Induction of Dependency Structures Based on Weighted Projection

Alina Wróblewska and Adam Przepiórkowski

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{alina,adamp}@ipipan.waw.pl

Abstract. This paper describes a novel weighted projection method of inducing grammatical dependency structures for Polish. Using a parallel English-Polish corpus, the English side is automatically annotated with a syntactic parser and the resulting annotations are projected to Polish via word alignment links. Projected arcs are weighted according to the certainty of word alignment links used in the projection, where arcs projected via intersection links are weighted with the lowest value (corresponding to the highest certainty). Minimum spanning trees induced from such graphs are used to train a parsing model with a publicly available parser-generation system.

Keywords: weighted projection, cross-lingual projection, dependency structure, dependency parsing

1 Introduction

Dependency parsing is becoming quite important in different language processing tasks. The predicate-argument structure transparently encoded in dependency-based syntactic representations may support machine translation, question answering, information extraction, etc. Supervised methods are very well-established in data-driven dependency parsing, and they give the best results so far. Supervised dependency parsers trained on correctly annotated data may have high parsing performance, even for languages with relatively free word order, such as Czech [15], Bulgarian [15] or Polish [26].

However, the manual annotation of training data is a very time-consuming and expensive process. That is why unsupervised techniques of training dependency models have been proposed, e.g., [9], [22], [7]. Unfortunately, unsupervised dependency parsers achieve the inferior accuracy of up to 50% so far.

As there are still many languages without any manually annotated data, and unsupervised training – with its low performance and high complexity – is not often a viable solution, we consider an alternative method of cross-lingual projection of linguistic information. This method is built on the assumption that the linguistic analysis of a sentence largely carries over to its translation in an aligned parallel corpus. Projected annotations can then be used to train NLP tools for the target language. The cross-lingual projection method has been successfully applied to various levels of linguistic analysis and corresponding NLP tasks, such as part-of-speech tagging [27], semantic role labelling [19], as well as syntactic dependency annotation and parser induction [8], [21], [6].

Experiments with dependency projection were initiated by [8]. Based on the *Direct Correspondence Assumption* (DCA), [8] assume that dependencies in one language directly map to dependencies in other language. It follows that dependencies may be directly projected across the word alignment linking two corresponding sentences. [8] project dependencies from English into Spanish and Chinese, induce some target dependency structures and finally train dependency parsers on them. However, the DCA underlying the annotation projection approach is an idealisation and projected dependency relations may be incorrect due to poor accuracy of the automatic word alignment, true mismatches of dependency structures between languages and translational divergences. As these error sources radically impair the quality of induced dependency structures, [8] apply some correction rules that locally transform induced structures.

In the current paper, a novel weighted projection method is presented. Using a parallel English-Polish corpus, the English side is annotated with a syntactic parser and the resulting annotations are projected to Polish equivalent sentences via word alignment links. Projected arcs are weighted according to the certainty of links used in projection, where arcs projected via intersection links are weighted with the lowest value (corresponding to the highest certainty). Minimum spanning trees (MSTs) induced from such graphs (i.e., trees spanning over all nodes in projected graphs, minimising the total weight of arcs) are used to train a parsing model.

This work is set within the mainstream of the study on cross-lingual information projection. Our aim is to project dependencies from English into Polish, in order to induce a bank of Polish dependency structures. While the work reported here bears resemblance to [8], we use a novel weighted projection method (see Section 2). The current work is also inspired by [14], whose projection scenario is based on diversely symmetrised bidirectional word alignments. In contrast to [14], we use differently symmetrised alignment links to weight projected relations, not to sort the target tokens corresponding to a source token. The induction process consists in the search for a minimum spanning tree in a graph containing all weighted projected arcs labelled with dependency types (see Section 3). The set of induced dependency MSTs is used to train a dependency parser for Polish with a publicly available parser-generation system (see Section 4).

