Shallow Parsing for Automatic Speech Recognition in Polish

Alicja Wójcicka[1], **Bartosz Zaborowski**[2]

Institute of Computer Science, Polish Academy of Sciences
[1]alicja.wojcicka@uw.edu.pl, [2]b.zaborowski@ipipan.waw.pl

Abstract

The paper presents the current results of an attempt to develop a grammar model, which can be used in the process of automatic speech recognition as a means for choosing the best candidates for transcription (rather than for purely statistical methods). In order to achieve the goal, the authors use a shallow parser Spejd and a set of grammar rules developed specifically for the needs of the National Corpus of Polish project, and adjust them to the specific ASR needs. The applied method combines an n-gram language model with a rule-based approach. The article describes modifications of mentioned tools, necessary to resolve this question, as well as results of an evaluation of the method.

1. Introduction

Two main phases can be distinguished in a typical approach to speech recognition. In the first phase a signal is decoded to a candidate or a set of candidates for transcription. This set is often compactly encoded as a word lattice. The second phase aims at selecting the best candidate. Generally the second phase is performed in a statistical way by means of n-grams and Viterbi Algorithm.

The idea behind this project is to use a different method for scoring the best candidates. Instead of relying on an n-gram language model we check if the candidate abides with the rules of the grammar of the natural language.

Good examples of a grammar model for a natural language are models used in syntactic parsers. For this project shallow parsing was chosen, due to better performance. The parser tool being used for this project is a rule-based parser Spejd (Buczyński and Przepiórkowski, 2009). To be precise, we used a modified version of the parser described in (Zaborowski, 2014). The modification enables the parser to process word lattices directly, and thus more efficiently than by parsing utterance candidates sequentially.

The syntax rules used in this project has been slightly expanded over that of the original Spejd tool. In short, a Spejd grammar is a cascade of rules. Each rule is a regular expression pattern equipped with a list of modifying operations. Match patterns allow to specify desired morphosyntactic data of individual segments as well as values of attributes of syntactic structures (e.g. type or heads of syntactic groups). The data model allows to operate on multiple interpretations[1] per segment and marking more than one of them as selected. This helps in carrying out morphosyntactic disambiguation simultaneously with the parsing.

In this article we describe some interesting elements of a parsing grammar which are to be suitable for the assessment of grammatical correctness of candidates.

2. Starting point, preprocessing

2.1. Starting point: the NKJP Grammar

The parsing grammar is based on the Spejd grammar created by (Głowińska, 2012) for the purposes of automatic syntactic annotation of the National Corpus of Polish (**N**arodowy **K**orpus **J**ęzyka **P**olskiego, (Przepiórkowski et al., 2012)). The NKJP Grammar is chosen because it was developed during manual syntactic annotation of the 1-million-words subcorpus of the NKJP and it was tested by linguists who worked on that project. However, the Grammar has to be modified for two reasons. First, it was intended to work on fully and correctly disambiguated texts, since every tag in the 1-million-words subcorpus was checked manually. Second, it was used to parse existing texts; even if they were not completely correct, they were actually said or written by some native speakers of Polish. In our project, there are hardly any established texts to be parsed, since there is only a word lattice of candidates. Most of the proposed word strings are incorrect. The main problem is to correctly identify such candidates that may represent a Polish utterance. At this stage, we focus purely on grammatical correctness and do not evaluate the meaning of the chosen strings, so nonsensical candidates are admissible.

2.2. Morphosyntactic tagging of word lattices

As stated before, the grammar used as a starting point requires the input text to be fully morphosyntactically tagged. Unfortunately, to the best of our knowledge there are no morphosyntactic taggers which tag correctly on word lattices. We took advantage of the ability of Spejd formalism to perform morphosyntactic disambiguation beyond parsing in the same grammar. We started from a morphosyntactic analysis – we used a morphosyntactic analyzer Morfeusz ((Woliński, 2006)) which was already built into the original Spejd version. Disambiguation rules were generated using a Brill tagger for Polish named PANTERA ((Acedański, 2010)). A model for this tagger was built on a manually annotated part of the National Corpus of Polish (1.2 million token, mostly written texts) and then it was converted to a set of Spejd rules. Since Brill taggers need some bootstrap tagging, we equipped the modified Spejd implementation with a simple unigram tagger (trained on the same corpus). As a result, we got an efficient tagger working directly on word lattices, consisting of a few hundreds of Spejd rules. The impact of the tagging on parsing results is discussed in the Section 3.2..

