

Problems of Disambiguation in the Thetos-3 System[†]

Nina Suszczańska, Przemysław Szmal

Silesian University of Technology, Institute of Informatics, Gliwice, Poland

Nina.Suszczanska@polsl.pl

Przemyslaw.Szmal@polsl.pl

ABSTRACT

This paper discusses problems connected with ambiguity of automatic morphological and syntactical text analysis, as well as with ambiguity reduction techniques applied in Thetos-3, a system for Polish text translation into the Polish Sign Language. Special attention has been paid to the morpho-syntactic ambiguity since it influences all the remaining translation stages. It should be stressed that problems of processing are considered in respect to any text in electronic form independently of its nature: hand typed-in, issuing from speech or sign language utterance recognition, automatically generated, and the like.

1. Introduction

Thetos-3 is the next version of the Thetos system [10]. The main task of the Thetos family is translation of Polish texts into the Polish Sign Language. The translation proceeds in two phases. In the first, the linguistic one, on the basis of the input text another text, which is a record of the output utterance, is generated. It consists of Polish words arranged according to the sign language syntax rules. The input text is sequentially passed through the morphologic, syntactic, and semantic analysis. The transfer is done on the predicate-argument representation level. Then a syntactic structure of the output utterance is built, and finally its surface structure is generated. In the second, multimedia translation phase, the generated text is interpreted word by word, and in effect an animated gesture sequence, shown by a specially designed cartoon character, appears on the screen.

We assumed the system to generate a single variant of the output utterance. Yet the interpretation of individual utterance fragments when they are considered out of their context is ambiguous. That holds for all the analysis stages. It's then necessary to equip the translator with mechanisms that could reduce ambiguity on individual stages. We succeeded to solve – at least on the theoretical level – some problems related to that. Unfortunately, the mechanisms we applied are not perfect, what implies the scope of translations to be limited.

[†]This work was supported in part by the Polish Ministry of Science and Higher Education in 2008-2010 under Grant No. N N114 208334.

This paper is devoted to a discussion of problems connected with ambiguity of automatic morphologic and syntactic text analysis and with techniques of reducing it. A special interest has been paid to the morpho-syntactic ambiguity since it influences all the remaining translation stages. It should be stressed that problems of processing are considered in respect to any text in electronic form independently of its nature: hand typed-in, issuing from speech or sign language utterance recognition, automatically generated, and the like.

2. Ambiguities in morphologic analysis results

2.1. General premises

Polish, as every Slavic language, has an extensive flexion. In this connection, the main problem at building a morphologic analyzer is to prepare a fast algorithm working with a large-in-size dictionary. Modeling principles for Polish morphology are elaborated; there exist several morphologic analyzers. They differ both in algorithms they use and – what is the main factor – in the size of their dictionary database, in exactness of the analysis, and in performance (see e.g. [3, 5]). The performance of Morf [7, 8], the morphologic analyzer of the Thetos system, reaches 1670 words per second while its dictionary contains more than 68700 entries (words in basic form); it has been measured using the “Morphology” service of the LAS server¹ [4].

In most cases, the morphologic analysis provides ambiguous results. We can meet ambiguity of three basic types: ambiguity of morpho-syntactic features, lemmas, and analysis methods. Morphologic disambiguation consists in filtering ambiguous morphologic analysis results in order to select a single, proper morphologic interpretation. The task of disambiguation in the morphologic analyzer is in practice almost impossible since the morphologic analyzer has no sufficient knowledge to unambiguously select a set of tags. Moreover, as it can be observed in practice, disambiguation on the morphologic level may prove faulty – only some ambiguity types are worth solving. Ambiguous results are usually passed to next processing levels, where – in a wider context, with higher-level knowledge at hand – it’s easier to find and reject solutions which in given context are improper. In the Morf analyzer, the disambiguation has been executed in a very limited form, only in such degree to not damage the syntactic analysis, which is the next text processing stage. Almost all the weight of resolving the morpho-syntactic ambiguity has been carried on the syntactic level.

2.2. Examples of morpho-syntactic features ambiguity

As an example of morpho-syntactic features ambiguity let’s consider the word *kopiec* (*mound*). The result of the analysis is: *masculine noun in nominative or accusative* (see Fig. 1; in the number that denotes features of words subject to declination, the code for gender occupies the highest position, for case – the middle one, and for number – the lowest); ambiguity is connected with the case.

