ON ARGUMENT REALIZATION IN THE APERTIUM PLATFORM

Nikolay Zhelyazkov

Department of Computational Linguistics
Institute for Bulgarian Language
Bulgarian Academy of Sciences

There is a wide range of theories, approaches and interpretations dedicated to the realization of the arguments of a verb. On the one hand, the syntactic and semantic facets of argument realization still pose theoretical challenges to linguists. On the other hand, in the present era of information there is the practical question of how to represent the argument structure in a machine-readable form.The current paper focuses on Apertium, one of the most popular platforms for rule-based machine translation (RBMT).

Key words: argument realization, Apertium, RBMT

Introduction

The predicate is at the core of the clause structure, and hence the relation between verbs and their arguments has been widely discussed. There are a number of theories put forward by Fillmore, Grimshow, Dowty, Jackendoff, Levin, Hovav, Penchev and Koeva, just to mention a few of the most authoritative scientific treatments of the subject (see Levin, Hovav 2005 for a detailed overview of the various theories for English, as well as Penchev 1993 and Koeva 1998 for Bulgarian).

Verbs take a given number of arguments, which is in the realm of argument realization. The argument structure possesses both lexical semantic and syntactic properties. In this respect verbs fall into subclasses, and argument realization also deals with the so-called alternations, e.g. (Levin, Hovav 2005: 2):

1a. The boy broke the window with a ball.
1b. The boy hit the window with a ball.
2a. The window broke.
2b. *The window hit.
3a. Perry broke the fence with the stick.

3b. Perry broke the stick against the fence.
4a. Perry hit the fence with the stick.
4b. Perry hit the stick against the fence.

where one can see that some of the pairs of sentences are not grammatically acceptable, or at least are not complete paraphrases. Furthermore, in these examples we have the subclasses of *break* verbs and *hit* verbs. Despite the vast research so far, the above-mentioned theories diverge in their views on the specific semantic facets involved in the formation of the exact meaning of the resulting structures.

Additionally, the lexical items are contained in the lexicon. It provides categorial information about parts of speech and subcategorial information in the following form:

eat [NP __ (NP)]

The example above shows a subcetegorization frame. A key concept introduced by Chomsky is that the implementation of subcategorization frames makes the use of other selective rules redundant (Koeva 1998: 212), which is of uppermost importance to NLP as well, providing „scaffolding“ for the machine-readable representation of argument structure.

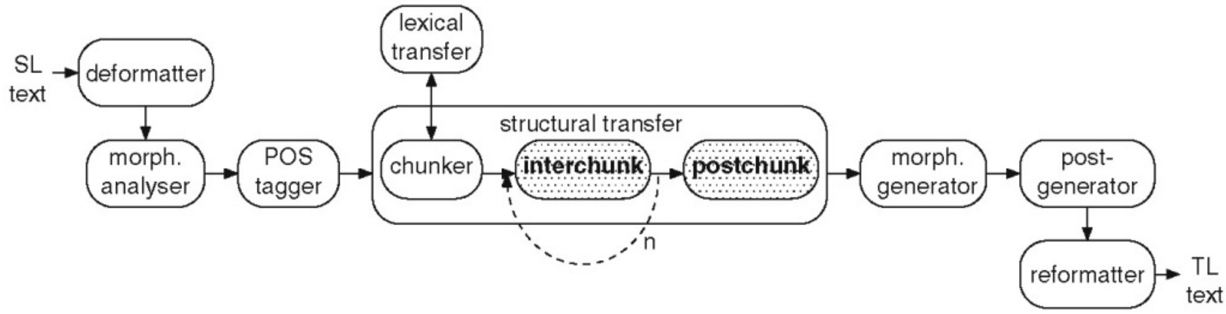
Apertium

Machine translation (MT) is the ultimate goal in NLP and it comes as no surprise that a slew of technologies compete in this field. The present age of big data and the wide availability of language data resources presumably make statistical machine translation the obvious choice. Understandably, here the typical example is Google Translate. In fact, analyzing large data sets and drawing patterns, trends and conclusions from them have become a key focus not only in the realm of NLP. Large organizations amass considerable quantities of data – to a great extent of human origin – which should be structured and interpreted beyond the purpose of translation. Hence, it is not uncommon for NLP specialists to act as data analysts and vice versa. Next comes another popular method known as rule-based machine translation (RBMT). In RBMT systems, linguists define transfer rules, which entails a large number of hand-coded definitions. Hybrid MT systems constitute a third major approach occupying the middle ground between the statistical MT and RBMT platforms – an in-between solution, where statistically produced translation is calibrated implementing linguistic rules. Actually, some specialists consider Google Translate to be a hybrid system; however, since it is not an open-source platform, one can only guess.

