Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5308–5315
Marseille, 11–16 May 2020

Towards the Conversion of National Corpus of Polish to Universal Dependencies

Alina Wróblewska
Institute of Computer Science, Polish Academy of Sciences
Warsaw, Poland
alina@ipipan.waw.pl

Abstract

The research presented in this paper aims at enriching the manually morphosyntactically annotated part of National Corpus of Polish (NKJP1M) with a syntactic layer, i.e. dependency trees of sentences, and at converting both dependency trees and morphosyntactic annotations of particular tokens to Universal Dependencies. The dependency layer is built using a semi-automatic annotation procedure. The sentences from NKJP1M are first parsed with a dependency parser trained on Polish Dependency Bank, i.e. the largest bank of Polish dependency trees. The predicted dependency trees and the morphosyntactic annotations of tokens are then automatically converted into UD dependency graphs. NKJP1M sentences are an essential part of Polish Dependency Bank, we thus replace some automatically predicted dependency trees with their manually annotated equivalents. The final dependency treebank consists of 86K trees (including 15K gold-standard trees). A natural language pre-processing model trained on the enlarged set of (possibly noisy) dependency trees outperforms a model trained on a smaller set of the gold-standard trees in predicting part-of-speech tags, morphological features, lemmata, and labelled dependency trees.

Keywords: Polish Dependency Bank, National Corpus of Polish, Universal Dependencies, conversion

1. Introduction

Natural language pre-processing (i.e. tagging, morphological analysis, lemmatisation, syntactic parsing, etc.) is a crucial issue in natural language understanding (NLU). It is an open question whether it is possible to infer syntactic and semantic knowledge essential for NLU without any supervision. If effective language-independent unsupervised algorithms are developed, the quality of pre-processing will undoubtedly improve and it will also be possible to pre-process less resourced languages. However, the state-of-the-art natural language processing (NLP) tools are still data-driven and a lot of attention and financial support is devoted to building high-quality training and testing data.

Manual annotation of resources that can be used for training NLP models is expensive and time-consuming. Therefore, alternative ways of obtaining training data are being sought. One of them is *self-labelling* technique, e.g. Triguero et al. (2015), in which a model trained on a gold-standard data set (mostly a small set of training examples) is used to predict pseudo-training instances. A new model is then trained on both gold-standard data and automatically predicted data, assuming that the model will provide more accurate predictions, because it is trained on a larger set of (possibly noisy) training instances.

We use the idea of self-labelling to build a large collection of Polish dependency trees. A set of Polish sentences is automatically parsed with a dependency parser trained on the largest bank of Polish gold-standard trees. A new parsing model is then estimated on both the gold-standard trees and the automatically predicted ones. In order to reduce the number of errors, the dependency trees are predicted for sentences from the largest corpus of the Polish texts which are manually annotated on the morphosyntactic level. The gold-standard dependency treebank used for training an initial parser and the morphosyntactically annotated corpus are described in Section 2. The final dependency trees are annotated according to the Universal Dependencies (UD) tree annotation schema (Nivre et al., 2016), which is cross-linguistically consistent and underlies the largest multilingual collection of dependency treebanks. Section 3. presents the resulting dependency treebank and the semi-automatic procedure of building it, reducing automatic predictions to a minimum. In the experimental part (Section 4.), it is evaluated whether a high-quality pre-processing model can be trained on the induced data set.

The contributions of this paper are twofold:

- We build a large Polish dependency treebank using a semi-automatic procedure and store it in the CoNLL-U format. The new Polish dependency treebank is publicly available.[1]
- We evaluate the quality of a pre-processing model trained on this large but possibly noisy treebank.

2. Selected NLP Resources for Polish

There are many NLP resources for Polish, but presenting them all is beyond the scope of this paper. We will thus focus on two data sets: the largest corpus of Polish texts and the largest bank of Polish dependency trees, as our research goal is to combine information from these two resources and to build a new dependency treebank for Polish.

2.1. National Corpus of Polish

National Corpus of Polish (NKJP, Przepiórkowski et al. (2012)) is a large collection of texts which are diverse in terms of theme and genre (i.e. literature, daily newspapers, specialist periodicals, journals, transcripts of conversations,

[1] http://git.nlp.ipipan.waw.pl/alina/PDBUD/tree/master/NKJP1M-UD_current

and Internet texts). The entire corpus consists of 1800M tokens, its balanced subcorpus has 300M tokens, and its manually annotated part (henceforth NKJP1M) includes 1.2M tokens.

