

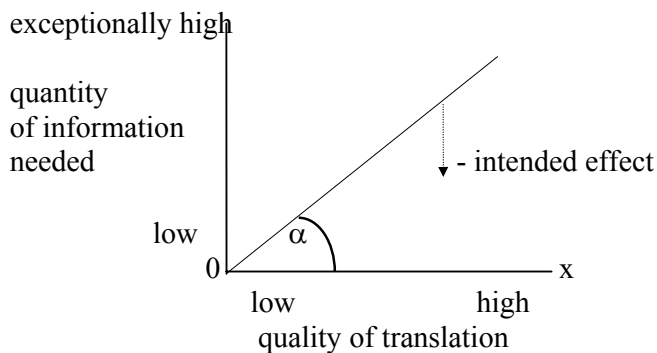
Slavic as a Source and Target Language in NeuroTran® Sentence Translation

0. Introduction

The following outlines the strategy of translating sentences between Slavic languages and English using the NeuroTran® MT (Machine Translation) system. We will first describe the manner in which NeuroTran copes with basic machine translation problems, then, the general rules of MIG (Minimal Information Grammar) -- the grammatical rules used by NeuroTran® to process both individual words as well as sentences. Finally, we will present actual translation examples using a Slavic language as both the source and the target language.

1. MT and NT

With a certain degree of simplification, the ratio between the quantity of information needed and quality of any sentence-to-sentence Machine Translation (MT) endeavor can be capsuleized as:



The main goal in developing NeuroTran® was to apply solutions in such a way as to improve this ratio by reducing the quantity of information the program requires to perform high quality MT tasks. Three ideas are crucial to the realization of this goal. The first, that one should strive to minimize the amount of information required by the program and, then, allow it to acquire new information by reading text and communicating with the user via artificial neural networks. The second foundation of NeuroTran® is that one needs to attempt to reduce the effort on the part of the user by requiring minimum information from them and, instead, using the neural network learning mechanism to generalize the information provided directly to the neural network so that it can generate

the balance of the information required. Finally, the data embodied in NeuroTran® should be reusable and thereby usable from one piece of text to another and in various situations and endeavors. When released, NeuroTran® will be typically post-fordist, simultaneously serving as a bilingual, bi-directional and bi-medial dictionary, thesaurus, translator, parser and text analyzer at the same time. Further information about the general design of the software and its morphological module can be found in Šipka & Končar (1997) as well as more the up-to-date information at: <http://www.tranexp.com>.

Sentence-to-sentence translation is one of the most complex NT modules. It is interwoven with other modules, both from the standpoint that it uses data directly from them (like morphological paradigms) and that it produces the data they require (like relations within clauses which are used in the qualitative analysis module). When compared with the existing sentence-to-sentence translation systems, NeuroTran® exhibits several features which are distinct from mainstream thinking, for example, as presented in King (1987) or Newton (1992):

- a. The relation between the software program and linguistic theory is such that the only purpose of formalization of linguistic material is to make the software function more accurately. This differs greatly from systems which are rooted in certain other linguistic theories or approaches. Our goal is to ascertain the one true and correct translation for any given sentence -- nothing less and nothing more. This means that we are not interested in those questions linguistic theories attempt to answer such as whether the sentence has universal features, if any other correct translations exist, whether the source sentence can have alternative forms, whether it is related to another structure, etc.
- b. The relation between algorithmic and heuristic strategies is such that any solution which minimizes the quantity of information and enhances the quality of translation is adopted. We have no fixed preference toward either of the strategies.
- c. The software is capable of learning in that the quality of translation will increasingly improve after each translation it performs. It functions much like a living being who learns from and adapts to its environment.

The set of rules (or formal language) by which this is achieved we have termed Minimal Information Grammar (MIG). In the next section we will present the basic features of this formal language and emphasize those which are directly relevant to sentence-to-sentence translation.

