eng-kaz

tuning of the extraction script by declaring monolingual dictionary, bilingual dictionary, and bilingual corpora.

2 “Chunking” Rules for the Apertium Platform

The Apertium free/open-source rule-based shallow-transfer MT platform [3] includes the following modules: de-formatter, morphological analyzer, POS disambiguator, structural transfer, morphological generator, post-generator, re-formatter.

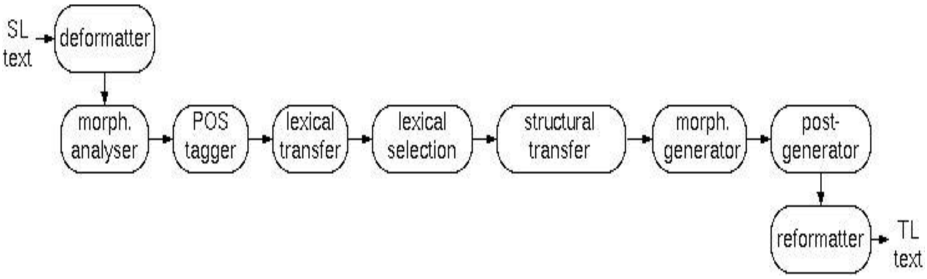

Fig. 1. Structure of the Apertium platform

Three stage structural transfer rules on Apertium platform is implemented by three stages, it follows the description in [4]:

1. A first stage of transformations (“chunker”) detects source language (SL) lexical form (LF) patterns and generates the appropriate sequences of target language (TL) LFs, which will be grouped in chunks representing simple constituents such as noun phrases, prepositional phrases, etc. These chunks bear tags that may be used for interchunk processing.
2. The second round (“interchunk”) reads patterns of chunks and produces a new sequence of chunks. This is the module where one can attempt to perform some longer-range reordering operations, interchunk agreement (for example, between noun and verb pharse, agreement in number and person), case selection, etc.
3. The third round (“postchunk”) transfers chunk-level tags to the lexical forms they contain and whose lexical-form-level tags are linked (through a referencing system) to chunk-level tags.

Structural transfer for English–Kazakh has an additional clean-up stage to remove tags.

Usually, three stage transfer uses different type of phrases, which helps to apply rules for concrete structures from stage to stage. For example, the current version of English–Kazakh MT system, for which experiments have been done, has 169 handwritten “chunker” rules, and is able to analyze the following kinds of phrases:

1. Noun phrases (NP). Sequences with nouns (case is nominative or accusative) are analyzed as noun-phrase, for instance, phrase *two little cars* – `<NP>` `{`*two (eki)*

<numeral> *little (kiskentai)* <adjective> *car(autokolik)* <noun>} is grouped into one NP phrase. All NP phrases have not determined tags for cases and possessives, in case that in next interchunk stage it will be assigned: *I see* <NP>{*the sky*<case-to-be-determined>} – I (Men) <NP>{sky<accusitive> (aspn*DY*)} see(koremyn). As can be seen from the example, noun phrase “the sky” has not determined case, but in interchunk stage it should be determined as “accusitive” case and will be added ending -dy, which could be changed, depending on vowel harmony.