[1]or readings: lexical form + POS tag + MSD

2.3. Removal of rare words

Within the word lattices there were plenty of – mainly short – words that are very rare or entirely not used in contemporary Polish; among them are

1. archaic words, e.g. `iże, szwa, suć, mość`,
2. potential forms, included in the dictionary, but uncommon in texts, e.g. `pawić, susać, motylić`,
3. specialized names, not commonly known (for example biological terms), e.g. `ostrożeń, gać, ostan`.

Since their phonological form consists of very common phoneme sequences, they were often proposed as potential candidates of what has been said. In the initial part of the grammar we placed a set of rules that delete such interpretations as they were very improbable. There are two sorts of these rules:

1. removing whole lemmas (e.g. archaic words, entirely not used),
2. removing only specific forms of a lemma (e.g. Vocativus form of a word `alba`, Eng. *alb*, which is identical with a form of a common conjunction `albo`).

The same problem occurs with abbreviations that are commonly used in written language, but nobody pronounces them as abbreviations (they are uttered in their full form). Because the dictionary includes also abbreviated forms, we have to reject them before parsing.

3. Parsing

3.1. NKJP Grammar Structure

The NKJP Grammar consists of two groups of rules:

1. creating syntactic words,
2. creating syntactic groups.

The first set of rules detects idiomatic expressions, compound pronouns, compound tense forms of verbs, etc.; every token must represent either some syntactic word, or part of such a word. The second set of rules combines syntactic words into larger structures: nominal, adjectival, adverbial, numeral and prepositional phrases. Separate single words are also marked as syntactic groups. The best candidate can be chosen by searching a string that is covered by a minimal number of syntactic groups (words in such a string are grammatically connected). For example, a string *mały kot siedzi na zielonej macie*, Eng. *a small cat is sitting on a green mat*, should be annotated with two syntactic groups:

```
NGa(mały kot) siedzi PrepNG(na
zielonej macie)
```

and should be chosen as a better candidate than an ungrammatical string *małe kot siedzi na zielony macie*, annotated with four syntactic groups:

```
AdjG(małe[Pl]) NG(kot[Sg]) siedzi na
AdjG(zielony[Nom]) NG(macie[Loc])
```

A problem arose when there were many candidates with few groups, which did not build any grammatical structure. For example, we can find in a lattice of candidates a string with only three NG groups:

```
NG(fundusz) + NGs(zespołem, czym,
osią Andzi kła, CIA) + NGs(wrak, dach,
obecne popa wad, wolnego)
```

Eng.: NG(fund[Nom,Sg]) NGs(team[Inst,Sg], something[Inst,Sg], axis of a fang of Ann[Inst,Sg], CIA[Inst,Sg]) NGs(wreck[Acc,Sg], roof[Acc,Sg], present of a priest of faults[Acc,Pl], free[Acc,Sg])

The three NGs consist of a sequence of nouns in a specific case (Nom, Inst, and Acc); each of them is grammatical correct, but the whole string is unacceptable. In order to resolve the problem, we introduced the third group of rules that creates verbal (sentence) structures, marked with the symbol Sent.

The part of the grammar responsible for creating Sent groups is divided into two subparts: 1) ascribing syntactic requirement (of case or preposition) to verbs, 2) building sentence phrases. About 9000 most frequent verbs obtained a value of a new verb category: valency. We used The Polish Valency Dictionary *Walenty* (Przepiórkowski et al., 2014) as a source of information about subcategorization frames. 20 most frequent frames are fully included in the rules (e.g. a verb that governs Nom and Acc gets a tag np4, a verb that governs Nom, Acc and Inst, gets a tag np4np5 and so on), the 30 following are represented only partially (e.g. a verb that requires Nom, Dat and sentence clause gets a tag np3).

