English Translator –A Bi-directional Polish-English Translation System

Marek Łabuzek, Maciej Piasecki

Wrocław University of Technology, Division of Computer Science
Wybrzeze Wyspiańnkiego 27, 50-370 Wroclaw
{labuzek,piasecki}@ci.pwr.wroc.pl

Abstract. The paper presents the structure of a bi-directional Polish-English machine translation system. Since it was created mostly as a commercial product, such aspects as speed of translation, dealing with ungrammatical texts and time and cost of developing the system are of big importance. An attempt of solving these problems by the application of Machine Learning techniques in parsing and tagging is discussed.

1 Introduction

The paper describes translation process and data used by a commercial, wide-scale machine translation (MT) system called *English Translator* (shorten further to *ET*), created by *Techland* company (Poland, Wrocław). The system was planned from its very beginning to be fully automated and was designated for wide-market. Naturally, the work presented here has more technical than scientific character. However, some experimental techniques being applied in the construction of the system and created data sets, which can be further utilised in the research, makes the subject different from the mere technical report.

Because of the expectations of the short period of development and limited resources, the main assumption underlying the system construction was to apply *Machine Learning* methods in as large extent as possible. The starting point was the work of Hermjakob (1997) on parser and lexical transfer based on inductive learning. His approach allowed for having first result fast, but further improvement of quality of translation demands increasing number of resources, while slowing down the progress.

ET has the typical architecture of the *MT* system based on transfer with the following subsequent stages of processing:

1. *Text segmentation* based on grammar implemented as *Finite States Automata*.
2. *Morphological analysis*.
3. *Part of Speech Tagging* (PST).
4. *Parsing* almost completely based on inductive learning of 'parsing strategy'.
5. *Transfer*: implemented in the form of a set of rules.
6. *Target syntactic structure synthesis*.
7. *Word form generation.*

They will be described in following sections.

2 Text Segmentation

The text is first segmented into blocks, which are words, punctuation marks, numbers and others. Each kind of block is described by a regular expression, which is compiled into a finite state automaton. The text is given as input to all automatons and the one, which accepts the longest part of the text, is chosen. It also returns the position of a last accepted character and searching for a next block starts with successor of this character. For each block, the information about its format (font size and type, colour) is stored with it. It is used to preserve the layout of the text.

The blocks most difficult to describe are abbreviations with periods and various symbols. The former have to be listed one by one for each language. The latter are further divided into subcategories (e.g. Internet addresses, dates, Roman numbers) and for each there is one or more regular expressions describing it.

After obtaining each block, it is checked whether it terminates the sentence. It is determined by one hand-crafted finite state automaton. It deals with balanced delimiters (parenthesis, quotes). The very hard problem is period not terminating a sentence (after ordinal numbers e.g. "1." or in abbreviation at the end of a sentence, where the period plays two roles: being part of an abbreviation and terminating a sentence).

3 Morphological Analysis

The morphological analysis unit uses a monolingual dictionary storing all known words together with morphosyntactic information. It will be called a compressed layer. To present the information in the lexicon to other parts of the system in a consistent way, a second universal layer is created on demand for given words.

The compressed layer groups words into parts of speech and lexemes. It consists of three parts: inflection, lexical information and derivations. For each part of speech, there is a list of inflection forms and a list of lexical information aspects, which mostly describe its syntactic properties such as e.g. reflexiveness of the verb or gradation type of adjectives and adverbs. Lexemes have unique identifiers, which are numbers and are used in other databases of the system. They also have a list of values of lexical aspects corresponding to the part of speech to which they belong.

The inflection part of the monolingual dictionary is implemented as a transducer translating between a word form and a pair: an identifier of a lexeme and a code of inflection form. The construction of the transducers is based on simplified Daciuk (1998) algorithm and dedicated techniques of binary compression of the files. In the Polish monolingual dictionary, we have currently above 180,000 lexemes which gives above 2.5 million forms. The compiled file of transducer has 20MB, which is 30% of a source file. English dictionary has 55,000 lexemes, 150,000 forms and takes 2.2.MB (64% ratio). The lexical information part is a simple table of compressed values of lexical aspects.