2. Noun phrases as gerund(NP-ger). For verbs, which appear after verbs: like, love, finish, start, hate, etc. and takes -ing form, it is translated as NP phrase too, and on the Kazakh side, it has gerund tense: *I like playing* - <NP> {*I (men)* <subject pronoun>} <VP> {*like*(zhaqsy koremin) <verb>} <NP>{*playing(oinau**dy**)* <verb gerund>}. Such type of verb is decided to be noun phrase, because on Kazakh side it could have case, possessive such like noun phrases with noun has.
3. Verb phrases (VP). All kind of verbs: simple verbs (only one word), complex verb tense (continuous, perfect), modal verbs (with assigning genitive case for subject – *I must play – Meniŋ (Ме**нің**) oinauym (ойнауым) керек (kerek)*), etc. Modal verbs have special phrases, for instance, “VP_must_inf” or “VP_should_inf”, it helps to assign possessive from subject by rule, which is written in interchunk stage.
4. Prepositional phrases. These phrases feature locative (-да/da – *in house* – үй*де/*ui*de*), ablative (-*нен/nen – from river* – өзен*нен/*ozen*nen*), genitive (-ның/niŋ – *of city* – қала*ның*/kala*nyŋ*), postpositions, as well as complex postpositional phrases with words үст/аст + possessive + locative (*under table* – үстелдің аст*ында/*ustel*diŋ* astynda).
5. Question verbs phrase (VP_Q) are used to detect question, which started with *did/do, was/were*, etc., where auxiliary verbs analyzed as VP_Q, and will be processed in interchunk, to generate question particles in Kazakh (-ma/me, etc.). For instance, “*Do you remember?” – Sizdiŋ (Сіздің) esiŋizde (есіңізде) me (**ме**)?*).
6. Auxiliary verb phrases(be/have/do,etc). Such kind of phrases are used in the next structures: <VPQ> {*Do* <verb do>(only tense)} *you play?* – to translate questions, where rule only detects tense and transfer it to the next stage in <VPQ> phrase; *I* <VP_be>{*am* <vbser>(*e* <copula>)} *a teacher* – to generate copula “e”(edi,edim) and move it at the end of noun phrase(a teacher – mugalim[e+**myn]**), then assign person and number from subject at the interchunk stage.
7. Adjectival phrases: single adjective (AdjP *big*) and comparative adjective (AdjP *bigger*), superlative adjectives have different phrase, because, for instance, the translation of “SupP the most beautiful” is “SupP *eŋ* (ең) ædemi (әдемі)”, but it could get possessive (SupP {*the most interesting} of these books – kitaptardyŋ* (SupP {*eŋ ædemi**si***}), so it can not be treated as regular adjective phrase AdjP.

As can be seen from phrases above, each type of phrase has concrete operations, which could be done at the interchunk stage: determining case, possessives, assign person and number, moving positions. Without certain phrase names, it is impossible to have well-worked interchunk stage.

3 Extracting “Chunker” Rules from Corpora

The method described by Sánchez-Cartagena et al. (2015) was inspired by the work of Sánchez-Martínez and Forcada (2009) [5] where alignment templates were also considered for structural transfer rule inference. However, this new approach overcomes the main limitations of that by Sánchez-Martínez and Forcada (2009). Firstly, choosing the appropriate generalization level for the alignment templates (AT), contained word alignment and use word classes instead of the words themselves [6,7,8], from which rules will be generated. Second, a different treatment words which have difficulties with context-dependent lexicalizations and are incorrectly translated by more general ATs. Third, the automatic selection of ATs to be used for generating convenient rules.

To adapt the method by Sánchez-Cartagena et al. (2015) for the English–Kazakh language the following steps were performed:

1. Building English–Kazakh parallel corpora by using Bitextor[2], a web crawler for parallel texts, and manually collected texts from fiction literature. Manually collected corpus consist of ~3200 parallel sentences, and with crawled texts, parallel English-Kazakh corpus contains 5625 sentences. Experiments were done on a corpus consisting of 140 sentences, and big corpus is used for testing and for tuning.
2. Creating tag groups file for the Kazakh language. Sánchez-Cartagena et al. (2015) method had not been tested on Turkic languages, which have rich-morphology. As a result, this file for the Kazakh language will have more morphological tag groups. Groups have the following format, for instance, group for numerals: `numtype:ord,coll,year:num`, where `numtype` is name of variable used to identify different types of numerals `ord` *(first, second)*,`coll` (using in Kazakh to identify number of objects or subjects without followed noun: two person – eki adam(екі адам), two came – ekey keldi (екеуі келді))`,` `year` (numerals, coming after prepositions: *in 1992*), and at the end after “:” is put name of part of speech “numeral” – “`num`”. If some tags belong to several part of speeches, they are put after “:” and is divided by comma: `tense:present:`*`vblex,v`*. This file will be used to generate an appropriate group of tags for each part of speech of English and the Kazakh language side, all necessary tags could be found from morphological analyses of English–Kazakh MT system on Apertium platform.
3. Creating a tag sequences file, where the defined tag groups are combined into appropriate sequences of tags, accordingly to morphological analysis. The sequences will be used to generate target language sequences of tags, which are the lexical categories of each lexical forms. If the morphological analysis of word “do” is: `do<vbdo><pres><p3><sg>,` in format of tag sequence it will be look like as follows: `vbdo:verbtime,person,numberat`, where `vbdo` is name of lexical category, and `verbtime,person,numberat` name of tag groups, defined in tag group file.

