Two Approaches to Aspect Assignment in an English-Polish Machine Translation System

Anna Kupść
Polish Academy of Sciences, Institute of Computer Science and
Carnegie Mellon University, Language Technologies Institute
aniak@cs.emu.edu

Abstract

The paper presents two approaches to aspect assignment in a knowledge-based English-Polish machine translation (MT) system. The first method uses a set of heuristic rules based on interlingua (IR) representation provided by the system, whereas the other employs machine learning techniques. Both methods have similar performance and obtain high accuracy of over 88% on test data. The crucial difference, however, is the development effort: the machine learning technique is fully automatic, whereas heuristic rules are derived manually.

1 Introduction

The paper presents two methods to deal with aspect assignment in a prototype of a knowledge-based English-Polish machine translation (MT) system. Although there is no agreement among linguists as to its precise definition, e.g., Vendler(1967), Comrie (1976), Dowty (1986), aspect is a result of complex interplay of semantics, tense, mood and pragmatics and it strongly affects overall text understanding. In English, aspect is usually not explicitly indicated on a verb. On the other hand, in Polish it is overtly manifested and incorporated into verb morphology. This difference between the two languages makes English-Polish translation particularly difficult as it requires contextual and semantic analysis of the English input in order to derive aspect value for the Polish output.

The MT system presented in this paper takes advantage of a knowledge-based interlingua (IR) representation in order to assign aspect in Polish translation. We propose two approaches based on this representation. First, we provide a set of human-defined heuristic rules (similar to 'cues strategy' presented in Gawrońska (1993)), and second, we use machine learning techniques to learn aspect assignment rules. The former approach has been incorporated into the system, whereas the latter has been, so far, run separately as an experiment. The results obtained by both methods are quite similar. The crucial difference, however, is the effort put into their development: the machine learning approach is fully automatic and rules are derived from examples rather than hand-coded.

The organization of the paper is as follows: section 2 briefly presents the system architecture, sections 3 and 4 describe heuristic rules and the machine learning approach, respectively. Finally, section 5 contains conclusions.

2 System description

The English-Polish MT project presented in this paper is an extension of the existing multilingual KANTOO system (a reimplementation of the KANT system, cf. Mitamura et al. (1991), Mitamura and Nyberg (1992)) developed at Carnegie Mellon University. KANTOO is a knowledge-based, high-quality, domain-specific MT system (in the English-Polish MT project, the domain is restricted to printer manuals) and it uses Interlingua (IR) as a semantic representation, see Leavitt et al. (1994). The system takes as an input a text written in constrained English (controlled language), which limits vocabulary and grammar of sentences accepted by the system, cf. Kamprath et al. (1998). Example (1) presents a sample English input along with the IR representation and its Polish translation provided by the system.

```
(1)   The printer prints pages.

      (*A-PRINT
        (agent
          (*O-PRINTER
            (number singular)
            (reference
                  definite)))
        (argument-class
                 agent+theme)
        (mood declarative)
        (punctuation period)
        (tense present)
        (theme
          (*O-PAGE
            (number plural)
            (reference
              no-reference))))

      Drukarka drukuje strony.
```

IR illustrated in (1) is the input for the Polish generation module. The module consists of four components: a mapper, a unification grammar (a type of context-free grammar), a morphological generator and a post-processing module. Mapping rules transform the IR semantic representation into a syntactic structure corresponding to the Polish output. The structure is a functional structure or FS in the LFG (Lexical Functional Grammar) formalism, cf. Bresnan (1982). Generation grammar rules convert this FS into a list of lexical tokens (FS frames), which are then fed to the morphology module responsible for generating appropriate inflected forms. Finally, a set of post-processing rules is applied to produce the resulting surface form of translation by cleaning up spacing, adding capitalization, inserting punctuation, etc. In order to develop the current system, a small corpus of about 280 English sentences from a printer manual has been examined. This corpus served as a baseline to develop the two approaches to aspect assignment presented in the paper.

As mentioned above, aspect is incorporated into verb morphology in Polish. Polish verbs may have two aspect forms: imperfective, e.g., *drukuje* 'prints', or perfective, e.g., *wydrukuje* 'will print$_{3.sg}$ (out)'. Aspect is independent of tense or mood as it is also present on infinitives: *drukować* 'to print$_{imperf}$' and *wydrukować* 'to printer$_{perf}$ (out)', or on gerunds: *drukowanie* 'printing$_{imperf}$' and *wydrukowanie* 'printing$_{perf}$ (out)'.