¹<http://las.aei.polsl.pl/las2>. The services of the LAS server are free, accessible for anonymous users too; however, we recommend to contact the Administrator and register to get password what lifts some using restrictions.

Word	Lemma	Class	Features	Feature description
kopiec	kopiec	1	111	masculine gender, nom. , sg.
	(mound)	noun	141	masculine gender, acc. , sg.

Figure 1. Analysis results for the word *kopiec*: case ambiguity.

The analysis of the word *pewnego* (*certain, sure*) shows ambiguity that refers to the gender; analysis gives four variants for gender: *masculine, neuter, virile, non-virile* (see Fig. 2).

In case of the word *babci* (*grandma*), there appears case and number ambiguity. The analysis gives results shown in Fig. 3.

Word	Lemma	Class	Features	Feature description
pewnego	pewny (certain, sure)	21 adjective	121	masculine gender , gen., sg.
			321	neuter gender , gen., sg.
			441	virile gender , acc., sg.
			641	non-virile gender , acc., sg.

Figure 2. Analysis results for the word *pewnego*: gender ambiguity.

Word	Lemma	Class	Features	Feature description
babci	babcia (grandma)	1 noun	221	feminine gender, gen., sg.
			231	feminine gender, dat., sg.
			261	feminine gender, loc., sg.
			222	feminine gender, gen., pl.

Figure 3. Analysis results for the word *babci*: number and case ambiguity.

Ambiguity in lemmas appears e.g. in case of the word *szybko* – there are two analysis variants here: *adverb* (*szybko* – *quickly*) and *feminine noun* (*szybka* – *small pane*) in *vocative* (Fig. 4). A similar thing holds in case of the word *dobrze* – the analysis defines it as *adverb* (*dobrze* – *well*) and *noun* (*dobro* – *good*) in *locative* (Fig. 5). The word *i* is *conjunction* (*i* – *and*) and *numeral* (Roman “1”) (Fig. 6). As to the word *mamy* – such form has a *verb* (*mieć* – *have*) and a *noun* (*mama* – *mum*); in addition, morpho-syntactic features of the noun are ambiguous (*genitive sg.*, *nominative*, *accusative*, or *vocative pl.*) (see Fig. 7).

A distinct ambiguity kind is connected with the Polish language property that allows for writing some types of word pairs as one word. Let’s take for example the word *jeżeli*, for which the analysis gives the result in two variants: *conjunction*: *jeżeli* (*if*) and *noun+particle*: *jeże li* (*hedgehog ~if*) (comp. Fig. 8). As another example can serve the word *nazywali*, classified either as *verb*: *nazywali* (*they called*) or *verb+particle*:

Word	Lemma	Class	Features	Feature description
szybko	szybko (quickly)	52 adverb	–	–
	szybka (small pane)	1 noun	271	feminine gender, voc., sg.

Figure 4. Analysis results for the word *szybko*: lemma ambiguity.

Word	Lemma	Class	Features	Feature description
dobrze	dobrze (well)	52 adverb	–	–
	dobro (good)	1 noun	361	neuter gender, loc., sg.

Figure 5. Analysis results for the word *dobrze*: lemma ambiguity.

Word	Lemma	Class	Features	Feature description
i	i (and)	unknown or error	–	–
		31 numeral	–	–
		81 conjunction	–	–

Figure 6. Analysis results for the word *i*: class ambiguity.

Word	Lemma	Class	Features	Feature description
mamy	mieć (have)	4 verb	11120	indicative mood, present tense, 1. person, pl., indefinite gender
		221	feminine gender, gen., sg.	
	mama (mum)	1 noun	212	feminine gender, nom., pl.
		242	feminine gender, acc., pl.	
		272	feminine gender, voc., pl.	

Figure 7. Analysis results for the word *mamy*: lemma and case ambiguity.

nazywa li (*it calls ~if*) (comp. Fig. 9). A similar thing is with movable particle gluing. Examples are shown in Figs. 10, 11, and 12.

The background of *analysis method ambiguity* is technical instead of linguistic, which is specific for the previous cases. It is caused by the multi-step organization of the morphologic analysis [7, 8]. Each next step can add details to the previous analysis

Word	Lemma	Class	Features	Feature description	Ending	Remarks
	jeż (hedgehog)	1 noun	112	masculine gender, nom., pl.	-li	
			172	masculine gender, voc., pl.	-li	
jeżeli	jeżeli (if)	8 conjunction	-	-	-li	analysis error
		8a conjunction	-	-	-	

Figure 8. Analysis results for the word *jeżeli*: ambiguity connected with possible gluing of two words.