2. NT and MIG

Minimal Information Grammar (MIG) is designed in such a way as to allow the concepts crucial to NeuroTran® to be realized. The grammar has been named minimal because of the fact that it reduces the information required by the software. It does this by balancing grammatical rules and dictionary label information -- both being components of the grammar -- by using different classes of rules (constructors, mutators, selectors, etc.) to manipulate the already existing linguistic material and, then, by using neural networks to add new information obtained from the learning process initiated when the software reads a text or communicates with users. All existing information is virtually "recycled" with each new piece of text. MIG operates with the following classes of rules: a. *constructors* -- using dictionary labels to construct all possible forms a word; b. *mutators* -- to change already generated forms; c. *finders* -- to find the form or word required; d. *definers* -- to declare what is what; e. *coordinators* -- they determine how one form coincides with others; f. *choppers* -- to divide larger units into smaller ones; g. *binders* -- to build smaller units into larger ones; h. *transformers* -- to replace one word or form with another (for example by translating a word in one language by a word in the other); i. *counters* -- to track statistics; j. *doubters* -- to detect situations where there are numerous possibilities; k. *gamblers* -- to choose a single solution that is most probable where other options are available; l. *teachers* -- to change existing information (rules and figures) after reading different texts and translations; m. *chatters* -- to ask the user when they need a piece of information or if they want to change something; n. *conductors* -- to direct the order in which the rules are applied. Every rule consists of a head (stating the input of the rule) and a body (providing details of how the output is calculated). The rule format along with an example from the sentence translation module is demonstrated in the Table 1

Rule	Example
<rule head> =>	ENGSCR GRM N[ADJECTIVE PRONOUN] NOUN =>
<rule body line 1>;	COPY(2->1:NUMBER,GENDER)
<rule body line 2>;	
....	
<rule body line N>	

Table 1

The example above accounts for the situation when, for example, translating from English to a Slavic language where one must coordinate the gender and number of the modifier with the one of the head of a NP. The head of the rule states: "if the sequence of any number of adjectives or pronouns and one noun has been found" and the body states "copy the gender and the number from the second element to the first". If only two elements exist, the number and gender will be copied immediately, if not, the program will continue until it finds an element having its own gender and number (a noun) and then copy these values

to all elements preceding the noun. This example demonstrates how these requirements of minimalism are incorporated in the program. We do not need a separate rules for the number and quality of the NP modifiers. A single rule is sufficient.

Another minimization strategy relevant to the sentence translation was to reshuffle traditional morphological and syntax rules in order to place main clausal variation types within the verb inflection. Any finite verb form (in every tense and both active and passive), also has a negative, interrogative and negative-interrogative form. Thus, verbal inflection generates more than one thousand forms for each transitive verb. This can be see from the fragment of the Serbo-Croatian verb paradigm for the verbs like "kucnuti" = "knock" (dots/periods denote omitted lines):

<i>Fragment of the Rule</i>	<i>Explanations and Examples</i>
SCR PARA *nuti,*nem,*nu v; =>	head, if the dictionary entry has this form
VERB;	then it is a verb (our example is kucnuti,nem,nu v; 'knock')
ACTIVE;	active forms are generated as follows:
...	
PRESENT;	Present Tense
O1=(1->SAMEAS(1','+1))+(1','->2','-1);	uses the following stem one
O2=(1->SAMEAS(2','+1))+(2','->' ');	and the following stem two
AFFIRMATIVE;	Affirmative form
SINGULAR;	in Singular
FIRST=O1+m;	First Person: kucnem
...	
INTERROGATIVE 1;	First possible Interrogative form
SINGULAR;	in Singular
FIRST=O1+m li;	First Person is: kucnem li
...	
INTERROGATIVE NEGATIVE;	Interrogative-Negative form
SINGULAR;	In Singular
FIRST=ne O1+m li;	First Person is: ne kucnem li
...	
PASSIVE VOICE;	Passive voice
O1=(1->'','-2);	uses the following stem one
...	
PRESENT;	in Present tense
AFFIRMATIVE;	Affirmative form
SINGULAR;	in Singular
MASCULINE;FIRST=sam O1+t;	First Person, Masculine Gender is: sam kucnut
...	

Table 2

Obviously, some of the forms presented in the table are a part of clausal variation rather than (parts of verbs) inflection but in this manner we reduce the complexity of the sentence translation task.