Since English verbs do not have morphological aspect, we consider lexical concepts, e.g., `*A-PRINT` in (1), ambiguous: they can be translated by either a perfective or an imperfective verb, see (2).

```
(2)   *A-PRINT = ([?verb] drukowac)
      drukowac =
      (*OR* ((morph verb-imperf)
             (root drukuje))
            ((morph verb-perf)
             (root wydrukuje))),
```

The role of aspect assignment rules is to specify which form to use in translation. The next two sections describe two methods which provide such rules based on IR specification.

3 Heuristic rules

Heuristic rales are specified in the mapper and they assign aspect according to attributes found in IR. The rales are ordered so that more general cases are considered first and if they do not hold, more specific rules are applied. Aspect assignment rules proposed in the system are presented below.

3.1 Declarative Mood

For finite verbs in declarative mood, aspect primarily depends on tense. First, all continuous forms, marked as `(progressive +)` in IR, are translated as imperfective. Next, forms of perfective tenses, i.e., `(perfective +)`, are translated as perfective. Then, verbs in simple past, (`tense past`), or future simple tenses, (`tense future`), are translated as perfective. Similar assignment rules have been proposed in Gawronska (1993).

Additionally, we assume that certain types of subordinate conjunctions, e.g., 'while', 'once', 'before', etc., impose aspect requirements on a verb in the subordinate clause. The following assignments have been proposed:

- 'while': imperfective

(3) You can send an electronic fax **while** the printer makes copies.

Można wysłać elektroniczny faks,
can send_{inf} electronic fax
podczas gdy drukarka robi
while printer makes_{imperf}
kopie.
copies

- 'once', 'after', 'before', 'until': perfective; additionally clauses introduced by the conjunction 'until' have to be negated in Polish

(4) Jobs also queue and wait **until** another job finishes.

Zadania także ustawiają się w kolejce
jobs also stand REFL in queue
i czekają, dopóki inne zadanie nie
and wait until another job not
skończy się.
finishes_{perf} REFL

- 'by'+gerund: imperfective; such clauses are translated into Polish by a contemporary adverbial participle derived only from imperfective verbs, see Saloni and Świdziński (1985)

(5) Close the document **by selecting** `Close` from the `File` menu.

Zamknij dokument wybierając
close document $\text{selecting}_{imperf}$
`Zamknij` **z** menu `Plik`.
`Close` from menu `File`

If none of the above cases hold, we assume that aspect of present tense verbs is imperfective. This assignment is valid also for gerunds as they are represented in IR as present tense verbs with an additional attribute (`nominal` +).

3.2 Imperative Mood

After a brief analysis of Polish technical documentation, we decided to condition aspect in imperative mood on negation. Negated imperatives more often appear with imperfective forms (86.2%), whereas perfective aspect prevails with non-negated imperatives (83.5%).

Heuristic rules used in the system conform with the above statistics: we translate non-negated imperatives as perfective, (6), and negated imperatives as imperfective verbs, (7).

(6) Print a test page.

Wydrukuj stronę próbną.
print_{perf} page test

(7) Do not move the lever after the scanner has begun sending the page.

Nie przesuwaj dźwigni, gdy skaner
not move_{imperf} lever when scanner
zaczął wysyłanie strony,
started sending page

3.3 Infinitives

Infinitives have no mood or tense specified and we need separate rules to resolve aspect of these forms. In general, English infinitives appear as either complements of other verbs, e.g., modals, or they head infinitive clauses introduced by a conjunction such as 'in order to'. We assume that in the former case, aspect of the infinitive depends on the governing verb while in the latter — on the subordinate conjunction.

For the conjunction 'in order to', we assume that it requires a perfective infinitive argument, (8).

(8) You must unhook the other device **in order to** connect the printer.

Trzeba wyłączyć inne urządzenie,
need unhook_{perf} another device
aby podłączyć drukarkę,
in order to connect_{perf} printer

Modal verbs are represented in IR by a set of semantic attributes such as ability, possibility, tentativity, necessity, obligation, see Leavitt et al. (1994). The following aspect assignment has been adopted in the system:

- 'can', (ability +) or (possibility +) : perfective;
- 'cannot', (ability +) (negation +) : imperfective;
- 'cannot', (possibility +) (negation +): perfective;
- 'could', (possibility +) (tentativity low) : perfective;
- 'may', (possibility +) (tentativity medium) : perfective;
- 'must', (obligation medium): perfective;
- 'should', (expectation +) : perfective.