2 Weighted Projection Method

A sentence in one language and its equivalent in the other language tend to have parallel semantic structures and correlated syntactic structures. For this reason it makes sense to apply cross-lingual projection of English dependencies onto Polish via word alignment. As the aim of this work is to induce Polish dependency structures spanning over all Polish tokens, Polish-to-English word alignment – in which each Polish token is aligned to one English token or a NULL-token – seems to be more functional than English-to-Polish word alignment, in which some Polish tokens may remain unaligned.

As dependencies are projected in one direction (English→Polish), unidirectional word alignment seems to be sufficient. However, statistical word alignment can only handle one-to-one and many-to-one links. There are well-known typological differences between considered languages,[1] and one(many)-to-one links are not sufficient to cover relevant linguistic phenomena. For example, a Polish noun phrase marked for the genitive case may be realised as an *of*-prepositional phrase in English (Pol. *Statua Wolności*, literary *Statue*.NOM *Liberty*.GEN, vs. Eng. *Statue of Liberty*), a Polish prepositional phrase may be realised as an adverb in English (Pol. *po prostu*, lit. *by just*.ADJ, vs. Eng. *simply*), the number of tokens in multiword expressions may differ (Pol. *zgodnie z*, lit. *appropriately*.ADV *with*, vs. Eng. *in accordance with*). In order to improve the quality of word alignment and overcome the limitation of alignment scenarios, an idea of merging bidirectional word alignments – symmetrisation – was proposed by [17]. Bidirectional word alignments derived from trained IBM models are combined according to a symmetrisation heuristic in order to achieve one-to-many and many-to-many alignment links. There are different symmetrisation heuristics, e.g., `union` [17], `intersection` [17], `grow-diag-final-and`[2] [11]. We make use of diversely symmetrised bidirectional word alignments in our projection scenario in order to widen the scope of alignment links. Four sets of alignment links composed of `intersection`, Polish-to-English links and `grow-diag-final-and` links are defined:

1. `intersection` of bidirectional alignments: $IS = AL_{PL\text{-}to\text{-}EN} \cap AL_{EN\text{-}to\text{-}PL}$,
2. union of `intersection` links (IS) and intersected Polish-to-English and `grow-diag-final-and` links: $UN_1 = IS \cup (AL_{PL\text{-}to\text{-}EN} \cap AL_{GDFA})$,
3. union of UN_1 links and Polish-to-English links: $UN_2 = UN_1 \cup AL_{PL\text{-}to\text{-}EN}$,
4. union of Polish-to-English links and `grow-diag-final-and` alignment links: $UN_3 = AL_{PL\text{-}to\text{-}EN} \cup AL_{GDFA}$.

Considering the certainty of alignment links, projected arcs are weighted with appropriate scores. `Intersection` links are supposed to be characterised by high precision scores, so projected relations which hold between two English tokens aligned with two Polish tokens by `intersection` links are weighted with the lowest value in the target graph: 1. Relations projected via UN_1, UN_2 and UN_3 links are weighted with 4, 5 and 8, respectively.[3] A schema of the weighted projection heuristic is given in Figure 1.

[1] Polish is an inflectional language with a relatively free word order; English is an isolating language with the topological argument marking.

[2] The `grow-diag-final-and` heuristics consists in the progressive selection of desirable alignment links. In the first step, all `intersection` alignment points are selected. In the `grow-diag` step, neighbouring and diagonally neighbouring alignment points which are in the `union` but not in the `intersection` are selected. In the `final-and` step, alignment points for words which remained unaligned are selected.

[3] These values were determined experimentally; further research should include investigation into more principled ways of determining weights, e.g., training them.