[2] https://svn.code.sf.net/p/bitextor/code/trunk

4. Adapting the rule extraction script: defining installed the English–Kazakh language pair on Apertium machine translation system, morphological and bilingual dictionaries, size of corpora.
5. Problem of adapted method of extracting chunker rules from corpora. Some MT systems, like English-Kazakh machine translation system on Apertium, uses three-stage structural transfer, which means that adapted method needs improvement, because the rule learning algorithm is designed to work only with 1-level Apertium transfer (only apertium-transfer module and not apertium-interchunk). Generated chunks have no special phrases (it generates "LRN" phrases), as NP, VP, etc., showed in section 2, this fact prevents correct usage of this phrase in interchunk stage.

4 Automatic Detection of Chunk Type

To improve quality of translation and to do work of generated rules more usable in interchunk stage additional step was added: detect name of phrase for generated chunks. For instance, if chunk named "__n__" and deal with nouns, "NP" phrase should be assigned.

To assign a phrase to each chunk, first, there are defined part-of-speech sequences to each phrase and will consider sequences of POSes by using X'-theory [9], where has been defined the X'-equivalences shown as Table 1.

Table 1. X'-Equivalences

X	**X'**	**X"**
N	N'	NP
V	V'	VP
A	A'	AP
P	P'	PP
INFL	S(I')	S'(IP)

As can be seen from Table 1, there are defined five phrases. First level defined as X, next levels are will modified with other grammatical constituents function as the specifiers: X' defines X+X phrase and X'' could define X+X' or X'+X' phrases. There are could be defined next primary priority of POS [10]:

Primary POS priority : V > N > A > P

According to this priority, for example, for English-Kazakh language pair, POS sequences will be defined for each phrase as follow:

Table 2. POS sequences for X'-Equivalences

X	**X'**	**X''**
pr	X+n, X+adj+n, X+num+n, X+det+n X+det+adj+n	**PP**

vblex, vbser, vbhaver, vbmod	X+adv	**VP**
n, det, prn	adj+X, num+X, num+adj+X	**NP**
adv	preadv+X	**AdvP**
adj	more/the most+X less+X	**AdjP**

And, POS priority for English-Kazakh pair will be looked like that:

Primary POS priority : P > V > N > A

Choosing that priority based on highest score got from evaluation, which will be showed in Results section, and also on that P-prepositions could be only modifiers of noun and in Kazakh they will transform into postpositions or case. In that case, PP phrases include noun in their structure that took them in priority before N.

Described phrases are written in additional file, where user can specify phrases by priority, accordingly to each language pair's features. This file is called "phrase.txt", where described priority is written in the Apertium chunk names format.

Table 3. POS sequences for X'-Equivalences

X	X'	X''
pr	X+n, X+adj+n, X+num+n, X+det+n, X+det+adj+n	PP
vblex, vbser, vbhaver, vbmod	X+adv	VP
n, det, prn	adj+X, num+X, num+adj+X	NP
adv	preadv+X	AdvP
adj	more/the most+X, less+X	AdjP

As can be seen from the Table 3, first, user writes name of phrase, then part-of-speech, which defines this phrase: `VP,vblex,vbser,vbhaver,vbmod.` Phrase detection program reads this file and generated file with rules, and assigns phrases. To do this application more usable, templates of rules were changed by adding one-word rules. Evaluation of this method is described in the next section.