3.4 Results

As mentioned above, aspect strongly depends on semantic and pragmatic context. Since such information is impoverished in KANTOO, the proposed rules cannot be perfect. In order to evaluate their performance, results obtained by the system have been compared with human translations of the initial (training) English corpus (280 sentences). The heuristic rules have been developed in order to accommodate data in the training corpus. Therefore, in order to obtain a more objective verification of the proposed rules, we additionally tested the system performance on a separate set of 24 (test) sentences taken from the same manual. The results obtained on training and test sets are summarized in Fig. 1.

result	train		test	
	#verbs	%	#verbs	%
correct	430	88.1%	53	88.3%
incorrect	58	11.9%	7	11.7%

Figure 1 : Performance of heuristic rales

4 Machine Learning

The machine learning approach described in this section is also based on the IR representation provided by the MT system. In this experiment, we used the C4.5 software to build a decision tree. Training and test data have been derived from the same sentences the heuristic rules have been proposed for and evaluated on. We have run the experiment twice, using two different measures to build the decision tree: information gain and gain ratio. Performance of both algorithms has been evaluated on unpruned and pruned trees. Additionally, the optimal (pruned) trees have been transformed into rules and their accuracy has been measured as well. Details of the experiment and its results are presented below.

4.1 Data

Data used for training and testing were taken from the same set of sentences the heuristic rules have been applied to. All sentences have been analysed by the KANTOO analyser and the resulting IR served as an input for preparing the data. In particular, we have selected 12 attributes which had been crucial for development of the heuristic rules: ability, expectation, marker, mood, necessity, negation, obligation, perfective, possibility, progressive, tense, tentativity.

Most of these attributes are taken directly from IR, with an exception to marker, which has been introduced to indicate the type of subordinate conjunction, e.g., 'while', 'unless', 'once', etc. Note that not all attributes are specified in IR for every verb, e.g., infinitives do not have the mood attribute. We have slightly modified the mapper to make sure that all 12 attributes required for learning are present for every verb and have their values specified. Values of attributes missing in IR are either set to '-' or none, depending on whether the attribute is binary or has more values. In addition, every verb in the data set has been labelled with the correct aspect value based on the human translation. The resulting 13-tuples served as training data for the decision tree. The target concept (aspect) has been represented by a binary attribute: 0 corresponds to imperfective, 1 to perfective aspect. The test data has the same format.

Due to changes in the mapper, the final number of examples used in the experiment was smaller than in the original system. The decision tree was trained on 417 and tested on 55 examples.

4.2 Decision Trees

As mentioned above, we employed two measures to build a decision tree: information gain, Quinlan (1986), and gain ratio, Quinlan (1986; Quinlan (1993). The main difference between the two techniques is in the size of the resulting tree: the former favours attributes with multiple values, which results in a wider (and usually bigger) tree. Indeed, the tree built according to gain ratio is smaller (31 nodes), Fig. 2, whereas the one based on information gain is slightly bigger (33 nodes), Fig. 3.

```
mood = none: 1
mood = declarative:
|   tense = past: 1
|   tense = future: 1
|   tense = none: 0
|   tense = present:
|   |   progressive = +: 0
|   |   progressive = -:
|   |   |   possibility = +: 1
|   |   |   possibility = -:
|   |   |   |   marker = to-inf: 1
|   |   |   |   marker = while: 0
|   |   |   |   marker = because: 0
|   |   |   |   marker = if: 0
|   |   |   |   marker = until: 1
|   |   |   |   marker = by_ing: 0
|   |   |   |   marker = in-order-to: 0
|   |   |   |   marker = after: 1
|   |   |   |   marker = when: 1
|   |   |   |   marker = unless: 1
|   |   |   |   marker = once: 1
|   |   |   |   marker = so_that: 1
|   |   |   |   marker = none:
|   |   |   |   |   ability = +: 1
|   |   |   |   |   ability = -:
|   |   |   |   |   |   obligation =
                                 none: 0
|   |   |   |   |   |   obligation =
                               medium: 1
mood = imperative:
|   negation = +: 0
|   negation = -: 1
```

Figure 2: Decision tree based on gain ratio

The produced trees turned out to be optimal with respect to the learning algorithm (every node in the tree produced an improvement over the training data) and no nodes were pruned. Evaluation of the decision trees on the training and test data is summarized in Fig. 4.

