Projection-based Annotation of a Polish Dependency Treebank

Alina Wróblewska, Adam Przepiórkowski

Institute of Computer Science, Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland
{alina, adamp}@ipipan.waw.pl

Abstract

This paper presents an approach of automatic annotation of sentences with dependency structures. The approach builds on the idea of cross-lingual dependency projection. The presented method of acquiring dependency trees involves a weighting factor in the processes of projecting source dependency relations to target sentences and inducing well-formed target dependency trees from sets of projected dependency relations. Using a parallel corpus, source trees are transferred onto equivalent target sentences via an extended set of alignment links. Projected arcs are initially weighted according to the certainty of word alignment links. Then, arc weights are recalculated using a method based on the EM selection algorithm. Maximum spanning trees selected from EM-scored digraphs and labelled with appropriate grammatical functions constitute a target dependency treebank. Extrinsic evaluation shows that parsers trained on such a treebank may perform comparably to parsers trained on a manually developed treebank.

Keywords: dependency annotation, cross-lingual projection, weighted induction

1. Introduction

In recent years, dependency parsing has become quite important in complex NLP tasks, such as machine translation, question answering, information extraction. Most of contemporary dependency parsing systems are based on statistical methods. Using training data, parsers learn how to analyse sentences and predict appropriate syntactic structures for them. Different statistical methods have been applied to data-driven dependency parsing, but the best results are given by supervised methods so far.

Supervised methods of training dependency parsers require a sufficient amount of reliable data, i.e., manually annotated dependency trees. However, there are still many resource-poor languages without treebanks that could be used for training dependency parsers. Furthermore, even if a treebank exists for a language, it may contain very simple structures and a parser trained on them may not be able to cope efficiently with real textual data. For example, there is a Polish dependency treebank[1] (Wróblewska, 2012) that can be used for training purposes. If a dependency parser[2] trained on 7405 treebank trees is evaluated against 822 other trees from this treebank, results are remarkable – 92.9% UAS and 87.4% LAS. However, if the parser is tested against a small set of 50 manually annotated trees,[3] parsing performance drastically decreases – 74.7% UAS and 66.4% LAS.

The manual annotation of training data required by supervised frameworks is a very time-consuming and expensive process. For this reason, intensive research has been conducted on unsupervised grammar induction. However, the performance of unsupervised dependency parsers is still significantly below the performance of supervised systems. Moreover, the performance of unsupervised parsers is also substantially below the performance of systems based on cross-lingual projection methods (McDonald et al., 2011).

The cross-lingual projection builds on the assumption that a linguistic analysis of a sentence largely carries over to its translation in an aligned parallel corpus. Annotations projected to a target language can constitute data for training NLP tools for this language. The cross-lingual projection method has been successfully applied to various levels of linguistic analysis and corresponding NLP tasks, e.g., dependency projection pioneered by Hwa et al. (2005).

The cross-lingual projection of linguistic information (so-called *annotation projection*) is an alternative method of annotating sentences with dependency trees in less researched languages. The method builds on the assumption that the dependency analysis of a sentence largely carries over to its translation since valency relations encoded in dependency structures are relatively invariant across languages. Moreover, a sentence in one language and its trans-

[1] The Polish dependency treebank consists of 8227 dependency trees (10.2 tokens per sentence on average) automatically converted from relatively homogeneous constituent trees of the *Składnica* treebank (Woliński et al., 2011). Even if Polish is a language with relatively free word order and, hence, multiple crossing edges are possible, there is only 0.15% of non-projective arcs in the entire dependency treebank. A constituency grammar that accompanied the creation of the source constituent trees did not allow to annotate linguistic phenomena resulting in crossing edges. Single non-projective arcs are results of the manual correction of a part of the dependency treebank.

[2] We use the *Mate* dependency parser (v. 3.5) downloaded from `http://code.google.com/p/mate-tools/`.