NKJP1M subcorpus consists of 86K sentences and is the largest manually annotated corpus of Polish. There are 14.2 tokens per sentence on average in NKJP1M. 44% of all sentences are short (1–10 tokens), 35% are medium length sentences (11–20 tokens), and 21% are long sentences (above 20 tokens). NKJP1M is manually annotated on the word level with part-of-speech tags, morphosyntactic features, and named entities. NKJP1M is also annotated on the sentence level with syntactic chunks (the entire NKJP1M is not annotated with dependency trees or another kind of syntactic structures).

NKJP1M subcorpus, which is publicly available[2] on GNU GPL v.3, is the main resource for training NLP tools (e.g. taggers, named entity recognisers) for Polish.

2.2. Polish Dependency Bank

Polish Dependency Bank (PDB, Wróblewska (2014)) is the largest collection of Polish dependency trees. The treebank sentences come from various sources: (1) NKJP1M (14K trees; 217K tokens), (2) parallel Polish-English corpora: *Europarl* (Koehn, 2005), *Pelcra Parallel Corpus* (Pęzik et al., 2011), *DGT-Translation Memory* (Steinberger et al., 2012), OPUS (Tiedemann, 2012), (3) *CDSCorpus* (Wróblewska and Krasnowska-Kieraś, 2017), and (4) modern literature and NKJP corpus excluding NKJP1M.

The entire PDB treebank consists of 22K trees (350K tokens). There are 15.8 tokens per sentence on average in PDB. 34% of all sentences are short (1–10 tokens), 42% are medium length sentences (11–20 tokens), and 24% are long sentences (above 20 tokens). All PDB trees are manually annotated.

Most of the newly developed parsing systems are adapted to the CoNLL-U format of Universal Dependencies. Therefore, PDB trees are automatically converted into UD trees (Wróblewska, 2018) and stored in the CoNLL-U format. The converted trees meet the requirements of the UD v2 guidelines[3] and the Polish PDB-UD treebank[4] (henceforth UD-Polish-PDB) has been included in the UD collection as of release 2.4.

3. NKJP1M-UD

The objective of this research is to enrich NKJP1M with a syntactic layer, i.e. dependency trees of all sentences, and to store both dependency trees and gold-standard morphosyntactic annotations of particular tokens in the CoNLL-U format. We attempt to achieve these goals using the following procedure. First, NKJP1M sentences are parsed with a dependency parser trained on PDB. The predicted trees and the morphosyntactic annotations of tokens are then automatically converted into UD dependency graphs. If a tree is poorly predicted and it is not covered by the conversion rules, no UD equivalent of this tree is generated. Since we strive to have the UD-style representations of all NKJP1M sentences, we decided to manually correct the poorly predicted trees and to include the corrected trees in the final dataset. Finally, as NKJP1M sentences are an essential part of PDB and thus also UD-Polish-PDB, in the final NKJP1M-UD dependency treebank, some automatically predicted dependency trees are replaced with their manually annotated equivalents from UD-Polish-PDB.

As indicated by a reviewer of the current paper, another procedure of annotating NKJP1M sentences with UD representations is also possible: the gold-standard morphosyntactic annotations can be converted to universal part-of-speech tags and universal morphological features, but UD dependency trees can be predicted by a dependency parser trained on UD-Polish-PDB trees (the conversion of dependency trees is unnecessary in this case). We considered this option, but eventually decided not to use it. Assuming that correctly predicted trees are convertible, and incorrect parse trees cannot be converted using the existing rule, the rule-based conversion is treated as a verification step. The unconverted dependency trees are considered unreliable and need to be verified (and possibly corrected). In case of direct UD parse trees, possibly incorrect trees are not indicated and we have to verify all parses (this process is prohibitively expensive).

3.1. Initial Parsing

NKJP1M sentences are parsed with Combo parser[5] (Rybak and Wróblewska, 2018) with the publicly available dependency parsing model[6] trained on PDB. Combo is a preprocessing system with a biLSTM feature encoder and a graph-based parsing module. The feature encoder takes various combinations of input features, e.g. word embeddings, part-of-speech tags, lemmata and morphological features, and produces contextual word embeddings. Combo can be trained to predict not only labelled dependency trees, but also part-of-speech tags, lemmata, morphological features and/or semantic labels.