The derivation part of the monolingual dictionary stores links between lexemes. Currently, we have links between verbs and deverbal nouns, verbs and four participles available in Polish (of course concrete lexemes can have less than five mentioned

links) and perfective and imperfective version of a verb. These links are necessary for a proper transfer of tenses and other grammatical structures.

The source for compressed layer is generated from a set of text files. They contain a list of parts of speech, inflection and lexical aspects proper for each part of speech, inflection tables and a list of base forms annotated with a part of speech, values of lexical information aspects and names of inflection patterns. These descriptions have also annotations, which describe how to convert the information into universal layer.

The universal layer describes words in a hierarchical way. One syntactic element, describing one word, consists of a word form and a set of syntactical alternatives. A syntactic alternative consists of an identifier of lexeme, a code of basic syntactic category and a set of morphological alternatives. And a morphological alternative is a set of pairs: attribute and value, where attribute can be either inflectional or lexical. An example of the representation is presented below:

```
Surface: "chodzenie"
  (    Syntactic Category: ODS-NOUN
       Semantic Class: 14060
       ( PERSON: F-THIRD-P, CASE: F-NOM, NUMBER:
                F-SING, GENDER: F-NEUT, NEG: F-NEG-N )
    ( PERSON: F-THIRD-P, CASE: F-ACC, NUMBER: F-SING,
                         GENDER: F-NEUT, NEG: F-NEG-N )
    ( PERSON: F-THIRD-P, CASE: F-VOC, NUMBER: F-SING,
                        GENDER: F-NEUT, NEG: F-NEG-N ) )
```

A set of *basic syntactic categories* (BSC) was proposed. The set is extended in comparison to a typical list of Polish *parts of speech* and the categories from the set are also a part of the grammar of the parser. All syntactic categories are organised into hierarchy by explicitly defined *subsumption* relations, e.g.:

```
NOUN: ODS-NOUN, PN, PRON
PRON: PER-PRON, PRON-NPER, PRON-ZPR, PRON-ZWR, PRON-
NEG, PRON-DEM.
```

In the example NOUN has three subcategories: deverbal noun, proper noun and pronoun. PRON, in turn, has six subcategories: personal pronoun, indefinite pronoun, interrogative pronoun and others.

The subsumption relations determine the set of morpho-syntactic attributes assigned to the categories. Some categories of the higher levels are motivated by the correspondence between Polish and English grammar or have a character of semantic subcategories. The subsumption relation is next used in machine learning algorithms.

4 Part of Speech Tagging

The initial version of *ET* did not have a tagger – it was implicitly included in the parses (see next section). However, problems with quality of the parser, forced to look for the improvements by the introduction of the tagger. This late decision have made a lot of problems with the adaptation of the purchased *Penn Tree Bank* (PTB), which had been chosen as the base for the construction of the tagger. The system of syntactic

categories of the parser, being close to the one proposed in *XTAG* (1999), had been defined before purchasing PTB. It was necessary to convert syntactic categories of PTB to *ET* standard. The problem was so serious that a sophisticated expert system had to be constructed to perform the task, which anyway could not be done completely by it. The patterns of subcategorisation of multiword verbs and phrasal verbs concerning the tagging of words as prepositions and adverbs are not explicitly given in *PTB*. Moreover, they seem to be very unstable across the corpus.

PTB includes also a lot of mistakes of different types strongly influencing the final quality of the tagger. There are quite a big number of ambiguities left in PTB tags, at least two different historical versions of the system of taggs can be met (e.g. word "to" earlier tagged as TO, now having two different tags) and, finally, many simple mistakes (e.g. pronouns tagged as determiners and vice versa).