In creating the second subpart of the verbal rules it was taken into consideration that the word order in Polish is not fixed, that an ellipsis of some element can appear, and that some free adverbial or prepositional groups can occur between phrases that are governed by the verb. There are also rules responsible for sentence coordination and subordination, which allows to detect more complex structures, e.g. such as

```
ten, który na mnie rzucił się,
niewiele szczęścia miał, bo wpadł
prosto mi na kły i krew trysnęła z
rany,
```

Eng. *that (one) who threw oneself at me, was not very lucky, because he fell straight on my fangs and his blood gushed out of the wound.*

The new set of rules allowed us to find the best candidate in the word lattice, but only if the text which has to be recognized represents a sentence (and not just some loose syntactic groups). For example, the system produces several dozen potential word strings for the uttered sentence `Fundusz społeczny podjął działania w ramach obecnego prawa cywilnego`, Eng. *A social fund took actions according to contemporary civil law*. After shallow parsing with the new grammar, we could identify four of them as full sentences (the sentence phrase covers a whole candidate).

3.2. Disambiguation problems

At the beginning of the work on the new grammar, we tried to achieve the goal without a set of disambiguation rules. The reason was that we do not parse texts that have

been actually uttered by someone, but propositions created by our signal decoder. Most of the candidates in a word lattice are both grammatically and semantically unacceptable, so it is hard to state, which tag proposed by the tagger should be chosen. On the other hand, usually at least one of the candidates is grammatically correct (the one that had been said by the speaker), and within the correct string the process of disambiguation should produce a positive result. During the evaluation of lattices parsed with the first versions of the grammar, we noticed a few problems that arose due to the absence of a set of disambiguation rules.

The main problem is connected with nested nominal groups. There are several NG types in the grammar:

- NG (single noun);
- NGa (nominal-adjective);
- NGs (nominal-nominal in other case than Genitive, apposition);
- NGg (nominal-nominal in Genitive);
- NGe (elective construction);
- NGk (coordinated);
- NGn (nominal-numeral).

Some of them can be nested, so they can consist of some other NGs (e.g. NGk can be made up of NGa and NG in the same case). Since syncretic forms of nouns in Polish are common, in some cases it may occur that a NGa will be created on the basis of a different noun tag than a whole NGk. For example, the form `kobiety` (a form of the word *woman*) can be interpreted as Gen Sg, Nom Pl, Acc Pl, or Voc Pl. The word `mężczyzny` (a form of the word *man*) has only one possible tag: Gen Sg. The string
`kobiety i mężczyzny`
(of) woman – and – (of) man
should be interpreted as NGk in Gen Sg, but if the system propose a candidate
`miłe kobiety i mężczyzny`
nice – women – and – (of) man
the grammar should create a group NGa `miłe kobiety` (*nice women*, Nom, Acc or Voc Pl, because the form `miłe` must be interpreted as Pl) and should not build a group NGk (since there is no agreement between `miłe kobiety` and `mężczyzny`). In spite of that a NGk is created, because the parser checks the agreement of cases between the heads of two groups: NGa and NG, and a noun is always both a syntactic and semantic head of a nominal group. The problem has been resolved by introducing disambiguation rules that can establish the values of cases and numbers of a noun in NGa groups on the basis of the requirements of an agreement within an adjective. Since sequences of nouns and their subordinates occur very often in the lattices and they are in most cases meaningless, it is crucial to eliminate those which are grammatically incorrect. This task is much easier with a good set of disambiguation rules.