NKJP1M sentences are not parsed from scratch, because Combo parser has an access to morphosyntactic annotations of their tokens, i.e. the gold-standard part-of-speech tags, morphological features and lemmata. The parser therefore should be less prone to errors, because lower level errors (e.g. tagging errors) are not propagated to the tree prediction level. According to Wróblewska and Rybak (2019), given sentences with the gold-standard part-of-speech tags, morphological features and lemmata, Combo predicts labelled dependency trees with Labelled Attachment Score (LAS) of 88.92.

[2] http://clip.ipipan.waw.pl/NationalCorpusOfPolish?action=AttachFile&do=get&target=NKJP-PodkorpusMilionowy-1.2.tar.gz

[3] http://universaldependencies.org/guidelines.html

[4] https://github.com/UniversalDependencies/UD_Polish-PDB

[5] https://github.com/360er0/COMBO

[6] http://mozart.ipipan.waw.pl/~alina/Polish_dependency_parsing_models/190423_COMBO_PDB_nosem_parseonly.pkl

3.2. Conversion to UD

The automatically predicted dependency trees and the gold-standard morphosyntactic annotations of tokens are converted at once and stored in the CoNLL-U format of Universal Dependencies. The conversion is based on a set of rules designed for converting PDB into the largest Polish treebank in the UD collection, i.e. UD-Polish-PDB (Wróblewska, 2018). The conversion of the morphosyntactic layer is largely an independent process. In the first place, it is based on the original part-of-speech tags and morphological features. However, some tags can only be converted, if we have access to the lemma or the syntactic context. The morphosyntactic conversion rules also use dependency labels in ambiguous conversion cases. For example, all punctuation marks are annotated with the interp tag in PDB, but they should be converted into either PUNCT or SYM UD tags. The rule converting punctuation marks looks at their dependency labels, and the punctuation marks labelled punct are converted as PUNCT UD tags and the punctuation marks labelled with other dependency labels are converted as SYM UD tags (see Figure 1).

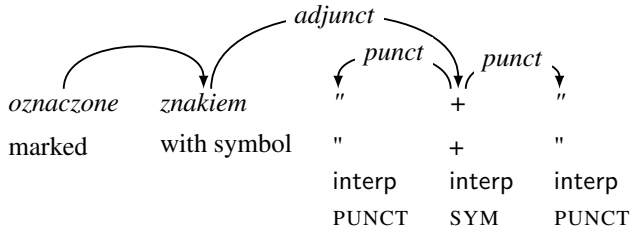

Figure 1: The snippet of a predicted tree of the sentence *Tendencje jang zostały* ***oznaczone znakiem "+"*** ('Yang tendencies have been **marked with the symbol "+"**').

The conversion rules are very strict and do not allow to convert some erroneous trees (e.g. trees without a ROOT edge or with multiple ROOT edges, trees with incorrect annotation of predicative expressions, subordinating conjunctions without a subordinating clause). Therefore, the conversion can be seen as a process of verifying the quality of the predicted trees. If a predicted parse tree contains any major error, it cannot be converted into a UD tree.

3.3. Manual Annotation

There are 755 parse trees that could not be converted due to errors. The underlying sentences contain 17.4 tokens on average (291 short sentences with 2–10 tokens, 235 medium length sentences with 11–20 tokens, and 249 long sentences containing above 20 tokens). Even if the sentences are relatively long, two-token sentences build the largest group (85 sentences) of the unconverted trees.

The trees which are not converted into the UD trees are only a small part of the entire set of all predicted PDB-like trees (i.e. 85,663) and could be excluded from the further experiment. On the other hand, they contain some problematic linguistic phenomena which prevent their conversion. Therefore, we decided to analyse the unconverted parse trees, to correct errors in these trees, and to enlarge the set of training instances to those that have been poorly predicted.

Our error analysis shows that many unconverted parse trees do not have a ROOT edge, while a particle or an interjection should be selected as the ROOT element (see Figure 2). There are no examples of such trees in PDB and Combo parser cannot learn this scenario.

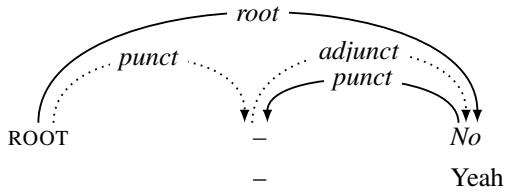

Figure 2: The predicted parse tree with the wrong dependencies (dotted lines) and the corrected dependencies (solid lines).

Even if NKJP1M is a manually annotated, it still contains some problematic cases and thus also questionable annotations, e.g. the token *to* can be either a pronoun ('it'), a pseudo predicative verb ('be'), a subordinating conjunction or a particle (incorrect annotations of *to* are problematic especially in the conversion of predicative expressions). Furthermore, some words belong to different part-of-speech classes in NKJP1M and in PDB, e.g. *więc* ('so') and *zatem* ('therefore') are annotated as coordinating conjunctions in NKJP1M and subordinating conjunctions in PDB. These conjunctions govern subordinating clauses in the predicted parse trees, but this dependency level is not compatible with the morphosyntactic level of NKJP1M and the conversion rules fail to convert such trees.

3.4. Final Dependency Treebank

The final NKJP1M-UD dependency treebank contains 86K trees. The automatically converted dependency trees are replaced with their manually annotated equivalents from UD-Polish-PDB or manually corrected 755 trees if available (see Table 1 for more details).

Some automatically converted dependency trees are replaced with their manually annotated equivalents from UD-Polish-PDB and manually corrected 755 trees (see Table 1 for more details). All gold-standard trees (UD-Polish-PDB trees and 755 corrected trees) are marked with PDB = True in metadata of NKJP1M-UD. More than 9% of trees are non-projective and more than 35% of all trees contain enhanced edges. The edges are labelled with 32 universal dependency types and 35 Polish-specific subtypes.[7]

4. Evaluation

For the purpose of evaluating the converted annotations, we train Combo model for predicting universal part-of-speech

[7]The Polish-specific UD subtypes are described in the UD guidelines https://universaldependencies.org/pl/dep/.

NKJP1M	Sentences	Tokens	Avg. toks/sents	Length of sentences (in tokens): 1–10	11–20	above 20
gold-standard	15,101	230,902	15.29	5902 (39%)	5655 (37%)	3544 (24%)
automatic	70,562	984,439	13.95	31,768 (45%)	24,747 (35%)	13,047 (20%)

Table 1: Statistics of NKJP1M-UD dependency treebank. UD-Polish-PDB trees and 755 corrected trees are grouped under *gold-standard* and automatically predicted UD trees are indicated as *automatic*.

tags, language specific tags, lemmata and labelled dependency trees on NKJP1M-UD. The quality of NKJP1M-based model is compared with the quality of two other models trained on UD-Polish-PDB and on the trees from both NKJP1M-UD and UD-Polish-PDB.

4.1. Data

Three Combo models are trained: (1) **nkjp** trained on NKJP1M-UD without the trees that are part of the test and development sets of UD-Polish-PDB, (2) **pdb-train** trained on the training subset of UD-Polish-PDB from Universal Dependencies release 2.5 (Zeman et al., 2019), and (3) **nkjp+pdb-train** trained on the data set resulting from the combination of NKJP1M-UD and UD-Polish-PDB-train. The development subset of UD-Polish-PDB is used for tuning the hyperparameters of all three models. The models are tested on the test subset of UD-Polish-PDB and on the Polish part (UD-Polish-PUD, Wróblewska (2018)) of Parallel Universal Dependencies treebank (PUD, Zeman et al. (2019)). All NKJP1M sentences which are in the test and development subsets of UD-Polish-PDB are excluded from **nkjp** and **nkjp+pdb-train** training sets. See Table 2 for some statistics on the training, testing and development data sets.

Dataset/Model	train	test	dev
pdb	18K (282K)	2K (34K)	2K (35K)
nkjp	83K (1.2M)		
nkjp+pdb-train	89K (1.3M)		
pud		1K (18K)	

Table 2: Statistics of the data sets used for estimation and evaluation of Combo models. The numbers before parentheses refer to sentences and the numbers in parentheses refer to tokens. Explanation: **pdb** – UD-Polish-PDB treebank from Universal Dependencies, **nkjp** – the set of NKJP1M trees in the UD format without the trees that are part of the test set (pdb-test) and development set (pdb-dev), **nkjp+pdb** – the union of nkjp set and pdb-train, **pud** – the Polish part (UD-Polish-PUD) of the parallel UD treebank.

4.2. Training Setup

The models are trained with Combo system using the same settings. Combo's feature encoder takes words represented with internally estimated character-based word embeddings as input, and outputs contextual word embeddings. Internal character-based word embeddings can be concatenated with external word embeddings. As shown in Wróblewska and Rybak (2019), external word embeddings have a positive impact on the morphosyntactic prediction of dependency trees. In the current experiment, we do not use external word embeddings, because we want to validate the impact of the treebank size and the treebank quality on the parsing performance. Combo models are jointly trained for tagging, lemmatisation and dependency parsing, and they predict universal part-of-speech tags, language-specific part-of-speech tags, lemmata, morphological features and labelled dependency trees.

4.3. Evaluation Setup

We apply the evaluation measures defined for the purpose of the CoNLL 2018 UD shared task (Zeman et al., 2018). The individual metrics evaluate different aspects of dependency predictions:

- UAS (unlabelled attachment score) measures how many words are assigned a correct head,
- LAS (labelled attachment score) measures how many words are assigned a correct head and a correct dependency label,
- MLAS (morphology-aware labelled attachment score) – labelled attachment score of content words extended with the evaluation of part-of-speech tags and morphological features,
- BLEX (bi-lexical dependency score) – labelled attachment score of content words extended with the evaluation of lemmata.

The measures are implemented in the official evaluation script `conll18_ud_eval.py`[8] of the CoNLL 2018 UD shared task. The script also evaluates predictions of universal part-of-speech tags, language-specific tags, morphological features, and lemmata.

4.4. Results

The quality of predicting universal part-of-speech tags (UPOS), language-specific tags (XPOS), morphological features (FEATS) and lemmata is presented in Table 3. Table 4 shows the dependency parsing quality measured with UAS, LAS, MLAS and BLEX.

The results show that enlarging the training data set with automatically predicted trees (i.e. possibly noisy trees) increases Combo's prediction quality. The best results are achieved by Combo's model trained on NKJP1M-UD enlarged with 6K gold-standard trees from UD-Polish-PDB

[8] `http://universaldependencies.org/conll18/conll18_ud_eval.py`

Training data	UD-Polish-PDB-test				UD-Polish-PUD			
	UPOS	XPOS	FEATS	LEMMA	UPOS	XPOS	FEATS	LEMMA
pdb-train	**98.19**	93.06	93.00	96.85	97.30	90.72	91.10	95.94
nkjp	98.06	94.10	94.74	97.67	97.34	92.58	93.00	96.67
nkjp+pdb-train	98.18	**94.62**	**95.19**	**97.89**	**97.42**	**92.75**	**93.23**	**96.89**

Table 3: The quality (F_1-scores) of predicting UPOS, XPOS, FEATS, and LEMMA by Combo models trained on the following data sets: **pdb-train** (the training set of UD-Polish-PDB), **nkjp** (the set of NKJP1M-UD trees without the trees that are part of UD-Polish-PDB-test and UD-Polish-PDB-dev), **nkjp+pdb-train** (the union of nkjp set and pdb-train), and tested on the test subset of UD-Polish-PDB and on UD-Polish-PUD.

Training data	UD-Polish-PDB-test				UD-Polish-PUD			
	UAS	LAS	MLAS	BLEX	UAS	LAS	MLAS	BLEX
pdb-train	92.74	89.95	78.85	83.99	93.00	89.79	75.86	82.48
nkjp	93.30	90.88	81.82	85.99	93.83	91.17	80.11	85.34
nkjp+pdb-train	**93.60**	**91.26**	**82.73**	**86.71**	**94.06**	**91.54**	**80.86**	**86.10**

Table 4: The quality of predicting unlabelled dependency trees (UAS F_1), labelled dependency trees (LAS F_1), correct heads, dependency labels, UPOS and FEATS of the content words (MLAS F_1), and correct heads, dependency labels and LEMMA of the content words (BLEX F_1) by Combo models trained on the following data sets: **pdb-train** (the training set of UD-Polish-PDB), **nkjp** (the set of NKJP1M-UD trees without the trees that are part of UD-Polish-PDB-test and UD-Polish-PDB-dev), **nkjp+pdb-train** (the union of nkjp set and pdb-train), and tested on the test subset of UD-Polish-PDB and on UD-Polish-PUD.

(nkjp+pdb-train). The second best model is trained on NKJP1M-UD (**nkjp**). The quality loss of the second best model compared to the best model is less than 1 pp. The both models trained on partially noisy data outperform the supervised model trained on UD-Polish-PDB (**pdb-train**).

The quality of morphosyntactic predictions is superior, especially the quality of predicting universal part-of-speech tags and lemmata is almost perfect, if tested against the test trees of UD-Polish-PDB. These results are not surprising, because Combo's modules predicting lemmata and tags are trained on a large set of gold-standard lemmata and UD tags converted from gold-standard tags. The quality of morphosyntactic predictions goes slightly down when tested against UD-Polish-PUD. The reason could be the discrepancy in textual content of both test sets. The UD-Polish-PDB test set contains 10% of sentences from four textual sources listed in Section 2.2. As NKJP1M is the largest source of texts, most sentences of UD-Polish-PDB-test come from this textual source. UD-Polish-PUD, in turn, contains other types of sentences, i.e. translations from different languages with numerous foreign inclusions. The foreign tokens, e.g. 'The', 'Rocket', 'Record', 'Company' are annotated with UPOS X (foreign token), but the model assigns them PROPN (personal name) UD tags. On the other hand, dependency trees are predicted more accurately in UD-Polish-PUD than in UD-Polish-PDB. The possible reason is that UD-Polish-PUD contains syntactically correct sentences and Combo's model learns to annotate these constructions correctly. UD-Polish-PDB test set contains not only properly built sentences, but also colloquial texts, e.g. from online forums, and the parser is not able to predict their gold-standard parses (manual annotation of colloquial language is also not a trivial task, and there are large discrepancies between gold-standard annotations).

4.5. Evaluation of Dependency Type Prediction

Apart from evaluation of automatically predicted trees, we propose to evaluate the quality of predicting particular dependency types. There are two dimensions in evaluating predicted dependency types:[9] the first one is the numer of training examples in the training set, and the second one is the test set used in the evaluation. We observe four different patterns among results for particular dependency types: (1) F-scores are similar across the training sets of various sizes and across tests sets: `acl`,[10] `amod`, `aux`, `case`, `cc`, `det`, `expl`, `iobj`, `mark`, `nummod`, `parataxis`, `punct` and `root`; (2) F-scores are similar across the training sets, but they vary across the test sets. Two dependency types – `cop` and `xcomp` – are more accurately predicted in UD-Polish-PDB-test, but their prediction quality is generally quite high (i.e. F-scores above 0.88). One dependency type – `advcl` – is more accurately predicted in UD-Polish-PUD; (3) F-scores increase as the number of training examples rises, and the upward trend is comparable in both test sets: `advmod`, `fixed`, `nmod`, `nsubj` and `obl`; (4) F-scores increase as the number of training examples rises, but they vary across the test sets: three dependency types

[9] Only universal dependency types are taken into account in the evaluation and not Polish-specific subtypes.

[10] The dependency types are explained in the Universal Dependency annotation guidelines: `https://universaldependencies.org/u/dep/index.html`.

Type	UD-Polish-PDB-test				UD-Polish-PUD			
	#	pdb-train	nkjp	nkjp+pdb	#	pdb-train	nkjp	nkjp+pdb
acl	682	0.83	0.83	0.83	427	0.83	0.84	0.84
advcl	376	0.78	0.77	0.76	174	0.82	0.82	0.83
advmod	1897	0.86	0.88	**0.89**	854	0.86	0.89	**0.90**
amod	2380	0.95	0.96	0.96	1704	0.95	0.96	0.96
appos	209	0.68	0.72	0.72	136	0.53	0.59	0.61
aux	522	0.97	0.98	0.98	240	0.96	0.97	0.97
case	3430	0.98	0.98	0.99	1993	0.98	0.98	0.98
cc	1063	0.94	0.94	0.94	572	0.93	0.93	0.93
ccomp	319	**0.83**	0.82	0.82	136	**0.85**	0.81	0.82
conj	1546	0.77	0.80	0.81	714	0.83	0.84	0.85
cop	318	0.93	0.94	0.94	215	0.88	0.89	0.89
csubj	17	0.61	0.82	0.89	6	0.77	0.73	0.80
det	604	0.96	0.97	0.97	336	0.98	0.99	0.98
discourse	9	0.67	0.43	**0.75**	0	–	–	–
expl	596	0.98	0.98	0.98	271	1.00	1.00	1.00
fixed	348	0.85	0.89	**0.90**	195	0.86	0.89	**0.90**
flat	269	0.87	0.90	0.90	338	0.84	0.84	0.85
iobj	671	0.81	0.83	0.83	291	0.79	0.81	0.81
list	25	**0.96**	0.87	0.83	0	–	–	–
mark	698	0.94	0.92	0.92	341	0.93	0.93	0.93
nmod	2480	0.81	0.83	0.84	1724	0.78	0.82	0.82
nsubj	2038	0.91	0.93	0.93	1189	0.90	0.93	0.94
nummod	232	0.94	0.93	0.94	166	0.94	0.95	0.94
obj	1478	0.89	0.90	0.91	817	0.89	0.93	0.94
obl	2631	0.85	0.87	0.88	1511	0.83	0.86	0.87
orphan	7	0.00	0.00	0.00	2	0.50	0.00	0.00
parataxis	383	0.68	0.73	0.72	135	0.71	0.70	0.72
punct	5633	0.93	0.93	0.94	2658	0.94	0.95	0.95
root	2215	0.97	0.97	0.96	1000	0.98	0.98	0.98
vocative	36	0.65	0.76	**0.82**	1	0.00	1.00	1.00
xcomp	505	0.93	0.94	0.94	243	0.88	0.91	0.90

Table 5: Evaluation of the individual dependency labels assigned by Combo parser trained on the following data sets: **pdb-train** (the training set of UD-Polish-PDB), **nkjp** (the set of NKJP1M-UD trees without the trees that are part of UD-Polish-PDB-test and UD-Polish-PDB-dev,), **nkjp+pdb-train** (the union of nkjp set and pdb-train), and tested on the test subset of UD-Polish-PDB and on UD-Polish-PUD. F_1-scores are provided. The second and six columns contain frequencies of the dependency relation types in UD-Polish-PDB-test and UD-Polish-PUD, respectively.

– appos, csubj, and flat – are more accurately predicted in UD-Polish-PDB-test, and two dependency types – conj and obj – are more accurately predicted in UD-Polish-PUD. The best prediction scores are high (i.e. F-scores above 0.85), even for the under-represented csubj type, except for appos, which can be predicted with F-score of 0.72 at most.

There is one dependency type – ccomp – with decreasing F-scores as the numer of training examples rises. The decrease is not very pronounced, but noticeable in both test sets.

Finally, it is difficult to interpret the quality of predicting the discourse, list, orphan and vocative types, as they are not only under-represented, but also not evenly distributed in the test sets. These dependency types are relatively sparse also in training data.[11]

5. Conclusion

The morphosyntactically annotated part of National Corpus of Polish (NKJP1M) was enriched with a syntactic layer, i.e. dependency trees, using a semi-automatic annotation procedure. Furthermore, both dependency trees and morphosyntactic annotations of particular tokens were converted to Universal Dependencies, i.e. a cross-linguistically consistent annotation schema that underlies the largest multilingual collection of dependency treebanks. The final dependency treebank – NKJP1M-UD – consists of 86K trees (including 15K gold-standard trees).

The evaluation results indicate that the size of the train-

[11] The frequency of discourse: 88 in pdb-train, 1073 in nkjp, and 1116 in nkjp+pdb-train. The frequency of list: 294 in pdb-train, 2066 in nkjp, and 2168 in nkjp+pdb-train. The frequency of orphan: 69 in pdb-train, 143 in nkjp, and 151 in nkjp+pdb-train. The frequency of vocative: 211 in pdb-train, 999 in nkjp, and 1081 in nkjp+pdb-train. The frequency of case for comparison: 29,008 in pdb-train, 106,557 in nkjp, and 119,349 in nkjp+pdb-train.

ing set has a positive impact on the quality of natural language pre-processing, even if the training examples are possibly noisy. The question is however whether the reason for the quality gain is the higher number of training instances or the increased vocabulary size. In the future research, we are going to conduct some experiments on enriching Combo's models with external word embeddings. We are going to test whether external word embeddings have a similar impact on natural language pre-processing models trained on data sets of different sizes.