Besides construction of the expert system, in order to correct mistakes, a lot of manual disambiguations and corrections had to be done (up to 1.5% of all words of PTB, even, still leaving the problem of subcategorisation translation unresolved) resulting in, at most, a half of PTB being usable.

The constructed English tagger is combination of the purely statistical *Hidden Markov Model* based solution in initial phase and Brill's tagger associated with hand coded rules in the main phase. The overall accuracy of the tagger is almost 97%.

In the case of Polish, there are many morpho-syntactic ambiguities of three kinds – a word: is a form of different lexems of the same BSC, is a form of lexems of different BSC, represents different forms of the same lexems. The first two kinds are significantly less frequent than in English but the third one is very frequent. However, the construction of the Polish tagger appeared to be much more difficult task, mainly because of the lack of big, annotated corpus of Polish. Some activities in order to build the Polish corpus have been undertaken. Firstly, the 'rough' corpus (~2GB) has been collected from all publicly available sources of electronic texts, mainly from web pages. Then, all words in a part of it have been manually annotated and next assessed by a human supervisor. Annotation of the corpus gave also a good opportunity for elimination of mistakes from the monolingual dictionary together with the introduction of many new lexemes of 'internet jargon' (what is positive according to the typical area of *ET* application). Nevertheless, the present size of the corpus being about 65.000 tagged and corrected words appeared to be too small for the construction of the tagger. The first estimation gave the size of the full tag set about 1600 different tags. The initial experiments with statistical tagger (similar to the English one) resulted in about 86% accuracy counted in a standard way (in relation to all words), but the percentage of mistakes among the ambiguous words was very high. It seems that the better solution would be introduction of more hand-crafted, disambiguating rules.

5 Parsing

Parsing in *ET* is being done during the three subsequent phases: preprocessing (identification of some phrases), main parsing (based on inductive learning) and 'correcting' parsing (trying to amend some mistakes of the main parser).

During preprocessing, a sequence of analysed blocks is checked whether it contains simple phrases (mostly idioms), words from a user dictionary, words which user changed translations for and syntactic alternatives with small probability (e.g. “take” as noun, “father” as verb). The simple phrases are changed to one block with attributes properly set and information about their translation remembered. For words from a user dictionary and words with a changed translation, the proper syntactic alternative is chosen (helping the tagger and the parser) and information about a translation is also remembered. Morphological alternatives with small probability are simply removed.

The main parser is based on the architecture proposed by Hermjakob (1997). It preserves the general shift-reduce scheme but uses the hierarchical structure of *decision trees* instead of a control table. The decision trees are constructed by the application of a version of C4.5 algorithm of inductive learning. The learning set includes pairs consisting of a vector of values of *features* and a parsing *action*, which was performed. The feature values in each case describe partially the state of the parser (the stack and the input list) in which the given action was performed. Leaf nodes on the input list can be ambiguous according to the syntactic category, all others can be ambiguous only according to morphological attributes.

There are four main types of actions. Two of them are ‘standard’: shift and reduce. However restrictions put on reduce are very weak. It can be applied to many arguments, not necessarily located on the top of the stack (even changing their order). The ‘non-standard’ *add into* action is similar to reduce, but it can insert one node into any place of the structure of another. Finally, the *gap creation* action can create an ‘empty copy’ of some node (i.e. a version of it without a lexem identifier in the head). Some examples of different actions:

```
S I-EN-HAVE
R (-3 -1) AS AUX PRED AT -2
A (-2 -1) TO (NP -1 BEFORE -2) AS CONJ COMPL
```

The features express such syntactic information as:

- values of morphological attributes of number, gender, verb form etc. (some of them ambiguous in most of the parse nodes),
- details of the structure of nodes e.g. presence of some branch described by the syntactic (and semantic) role or values of attributes of some role filler,
- possible agreement in values of attributes between some nodes,
- *matching*: between subcategorisation pattern of some node and a possible argument (the value of the feature is syntactic role or category of the filler).

The last type is based on detailed subcategorisation dictionary (SCD) and, in the case of ambiguity, a ranking of patterns is heuristically calculated. The matching features together with features based on relatively rich semantic information (semantic class of the lexem and semantic role according to the matching of subcategorisation) decide about a good quality of parsing of sentences from *Wall Street Journal* corpus, reported by Hermjakob (1997). Examples of some features are given below:

```
synt of -3 at verb
np-vp-match of 1 with 2
```

```
syntrole of vp -1 of –2
morphp of f-ger of mod of -1
```

However, creation of the detailed, hand-coded semantic dictionary for unlimited domain in limited time is very unrealistic. The semantic information used in *ET* parsing is reduced only to some classes based on *WordNet* categories of location, time etc. The only 'advanced' features are the ones based on syntactic SCD. The size of SCD for unlimited domain must be relatively very large[1]. Unfortunately, increasing number of ambiguities is correlated with decreasing quality of heuristic matching. Entries in SCD are tree structures with distinguished leaf (signed PRED) node. The identifier of a lexeme kept in the PRED node is used as an index for retrieval. Besides the structure, each tree describes several requested elements like *subtrees*, *syntactic roles*, requested values of morphological attributes (e.g. `_G` annotating the case) and *specific lexems* (extending the key for retrieval). The Polish subcategorisation dictionary, based on Polański (1984), codes additional information concerning the optional elements and groups of optional elements (where at least one element of the group must be realised) and sequences with fixed order. Examples of entries from Polish SCD:

```
SNT { SUBJ NP_N}{ PRED VP {strzec PRED VERB }{się MOD
PART }( OBJ NP_G)}
SNT { SUBJ NP _N}{ PRED VP {dawać PRED VERB }{ IOBJ
NP_D}{ OBJ NP_A}}}
```

The decision structure is built on the base of examples prepared by a human operator. Theoretically, there is no need to create a detailed grammar of the language being parsed. But in practice, it appeared with the increasing number of examples[2] that the probability of inconsistency is very high. Each inconsistency in decision made during teaching causes formation of a 'strange rule' e.g. a decision of the reduction of an object to VP can be activated by the presence of an adverb in some remote position on the stack. Obviously, such strange associations result from the large number of features (135 with the tagger, above 256 without the tagger) in comparison to the number of examples. The teacher must remember all the time what the parser 'sees'. The feature selection is very difficult and probably the best way is by empirical reduction of them. The initial fast improvement in syntax covering of the parser, slows down quickly. Monitoring of the constructions being presented to the parser and consistency preserving forces to maintain some kind of the grammar. The parser does not backtrack. It generates always only one analysis and stops often encountering unknown combinations of feature values. A mechanism of 'pushing forward'by a special shift action had to be introduced. The positive is that after 'pushing' some parts of the sentence, the words following the problematic construction can be analysed properly. The speed of the parser is very high: it consumes much less time than other *ET* parts.

However, all the negatives mentioned above (especially the lack of semantic information) have caused that the quality of the parser is very unsatisfactory: the parsing of only 147 on 325 test sentences (typical 'textbook' examples) was assessed positively.

---

[1] The present state is more than 18 000 entries in English SCD.

[2] More than 2000 different English sentences, where even the parsing of the sentences consisting of several words can include more than 30 actions – learning cases.

Original, Hermjakob's version of the parser implicitly includes a POS tagger in its decision structure: nodes on the input list are ambiguous according to their categories and the shift action has to choose between the alternatives. The quality of this implicit tagger is comparable to 'stand alone' taggers (~95%) but it needs a lot of additional learning features. The application of the tagger (described earlier) before parser allowed for significant reduction of the features (256->135), what resulted in identification of many inconsistencies and elimination of many 'strange rules'. Anyway, the relatively good quality of the tagger has not brought any significant improvement in the parser: 97% of accuracy still means that there is almost one mistake in each sentence!

Because the final state of the parser's stack contains very often not a single tree of complete analysis, it was necessary to introduce a special, simplified, *correcting parsing*. It is based on some kind of expert system with powerful rules which try to recognise some more obvious mistakes and join all partial structures into one tree. The last operation facilitates transfer in assigning the proper case to arguments of the verb.

6 Transfer

The main goals of the transfer are to transform a tree created by a parser to a tree, which roughly has target language structure, and to translate words and phrases comprising the tree. It is implemented as recursive functions assigned to syntactic categories. They start from the root of the tree and are translating lower and lower parts of the tree. In this phase some typical parser errors can also be corrected.

Translation of the lexeme only on the base of a bilingual dictionary is very often ambiguous or simply wrong. The results can be significantly improved when we use the subcategorisation context during translation. A bilingual subcategorisation dictionary suits these needs. Moreover, during the transfer we should not only translate a lexeme on the base of its subcategorisation but also we should transform, during the transfer, the whole structure described by the tree from the dictionary.

The bilingual subcategorisation dictionary associates pairs of the trees and delivers additional information controlling the transfer e.g. the information defining the corresponding pairs of arguments, identifying arguments to be deleted or controlling the process of transformation of the source argument into the target in case when their categories differ significantly. A special tool with a graphical interface has been created in order to facilitate the process of definition of the entries in the bilingual subcategorisation dictionary.

Presently, the subcategorisation dictionary contains mainly the verb trees (probably the most important between all other types for the parsing). But also, there are some trees describing adverbial constructions, some compound adverbs, prepositions and conjunctions. An important part of the dictionary, constantly growing, is formed by trees describing *multiword idioms*. Presently dictionary contains about 18.000 trees.

7 Target Syntactic Structure Synthesis and Word Form Generation.

In this phase a tree produced by the transfer is modified to fully conform to target language syntax. The modifications are of three kinds: adding function words like particles (Polish reflexive "się" or English "do"-support), correcting the order of words (e.g. in questions) and setting morphological attributes. The last task is the most complicated: it must not only take into account values set by transfer but also all agreements which especially in Polish a very plentiful and complicated. The values set by transfer are usually propagated to the proper child nodes of a given tree while in case of agreements, morphological values of a main word are firstly raised to the root of the phrase and then propagated to proper child nodes. On the base of values of attributes set in leaves of the whole tree, word forms of the output sentence are generated.

8 Further Development

The system is constantly developed. Various works are being conducted. Some of them concern dictionaries and the transfer rules are also improved. New solutions for parsing are being sought, too. One of the biggest problems is to find the proper balance between the usage of Machine Learning techniques and hand coding. Here important are such aspects like resources both human and linguistic necessary to develop the parser, the quality and speed of parsing and the easiness to improve it. It seems that for a wide-scale translation system Machine Learning techniques are indispensable but they should be carefully designed and supported with significant amount of hand-crafted knowledge.

References

Bień J.S. "Koncepcja słownikowej informacji morfologicznej i jej komputerowej weryfikacji", Wyd. UW, Warszawa (1991).

Daciuk J. Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing. PhD thesis, Technical University of Gdańsk, (1998).

Hermjakob Ulf. Learning Parse and Translation Decisions From Examples With Rich Context PhD dissertation, University of Texas, Austin (1997).

"Nowy słownik poprawnej polszczyzny", red. Markowski A., PWN, Warszawa, (1999).

„Słownik syntaktyczno-generatywny czasowników polskich", red. Polański Kazimierz, Instytut Języka Polskiego PAN, (1984).

Saloni Zygmunt, Świdziński Marek. Składnia współczesnego języka polskiego. PWN, (1998).

Szpakowicz S. Formalny opis składniowy zdań polskich. Wyd. UW, Warszawa, (1983).

Świdziński Marek. Gramatyka formalna języka polskiego. Wyd. UW, Warszawa, (1992).

The XTag Group of Institute for Research in Cognitive Science, University of Pennsylvania. A Lexicalized Tree Adjoining Grammar for English. www.cis.upenn.edu/~xtag (1999)