5 Results

Results of improved method are performed by using English-Kazakh MT system on Apertium. From GATs, extracted from corpora with 140 sentences, 13 rules are generated. In the next table, some of the translation rules obtained with handwritten and extracting processes are compared:

Table 4. Comparing translation

Input sentence	Handwritten rule	Generated rule
in the garden	бақшада - ^prep-nom<PP><sg><p3>**<PXD>**<loc>{^бақша<n><2><4> <5> $}	бақшада - ^__n_<LRN>{^бақша<n><loc>$}
play	ойнайды - ^pers-verb<VP>**<ND><PD>**<aor>**<PXD><NXD><CD>**{^ойна<v><tv><6><4><5><3><2><7>$}$	ойнайды - ^__vblex_<LRN>{^ойна<v><tv><aor><p3><sg>$}
from place	орыннан - ^prep-nom<PP><sg><p3>**<PXD>**<abl>{^орын<n><2><4><5>$}$^sent<SENT>{^.<sent>$}$	do not generated
big house	үлкен үй - ^adjec-noun<NP><sg><p3>**<PXD><CD>**{^үлкен<adj>$ ^үй<n><2><4><5>$}	**үйдің** - ^үлкен<adj>$ ^__n_<LRN>{^үй<n>**<gen>**$}$ ^.<sent>$
Conan Doyle	noun<NP><sg><p3>**<PXD><CD>{^**Конан<np><cog><mf><2><4><5>$}$ ^noun<NP><sg><p3	^Конан<np><cog><mf><sg>$ ^Дойл<np><cog><mf><sg>

	><**PXD**><**CD**>{^Дойл<np><cog><mf><2><4><5>$}		
A dog is also in the garden	Ит тағы бақшада	Ит ол бақшада	**<missed “also”>**

As can be seen from Table 4, a few generated rules works correctly, but number of generated rules is not big, because of small volume of corpus. The main differences between rules are that in handwritten rules some of tags undetermined (<PXD>, <ND>, <PD>, <NXD>) or could be changed in next interchunk stage, whereas generated rules assign all tags constantly. Also, generated rules while translating could miss some words, as can be seen from the last translation of sentence “A dog is also in the garden”, where generated rule translated it without adverb “also”. Such problems appears because of low generalization level, that problem could be solved by using bigger corpus for extracting rules.

In next table quality of translations will be compared with rules, which were got after using rules with application for specifying phrases and rules, without it:

Table 5. Quality of translated texts

Evaluated phrase types in generated rules	BLEU	unigram	bigram	trigram
Default phrases (LRN)	6,07	27,16	9,41	3,87
Specific phrases (NP, VP, etc.) and improved rule templates	10,26	39,71	18,06	8,89

As can be seen from Table 5, adding phrase detection step and improvements of rules templates helped to raise quality of translation for 4%, for unigrams it is raised for 12.55, bigrams for 8.65 and trigrams for 5,02. In the next Table 6 was showed translated sentences by Sánchez-Cartagena et al. methodology and by proposed improved methodology:

Table 6. Translated texts

Evaluated phrase types in generated rules	Input sentence	Chunker stage	Interchunk stage	Output
Default phrases (LRN)	Dog plays in the garden	^__any__<**LRN**>{^Ит<n><sg>$}$ ^__vblex_<**LRN**>{^ойна<v><tv><aor><p3><sg>$}$^*executedtule11$ ^__pr___det___n_<	^__any__<**LRN**>{^Ит<n><sg>$}$ ^__vblex_<**LRN**>{^ойна<v><tv><aor><p3><sg>$}$ ^*executedtule11$ ^__pr___det___n	Ит *ойнайды* бақшада