[3] Additional test sentences were randomly selected from some Polish newspapers. The selected sentences are quite long and contain 15.3 tokens per sentence on average. They were first automatically tokenised, lemmatised and part of speech tagged, and then manually annotated with dependency trees by two experienced linguists. These linguists also corrected possible errors in lemmatisation and tagging, but not discrepancies in tokenisation. Possible tokenisation discrepancies relative to the tokenisation of treebank sentences and a relatively high complexity of additional test trees (i.e., they contain 2.2% of non-projective arcs) may cause problems for a dependency parser trained on trees with the majority of projective arcs.

lation in another language tend to have not only parallel semantic structures but also correlated syntactic structures. Furthermore, the cross-lingual projection of source language dependencies to a target language does not take into account the order of words. It is thus perfectly suited for projection between languages with different word orders.

Originally, the cross-lingual dependency projection resulted in target dependency trees assuming that some additional smoothing techniques and aggressive filtering methods were applied (Hwa et al., 2005; Jiang and Liu, 2009). Some experiments have been carried on projecting only reliable relations and training parsers on partial dependency structures (Spreyer, 2011; Täckström, 2013). There are also some constraint-driven learning approaches (Ganchev et al., 2009; Smith and Eisner, 2009) which apply projected information to constrain estimation of dependency parsing models. Other related approaches consist in transferring delexicalised parsers between languages (Zeman and Resnik, 2008; McDonald et al., 2011; Søgaard, 2011; Naseem et al., 2012; Täckström et al., 2013).

The study presented in this paper continues the trend of acquiring well-formed dependency structures that could build a dependency treebank. We present an induction-based approach aimed at gathering a large amount of unlabelled dependency trees, which may then be labelled and corrected with predefined rules. The entire process of acquiring labelled dependency structures is hence partially automatic (induction of unlabelled dependency trees) and partially manual (design of comprehensive labelling and correction rules). However, the manual construction of rules requires much less manual work than annotation of thousands of sentences of a treebank. The application of this annotation approach led to the creation of a Polish treebank with nearly 4 million dependency structures.

2. Automatic Induction of Unlabelled Dependency Structures

This paper presents an approach of inducing Polish dependency structures which is set within the mainstream of the study on dependency projection. The approach builds on the idea of weighted projection (Wróblewska and Przepiórkowski, 2012). However, we involve a weighting factor not only in the process of projecting dependency relations (weighted projection), but also in the process of inducing dependency trees (weighted induction).

2.1. Weighted Projection

Weighted projection is the first step in the entire process of acquiring valid Polish dependency structures. Using a parallel English–Polish corpus, its English side is automatically annotated with a syntactic parser. Then, dependency relations making up an English tree are projected via an extended set of word alignment links, i.e., *complete bipartite graph*.

Since we aim to project relations which are restricted to sentence boundaries, only alignment links within a pair of aligned parallel sentences are considered in the projection of dependency structures. In our projection scenario, we make use of two unidirectional word alignments (Polish-to-English and English-to-Polish) and a set of bidirectional word alignment links combined with a heuristic `grow-diag-final-and` described in Koehn (2010). These three word alignment sets are referred to while scoring edges of complete bipartite graphs built for each sentence pair. Bipartite edges are weighted with the number of their occurrences in three sets of automatic word alignment links. Intuitively, these scores indicate the certainty of bipartite edges.

English relations are then projected via bipartite edges and – this way – used to build Polish directed graphs (henceforth, digraphs). The schema of the projection procedure is presented in Figure 1. Since it is possible that many bipartite edges are assigned a score 0, i.e., they are not represented in any set of automatic word alignment links, there is a restriction that an English relation can be projected only if at least one of two bipartite edges to be used to project this relation is assigned a score greater then 0.

In our approach, we assume that there is no manually annotated data to train a model that scores arcs in projected digraphs. All arcs are projected with the same significance and all of them may equally likely be selected as arcs of final dependency trees. Since only some of projected arcs correspond to correct dependency relations, it is essential to identify the most probable arcs and assign them appropriate scores. To do this, we first assign initial weights to arcs in projected digraphs and then optimise these weights as described in Section 2.2.