3. MIG and Sentence Translation

Following general NeuroTran® strategy, we are interested in finding one correct L2 equivalent for one L1 sentence where both L1 and L2 are concrete languages. That is to say, we are not interested in building a general linguistic model, because our rules are restricted to the concrete language pairs and do not account for alternative forms of the same content. Furthermore, we believe that perfect machine translated sentence translation is impossible. Our goal is to make the system increasingly less imperfect as it develops. Finally, we believe that sentence translation must take into account the concrete environment in which it functions. In turn this means that our goal is to create a living organism which translates better with each new translation task which is performed.

The general strategy of sentence translation consists of the following seven steps.

- A. Breaking up text into sentences
- B. Identifying finite verbs and predicates
- C. Breaking up sentences into clauses
- D. Identifying the subject
- E. Identifying and transferring phrases
- F. Performing transformations of translated clauses
- G. Binding translated clauses into an output sentence

It is important to stress that (in the final two steps above) the bulky corpus of the target language is used to transform and bind the clauses and sentences in such a manner so as to be consistent with the findings of the corpus. For example, if we are to find the English equivalent of Polish "w pracy" or Serbo-Croatian "na poslu", it will first be translated as "in work" and "on work" in Step E. Next, the corpus will be consulted and both translations will be changed into "at work".

This is demonstrated in Diagram 2

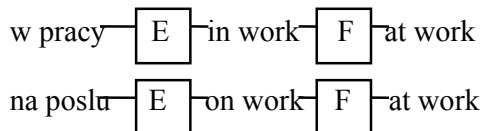


Diagram 2

Step A is dependent on the orthographical rules of the source language, Steps B-D depend on the source language, Step E on the both languages, whereas Steps F and G are dependent on the target language. We will therefore describe the sections A-E in Slavic as the Source Language sections and Steps E-G will be described in Slavic as the Target Language sections.

4. Slavic as Source and Target Language

In the phase A, the basic problem with the Slavic (just as with any other language which uses a period to delimit the sentence) is the existence of abbreviations which can be either at the end of the sentences or at any other point within the sentence. Abbreviations which are not likely to be found at the end of the sentence need not be addressed in this step. In such cases, the period at the end of the abbreviation is temporarily declared the end of sentence and (in the Step B) is examined to ascertain if the text preceding or following the period contains a predicate. Possible situations are presented in the Table 3

Example	Explanation
Marko, Petar i dr. 1 Markovići čitaju u kući. 2	only 2 is the ESM, finite verb in one sentence only.
Unosimo kutije, pakete i sl. 1 Markovićima u novi stan. 2	only 2 is ESM, finite verb in one sentence only.
Unose kutije, pakete i sl. 1 Uselili su se u novi stan. 2	two ESMs, finite verb in both sentences
Marko, Petar i dr. 1 Markovići uz pakete, kutije i sl. 2 Petrovićima unose i televizor u novi stan. 3	only 3 is the ESM

Table 3

A special problem arises in languages where orthographic tradition allow writing other ESMs in the middle of the sentence. For example, the Russian sentence *Зима! Крестьянин, торжествуя, на дровнях обтовляет путь* exemplifies such a problem.

In the Step B, provided the sentence contains a predicate, there are two types of problems to be tackled. Both are related to the features of each particular language.

One task in identifying finite verbs and then linking them to the other possible parts of the predicate (as understood in traditional Slavic linguistics) is to identify discontinued verb forms. This situation differs with different Slavic languages. For example, Past Tense is discontinued in Serbo-Croatian but is not discontinued in Russian. In Polish it can be either discontinued or not.

This problem is solved because there exists a general rule to link the two discontinued forms. The forms themselves are defined for each language separately. The discontinued finite verb forms are determined based on the morphology rules presented in Table 2 and complex predicates (just like modal verbs) are solved because there exist general rules to link the elements as well as language specific rules describing what the elements are. This can be seen in Table 4:

Rule	Explanation
SCR GRM MOD => 'moći', 'htjeti', 'smjeti', 'smeti', 'trebati', 'morati', 'voljeti', 'voleti'	The modal Serbo-Croatian verbs are the verbs stated in the body of the rule
SCR GRM REST => (VERB(INFINITIVE)) ('da' & VERB(PRESENT))	The rest of the predicate is either Infinitive or the Present Tense of another verb
UNI GRM MOD * REST => ASSIGN(1&2,PREDICATE)	If you find the modal verb and the rest of the predicate (and possibly something between these two elements) declare the both parts predicate
UNI GRM REST * MOD => ASSIGN(1&2,PREDICATE)	If you find it in reversed order do the same

Table 4

This accounts for cases such as *Mogu raditi* ('I can work') *Raditi danas ne mogu*. ('I cannot work today') (reversed order and third lexical in between), etc.