		LRN>{^бақша<n><loc>$ }$^.<sent>$	_<**LRN**>{^бақша<n><loc>$ }$^.<sent>$	
Specific phrases (NP, VP, etc.) and improved rule templates	Dog plays in the garden	^__n__<**NP**>{^Ит<n><sg>$}$ ^__vblex_<**VP**>{^ойна<v><tv><aor><p3><sg>$}$ ^__pr___det___n_<**PP**>{^бақша<n><loc>$ }$^.<sent>$	^__n__<**NP**>{^Ит<n><sg>$}$ ^__pr___det___n_<**PP**>{^бақша<n><loc>$ }$ ^__vblex_<**VP**>{^ойна<v><tv><aor><p3><sg>$}$ ^.<sent>$	Ит бақшада *ойнайды*

After chunker stage, input text is transferred into sequences of tags, where phrase type tags in bolt. In the interchunk stages, as can be seen from the fourth column, chunks with phrase types "<LRN>" did not changed their position, whereas specified phrases NP, VP, PP changed as follow: NP VP PP → NP PP VP. The last columns show output of translation. In the result, new method performed right sequences of phrases, verbs phrase in italic in the end of the sentence.

6 Conclusion

In the paper is proposed automatic detection method of "chunks" type improving methodology Sánchez-Cartagena et al. (2015) of extracting grammatical translation rules from bilingual corpora. Results of this paper could use for others morphology rich languages. Proposed improvement of methodology of extracting grammatical translation rules from corpora allows improving of machine translation quality. For future works it is planned to use improved methodology for more biggest English-Kazakh corpus, using proposed improved methodology for other kind of languages pair, Kazakh-Russian.

Acknowledgements

Authors thank Prof. Mikel L. Forcada and Miquel Esplà-Gomis from Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant (Alacant, Spain) for the continuous consult and help in the researching and development of this project. This research work is doing in frame of project 0749/GF financed by Ministry of Education and Science of Republic Kazakhstan.

References

1. Hutchins, William John, and Harold L. Somers. An introduction to machine translation. Vol. 362. London: Academic Press, 1992.
2. Víctor M. Sánchez-Cartagena, Juan Antonio Pérez-Ortiz, and Felipe Sánchez-Martínez. 2015. A generalised alignment template formalism and its application to the inference of

shallow-transfer machine translation rules from scarce bilingual corpora. Comput. Speech Lang. 32, 1 (July 2015), 46–90.

3. Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, Francis M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. In Machine Translation (Special Issue on Free/Open-Source Machine Translation), volume 25, issue 2, p. 127–144.
4. Sundetova, A., Forcada, M. L., Shormakova, A., Aitkulova, A.:Structural transfer rules for English-To-Kazakh machine translation in the free/open-source platform Apertium.Proceedings of the International Conference on Computer processing of Turkic Languages, pp. 317–326. L.N. Gumilyov Eurasian National University, Astana(2013)
5. F. S´anchez-Mart´ınez and M. L. Forcada. Inferring shallow-transfer machine translation rules from small parallel corpora. Journal of Artificial Intelligence Research, 34(1):605–635, 2009. ISSN 1076-9757.
6. F. J. Och and H. Ney. The alignment template approach to statistical machine translation. Computational Linguistics, 30(4):417–449, 2004.
7. F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51, 2003.
8. Y. Xu, T. K. Ralphs, L. Lad´anyi, and M. J. Saltzman. Computational experience with a software framework for parallel integer programming. INFORMS Journal on Computing, 21(3):383–397, 2009.
9. Sells, Peter (1985), Lectures on Contemporary Syntactic Theories, Lecture Notes, No. 3, CSLI.
10. Kuang-hua Chen and Hsin-Hsi Chen. 1994. Extracting noun phrases from large-scale texts: a hybrid approach and its automatic evaluation. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94). Association for Computational Linguistics, Stroudsburg, PA, USA, 234-241. DOI=http://dx.doi.org/10.3115/981732.981764