Arcs of projected digraphs are scored with initial weights that are estimated based on scores of bipartite edges used in the projection of a particular arc and a projection frequency. We define the scoring function

$$s = w_d + w_g + 2w_d w_g f$$

where w_d is the score of a bipartite edge used to project the dependent of the relation, w_g is the score of a bipartite edge used to project the governor of the relation and f is the projection frequency, i.e., a number of projecting English relations of the same kind via bipartite edges with the same scores. Initially weighted digraphs provide a starting point to induce final dependency trees.

2.2. Weighted Induction

Weighted induction is the second step in the process of acquiring Polish dependency trees. The main idea behind weighted induction is to identify the most likely arcs in initially weighted projected digraphs and assign them optimised weights. Using methods of selecting maximum spanning trees from weighted directed graphs, final dependency structures are inferred from projected digraphs with recalculated weights on arcs.

A heuristic of recalculating arc weights is based on the probability distribution over relation types[4] in k-best maximum spanning trees (MSTs), which are selected from initially weighted projected digraphs using the algorithm by Camerini et al. (1980). In order to increase the chance of selecting all accurate arcs that constitute a dependency tree,

[4]The type of a relation is defined by its features, i.e., tokens, lemmata and/or parts of speech of related lexical nodes, and projected English grammatical function.

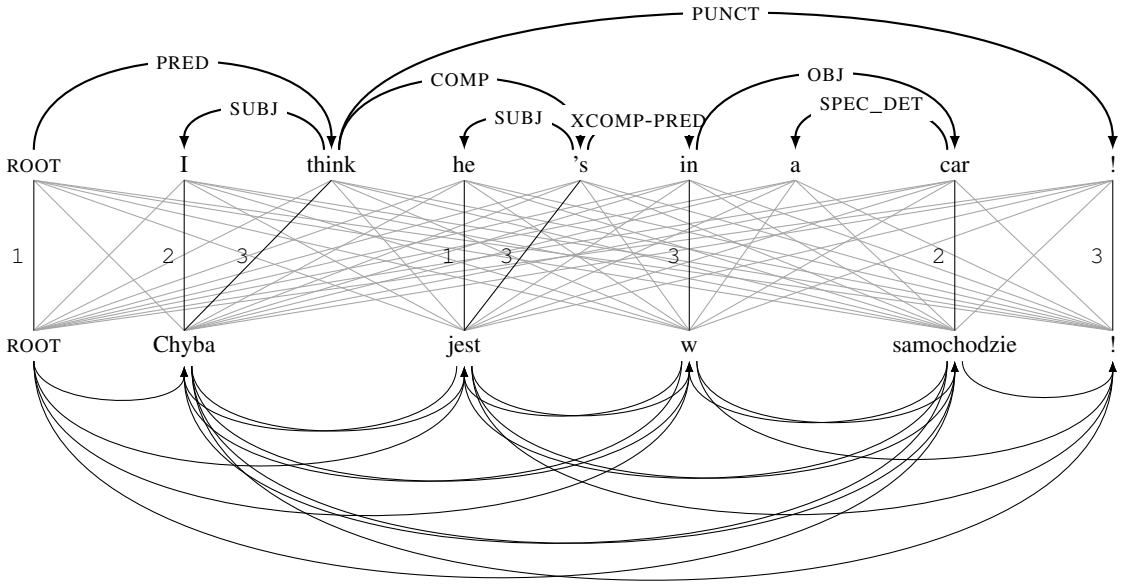

Figure 1: The Polish digraph (bottom arcs) projected from the English dependency structure (top arcs) via edges of the bipartite alignment graph (the middle edges between English and Polish nodes). Labels and weights of arcs in the Polish digraph are not displayed to preserve the clarity of presentation.

we do not confine the set of arcs to those building the maximum spanning tree. Instead, we enlarge this set with arcs which are in k-best maximum spanning trees assuming that each required arc is contained in at least one of these trees. The set of arcs in k-best maximum spanning trees is larger than the set of arcs in a maximum spanning tree. It may even contain all arcs required to build a final dependency tree. On the other hand, k-best MSTs contain less noisy arcs then projected digraphs. Therefore, arcs in k-best MSTs are more suitable for estimation of the probability distribution than entire projected digraphs.