Word	Lemma	Class	Features	Feature description	Ending
nazywali	nazywać (call)	4 verb	12320	indicative mood, past tense, 3. person, pl.	-
			11311	indicative mood, present tense, 3. person, sg.	-li

Figure 9. Analysis results for the word *nazywali*: ambiguity connected with possible gluing of two words.

Word	Lemma	Class	Features	Ending
małym	mały (small)	21 adjective	151	
			351	
			161	-
			361	
			-32	
			111	-m
			741	

Figure 10. Analysis results for the word *małym*: ambiguity connected with possible movable particle appearance

without deleting the old results. In such case several analysis variants arise; there are both proper and improper ones among them. Let's take for example the analysis results for the word *że* (*that*), shown in Fig. 13.

The analysis basing on the main Morf dictionary gives the result as a conjunction with a type tag 8. The module that completes features on the basis of a semantic dictionary qualifies the type as 8b. Old results will be deleted on the syntactic analysis level.

Word	Lemma	Class	Features	Feature description	Ending
używać	używać (use)	4	4----	infinitive	–
		verb	11311	indicative mood, present tense, 3. person, sg.	-ć

Figure 11. Analysis results for the word *używać*: ambiguity connected with possible movable particle appearance.

Word	Lemma	Class	Features	Feature description	Ending
być	być (be)	4	4----	infinitive	–
		verb			
	bycie (being)	1	322	neuter gender, gen., pl.	–
	noun	8	–		-ć
by (to)	by (to)	conjunction	–		-ć
		a	–		-ć
		particle	–		-ć

Figure 12. Analysis results for the word *być*: ambiguity connected with possible movable particle appearance.

Word	Lemma	Class
że	że (that)	8
		conjunction
		8b
		conjunction
		a7
		particle

Figure 13. Analysis results for the word *że*: class ambiguity.

There also happen more complex cases, where ambiguities of several different types overlap. An example for that is the word *także*; the analysis results are shown in Fig. 14.

2.3. Morphologic disambiguation

As mentioned in the beginning of this section, disambiguation of the morphologic analysis results is done very carefully. However, even on this level, ambiguity can be in some cases reduced. There are a few dozen rules on this level. Let's apply to the example from Fig. 14 two rules, one by one: "If word's class is unknown and the word has the suffix *-że*, reject this variant" and "If the word's class is described in less detail, eliminate it". In effect of such filtration a unique morphologic interpretation of the word *także* will remain: *lemma – także (also), class – 85, conjunction*. In the example in Fig. 13 the

Word	Lemma	Class	Features	Ending
	tak	unknown	–	–ze
	(so)	or error		
także		8	–	
	także	conjunction		
	(also)	85	–	–
		conjunction		

Figure 14. Analysis results for the word *także*: ambiguity connected with possible movable particle appearance.

word *ze* will be passed to the syntactic analysis in two variants: *85, conjunction* and *a7, particle*. The rule: “If word’s class is particle and its ending is *–ć*, mark the interpretation to be eliminated” allows for leaving in the example from Fig. 12 two interpretations instead of four. Similar rules act in cases shown in Figs. 8, 9, 10, and 11; their application allows for passing to the syntax analysis only one morphologic interpretation.

Calling filters has a negative influence on analyzers’ speed of work. That is why in the Polsyn parser of the Thetos system we applied mechanisms for speeding up the execution of rules. For example, such mechanisms are used to group rules, which are similar in action, and to apply principles of minimalism. Ambiguities in interpretation that refer to gender (as e.g. for the word *pewnego*, comp. Fig. 2), number (the word *babci*, Fig. 3), case and to other things, are often reduced on the first level of the syntactic analysis while performing the generalization operation on the features of individual words.