It is clear, both to common users and to linguists alike, that MT systems are far from being immaculate and each technology has its pros and cons (Forcada 2011: 128):

- Statistical MT systems often output translations which are more natural than those produced by RBMT systems, but less faithful to the original. Statistical MT attempts to balance, on one hand, the probability that the words of the translation correspond to those of the original sentence (fidelity) and, on the other hand, the probability that the words of the translated sentences are those and in that order in the target language (fluency). It happens sometimes that the latter outweighs the former: the result is a deceptively fluent translation which, however, is not faithful to the original. This is very unlikely to happen with RBMT systems.
- RBMT systems tend to produce translations which are more mechanical, sometimes less fluid and more repetitive, so that their errors tend also to be more repetitive and usually very evident, due the absence of any mechanism for smoothing the resulting translation to make it more fluent. This eases the work of posteditors, who tend to prefer MT systems that are predictable because of being repetitive. Another advantage of the RBMT systems is terminological consistency. Whereas RBMT systems produce the same equivalent (or an equivalent from a small list of candidates if the system includes a module for lexical selection) for the same words across the text, statistical MT systems may translate the same word in different seemingly random ways as they choose translation equivalents according to the translation probability of the whole sentence, or may have been trained on corpora which are not entirely parallel.

Apertium is a member of the rule-based branch. Moreover, it is among the most popular systems for RBMT. Apertium is an open-source RBMT platform, with a huge community developing and fine-tuning the following key modules (Forcada 2011:131):



- A deformatter which encapsulates the format information in the input as superblanks that will then be seen as blanks between words by the rest of the modules.
- A morphological analyser which segments the text in surface forms (SF) and delivers one or more lexical forms (LF) consisting of lemma, lexical category and morphological information.
- A statistical PoS tagger which chooses, using a first-order hidden Markov model (HMM), the most likely LF corresponding to an ambiguous SF.
- A lexical transfer module which reads each SL LF and delivers the corresponding target-language (TL) LF by looking it up in a bilingual dictionary encoded as an FST compiled from the corresponding XML file.
- A structural transfer module, which consists of three sub-modules:
  - A mandatory chunker.
  - An optional interchunk module which performs longer-range operations with the chunks and between them.
  - An optional postchunk module which performs finishing operations on each chunk and removes chunk encapsulations so that a plain sequence of LFs is generated.
- A morphological generator which delivers a TL SF for each TL LF.
- A post-generator which performs orthographic operations using an FST generated from a rule file.
- A reformatter which de-encapsulates any format information.

The lexicon is represented as monolingual and bilingual dictionaries:

- The monolingual dictionaries contain the morphological information used in determining the correspondence between lexical forms and surface forms.
- The bilingual dictionaries are used in the lexical selection rules. The dictionary resources of Apertium are grouped into language pairs.

Argument structure and rules

The argument structure and the rules in Apertium are implemented in the structural transfer stage. It involves chunking and shallow transfer. For instance, the simple sentence:

I saw a signal

becomes

```
^prpers<prn><subj><p1><mf><sg>$
^see<vblex><past>$
^a<det><ind><sg>$
^signal<n><sg>$.
```

and could be further transformed by the rule:

```
SN SV SN<nom> -> SN SV SN<acc>
```

As a matter of fact, Apertium uses its own nomenclature: SN, SV, AdjP and PP for NP, VP, AP and PP.

The language resources of Apertium (dictionaries, rules, etc.) are defined in XML format. Therefore, an example scaffolding of the above rule is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<transfer>
<section-def-cats>
<def-cat n="some_word_category">
<cat-item tags="mytag.*"/>
</def-cat>
</section-def-cats>
<section-def-attrs>
<def-attr n="some_feature_of_a_word">
<attr-item tags="myfeature"/>
<attr-item tags="myotherfeature"/>
</def-attr>
</section-def-attrs>
<section-def-vars>
<def-var n="blank"/>
</section-def-vars>
<section-rules>
<rule>
<pattern>
<pattern-item n="some_word_category"/>
</pattern>
<action>
<let><clip pos="1" side="tl" part="some_feature_of_a_word"/><lit-tag v="myotherfeature"/></let>
<out>
<lu><clip pos="1" side="tl" part="whole"/></lu>
</out>
</action>
```

```
</rule>
</section-rules>
</transfer>
```

Conclusion

The subject in question in the current paper is part of ongoing research based on Apertium. The provided theoretical framework, coupled with the popular RBMT platform, should contribute to the main goal of the research, which is to define transfer rules for verb phrases in the English-Bulgarian language pair.

There is no doubt that in the realm of big data the NLP methods are predominantly statistical. Nevertheless, the above-mentioned work on Apertium is conducted with the true conviction that RBMT systems have their rightful place both as stand-alone platforms and as fine-tuning tools in hybrid systems.

References

Forcada 2011: Forcada, M. et al. Apertium: a free/open-source platform for rule-based machine translation. // *Machine Translation*. Volume 25, Issue 2, 2011, 127–144.

Koeva 1998: Коева, Св. Аргументна структура, тематични отношения и синтактична реализация на аргументите. // *Езиково съзнание*. Ред. Ст. Димитрова. София: Наука и изкуство, 1998, 206–230.

Levin, Hovav 2005: Levin, Beth and Malka Rappaport Hovav. *Argument Realization*. Cambridge: Cambridge University Press, 2005.

Penchev 1993: Пенчев, Й. *Български синтаксис – управление и свързване*. Пловдив: УИ на Пловдивския университет, 1993.