6. Acknowledgements

The research presented in this paper was founded by the Polish Ministry of Science and Higher Education as part of the investment in the CLARIN-PL research infrastructure. The computing was performed at Poznań Supercomputing and Networking Center.

7. Bibliographical References

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit Conference*, pages 79–86.

Nivre, J., de Marneffe, M., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*, pages 1659–1666.

Pęzik, P., Ogrodniczuk, M., and Przepiórkowski, A. (2011). Parallel and spoken corpora in an open repository of Polish language resources. In *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 511–515.

Adam Przepiórkowski, et al., editors. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Rybak, P. and Wróblewska, A. (2018). Semi-supervised neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54, Brussels, Belgium, October. Association for Computational Linguistics.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., and Schlüter, P. (2012). DGT-TM: A freely Available Translation Memory in 22 Languages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 454–459.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2214–2218.

Triguero, I., García, S., and Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42:245–284.

Wróblewska, A. and Krasnowska-Kieraś, K. (2017). Polish Evaluation Dataset for Compositional Distributional Semantics Models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 784–792. Association for Computational Linguistics.

Wróblewska, A. and Rybak, P. (2019). Dependency parsing of Polish. *Poznań Studies in Contemporary Linguistics*, 55(2):305–337.

Wróblewska, A. (2014). *Polish Dependency Parser Trained on an Automatically Induced Dependency Bank*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Wróblewska, A. (2018). Extended and Enhanced Polish Dependency Bank in Universal Dependencies Format. In Marie-Catherine de Marneffe, et al., editors, *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 173–182. Association for Computational Linguistics.

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Zeman, D., Nivre, J., Abrams, M., Aepli, N., Agić, Ž., Ahrenberg, L., Aleksandravičiūtė, G., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Basmov, V., Batchelor, C., Bauer, J., Bellato, S., Bengoetxea, K., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Bielinskienė, A., Blokland, R., Bobicev, V., Boizou, L., Borges Völker, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Brokaitė, K., Burchardt, A., Candito, M., Caron, B., Caron, G., Cavalcanti, T., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cignarella, A. T., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., de Souza, E., Diaz de Ilarraza, A., Dickerson, C., Dione, B., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eckhoff, H., Eli, M., Elkahky, A., Ephrem, B., Erina, O., Erjavec, T., Etienne, A., Evelyn, W., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Fujita, K., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., González Saavedra, B., Griciūtė, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hämäläinen, M., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Heinecke, J., Hennig, F., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ikeda, T., Ion, R., Irimia, E., Ishola, Ọ., Jelínek, T., Johannsen, A., Jørgensen, F., Juutinen, M., Kaşıkara, H., Kaasen, A., Kabaeva, N., Kahane, S., Kanayama, H., Kanerva, J., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Klementieva, E., Köhn, A., Kopacewicz, K., Kotsyba, N., Kovalevskaitė,

J., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lam, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Liovina, M., Li, Y., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., McGuinness, S., Mendonça, G., Miekka, N., Misirpashayeva, M., Missilä, A., Mititelu, C., Mitrofan, M., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, K. S., Morioka, T., Mori, S., Moro, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Munro, R., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikaido, Y., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Ojha, A. K., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peljak-Łapińska, A., Peng, S., Perez, C.-A., Perrier, G., Petrova, D., Petrov, S., Phelan, J., Piitulainen, J., Pirinen, T. A., Pitler, E., Plank, B., Poibeau, T., Ponomareva, L., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Qi, P., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Riabov, I., Rießler, M., Rimkutė, E., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O., Rueter, J., Sadde, S., Sagot, B., Saleh, S., Salomoni, A., Samardžić, T., Samson, S., Sanguinetti, M., Särg, D., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shirasu, H., Shohibussirri, M., Sichinava, D., Silveira, A., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Spadine, C., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Suzuki, S., Szántó, Z., Taji, D., Takahashi, Y., Tamburini, F., Tanaka, T., Tellier, I., Thomas, G., Torga, L., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Utka, A., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Wallin, L., Walsh, A., Wang, J. X., Washington, J. N., Wendt, M., Williams, S., Wirén, M., Wittern, C., Woldemariam, T., Wong, T.-s., Wróblewska, A., Yako, M., Yamazaki, N., Yan, C., Yasuoka, K., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zhang, M., and Zhu, H. (2019). Universal Dependencies 2.5. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.