3. Disambiguation on the syntactic processing level

3.1. General premises

Polish belongs to the languages that admit a free word order in a sentence. Polish syntax is complex, e.g. syntactic attributes (describing words) – depending on the content being expressed – may appear both to the left and to the right of the word they describe. In effect of a search for unchangeable rules, upon which an automatic syntax analysis could be based, a syntactic group grammar for Polish (SGGP) has been elaborated [9]. SGGP takes into consideration five analysis levels, five grouping levels that correspond to them, and 16 types of syntactic groups. On the basis of SGGP the Polsyn parser, applied in our system, has been elaborated. Parser’s input data are the results of work of the morphological analyzer; as it was signaled above, they sometimes happen to be ambiguous. Ambiguous results for the same word make so-called homonymic nodes. Ambiguity of results on any grouping level causes homonymic nodes to arise on the level, which is directly higher, and so on until the highest level is reached. In our analyzer, mechanisms aimed to reduce the number of homonymic nodes have been applied. It happens however that it is impossible to solve the ambiguity on the syntactic level – in such case this task is done during the semantic analysis.

When grouping proceeds, ambiguity can be not only transmitted but also multiplied. Algorithms of homonymic nodes elaboration are NP-hard; in consequence, their execution to a great extent slows down the analysis. Moreover, processing time increases in effect of continued analysis of erroneous syntactic groups, which are built from accumulating homonymic elements. In order to reduce the number of unnecessary operations that do not make the solution be nearer, we aim at minimizing the number of homonymic nodes. The reduction on the zero level, where the foundation of the entire syntax edifice is established, is especially important.

3.2. Disambiguation mechanisms

Mechanisms for ambiguity reduction use so-called homonymic filters. There are a few dozen filters in our parser. The filtering algorithms are based on different approaches. Syntactic, semantic, pragmatic, and heuristic filters can be distinguished. Syntactic filters make use of grammar rules, semantic ones – of lexical semantics. Pragmatic filters are based on limitations imposed on translated texts pragmatics. Filters used on different analysis levels differ each from other. One of the reasons for that is the fact that data, upon which filtering algorithms on different levels work, differ in format. Information about selected filters of individual kinds is contained in the next subsections.

Principles of ambiguity reduction can be defined as follows. In the ideal case, from the set of tags in a homonymic node only one correct variant, consistent with grammar rules, has to be selected, all the rest can be rejected as incorrect. Unfortunately, such an ideal scenario happens very rarely. It turns out that – usually – there are more than one correct (i.e. consistent with the respective level grammar) variants. Moreover, on the lower level the knowledge about higher level grammar rules is inaccessible. As a matter of fact, some productions from higher levels are used, but they are the simplest ones and sometimes they are insufficient to get an unambiguous choice. Introduction of more complex rules would be too costly, the time of handling them would be comparable with the time of preliminary syntactic analysis, and even of the semantic one. In our case, there are at least two reasons to give such costly analysis up. First, our system is intended to work in a real-time, and second, performing such a complicated analysis may reduce system reliability. In research for a golden center, a system of result filtering rules for each analysis level has been elaborated.

3.3. Syntactic filters

Disambiguation of syntactic analysis results is performed on the basis of rules. The rules have been implemented in the form of functions that filter tags assigned to syntax groups during the analysis process. Inspiration for writing the rules has issued from linguistic works, first of all from [1]. Other works, e.g. [2, 6], have served for the source while verifying the assumed disambiguation techniques. On five syntactic analysis levels in the Polsyn parser, there are more than hundred rules and a few dozen filters.

Some sample rules used in the first group of filters – the syntactic ones – are given below.

Rule 1. *If the word is a noun and its features require preposition, and if preposition does not appear directly before the word, then this noun in the given context is an incorrect result and should be eliminated.*

- The filter that implements Rule 1 allows e.g. to solve the homonymy of the word *dobrze* (Fig. 5) by elimination of the result that refers to a noun.
- Rule 2. *If the word is preposition, and the next node is a homonymic one, then in this node we select a noun with features that correspond to preposition's requirements, and all other components of the node should be eliminated.*
- Rule 3. *From two interpretations of a noun group that one should be selected as correct, whose derivation tree is more extensive.*
- Rule 4. *If a homonymic node consists of a consistent noun group and an inconsistent one, then the latter should be rejected.*
- Rule 5. *If – while parsing a sentence – two groups are aspiring to be the subject then that one should be selected, which has a unique interpretation.*

3.4. Semantic, pragmatic, and heuristic filters

The semantic filtering rules use lexical semantics and base on a semantic classification of words. The rule „Names marked X do not join with names marked Y” allows for choosing the only interpretation of the group *parki stolicy* (*parks of the capital*), as well as for not linking the words *Gliwice transportu* (*Gliwice of transport*) in the phrase *przejeżdżającego przez Gliwice transportu* (*of transport that passes Gliwice*).

Pragmatic filters work according to assumption that concerns the use of the Thetos system. We assume that the system serves for translation of contemporary Polish. That is why in Polesyn rules that reduce interpretations of endings have been used, what has been mentioned above. Homonymy of the word *jeźeli* from example in Fig. 8 is solved by means of one of pragmatic filters. Another filter rejects interpretation of the word *ulicy* as *masc.-animate gender, nom., pl.* (it is interpretation for the word *ulik* – a kind of *herring*).

Heuristic filters we call ones, which do not use grammar rules or use them in rather a loose way. An example of that can be the rules: “*If the numeral joins with the noun “lat”(years) then mark the interpretation with lemma “rok” (year) as correct*” or “*Any old one*”.

4. Conclusion

The Thetos system is intended to make translations into the sign language – among other things – in the process of dialog with the user. That means that the translation should be precise and it should not contain two- or multifold ambiguities with which Polish texts abound. Ambiguity is a cause for problems with translation of sentences. An example for that can be the analysis of the sentence: *Gdzie Pan mieszka?* (*Where do you live?*). The word *mieszka* has three meanings: the verb *mieszkać* (*to live*), the noun *mieszek* (*small bag*) and the men's name *Mieszko* or *Mieszek*. According to that, three different translations are generated. In the first, the question refers the location where the interlocutor lives. In the second – where the owner of “*mieszek*” (meant as *bag*) is. At last in the third one – where the master of the man named “*Mieszko*” or “*Mieszek*”. At the angle of formal analysis, all these parsing variants are correct (although variants involving the name could be eliminated on account of lowercase letter used to spell the word). To solve such kind of homonymy one needs to understand the text, or – at least – to profoundly analyze the context of the sentence.

In order to eliminate improper values of results produced by individual analyzers of the Polsyn parser, a system of so called homonymic filters has been elaborated. This paper has been devoted to problems of classification of ambiguities that occur during automatic morphologic and syntactic analysis, as well as to algorithmization of respective filters. It has been shown on examples how the filters work.

Calling filters has a bad influence on the syntactic analysis speed, therefore several mechanisms have been elaborated to speed up parser's action. The idea behind some of them is to use only a specified subset of rules on each grouping level, or to put into one filter several rules with similar action. Filtering makes also use from grouping rules.

BIBLIOGRAPHY

- [1] Brill, E. 1995. Transformation-based error-driven learning and natural language processing: A case study of part-of-speech tagging. [In:] *Computational Linguistics, Vol. 21(4)*, Poznań, 543–565.
- [2] Dębowski, Ł. 2001. Tagging and morphosyntactic disambiguation. A review of methods and software. *Prace IPI PAN 934*, Warszawa.
- [3] Hajnicz, E., Kupść, A. 2001. Przegląd analizatorów morfologicznych dla języka polskiego. *Prace IPI PAN 937*, Warszawa.
- [4] Kulików, S. 2003. Implementation of linguistic analysis server for Thetos – Polish text into sign language translator. [In:] *Studia Informatica, Vol. 24(3)*, Gliwice, 171–178.
- [5] Przepiórkowski, A. 2008. *Powierzchniowe przetwarzanie języka polskiego*, Warszawa: Exit.
- [6] Rudolf, M. 2003. Metody automatycznej analizy korpusu tekstów polskich: pozyskiwanie, wzbogacanie i przetwarzanie informacji lingwistycznych. *PhD Thesis, Uniwersytet Warszawski*, Warszawa.
- [7] Suszczańska, N., Forczek, M., Migas, A. 2000. Wieloetapowy analizator morfologiczny. [In:] *Speech and Language Technology. Vol. 4*, Poznań, 155–165.
- [8] Suszczańska, N., Lubiński, M. 2001. Polmorph, Polish Language Morphological Analysis Tool. [In:] *19th IASTED Int. Conf. Applied Informatics – AI'2001*, Austria, 84–89.
- [9] Suszczańska, N., Simiński, K. 2007. The Parser Polsyn. [In:] *Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics 2*, Poznań, 345–349.
- [10] Suszczańska, N., Szmaj, P., Kulików, S. 2004. Continuous Text Translation using Text Modeling in the Thetos System. [In:] *International Journal of Computational Intelligence, Vol. 1(4)*, 338–341.