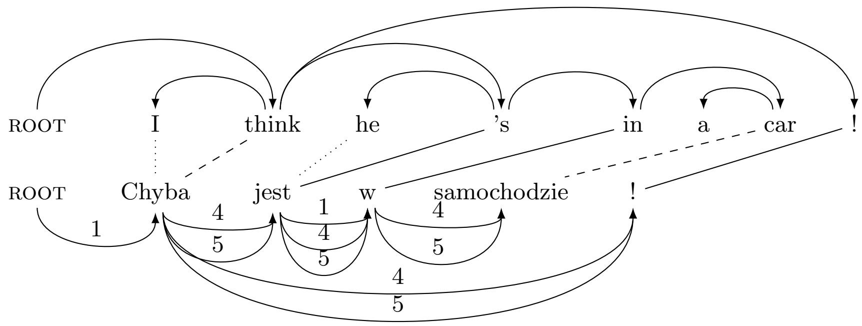

Fig. 1. Projection scenario. Alignment links are marked with different lines: *IS* (solid line), other UN_1 (dashed line), other UN_2 (dotted line). Arcs in the projected graph are weighted.

3 Induction of Dependency Structures

Given a graph built of all weighted arcs projected from the English dependency tree via word alignment links, we search for a minimum spanning tree (MST), i.e. the tree in which the sum of arc weights is minimised. The final dependency structure of a sentence is the lowest scored MST graph. We use the Chu-Liu-Edmonds' algorithm ([4], [5]) to find minimum spanning trees.

3.1 Mapping Rules

In order to adapt English dependency types to the Polish annotation schema described in [25], we define 36 mapping rules. While mapping, English dependency labels and morpho-syntactic features of Polish tokens (governor and its dependent) are taken into account. If no mapping rule applies because of different kinds of errors, e.g., in part-of-speech tagging, the default dependency type – *adjunct* – is assigned to the arc. An example mapping is given in Figure 2.

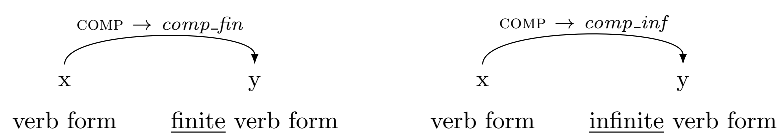

Fig. 2. Mapping of the English relation *COMP* (clausal complement) to appropriate Polish dependency types: *comp_fin* (clausal complement) or *comp_inf* (infinitival complement).

Mapping of English dependency types to Polish arcs results in a labelled dependency tree, as the one in Figure 3.

3.2 Modification Rules

According to our main assumption, parallel sentences tend to have correlated syntactic structures. This assumption seems to be true, if we consider semantic

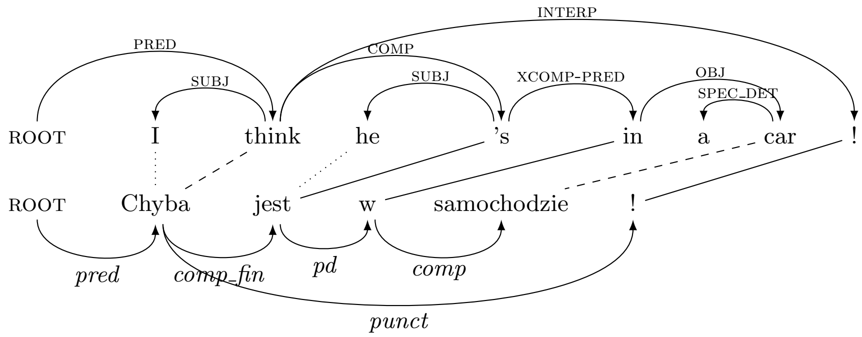

Fig. 3. Induction of a labelled Polish dependency structure.

predicate-argument structures of corresponding sentences. Even if the projection process seems to be straightforward, there are still some Polish morpho-syntactic phenomena whose annotation may not result from the English dependency structure, so some linguistic knowledge is necessary to annotate them correctly. Based on [25], the following phenomena are identified: mobile inflection,[4] conditional clitic,[5] reflexive marker,[6] abbreviation marker.[7] Four modification rules are defined to cover these phenomena. Due to lack of space only one rule is presented here:

Mobile inflection rule: *If the token adjacent to the left side of a mobile inflection is realised as a verb form or as a conditional clitic* by*, it constitutes the head of the mobile inflection. Else, the closest verb form on the right side of the mobile inflection is its head. Otherwise, if there is no such verb, the token adjacent to the left side of the mobile inflection is its head.*