The second major problem to be addressed in identifying the predicate is 'morphonymy'. One must decide if the homonym found within a sentence is a finite verb or something else. This can be seen with a simple Serbo-Croatian example. The four lexical units from the Table 5a (nothing extraordinary in this respect) generate a long list of morphonyms (one of which is presented in the Table 5b) where only one of ten forms is a finite verb:

lexeme	meaning
kos,a,o;	'steep'
kos,a # +ov m;	'blackbird'
kosa,e f;	'hair'
kositi,sim,se iv;	'mow'

Table 5a

morphonym	canonic form	morphological description
kose	kos	vocative,singular,masculine,noun
kose	kos	genitive,singular,feminine,adjective
kose	kos	accusative,plural,feminine,adjective
kose	kos	vocative,plural,feminine,adjective
kose	kos	nominative,plural,feminine,adjective
kose	kosa	genitive,singular,feminine,noun
kose	kosa	nominative,plural,feminine,noun
kose	kosa	accusative,plural,feminine,noun
kose	kosa	vocative,plural,feminine,noun
kose	kositi	3rd person,plural,present,verb,affirmative

Table 5b

In all such cases, all options are kept open and in further steps are reduced to the most probable choice.

In the Step C, the clause border is identified based on the end-of-clause marker between two predicates -- typically, a conjunction or another part of speech functioning as a conjunction. There are two major problems confronted in this step: a. homonymous end-of-clause markers and b. embedded clauses.

Consequently, in situations where an unambiguous end-of-sentence marker is not present, the odd comma is used as the delimiter if there is only one odd comma between the predicates. All options are kept open and in further steps are reduced to that which is most probable. The cases of embedded sentences are detected based on the fact that there are irregular predicate to end-of-sentence markers relations, for example: *Čovjek|1 koji je došao|2 radi na pošti|1* ($S1 = \text{Čovjek radi na pošti}$, $S2 = \text{koji je došao}$) where there is an end-of-sentence marker in front of the series of two predicates. Obviously, embedding can go along with ambiguous end-of-sentence markers.

After the clause is delimited, the program searches for the subject within each clause. First it checks to see if there is a nominal element preceding the congruent predicate. If there is not, it searches for those situations noted in Table 6:

Category	Explanation	Example
Contained subject	Due to the pro-drop feature in Slavic languages, there is no nominal element but the properties of the subject can be inferred from the verb form	<i>Radim u kući.</i>
Coordinated subjects	Two subjects are coordinated with one predicate. This is signaled by certain conjunctions (e.g. S-Cr „i”) and constructions (e.g. Russian Nom + sa + Ins)	<i>Отец с матерью ушли, Otac i majka su otišli</i>
Non-nominal subjects	In clauses where there is a be-predicate, in the subject position can be infinitive, adverb, or any other canonical form of any part of speech	<i>Raditi je glupo.</i>
Subject inversion	In certain focus/topic distribution configurations subject can be after the predicate.	<i>Marka je udario Petar. (a ne ja)</i>
Homonymous subject	There are some cases where two forms can be understood as subjects	<i>Zemljotres je srušio grad.</i>
No-subject situations	One number of constructions does not have a subject.	<i>Grmi. Zuji mi u ušima.</i>

Table 6

All identification and transfer rules (herein referred to as Step E) recurse/iterate until all sentence elements are detected. Not all target language equivalents are correct and/or grammatical. The idea is to translate everything using a minimal set of rules and then to transform such in the last two phases. In all case where no single and unambiguous solution exists advantage is given to the elements which are typical and/or closer to one another.