The error estimate presented in Fig. 4 indicates

Figure 3: Decision tree based on information gain

```
mood = none: 1
mood = declarative:
|   marker = to-inf: 1
|   marker = while: 0
|   marker = because: 0
|   marker = if: 0
|   marker = until: 1
|   marker = by_ing: 0
|   marker = in-order-to: 0
|   marker = after: 1
|   marker = unless: 1
|   marker = once: 1
|   marker = so_that: 1
|   marker = none:
|   |   tense = past: 1
|   |   tense = future: 1
|   |   tense = none: 0
|   |   tense = present:
|   |   |   possibility = +:
|   |   |   |   progressive = +: 0
|   |   |   |   progressive = -: 1
|   |   |   possibility = -:
|   |   |   |   ability = +: 1
|   |   |   |   ability = -:
|   |   |   |   |   obligation =
                              none: 0
|   |   |   |   |   obligation =
                            medium: 1
|   marker = when:
|   |   progressive = +: 0
|   |   progressive = -: 1
mood = imperative:
|   negation = +: 0
|   negation = -: 1
```

measure used in decision tree	error		
	train	estimate	test
gain ratio	9.8%	11.1%	10.9%
information gain	9.6%	10.8%	10.9%

Figure 4: Performance of decision trees

the predicted error rate on unseen examples (the so-called pessimistic estimate): the upper bound of the error based on the observed error on the training data for a given confidence level (set to 95% in the experiment). As shown in Fig. 4, the decision tree built according to gain ratio performed 0.2% worse on training data and it had 0.3% higher error estimate than the information gain tree. The gain ratio estimate overestimates the actual error on test (unseen) data by 0.2%, whereas the information gain estimate underestimates it by 0.1%. Hence, the results obtained by both classifiers are very similar and difference may be attributed to random noise. In order to eliminate this effect, they should be tested on a bigger sample, which was unavailable in the present experiment.

4.3 Automatically Learned Rules

The final part of the experiment consisted in converting the decision trees into rales and verify their performance. Initially, both trees were represented by the same number of rules (21) but after evaluation on the training data, one rule (Rule 18) has been removed from the gain ratio tree. The rules obtained from both decision trees are very similar but they appear in a different order and may have different accuracy, see Fig. 5 and Fig. 6. The rules are grouped according to their output class (i.e., aspect value), ordered with respect to accuracy within this class and applied in the obtained order. Examples to which none of the rules apply fall into the default class, computed individually for each tree. Performance of both sets of rules is identical: 9.4% errors on training and 10.9% errors on test data. Therefore, the learned rules score higher than the heuristic rules which have 11.9% errors on training and 11.7% errors on test data.

Note that the learned rules comprise the heuristic rules discussed in sec. 3. The only exception is Rule 5, which does not take into account negation and misclassifies complements of 'cannot' as perfective. Some of the heuristic rules do not have explicit counterparts among the learned rules. Heuristic rules referring to perfective, tentativity, expectation or the marker 'before' are not overtly present in the decision trees.

```
Rule 4:                  Rule 5:
 marker = to-inf          ability = +
 -> class 1 [99.7%]       -> class 1 [70.6%]
Rule 13:                 Rule 14:
 marker = after           marker = when
 -> class 1 [99.0%]       progressive = -
Rule 7:                   -> class 1 [62.0%]
 obligation = medium     Rule 2:
 -> class 1 [98.3%]       progressive = +
Rule 11:                  -> class 0 [99.8%]
 marker = until          Rule 12:
 -> class 1  [97.5%]      marker = by_ing
Rule 1:                   -> class 0  [99.4%]
 mood = none             Rule 8:
 -> class 1 [96.3%]       marker = while
Rule 15:                  -> class 0 [99.0%]
 marker = unless         Rule 20:
 -> class 1 [95.0%]       mood = imperative
Rule 16:                  negation = +
 marker = once            -> class 0 [98.3%]
 -> class 1 [95.0%]      Rule 9:
Rule 21:                  marker = because
 mood = imperative        -> class 0 [97.5%]
 negation = -            Rule 6:
 -> class 1 [93.2%]       ability = -
Rule 19:                  marker = none
 tense = future           mood = declarative
 -> class 1 [78.2%]       obligation = none
Rule 18:                  possibility = -
 tense = past             tense = present
 -> class 1 [74.6%]       -> class 0 [88.1%]
Rule 3:                  Rule 10:
 possibility = +          marker = if
 progressive = -          -> class 0 [78.9%]
 -> class 1 [73.6%]      Default class: 1
```