Another example of problems related to the lack of disambiguation is the parsing of often homonymous and word order in Polish is relative free (so the structure NP(Acc) - V - NP(Nom) is also possible). E.g. a sentence
`Koty lubią myszy`
Cats (Nom or Acc) – like – mice (Nom or Acc)
can be understood both as *Cats like mice* and *Mice like cats* (although an intonation will be different). Without a set of disambiguation rules, our grammar produces two Sent groups, each of them matching one of the two possible interpretations. One Sent group is redundant, since we only have to identify the best candidate and the actual syntactic structure of it is less important.

Because of the described problems, we decided to include disambiguation rules created on the basis of the Brill tagger PANTERA (see Section 2.2.) at the beginning of the grammar. In order to provide better results of disambiguation, we have created several additional rules that establish a correct grammatical value of forms that are dependent on each other, but are placed at a distance greater than two tokens. This in particular concerns dependencies between subject and predicate. For example:
`Wniosek rolniczego związku znajduje się w ministerstwie`
application – (of the) farming – union – is – in – (the) ministry
The PANTERA rules tag the form `wniosek` as Acc, instead of Nom (the distance between the subject `wniosek` and the predicate `znajduje się` is too far for the tagger). Additional PANTERA-liked rules resolve the problem.

3.3. Removing multi-constituent paths

In ideal circumstances, every grammatical path should be detected by the grammar and marked as one group (especially, if an utterance is a sentence, a single Sent group should be created). However, the expressive power of our grammar has significant limitations. Particularly, the Spejd parser is not well adapted to deal with non-continuous linguistic units (splitted syntactic words or groups). As the word order of Polish is relatively free, it is possible to build for example such sentences as (1) or (2). In the sentence (1), there is a splitted verbal syntactic word (`się bał`), in the sentence (2) occurs an non-continuous nominal group with an adjective (`ładną bluzkę`).

1. `On się bardzo myszy bał.`
   He – refl. pronoun – very – of mice – was afraid.
2. `Ładną masz bluzkę.`
   Pretty – you have – blouse.

It has to also be taken into consideration that the disambiguation tagger does not always choose the right interpretation of a token. Especially, if dependent constituents are not adjacent to each other, but at a distance of several tokens, misleading results of disambiguation may affect the outcome of syntactical parsing.

For that reason, we prepared two versions of the grammar: a more and a less restrictive form. In the first (the more restrictive version) there is (at the end of the grammar) a rule which removes all paths consisting of more than one syntactic group. Only paths recognized by the grammar as correct in their entirety are

represented in the output of the parsing. In the second (the less restrictive version), this final rule is disabled. The output is then greater, but the risk that the correct path has been removed due to the presence of non-continuous groups or disambiguation failures is eliminated. For example, our model generates two candidates of what has been said (the correct text reads as follows: `Fundusz społeczny podjął działania w ramach obecnego prawa cywilnego`, Eng. *A social fund took actions according to contemporary civil law*):

1. **`Fundusz`** `po meczu podjął działania w ramach obecnego prawa cywilnego.`

   A fund (Nom) – after the match – took – actions – according to – contemporary – law – civil

2. **`Funduszu`** `po meczu podjął działania w ramach obecnego prawa cywilnego.`

   A fund (Gen) – after the match – took – actions – according to – contemporary – law – civil

Only the first interpretation is a grammatical sentence. The parser creates such a structure of it:
Sent (NG + PrepNG (po + NG) + podjął + NG + PrepNG (w ramach + NGa))

The second string is unacceptable, since the case value (Gen) of the word `funduszu` does not agree with any other constituent of the path. The parser represents the structure of the candidate as follows:
NG + Sent (PrepNG (po + NG) + podjął + NG + PrepNG (w ramach + NGa))

Thus, the second path will be removed by the more restrictive version of the grammar (the path consists of NG+Sent, there is no group that consists of the whole), but will be left by the less restrictive one.

3.4. Final Grammar Structure

After the application of mentioned changes the structure of the grammar is as follows:

- Rules removing rare words.
- Disambiguation rules from the PANTERA tagger.
- Additional disambiguation rules.
- Rules creating syntactic words.
- Rules creating nominal groups.
- Rules ascribing syntactic requirement (of case or preposition) to verbs.
- Rules creating sentence phrases.
- The rule of removing multi-constituent paths (active only in the more restrictive version of the grammar).