Apart from linguistic phenomena which are not reflected in English dependency structures, there are also Polish linguistic constructions analysed differently in both languages, e.g., numeral phrases (a numeral combined with a noun phrase). In English, the noun phrase is the head of the numeral, treated as a number specifier. In Polish, in turn, the numeral depends on a verb form and governs the noun phrase, whose case either agrees with the case of the governing numeral or is determined as genitive. Induced dependency structures are modified appropriately, so that they meet the conventions of annotating Polish numeral phrases.

[4] The so-called *mobile inflection* is a verbal enclitic marked for number and person, which syntactically depends on the finite verb or a conditional clitic *by* appended to such a verb form, e.g., *walnąłby**ś*** (Eng. 'you would hit').

[5] The *conditional particle by* is used to underline the conditional modality of a sentence. It may be appended to the verb form or may appear anywhere in the sentence, e.g., *To* ***by*** *się nie udało.* (Eng. 'It would not have succeeded.').

[6] The *reflexive marker* realised as the particle *się* is used in Polish reflexive constructions, e.g., *Chronię* ***się***. (Eng. 'I protect myself.').

[7] The *abbreviation marker* is realised as a full stop that depends on the preceding abbreviation, e.g., *prof**.*** (Eng. 'Prof.').

3.3 Selection of Correct Dependency Trees

Any dependency structure is annotated as a graph with arcs representing directed binary relations between lexical nodes (tokens). One of related tokens is regarded as the head of the dependency relation, while the other one is its dependent. Arcs linking lexical nodes are named with dependency labels. A dependency tree has to meet some properties defined in [13]: *root property*, *spanning property*, *connectedness property*, *directness property*, *single-head property*, *acyclicity property*.[8] As the entire process of inducing dependency trees is automatic, it is possible that different errors occur at successive stages of processing, e.g., in tokenisation, morphological analysis, part-of-speech tagging, word alignment, English parsing. These errors may have the effect that the induced tree does not meet the properties of the correctly annotated dependency tree. In order to ensure that the dependency parser is trained on dependency structures which meet the properties of a correctly annotated dependency tree, we select only these MST graphs that do not violate any of the properties listed above.

4 Experiment and Results

4.1 Data and Preprocessing

In order to perform an experiment consisting in the induction of dependency structures using the weighted projection method, a large collection of parallel texts have been gathered from sources available on the Internet: *Europarl* [10], *DGT-TM* [23], *OPUS* [24] and *Pelcra Parallel Corpus* [20]. After filtering out some sentence pairs,[9] we have got a collection of nearly 5 million sentence pairs (Polish: 10.75 tokens/sentence; English: 12.51 tokens/sentence). These sentence pairs constitute our training corpus. For evaluation and comparison purposes, we use the Polish Dependency Bank (PDB) [25].

Word alignment and symmetrisations of bidirectional word alignments are performed with the statistical machine translation system *MOSES* [12], based on statistics captured from the entire corpus. English is a largely isolating language that makes use of function words. Polish, by contrast, is a highly inflecting language and often needs fewer words than English to express the same content,

[8] *Root property*: exactly one node ROOT in a tree is not governed by any other node. *Spanning property*: a dependency tree spans over all words of a sentence. *Connectedness property*: a dependency tree forms a connected syntactic structure with all word-forms (tokens) linked by syntactic dependencies. *Directness property*: a dependency structure contains only directed edges. *Single-head property*: any node in a tree depends on just one unique governor-node. *Acyclicity property*: a dependency tree may not contain cycles.

[9] We filter out: double sentence pairs, sentences in which at least 60% of Polish tokens have not been recognised by a part-of-speech tagger [1], sentences with the unsure word alignment and sentence pairs in which the English sentence has not been fully parsed.

but lexemes tend to have many more different forms, so there are more different token types on the Polish side. For this reason, for the purpose of training word alignment, the English side of the parallel corpus is only tokenised, while the Polish side is lemmatised.