The probability distribution over relation types is estimated using a version of the expectation maximisation algorithm defined by Dębowski (2009). This EM selection algorithm was originally designed to select the most probable valency frames from sets of valency frame candidates. Dębowski's algorithm is adapted for our purposes of identifying the most reliable arcs in sets of arcs in k-best MSTs found in initially weighted projected digraphs. According to the original procedure by Dębowski (2009), the most likely arc would be selected from the set of possible arcs coming into a node. However, the most probable incoming arcs for each lexical node do not have to necessarily constitute a well-formed dependency tree for a sentence (e.g., the resulting graph may contain a cycle). Therefore, our approach to recalculating weights does not directly build on the selected arcs but on the probability distribution over arc types estimated in the last iteration of the EM selection algorithm.

Based on the probability distribution over relation types, initial weights of projected arcs are optimised. The new weight s^* of an arc (v_k, v_i, l), for v_k being a governor, v_i being a dependent and l being a label of the arc, with the feature representation j, i.e., $fr(v_k, v_i, l) = j$, is calculated as the product of the square root of the initial arc weight s and the probability p_j of the relation type j:

$$s^* = \sqrt{s(v_k, v_i, l)} \times p_j$$

If an arc is not present in any of extracted k-best MSTs, its probability value is equal to 0. Because there is a risk that some digraph arcs would be assigned 0 and they would have the same priority in the extraction of final MSTs, we assign them the following value:

$$s^* = \sqrt{s(v_k, v_i, l)} \times \min_j p_j \times \alpha,$$

for some $0 < \alpha < 1$. The idea behind these modifications is to optimise initial weights of arcs in projected digraphs. Arcs of a particular type which have multiple instances in sets of k-best MSTs should get higher scores than other arcs represented or not in the probability distribution. Arcs with higher weights are more likely to be selected as part of final dependency trees.

Finally, MSTs corresponding to final dependency structures are selected from projected digraphs with recalculated arc weights. For the purpose of selecting one MST from a digraph with recalculated arc weights, we apply the k-best MST algorithm (Camerini et al., 1980) with $k = 1$. The entire induction procedure results in a collection of dependency structures labelled with English grammatical functions which need to be adapted to Polish dependency types.

3. Rule-based Adaptation of Dependency Structures

Automatically induced dependency trees are assumed to encode domination relations between Polish tokens. These relations should be accepted by the dependency annotation schema defined for Polish (Wróblewska, 2012). However, arcs in induced dependency trees are labelled with projected English grammatical functions which need to be adapted to Polish dependency types.

3.1. Labelling Rules

Adaptation of English grammatical functions to Polish dependency types is performed in a rule-based fashion. The definition of labelling rules is preceded by an analysis of the most common configurations of English grammatical functions and morphosyntactic properties of related Polish tokens in a developing part of induced trees (30,322 induced trees). Furthermore, the labelling procedure also applies information about the number and types of arguments subcategorised by some verbs or quasi-verbal predicates.[5] This information is extracted from the Polish valency dictionary[6] (Przepiórkowski et al., 2014).

The labelling process consists of three successive steps in which induced arcs are assigned labels. In the initial two steps, we consider only relations between those verbs that are represented in the valency dictionary and their dependents. In the first step, it is verified whether a dependent of a verb is its argument according to one of valency frames associated with this verb in the valency dictionary. A dependent is considered to be an argument fulfilling a particular grammatical function only if this dependent and its governing verb have some predefined morphosyntactic properties, and a candidate Polish dependency label directly maps to the projected English grammatical function, e.g., *subj*↦SUBJ, *comp_fin*↦COMP. Direct mapping between Polish dependency types and projected English grammatical functions is a distinguishing characteristic of the first step of the labelling process.

In the second step of the labelling process, we once again take into account only relations between verbs which are represented in the valency dictionary and their dependents. However, there are some additional restrictions on this labelling step. First, it is allowed to modify only these relation labels which correspond to English grammatical functions. Second, the set of frames which are taken into account is restricted to those which contain already assigned arguments. Finally, there is a restricted set of Polish arguments (i.e., *subj*, *obj*, *pd*, *comp_fin*, *comp_inf*) which are not allowed to be repeatedly governed by a verb. The idea is to label a dependent of a verb with a candidate Polish grammatical function which is in the set of possible and not multiplied arguments defined by a frame. However, the English grammatical function currently assigned to the dependent doesn't have to correspond to the candidate label as in the first labelling step.

Finally, all other unlabelled relations are assigned Polish dependency types in the third step. We distinguish relations between verb forms which are not covered by the valency dictionary and their dependents, and relations between governors which are not annotated as verb forms and their dependents. We iterate over unlabelled arcs and try to annotated them with the most appropriate labels based on a set of 45 predefined rules. These rules define what properties (e.g., morphosyntactic features, English grammatical functions) should be fulfilled by related tokens in order to assign a particular label to the relation between these tokens.

3.2. Correction Rules

Parallel sentences in Polish and English tend to have correlated dependency structures. This assumption seems to be largely true if we consider semantic predicate-argument structures of corresponding sentences. However, a syntactic realisation of the semantic predicate-argument structure may differ in both languages.

The already described labelling rules cover many discrepancies between Polish and English dependency types, e.g., the noun 'home' with the OBL function should correspond to the prepositional phrase 'do domu' labelled with the *comp* function. However, labelling rules assume that induced dependency structures are correct in terms of domination relations between tokens. Even if many Polish induced dependency structures are correct, there are still some induced structures that are noisy. For this reason, we apply the idea by Hwa et al. (2005) to use a predefined set of correction rules.

Even though the induction process seems to be straightforward, there are still some Polish-specific morphosyntactic phenomena (e.g., conditional clitic, mobile inflection,[7] reflexive marker) or linguistic structures diversely annotated in both languages (e.g., numeral complements, negation markers) whose correct annotations may not be induced based on English dependency structures. Moreover, noise in induced dependency structures may result from erroneous English dependency structures, incorrect word alignment or an inaccurate induction process. A linguistic analysis of trees from the developing set indicates types of errors or divergences that occur most frequently in induced dependency structures. These error types are covered with 31 correction rules, including the following, for correcting mobile inflections:

```
If a mobile inflection (lemma: 'być',
part of speech: aglt) is adjacent to a
verb form or a conditional clitic 'by'
on the left side, then the left token is
annotated as the governor of the mobile
inflection. Otherwise, if there is a verb
form immediately following the mobile
inflection, it is its governor.
```

After rule-based labelling and possibly correction, induced dependency trees build a projection-based Polish dependency treebank that can be used to train dependency parsers.

[5] Polish distinguishes between proper and quasi-verbs, e.g., *brak* (Eng. 'to miss, to fail, to lose'), *grzmi* (Eng. 'it's thundering'), *można* (Eng. 'it's allowed'), *należy* (Eng. 'it's necessary, should'), *szkoda* (Eng. 'it's pointless'), *trzeba* (Eng. 'it's necessary, should'), *warto* (Eng. 'it's worth'), *wiadomo* (Eng. 'it's known'), *wolno* (Eng. 'it's allowed'). Quasi-verbs do not inflect by number, person or gender, but they can be marked for mood and tense, e.g., *bedzie trzeba* (Eng. 'it will be necessary'), *byłoby warto* (Eng. 'it would be worth').

[6] The Polish valency dictionary is publicly available on `http://zil.ipipan.waw.pl/Walenty`.

[7] The mobile inflection is an agglutinate/clitic form marked for number and person. It is a characteristic feature of the mobile inflection that it may appear in different positions within a clause (e.g., *Wygraliśmy*., Eng. 'We won.').

4. Experiments and Evaluation

To test the annotation method outlined above, we conduct an experiment consisting in induction of a bank of Polish dependency structures which are labelled and corrected with predefined rules. Since there is no Polish-English parallel corpus annotated with gold-standard dependency trees, we may evaluate neither the induction procedure itself nor the quality of induced trees. Instead, we perform an extrinsic evaluation to see to what extent induced trees affect performance of a parser trained on them.

4.1. Data and Preprocessing

The experiment is conducted on a large collection of Polish–English bitexts gathered from publicly available sources: *Europarl* (Koehn, 2005), *DGT-Translation Memory* (Steinberger et al., 2012), *OPUS* (Tiedemann, 2012) and *Pelcra Parallel Corpus* (Pęzik et al., 2011).

After tokenisation, sentence segmentation and sentence alignment, bitexts are used to produce automatic word alignment links using the statistical machine translation system MOSES (Koehn et al., 2007). Three sets of word alignment links are generated: Polish-to-English, English-to-Polish and a set of links from both unidirectional word alignments gathered with the *grow-diag-final-and* method, the implementation of which is distributed as part of the MOSES system.

To parse the English side of the parallel corpus, we use the handcrafted wide-coverage English *Lexical Functional Grammar* (Dalrymple, 2001; Bresnan, 2001, LFG), using the *Xerox Linguistic Environment* (Crouch et al., 2011) as a processing platform. The most probable LFG analyses are converted into dependency structures using a conversion procedure similar as in Øvrelid et al. (2009). The conversion of permitted LFG analyses results in a collection of 4,946,809 English dependency structures, which constitute the subject matter of projection.

4.2. Automatic Induction of Dependency Trees

Given three sets of word alignment links, English dependency structures, and Polish sentences enriched with morphosyntactic information using the *Pantera* tagger (Acedański, 2010), the projection module (see Section 2.1.) outputs 4,946,809 initially weighted digraphs. Then, the induction procedure described in Section 2.2. acquires well-formed Polish dependency structures from these projected digraphs. After labelling and correcting induced trees (see Sections 3.1. and 3.2.), the final bank of Polish dependency structures may be applied to train a dependency parser.

4.3. Evaluation Experiment

We use the *Mate* system (Bohnet, 2010) in our evaluation experiment. The performance of the *Mate* parser trained on automatically induced trees is evaluated against a set of 822 dependency trees (*manual test*) taken from the Polish dependency treebank (Wróblewska, 2012). Furthermore, we provide a version of these test trees with automatically generated part of speech tags and morphological features (*automatic test*). In addition to test sets derived from the Polish dependency treebank, the parser is evaluated against a set of 50 relatively complex trees (*additional test*) mentioned in Introduction.

Table 1 reports parsing results of the *Mate* parser trained on induced dependency trees. Parsing performance is measured with two evaluation metrics: *unlabelled attachment score* (UAS) and *labelled attachment score* (LAS) as defined by Kübler et al. (2009). These results are compared with performance of a *supervised* parser trained on a part of the Polish treebank.

model	**training data**	**manual test**		**automatic test**		**additional test**	
		UAS	LAS	UAS	LAS	UAS	LAS
induced	3958556	73.7	–	72.8	–	65.4	–
labelled	3958556	74.6	69.4	74.0	68.1	65.1	61.2
modified	3958556	85.1	79.2	84.0	77.3	77.3	72.1
filtered	2352940	86.0	80.5	84.7	78.3	**78.5**	**73.6**
supervised	7405	**92.9**	**87.4**	**88.2**	**80.8**	74.7	66.4

Table 1: Performance of parsers trained with the *Mate* parsing system on Polish dependency trees acquired with the weighted induction method (*induced*), induced and labelled (*labelled*), labelled and modified with correction rules (*modified*), labelled, modified and filtered (*filtered*). Setting of model training: one iteration, the heap size of 100 million features, the threshold of non-projective approximation of 0.2. The *supervised* model is trained on 7405 trees form a Polish dependency bank. Setting of supervised model training: 10 iterations, the heap size of 100M, the threshold of 0.2. Validation data sets: manual test – the set of 822 gold-standard dependency trees; automatic test – the set of 822 test trees with automatically assigned morphosyntactic annotations; additional test – the set of 50 relatively complex sentences manually annotated with dependency trees.

A parser trained on automatically induced trees (*induced*) achieves 73.7% UAS if tested against the manual test set and 72.8% UAS if tested against the automatic test trees. The *Mate* parser trained on induced trees labelled with Polish dependency types (*labelled*) achieves 74.6% UAS and 69.4% LAS if tested against the manual test trees, and 74% UAS and 68.1% LAS if tested against the automatic test trees. The *Mate* parser trained on induced dependency trees modified with predefined rules performs significantly better – 85.1% UAS and 79.2% LAS if tested against the manual test trees and 84% UAS and 77.3% LAS if test against the automatic test set. These results are below parsing performance of the supervised parser trained on a part of the Polish dependency treebank – 92.9% UAS and 87.4% LAS if tested against the manual test trees and 88.2% UAS and 78.3% LAS if tested against the automatic test set. Note, however, that in the second – more realistic – scenario on evaluating the parser on automatically tagged data, the more useful measure LAS shows that the results of the semi-supervised procedure described here are directly comparably to the more costly supervised procedure.

The parsing models are also tested against a small set of 50 manually annotated sentences (*additional test*). Parsing results are generally worse than those reported above. However, the parser trained on corrected induced dependency trees (77.3% UAS and 72.1% LAS) significantly outperforms the supervised parser (74.7% UAS and 66.4%

LAS). The results show that dependency parsers developed in an automatic way as described here may rival fully supervised – and, hence, more costly – parsers.

Filtering is one of the most common optimisation techniques in projection-based approaches. However, our results show that filtering of possibly incorrect trees does not contribute significantly to improve parsing performance. Two filtering criteria are used: percentage of non-projective arcs and percentage of arcs labelled with a default function *dep* (dependent). The best parsing results (*filtered*) are achieved if we reject trees with more than 30% of non-projective arcs and with more than 10% of *dep*-labelled arcs – 86% UAS and 80.5% LAS if tested against the manual test set, 84.7% UAS and 78.3% LAS if tested against the automatic test trees, and 78.5% UAS and 73.6% LAS if tested against the additional test trees.

5. Conclusion

This article presented an approach of annotating Polish sentences with labelled dependency trees. The approach builds on an induction-based method of acquiring unlabelled dependency trees and on a rule-based adaptation of projected English grammatical functions labelling relations in induced trees to Polish dependency types. The process of inducing unlabelled dependency trees is fully automatic. The adaptation of induced dependency trees to the Polish dependency annotation schema requires a manual construction of a few dozen labelling rules and possibly correction rules. However, it is certainly much less manual work than in the annotation of thousands of treebank sentences.

Results of an extrinsic evaluation consisting in training a dependency parser on induced trees and evaluating results of this parser are very encouraging. When tested on a homogeneous set of rather short sentences from the Polish dependency treebank, performance of induced parsers is mostly a little below performance of the supervised parser which was trained on trees from the same source as the test trees. However, if tested against a small set of long and complex trees, a parser trained on induced trees may even exceed the supervised upper bound. A possible reason for the better performance of the induced parsers is that they were trained on automatically tokenised sentences annotated with induced dependency trees and they can cope with automatically tokenised test sentences better then the supervised parser. The supervised parser is helpless if it has to analyse divergently tokenised sentences. Hence, a parser trained on induced trees may be better suited for real NLP tasks. These results encourage us to continue our research on automatic induction of Polish dependency structures.

6. Acknowledgements

This research is supported by the POIG.01.01.02-14-013/09 project which is co-financed by the European Union under the European Regional Development Fund. It is part of the doctoral research of the first author, supervised by the second author, to be defended at the Institute of Computer Science, Polish Academy of Sciences.

7. References

Acedański, S. (2010). A Morphosyntactic Brill Tagger for Inflectional Languages. In *Advances in Natural Language Processing*, volume 6233 of *LNCS*, pages 3–14. Springer-Verlag.

Bohnet, B. (2010). Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97.

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell, Oxford.

Camerini, P. M., Fratta, L., and Maffioli, F. (1980). The K Best Spanning Arborescences of a Network. *Networks*, 10:91–110.

Crouch, D., Dalrymple, M., Kaplan, R., King, T., Maxwell, J., and Newman, P., (2011). *XLE Documentation*. Palo Alto Research Center.

Dalrymple, M. (2001). *Lexical-Functional Grammar. Syntax and Semantics*, volume 34. Academic Press.

Dębowski, Ł. (2009). Valence extraction using EM selection and co-occurrence matrices. *Language Resources and Evaluation*, 43(4):301–327.

Ganchev, K., Gillenwater, J., and Taskar, B. (2009). Dependency Grammar Induction via Bitext Projection Constraints. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1, pages 369–377.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping Parsers via Syntactic Projection across Parallel Texts. *Natural Language Engineering*, 11(3):311–325.

Jiang, W. and Liu, Q. (2009). Automatic Adaptation of Annotation Standards for Dependency Parsing – Using Projected Treebank as Source Corpus. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT '09)*, pages 25–28.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 177–180.

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

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Kübler, S., McDonald, R. T., and Nivre, J. (2009). *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

McDonald, R., Petrov, S., and Hall, K. B. (2011). Multi-Source Transfer of Delexicalized Dependency Parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, pages 63–72.

Naseem, T., Barzilay, R., and Globerson, A. (2012). Selective Sharing for Multilingual Dependency Parsing. In

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, pages 629–637.

Øvrelid, L., Kuhn, J., and Spreyer, K. (2009). Improving Data-Driven Dependency Parsing Using Large-Scale LFG Grammars. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 37–40.

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*, pages 511–515.

Przepiórkowski, Adam nd Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., and Świdziński, M. (2014). Walenty: Towards a comprehensive valence dictionary of Polish. In *Proceedings of LREC 2014*. Also available from `http://zil.ipipan.waw.pl/Walenty`.

Smith, D. A. and Eisner, J. (2009). Parser Adaptation and Projection with Quasi-Synchronous Grammar Features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 822–831.

Søgaard, A. (2011). Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11): Short Papers - Volume 2*, pages 682–686.

Spreyer, K. (2011). *Does It Have To Be Trees? Data-Driven Dependency parsing with Incomplete and Noisy Training Data*. Ph.D. thesis, Universität Potsdam.

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.

Täckström, O., McDonald, R., and Nivre, J. (2013). Target Language Adaptation of Discriminative Transfer Parsers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 1061–1071.

Täckström, O. (2013). *Predicting Linguistic Structure with Incomplete and Cross-Lingual Supervision*. Ph.D. thesis, Uppsala Universitet.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of LREC 2012*, pages 2214–2218.

Woliński, M., Głowińska, K., and Świdziński, M. (2011). A Preliminary Version of Składnica – a Treebank of Polish. In Vetulani, Z., editor, *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 299–303, Poznań, Poland.

Wróblewska, A. and Przepiórkowski, A. (2012). Induction of Dependency Structures Based on Weighted Projection. In *Proceedings of the 4th International Conference on Computational Collective Intelligence Technologies and Applications (Part I)*, volume 7653 of *LNAI*, pages 364–374, Berlin. Springer-Verlag.

Wróblewska, A. (2012). Polish Dependency Bank. *Linguistic Issues in Language Technology*, 7(1):1–15.

Zeman, D. and Resnik, P. (2008). Cross-Language Parser Adaptation between Related Languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.