4. Evaluation

4.1. Evaluation part I

The first part of the evaluation was done with the same kind of input data as the development of the grammar: on large lattices based on an output from the word decoder which is developed by the Signal Processing Group of AGH University of Science and Technology. In the process of evaluation, we used 40 relatively small texts (short sentences or strings of two nominal or prepositional groups; the utterances consisted of 3-26 words, with average length of approximately 10 words). In this part of the evaluation we compare performance of a basic bigram approach with a bigram method refined by parsing.

4.1.1. Baseline I: a bigram model

The baseline presented as a reference is a basic bigram approach. It uses a bigram model generated on a balanced, 300-million-words subcorpus of the National Corpus of Polish. The bigram model was refined using Kneser–Ney method with delta parameter equal to 0.5. As usual, it uses the Viterbi algorithm to compute the best path.

4.1.2. Baseline I with parsing

Our original input lattices are quite large consisting (10^6 up to over 10^{70} of candidates for an utterance, with geometric mean of 10^{27}). Because of the high expressive power of the Spejd parser, it is unsuitable for parsing such large input. To decrease the size of the lattices, we first obtained a $100-$best list of paths for each lattice by means of the bigram model. Then we removed all edges that do not belong to any of these paths. As a result we get a lattice containing a superset of the $100-$best list, precisely: the number of combinations of these paths. These intermediate lattices were then parsed by Spejd and our grammar. The last step for our method is choosing the best candidate out of those given by the parser thanks to a baseline bigrams.

Obviously the bigram method used with our parsing stage could be replaced with a more sophisticated algorithm capable of computing an $n-$best list from a lattice.

4.1.3. Results

In the evaluation we discuss the filtering capabilities of our approach in terms of a number of utterance candidates present in a lattice. Additionally, we present a comparison of a simple baseline approach and a baseline enhanced with our method, with the use of two versions of grammar: more restrictive (with the final rule that removes multi-constituent paths) and less restrictive (with the final rule disabled; see Section 3.3.).

Statistics regarding the size of the lattices before and after parsing are presented in table 1. It is clear that the grammar has good filtering capabilities.

Table 1: *Size of the lattices before and after parsing (in number of paths), correct path preserve ratio.*

lattices	**# of paths (geom. mean)**	**# of files with correct path**
intermediate	110307	40
after parsing (no removing rule)	6094	38
after parsing (with removing rule)	121	35

A comparison of the baseline with our method is presented in table 2. It uses the Word Error Rate as a score.

This black-box comparison shows a clear gain from the parsing stage, especially if the multi-constituent paths removing rule will be taken into consideration. On average it gives about a 33% reduction of number of errors.

Table 2: *Word Error Rate comparison (evaluation part I).*

	Word Error Rate
bigram (Baseline I)	0.777
bigram + parsing (no removing rule)	0.686
bigram + parsing (with removing rule)	**0.515**

In our further research we will focus on improving the set of disambiguation rules in order to provide more consistent tagging of not adjacent, but grammatically connected tokens.

4.2. Evaluation part II

The second part of the evaluation aims at verifying if the parsing method described in the article is able to refine results of a State-Of-The-Art ASR system for Polish.

4.2.1. Baseline II: Sarmata 2.0

For the baseline we have used an ASR system Sarmata 2.0, built at the AGH University of Science and Technology. It is based on a Kaldi toolkit, involving a triphone HMM Gaussian mixture acoustic model and word-level trigram language model. A detailed description of the system goes beyond the scope of this article and can be found in (Ziółko et al., 2015). The system was trained on recordings collected by AGH and a selection of recordings from the Global Phone acoustic database (Vu et al., 2010).

The evaluation was performed on a subset of the Global Phone corpus. The tuning set included randomly chosen 10% of all recording, totaling in 249 recordings, while the testing set included 2240 recordings.

4.2.2. Baseline II with parsing

In this scenario we put the parsing as a last stage of the processing chain. The parser was fed with $10-$best-list results of the Sarmata system. Since in this case the input data was better quality, we needed a more fine-grained scoring method than the binary preserve/remove approach from the first part of evaluation. We used the first version of the grammar (that without the rule removing multi-constituent paths). The final score (cost) given to a candidate was $P_s + k * n + l$, where P_s was a cost given by Sarmata (negative log-probability), n was a number of constituents for an utterance which remained after the parsing, l was a constant penalty for cases completely dropped by grammar (zero otherwise) and k was a constant weight for the parsing stage. After tuning the k was set to 0.3 and l was set to 1.5 (tuning on random 10% of data).

4.2.3. Results

The Table 3 shows a comparison of the Sarmata system results with the combined approach on the test set (2240 recordings). As we can see, there is a slight improvement as a result of the parsing stage. As mentioned before it is observed on a different kinds of data than the ones used for the development of grammar. Hence, the results prove that the parsing approach with handwritten grammar may help with refining statistical ASR system. We observe that the grammar has to be further developed to handle better different kinds of data – in this test only approx. 5% of test results were altered by the grammar. Closer inspection of the results show that the parsing approach helps mainly if a recording consists of at least a verbless sentence. The parsing increases error rate mainly in cases where recording seems to be a randomly cut sentence. This kind of behavior was to be expected.

Table 3: *Word Error Rate comparison (evaluation part II).*

	Word Error Rate
Sarmata 2.0 (Baseline II)	0.320
Sarmata 2.0 + parsing	**0.316**

Although the improvement is quite low, the Wilcoxon statistical significance test shows *p-value* = $0.00672 < 0.05$, so the changes are not random modifications.

Acknowledgments

The project was funded by the National Science Centre allocated on the basis of a decision DEC-2011/03/D/ST6/00914.

5. References

Acedański, Szymon, 2010. A morphosyntactic brill tagger for inflectional languages. In Hrafn Loftsson, Eiríkur Rögnvaldsson, and Sigrún Helgadóttir (eds.), *Advances in Natural Language Processing*, volume 6233 of *Lecture Notes in Computer Science*. Springer.

Buczyński, Aleksander and Adam Przepiórkowski, 2009. Spejd: A shallow processing and morphological disambiguation tool. In Zygmunt Vetulani and Hans Uszkoreit (eds.), *Human Language Technology: Challenges of the Information Society*, volume 5603 of *Lecture Notes in Artificial Intelligence*. Berlin: Springer-Verlag, pages 131–141.

Głowińska, Katarzyna, 2012. Anotacja składniowa NKJP. In (Przepiórkowski et al., 2012), pages 107–127.

Przepiórkowski, Adam, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk (eds.), 2012. *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Warsaw: Wydawnictwo Naukowe PWN.

Przepiórkowski, Adam, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski, and Marek Świdziński, 2014. Walenty: Towards a comprehensive valence dictionary of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*. Reykjavík, Iceland: ELRA.

Vu, Ngoc Thang, Franziska Kraus, and Tanja Schultz, 2010. Multilingual a-stabil: A new confidence score for multilingual unsupervised training. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE.

Woliński, Marcin, 2006. Morfeusz — a practical tool for the morphological analysis of Polish. In Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski (eds.), *Intelligent Information Processing and Web Mining*, Advances in Soft Computing. Berlin: Springer-Verlag, pages 503–512.

Zaborowski, Bartosz, 2014. Shallow parsing on word lattices. In *2014 XXII Annual Pacific Voice Conference (PVC)*.

Ziółko, Bartosz, Tomasz Jadczyk, Dawid Skurzok, Piotr Żelasko, Jakub Gałka, Tomasz Pędzimąż, Ireneusz Gawlik, and Szymon Pałka, 2015. Sarmata 2.0 automatic polish language speech recognition system. In *Sixteenth Annual Conference of the International Speech Communication Association*.