Figure 5: Automatic rules for the gain ratio tree

Recall, however, that all these rules resolved aspect to perfective. In the machine learning approach, they are covered by the default rule. Finally, note that in the machine learning approach several new rules have been discovered: Rules 9, 10, 14 and 15 in Fig. 5 (11, 12, 16 and 17 in Fig. 6) do not correspond to any of the heuristic rules.

```
Rule 2:
 marker = to-inf
 -> class 1 [99.7%]
Rule 8:
 marker = none
 tense = past
 -> class 1 [99.0%]
Rule 15:
 marker = after
 -> class 1 [99.0%]
Rule 7:
 obligation = medium
 -> class 1 [98.3%]
Rule 13:
 marker = until
 -> class 1 [97.5%]
Rule 1:
 mood = none
 -> class 1 [96.3%]
Rule 17:
 marker = unless
 -> class 1 [95.0%]
Rule 18:
 marker = once
 -> class 1 [95.0%]
Rule 21:
 mood = imperative
 negation = -
 -> class 1 [93.2%]
Rule 9:
 tense = future
 -> class 1 [78.2%]
Rule 4:
 possibility = +
 progressive = -
 -> class 1 [73.6%]
Rule 5:
 ability = +
 -> class 1 [70.6%]
Rule 16:
 marker = when
 progressive = -
 -> class 1 [62.0%]
Rule 3:
 progressive = +
 -> class 0 [99.8%]
Rule 14:
 marker = by_ing
 -> class 0 [99.4%]
Rule 10:
 marker = while
 -> class 0 [99.0%]
Rule 20:
 mood = imperative
 negation = +
 -> class 0 [98.3%]
Rule 11:
 marker = because
 -> class 0 [97.5%]
Rule 6:
 ability = -
 marker = none
 mood = declarative
 obligation = none
 possibility = -
 tense = present
 -> class 0 [88.1%]
Rule 12:
 marker = if
 -> class 0 [78.9%]
Default class: 1
```

Figure 6: Automatic rales for the information gain tree

5 Conclusions

In this paper, we presented two approaches to aspect assignment in a knowledge-based English-Polish MT system: heuristic rules and machine learning. As for approaches which do not rely on semantics or pragmatics, accuracy of both methods is very high: heuristic rules achieve 88.3% and automatically learned rules 89.1% accuracy on test data. Although the final results turn out to be very similar, the crucial difference between the two methods is the development effort: the machine learning technique acquires rules automatically, while heuristic rales are hand-coded. Another advantage of the machine learning approach is that it allows for more concise encoding of the heuristic rules and discovering new rales.

It has to be noted that the success of the machine learning approach strongly relies on the choice of attributes used for learning. The heuristic rales and the decision trees employ the same attributes. Therefore, human knowledge is necessary to limit the search space in the automatic approach. Another factor which contributed to the high system performance is the restricted domain of translation and use of controlled language. Although some heuristics are quite general (e.g., the rules compatible with those independently proposed in Gawrońska (1993)), the system probably will not be fully scalable to an open-domain unrestricted natural language text. Providing reliable heuristics in a general purpose MT system will be much more difficult than for a domain-specific MT system. On the other hand, having set the learning attributes (or corresponding surface / syntactic patterns), machine learning methods can be successfully applied to automatically acquire rales from annotated data.

Acknowledgments

I wish to thank Krzysztof Czuba, Kathryn L. Baker and two anonymous reviewers of the EACL EAMT Workshop for their comments and suggestions to improve this paper.

References

Joan Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations.* MIT Press Series on Cognitive Theory and Mental Representation. The MIT Press, Cambridge, MA.

Bernard Comrie. 1976. *Aspect. An introduction to the study of verbal aspect and related problems.* Cambridge University Press, Cambridge.

David Dowty. 1986. The effects of aspectual class on the temporal structure of discourse: Semantics or pragmatics? *Linguistics and Philosophy,* 9:37-61.

Barbara Gawrońska. 1993. *An MT Oriented Model of Aspect and Article Semantics.* Lund University Press.

Christine Kamprath, Eric Adolphson, Teruko Mitamura, and Eric Nyberg. 1998. Controlled language for multilingual document production: Experience with Caterpillar technical English. In *Proceedings of the Second International Workshop on Controlled Language Applications (CLAW '98).* Available from http://www.lti.cs.cmu.edu/Research/Kant/claw98ck.pdf.

John R. Leavitt, Deryle W. Lonsdale, and Alexander M. Franz. 1994. A reasoned interlingua for knowledge-based machine translation. In *Proceedings of CSCSI-94.* Available from: http://www.lti.cs.cmu.edu/Research/Kant/.

Teruko Mitamura and Eric Nyberg. 1992. The KANT system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of COLING-92.*

Teruko Mitamura, Eric Nyberg, and Jaime Carbonell. 1991. An efficient interlingua translation system for multi-lingual document production. In *Proceedings of the Third Machine Translation Summit.*

J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning,* 1(1):81-106.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

Zygmunt Saloni and Marek Świdziński. 1985. *Składnia Współczesnego Jeżyka Polskiego.* Państwowe Wydawnictwo Naukowe, Warszawa, 2nd edition.

Zeno Vendler. 1967. *Linguistics in Philosophy.* Cornell University Press, Ithaca, NY.