The program searches for the patterns of words within a clause so as to match the phrases which exist in any particular language. Please note that transfer rules are used at the same time for parsing. The patterns at the head of each rule are assigned certain dependency values. This is also true for the target language patterns which enables the writing of a parse tree as a byproduct of the transfer. As noted previously, Slavic languages can be both in L1 and L2 position here. Let us see how this works using a simple example which transfers English cases like *father's house* into the Serbo-Croatian *kuća oca*.

Rule	Example
ENGSCR GRM NOMINAL1('s) NOMINAL2 =>	father's house
NOMINAL2 NOMINAL1(GENITIVE)	kuća oca

Table 7

Within each language there is a rule stating the dependency relations between the elements in the pattern, such as the one for Serbo-Croatian cases like *kuća oca*:

SCR GRM NOMINAL2 NOMINAL1(GENITIVE) =>
 ((NOMINAL1(NOMINAL2)), ATTRIBUTE, NON-CONGRUENT)

The rules can account for situations where there is a third element exists between the two elements of the pattern:

Rule	Example
SCRENG GRM PREDICATE * NOUN(ACCUSATIVE) => PREDICATE * NOUN	On čita dosadnu knjigu He reads boring book

Table 8

Of course, there exist a wide range of atypical situations (like "morphonymy") and atypical configuration of elements. Additionally there are a wide range of equivalency problems. The solution strategy for them is similar to what we have demonstrated up to this point. It should be stressed that not all translations are correct after this step and this is why the last two steps exist. The most common problems with equivalence (putting aside problems with lexical equivalency) are:

- a. multiple verb forms equivalence (one L1 verb form is the equivalent of several forms in L2),
- b. Slavic aspect (L2 has to be either perfective, imperfective or bi-aspectual)
- c. prepositional phrase equivalents are influenced by the noun modifier

Step F is highly dependent on the existence of a well organized corpus of target language text. Additionally, an appendable list of standard errors also exists so that, with each new translation, the program acquires more and more knowledge. Since all information from previous text is present, the option of backtracking exists.

The corpus is an integral part of the NeuroTran® system and statistical data are collected ad hoc. As an example, let's step through the process of translating the English "in Cetinje" (Cetinje is a proper name of a town) into the Serbo-Croatian "na Cetinju." In phase E, this prepositional phrase will be translated as "u Cetinju", which is incorrect because the noun is exception and takes the preposition *na* (literally *on*). Now, in the step F, the program will use the list of the following structure: *S-Cr form/pointer to the corpus 1/pointer 2/.../pointer n*. Using the PP patterns from Step E, the program performs a frequency analysis of different prepositions headings. This in turn indicates that it should be "na", and consequently, "u Cetinju" is transformed into "na Cetinju".

Phase G is devoted to solving such things as adding case and gender to relative pronouns, which in the most cases requires backtracking and establishing relationships between clauses.

5. Conclusions

Using NeuroTran® for sentence-to-sentence translation of Slavic languages follows a predetermined set of ideas. Many of the problems that must be tackled are all but universal (e.g. sentence delimitation), while some are common to all Slavic languages (e.g. adding gender to adjectives). Additionally, some problems, are common to most all Slavic languages (e.g. determining the appropriate synthetic case) and some are specific to only one particular Slavic language (exceptions with prepositional phrase valences). NeuroTran® sentence translation remains work in progress and testing it in a number of situations will determine the specific strategies that provide the best results.

Acknowledgements

We would like to express our thanks to Vladimir Šipka and Sławek Pawłowski for all their kind programming, to James Connolly who proofread this paper, and to the Alexander von Humboldt-Stiftung that made it possible for Danko Šipka to work on the usage label network.

References

- King, Margaret ed. (1987) *Machine Translation Today*, Edinburgh: Edinburgh University Press
Newton, John ed. (1992) *Computers in Translation: A Practical Appraisal*, London: Routledge
Šipka, D. & Končar, N. (1997) „Minimal Information Grammar (MIG): Serbo-Croatian and Polish Morphological Paradigms” in: Junghanns, U. & Zybatow, G. (eds.): *Formale Slavistik* (Leipziger Schriften zur Kultur-, Literatur-, Sprach- und Uebersetzungswissenschaft; 7), Frankfurt am Main: Vervuert Verlag, p. 427-436

Nenad Končar (n.koncar@tranexp.com) Danko Šipka (d.sipka@tranexp.com)