The English side of the parallel corpus is parsed with the handcrafted wide-coverage English LFG grammar,[10] which is enhanced with a statistical disambiguation component selecting the most probable analysis. Resulting constituency and functional structures are converted to dependency structures [18] and stored in the column-based data format of the CoNLL shared task [3].

4.2 Induction of Weighted Dependency Trees

Given differently symmetrised word alignments, English dependency structures and Polish tokens enriched with morpho-syntactic information (lemma, part-of-speech tag, morphological features), the projection module outputs weighted directed graphs. Then, the projected graphs are given to the induction module. This module finds an MST in each graph, maps English functions to Polish dependency types and decides if the resulting tree is correctly formed. As only valid dependency trees are output (see Section 3.3), the final bank contains more than 3 million Polish dependency trees (average sentence length 7.97 tokens/sentence). This data may be used to train a dependency parser for Polish.

4.3 Training of Dependency Parsers

The best results for parsing Polish reported by [25] were obtained using *MaltParser* [16]. Therefore, this parser is used for the purposes of this experiment. The transition-based dependency parser uses a deterministic parsing algorithm that builds a dependency structure of an input sentence based on transitions (shift-reduce actions) predicted by a classifier. The classifier learns to predict the next transition given training data and the parse history. We use the LIBLINEAR classifier, the built-in non-projective transition system *stackeager* and the history-based feature mode combined of static features (word form, lemma, part-of-speech tag, morphological features) available in input data and a dynamic feature (dependency relation) extracted from a partially built dependency graph and updated during parsing.

It turns out that it is impossible to train MaltParser on the entire dependency bank using the available machines because of limited memory. For this reason MaltParser is finally trained on a set of 180,000 structures (11.01 tokens/sentence). However, it is possible to train a dependency parsing model on the entire bank with the *Mate* dependency parser [2]. The Mate parser is an implementation of the second-order MST dependency parsing algorithm and uses the passive-aggressive perceptron algorithm – MIRA – to learn feature weights.

Results produced by both parsers are reported below.

[10] The used XLE parser is based on Lexical Functional Grammar (LFG) and uses the Xerox Linguistic Environment (XLE) as a processing platform.

4.4 Evaluation and Error Analysis

The performance of the dependency parsers trained on MSTs induced using the weighted projection method is evaluated against the set of 1000 dependency trees taken from PDB. The rest of PDB is used to train a fully supervised version of MaltParser, whose performance constitutes our point of reference.

The accuracy of the parsers trained on MSTs is also compared against the baseline parsing model (MaltParser) trained on 180,000 directly projected dependency structures (9.97 tokens/sentence),[11] i.e., structures induced from relations projected via basic Polish-to-English word alignment links. Table 1 shows the results.

Table 1. Evaluation of the projection-based dependency parser for Polish. Evaluation metrics: labelled attachment score (LAS) and unlabelled attachment score (UAS).

Parsing model	**PDB test sets (1000 sentences)**		**Additional test set (50 sentences)**	
	LAS	**UAS**	**LAS**	**UAS**
baseline: direct projection	60.0	70.8	54.5	62.1
PDB: supervised	**84.7**	**90.5**	68.5	75.2
Malt: weighted projection	70.6	79.8	63.7	71.2
Mate: weighted projection	74.0	83.5	**69.1**	**78.1**

According to the results, the parsers trained on the weighted projection-based dependency structures perform much better than the parser trained on directly projected dependency structures (70.6/74.0% LAS vs. 60% LAS), but still significantly worse than the supervised parser trained on the part of the semi-manually annotated PDB (70.6/74.0% LAS vs. 84.7% LAS), when evaluated against the set of 1000 PDB-structures.

The parsers were also tested against a small set of 50 sentences (Additional test set) randomly selected from the projection-based corpus (containing much longer and more complex sentences than PDB) and unseen by the parsers at the training phase. These sentences were manually annotated by two experienced linguists. The results are generally worse than those obtained in the evaluation against the PDB test set (mostly due to the relative complexity differences between the two test sets). The Mate parser trained on the entire MST-based dependency bank slightly outperforms the supervised MaltParser, but the MaltParser trained on the part of the bank does not.

We also performed the evaluation of individual dependency relations against the gold standard sets in terms of precision, recall and f-measure. According to the results, the most problematic constructions are apposition, coordination, multiword expressions and predicative complements, which are all annotated with the f-score below 50%.

[11] Since directly projected graphs which are unconnected or contain cycles (about 70% of all projected graphs) are filtered out, the training sample of 180,000 dependency trees is not the same as the sample for training on projected MSTs (hence the different tokens/sentence ratio).

5 Conclusion and Future Work

We have presented a novel weighted projection method used to induce dependency structures. The results show that this method performs significantly better than baseline projection. What is more, the performance of the resulting parser may be slightly higher than the performance of the fully supervised dependency parsing model, if the parser is trained on the sufficient amount of data. While our experiment considers the Polish-English language pair, the weighted projection method may be applied to the induction of dependency structures for other resource-poor languages which do not have any annotated data, but have a reasonable number of sentences parallel with their resource-rich translations. The weighted projection method was tested on the task of the induction of dependency structures, but may also apply to other projection tasks, e.g., semantic role labelling and word sense disambiguation.

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.

References

1. Acedański, S.: A Morphosyntactic Brill Tagger for Inflectional Languages. In: Advances in Natural Language Processing. Lecture Notes in Computer Science, vol. 6233, pp. 3–14. Springer-Verlag (2010)
2. Bohnet, B.: Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010). pp. 89–97 (2010)
3. Buchholz, S., Marsi, E.: CoNLL-X shared task on Multilingual Dependency Parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning. pp. 149–164 (2006)
4. Chu, Yoeng-Jin and Liu, Tseng-Hong: On the Shortest Arborescence of a Directed graph. Science Sinica 14, 1396–1400 (1965)
5. Edmonds, J.: Optimum Branchings. Journal of Research of the National Bureau of Standards 71B(4), 233–240 (1967)
6. Ganchev, K., Gillenwater, J., Taskar, B.: Dependency Grammar Induction via Bitext Projection Constraints. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. vol. 1, pp. 369–377 (2009)
7. Gillenwater, J., Ganchev, K., Graça, J., Pereira, F., Taskar, B.: Sparsity in Dependency Grammar Induction. In: Proceedings of the ACL 2010 Conference Short Papers. pp. 194–199 (2010)
8. Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., Kolak, O.: Bootstrapping Parsers via Syntactic Projection across Parallel Texts. Natural Language Engineering 11(3), 311–325 (2005)
9. Klein, D., Manning, C.D.: Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. pp. 478–485 (2004)
10. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: Proceedings of the 10th Machine Translation Summit Conference. pp. 79–86 (2005)

11. Koehn, P.: Statistical Machine Translation. Cambridge University Press (2010)
12. Koehn, P., Federico, M., Shen, W., Bertoldi, N., Bojar, O., Callison-Burch, C., Cowan, B., Dyer, C., Hoang, H., Zens, R., Constantin, A., Moran, C., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of ACL 2007 Demo and Poster Sessions. pp. 177–180 (2007)
13. Kübler, S., McDonald, R.T., Nivre, J.: Dependency Parsing. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2009)
14. Mareček, D.: Combining Diverse Word-Alignment Symmetrizations Improves Dependency Tree Projection. In: Computational Linguistics and Intelligent Text Processing: 12th International Conference (CICLing 2011). Lecture Notes in Computer Science, vol. 6608, pp. 144–154. Springer-Verlag (2011)
15. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 Shared Task on Dependency Parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007. pp. 915–932 (2007)
16. Nivre, J., Hall, J., Nilsson, J.: MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation. pp. 2216–2219 (2006)
17. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics 29(1), 19–51 (1993)
18. Øvrelid, L., Kuhn, J., