

Danko Šipka (Adam Mickiewicz University, Poznan, Poland)

Nenad Končar (Imperial College of Science, Technology and Medicine, London, UK)

Minimal Information Grammar (MIG): Serbo-Croatian and Polish Morphological Paradigms

1. NeuroTran

Minimal Information Grammar is a set of rules used by NeuroTran, a piece of software intended to „do things with words”. NeuroTran, when completed, will be a typical post-fordist, serving as a bilingual, bi-directional, and bi-medial dictionary, thesaurus, translator, parser, and text analyzer at the same time.

It has been planned that NeuroTran does the following things with words:

1. lookup a word and its L2 equivalent(s), with their respective grammar and usage labels,
2. reproduction of the sound of a word or phrase,
3. generation of all inflections (forms) of a word,
4. lookup all words with a common part of speech, or some other grammatical feature
5. lookup all words with a common subject-matter field or common usage features,
6. lookup synonyms and antonyms,
7. sentence-to-sentence translation,
8. determining the type and language of some text and based on this information to automatically choose between multiple possible translations for a given word
9. sentence parsing,

10. qualitative (content) text analysis.

Three ideas are crucial to the concept of NeuroTran. The first postulates that one should minimize the information required by the software to work, and then have it acquire new information by reading texts and communicating with the user, while using neural networks. The second idea underpinning NeuroTran is that one should reduce the effort on the part of the lexicographer, by requiring minimum information from her, and using neural network learning mechanisms in order to generalize from information provided to the neural network in order to generate the rest of the information. Finally, the third idea behind NeuroTran is that the data included in NeuroTran should be reusable, and thus functions performed by the software usable in various situations and fields. NeuroTran has a set of rules and procedures intended to implement those ideas. MIG is one of them.

2. MIG

Minimal Information Grammar is designed in such manner to allow the concepts crucial to NeuroTran to be realized. The grammar is named minimal for the fact that it tends to reduce the information needed for the software to perform functions stated above. It does so by balancing information in grammatical rules and dictionary labels, both being components of the grammar, by using different classes of the rules (constructors, mutators, selectors, etc.) to manipulate the existing linguistic material, and finally by using neural networks to add new pieces of information from a learning process which is initiated any time when the software reads a text or communicates with users. The existing information is virtually „recycled”. MIG operates with the following classes of rules: a. *constructors* - they use dictionary labels to construct all forms a word can have, b. *mutators* - they change already generated forms, c. *finders* - they find the form or word we need, d. *definers* - they say what is what, e. *coordinators* - they say how one form goes along with others, f. *choppers* - they chop larger units into smaller ones, g. *binders* - they unite smaller units into larger ones, h. *transformers* - they replace one

word or form by the other, for example by translating a word in one language by a word in the other, i. *counters* - they keep track of all statistics, j. *doubters* - they detect situation where there are more possibilities to proceed, k. *gamblers* - they choose the solution that is in their mind the most probable, even though the other options are still on the table, l. *teachers* - they change the existing information (rules and figures) after reading different texts and translations, m. *chatters* - they ask the user when they need a piece of information, or if user wants to change something, n. *conductors* - they direct the order in which the rules are to be applied. Every rule consists of its head, stating the input of the rule, and its body, giving details of how the output is calculated.

3. MIG Morphological Paradigms

Morphological paradigms within MIG are basic information used by neural network to initiate and investigate all possible solutions in translation, parsing, and qualitative analysis. The first two pairs of languages for which morphological paradigms have been constructed were English-Serbo Croatian and English-Polish. The other Slavonic languages for which the MIG is developed include Russian and Czech, and there are also rules for non-Slavonic languages such as German. In the nearest future, the MIG labels are to be implemented to a number of other languages, for which the dictionary used by Word Translator, a computational dictionary and word-for-word translator which was a first step in developing NeuroTranslator, has already been made. These languages include French, Swedish, Danish, Norwegian, Hungarian, Bulgarian, Spanish, Portuguese, and Old Church Slavonic.

We will present a general layout of the rules used in both languages, their features specific to Slavonic languages, and only occasionally point to those which are specific to Polish or Serbo-Croatian. It will also be shown how these paradigms are used by neural networks. First we will present those rules and functions used in all paradigms, and then turn to specific part of speech and their paradigms.

3.1 MIG Paradigm Rules: General Layout

In order to generate a paradigm MIG uses labels attached to the lexical entries in main dictionary text, and then applies a set of primarily constructor-type rules to the word bearing the label. For example, in

order to get all paradigmatic forms of the Serbo-Croatian word *selo*, we need the following two elements:

a. the dictionary entry

selo,a n;/village n;/countryside n;

where sequence *o,a n;* is the label used by the rules. The labels are attached within the procedure where lexicographer labels some 15 % of the entries, after which a procedure is used to find the correlation and label the rest of the text automatically, which is at the end passed to the lexicographer to correct it. Their form is designed not only to make it convenient for the rules, but also for the lexicographer, who has to invest minimal effort in attaching them - a native speaker can provide all information from the top of her head.

b. the set of rules

```
SCR PARA *o,a n =>
NOUN;
NEUTER;
O1=(1->','-1);
SINGULAR;
  NOM=O1+o;
  GEN=O1+a;
  DAT=O1+u;
  ACC=O1+o;
  VOC=O1+o;
  INS=O1+om;
  LOC=O1+u;
PLURAL;
  NOM=O1+a;
  GEN=O1+a;
  DAT=O1+ima;
  ACC=O1+a;
  VOC=O1+a;
  INS=O1+ima;
  LOC=O1+ima
```

Where the part in front of the sign '=>' is head of the rule, stating the input, and the part behind it is the body, defining how to produce the output. Speaking less technically, the rule states that if a dictionary item ends in "o,a n;" then rule tells us that this is a neuter noun, and generates all its cases in singular and plural. The cases are generated by chopping off one character before the comma, declaring that part a base, and simply attaching case endings to it.

This is of course not the only rule needed to generate paradigms. There are also additional procedures to handle alternations, like the Serbo-Croatian pair *vrabac:vrapče*, where we have three alternations b:p, a:o, c:č. We use three ways of handling these and other alternations:

- a. in the labels,
- b. through string cleanup rules,
- c. through functions applied to the bases

The Polish entry *jechać,jadę,jedziesz,ip*; shows how the problem of several alternations is eliminated through the label. The main principle here is to start the label at one character before the character which is different in two forms. In the example *jechać* it was necessary to repeat the whole form, while in some other cases, only part of the form is repeated, for example: *wykonywać,nuję,nujesz,ip*; The repetition of the forms is needed anyway, to generate the paradigm, so no additional effort has to be invested in handling alternations this way. The rule, then, has only to generate the paradigm, without having to deal with the alternation. The beginning of the rule for *jechać*, and similar cases looks as follows:

```
POL PARA *ć,*ę,*esz,ip =>
VERB;
O1=(1->');
INFINITIVE=O1;
PRESENT;
O1=(1->SAMEAS(1,'+1))+(1,'->2','');
O2=(1->SAMEAS(2,'+1))+(2,'->3',''-2);
O3=(1->SAMEAS(1,'+1))+(1,'->2',''-1);
SINGULAR;
FIRST=O1;
SECOND=O2+sz;
THIRD=O2;
.....
```

Now, due to the elimination of the problems with alternations in the label, this case is handled by the rule as any other verb having the same ending, no matter if it contains any alternations or not.

If an alternation appears in any string, i.e. the generation of a paradigm produces a non-acceptable sequence of characters it is settled by the so called string cleanup rules, which come at the end, after all other rules have fired, to check if a string contains any non-acceptable sequences. It can be seen from the Serbo-Croatian rule for voiced:voiceless alternation:

```
SCR STRING CLEANUP 2 =>
DEF Z=(b,d,g,z,ž,dž,đ);
```

```
DEF B=(p,t,k,s,š,ć,ć,f,h,c);
  IF (STRING[X] & X==*ZB*) THEN X=*BB*
```

```
SCR STRING CLEANUP 3 =>
  DEF Z=(b,d,g,z,ž,dž,đ);
  DEF B=(p,t,k,s,š,ć,ć,f,h,c);
  IF (STRING[X] & X==*BZ*) THEN X=*ZZ*
```

This of course is only the main cleanup rule, the exceptions are handled in separate rules.

Finally, those rules which change the base of a word, and appear only in certain forms are handled by functions, which are first stated, and then applied within the main rule, such as the Serbo-Croatian rule to handle palatalizations, which is stated as follows:

```
SCR FUN CZS => CZS[O]=LAST[O][(k,g,h)=>(c,z,s)]
```

and

```
SCR FUN CZS2 => CZS2[O]=LAST[O][(k,g,h)=>(ć,ž,š)]
```

and used with a paradigm generation rule as:

```
SCR PARA *K,a # m =>
  NOUN;
  MASCULINE;
  O1=(1->',');
  SINGULAR;
  ...
  VOC=CZS2(O1)+e;
  ...
  PLURAL;
  NOM=CZS(O1)+i;
  ...
  DAT=CZS(O1)+ima;
  ...
  VOC=CZS(O1)+i;
  INS=CZS(O1)+ima;
  LOC=CZS(O1)+ima
```

Several Polish functions, like POL FUN OU => OU[O]=[O][(*KoK_)=>(*KóK_)] which handle the alternation *koza-kóz* are implemented in this manner.

This in a nutshell this is the way how the paradigms are generated. Now, we will present the problems connected with different parts of speech in Slavonic languages.

3.2 Nouns

Although Slavonic nouns have typically 14 forms, and a maximum of 21 forms, it is due to their numerous alternations and peculiarities of particular units and/or declination types, in particular with masculine nouns, that their generation was one of the most serious problems we had to cope with. Most of these peculiarities are accounted for in the label. The main rule can be seen from the example 'selo' above, the additional information to solve problems stemming from peculiarities of lexical units or declination types and to make the rules more effective in all cases include sign * denoting any character, signs representing different phonemic categories, like K - consonant, V - vowel, etc., '-in' and '+ov'- to account for a short and a long plural in Serbo-Croatian, attaching the sequence 'pl' after information about the gender if the noun is plural only (mpl;, fpl;, npl;), sign # for animate masculine nouns, and ## for masculine personal (only in Polish). It is important that the form of the label is flexible and the amount of information in it depends on concrete language and lexical items. For example, we needed information about nominative plural of the Polish masculine nouns, which was not necessary with Serbo-Croatian. The use of such varying information in the labels can be seen from the following Serbo-Croatian dictionary entries and their corresponding rule heads

<u>Dictionary entry</u>	<u>Rule head</u>
kost,i f;	SCR PARA *K,i f =>
gost,a # m;	SCR PARA *K,a # m =>
Arapin,a # -in m;	SCR PARA *K,a # -in m =>
skok,a +ov m;	SCR PARA *K,a +ov m =>
novac,vca m;	SCR PARA *KaK,KKa m =>
sanke,i fpl;	SCR PARA *e,i fpl =>

The cases of so called morphological synonymy are accounted for by stating both (or several possible forms). For example we have:

```
SCR PARA *K,i f =>
  NOUN;
  FEMININE;
  O1=(1->');
  SINGULAR;
  ...
  INS=O1+i | PAL(O1)+u;
  ...
```

to account for kosti and košću, masti and mašču, etc. in the instrumental singular.

Comment: Ovoga još nema u pravilima, ali ću napraviti da bude. Ako ti ne odgovara ova forma, promijeni i javi mi da tako stavim i u pravila. Dobro naglasi kako je rješavana homonimija padeznih oblika dolje u dijelu o prevodu recenica.

3.3 Pronouns

Pronouns (ja,ti,on,ona,ono,mi,vi,oni,one,ona) are a closed system and therefore rules to generate them are simple. In most instances, the head of the rule consists of the pronouns themselves, and the forms listed in the body, like in the case:

```
SCR PARA ja =>
  PRONOUN;
  PERSONAL;
  FIRST PERSON;
  SINGULAR;
  NOM=ja;
  GEN=mene | me;
```

...

(only the beginning of the rule has been shown)

3.4 Adjectives

In general labels for adjectives are much similar than those used for nouns, and in most of the cases only endings for masculine, feminine, and the neuter form were necessary, like in the Polish examples: drogi,a,ie;. or młody,a,e; The rules used to generate their paradigms were very similar to those used for nouns, and problems with alternations are encountered primarily with the Serbo-Croatian comparative. The following two parts of the rule for the adjectives like młody, shows how this has been done.

```
POL PARA *VKy,a,e =>
  ADJECTIVE;
  O1=(1->','-1);
  SINGULAR;
  MASCULINE;
  NOM=O1+y;
  GEN=O1+ego;

  ...

  COMPARATIVE;
  O1=(1->','-1)+sz;
  SINGULAR;
  MASCULINE;
  NOM=O1+y;

  ...
```

It was necessary to attach the sign ‘-’ to those adjectives which have an analytic comparison, and the sign ‘--’ to those which cannot have any comparison at all to the dictionary label of such adjectives. This requires some additional effort on the part of lexicographers, but was necessary to make sure that comparison forms were generated only there where they exist.

3.5 Numerals

Since numerals form a closed system of basic units, and the number of complex units was extremely high, the rules are designed to produce complex numerals from the basic ones, rather than keeping both

basic and complex ones in the dictionary. This makes their place specific within other parts of speech.

Here is a part of rule used to generate cardinals in Serbo-Croatian

SCR CARDINAL number =>

```

ZERO;
  0=nula;
ONE;
  1=jedan;
  2=dva;
  3=tri;
...
TEEN;
  10=deset;
  11=jedandaest;
  12=dvanaest;
  13=trinaest;
...
TEN;
  20=dvadeset;
  30=trideset;
...
HUNDRED;
  100=sto;
  200=dvesta | dvjesto;
  300=trista | tristo;
....
THOUSAND;
  1000=hiljada | tisuća;
  2000=dvije (hiljade | tisuće);
  3000=tri (hiljade | tisuće);
...
number= THOUSAND+HUNDRED+(TEN+ONE | TEEN)
...

```

When generated, numerals which have morphological paradigm are passed through rules similar to the ones used with nouns - in some cases the head consists of the entry itself (Serbo-Croatian collective numeral *dvoje*, for example), and in some others are very similar or identical to existing paradigm rules; for example those for adjectives.

3.6 Adverbs

Adverbs use only the label 'ad;' and the signs '-' and '--' for the cases where there is no analytic comparison, or no comparison at all. The same rules as with adjectives are used to make a comparative out of a positive, and a superlative out of a comparative.

3.7 Verbs

Finally, the verb paradigms, with their complexity of forms, and their enormous importance in translation required our particular attention. There are two basic ideas in handling labels for verb entries in the dictionary. First, the one already mentioned - the label has to start with the last character shared by two different forms stated in the entry, and second that the label has to be flexible, to allow for additional information if the basic information is not enough to generate the paradigm. In case of the Serbo-Croatian verb *prigovarati,am,aju,ip*; only the infinitive, first person singular and third person plural of present tense was needed to construct the paradigm, whereas in the entry *naći,adem,ađu,ađ,aš,xp*; has also the information needed to generate the aorist and the active participle.

The labels for verb entries are language sensitive in the sense that the information in label is the optimal one for a given language. Serbo-Croatian labels for verbs, in their basic form include the infinitive, the first person singular and the third person plural in the present tense, whereas Polish labels include the infinitive, the first person and the second person singular in the present tense. Of course, labels for Slavonic languages also contain the information on whether the verb is perfective (p;), imperfective (ip;) or bi-aspectual (p+ip;).

The basic idea on how the rule is ordered could be seen from the Polish example *jechać*. It is important to stress that the base is being defined within any verb form. This, along with making the head of the rule more or less specified, depending on the concrete example helped in coping with the intricacies of verb paradigms.

Again, as the rules for all other parts of speech, rules were designed in accordance with principles on which MIG rests. The rules provide minimal information required for the software to operate, and require the lexicographer to invest minimal effort in specifying the labels.

4. MIG Morphological Paradigms at Work

For the purposes of using morphological paradigms described above in the processes of sentence translation, parsing, and qualitative analysis a more precise dictionary is automatically created from the main text dictionary using the MIG morphological rules. This design ensures that the software runs more efficiently.

5. Neural Networks

Neural networks can be used in various parts of a program like NeuroTran. The key is to use them in such a way as to maximize their strengths and minimize their weaknesses. First of all, our observation is that most computer hardware is based around a single processor or at the best around multi-processors. This being the case the natural playground in which neural networks thrive, i.e. purely parallel hardware has been assumed not available except for the most expensive implementations of NeuroTran. Now that we know that we have at least one hand tied behind our back we have to explore those good sides of neural networks which can provide us with a useful edge even on serial machines. We have a wide range of neural network paradigms to choose from: Feedforward neural networks (Back-propagation, etc.), Hopfield neural networks, Boltzman machines, Hebian learning, General Regression Neural Network, etc. As you can see there is a huge array of potential neural network candidates. In general neural network can model and generalize well in the presence of noisy or incomplete multi-dimensional inputs and output both in the binary and real valued domains. This useful property of neural networks is separate from the hardware on which they run.

Let us for a second imagine that we had the perfect parallel hardware on which to both train and use our neural network. By far the most useful future application of neural network in machine translation which is that of a having a single neural network be presented with input-output sentence pairs from already existing aligned translated texts in order to train it to learn the underlying orderliness or mapping that exists between the input-output sentence pairs. A small scale experiment had been done to test the hypothesis that this idea can be realized with some success [Končar 1994]. A slightly more feasible place to use such a huge neural network would be for the purpose of a very accurate, incredibly fast spell checker which would give the best possible spelling suggestion by training the neural network on a spell checking list with each word divided up into syllables given as both input and output during the training phase.

Coming back to reality we are faced with the fact that large networks are out of the question, at least for the time being, and that we must use small or relatively small neural networks trained on relatively small training sets. One place which we have clearly identified is that of using neural networks to learn grammar rules and to use them to suggest the best grammar rule if a rule does not exist for a particular situation. Also, the same neural network can be trained on examples generated from user interaction with NeuroTran in which the user can specify a new rule. Another place we have identified as a good place for the application of neural networks is that of automatically generating dictionary labels for

new vocabulary that the user enters into NeuroTran during its use. After being trained on the existing word endings and their corresponding dictionary labels the neural network learns to generalize and can automatically guess the dictionary label of a new word entered by the user. In both cases the number of training samples is manageable and the networks generated do not have to be large, in fact small ones do quite well. From all the different kinds of neural networks we have experimented with we have found that the General Regression Neural Network [Specht 1991] is the fastest learner and generalizes to a satisfactory degree of accuracy.

7. Conclusion

MIG is an evolving grammar which will encompass a wider and wider array of the world's languages as it grows. By this we mean that we are planning to encompass not only East European languages, but also Western European, Asian and later other languages. The NeuroTran software which uses MIG as its core representation and computing model is software which is evolving in step with MIG. NeuroTran is slowly starting to show signs of maturity in the sense that it is actually becoming a useful software tool. However, much remains to be done in order to start fulfilling our high aspirations and we hope that with the continued support of all who have helped thus far and others that we can bring NeuroTran to a stage where it is very useful.

8. Acknowledgments

We would like to express our thanks to Alexander von Humboldt-Stiftung who enabled Dr. Šipka to work on the usage label network, Vladimir Šipka for all his kind programming work on Dictionary Manager and speech recording, compression and reproduction, all the students and other people who have worked on the Polish and Serbo-Croatian NeuroTran dictionaries, Translation Experts Ltd. for kindly allowing us to present their software at this conference and for providing financial assistance, Grot for providing financial assistance and help with the Polish dictionary.

References

[Barić 1995] Barić, Eugenija et al. *Hrvatska gramatika*. Školska knjiga, 1995.

[Končar 1994] N. Končar, G. Guthrie. *A Natural Language Translation Neural Network*. International Conference of New Methods in Language Processing (NeMLaP), Centre for Computation Linguistics, UMIST, Manchester, United Kingdom, pp. 71-77, 14-16th September 1994.

[Quirk 1993] R. Quirk, S. Greenbaum. *A University Grammar of English*. Longman, 1993.

[Specht 1991] Donald F. Specht. *A General Regression Neural Network*. IEEE Transactions on Neural Networks, pp. 568-575, Vol. 2, No. 6, November 1991.

[Stanojčić 1992] Ž. Stanojčić, Lj. Popović. *Gramatika srpskoga jezika*. Zavod za udžbenike i nastavna sredstva, 1992.

[Šipka 1994] D. Šipka. *Usage Labels Network: An Approach to Lexical Variation*. *Linguistica*, XXXIV,2, p. 31-42, 1994.

[Tokarski 1989] J. Tokarski „Formy fleksyjne”, in Skorupka, Stanisław et al. *Mały słownik języka polskiego*, PWN, p. VIII-XXII, 1989.

[Urbańczak 1984] S. Urbańczak (ed.). *Gramatyka współczesnego języka polskiego*. Morfologia, PWN, 1984.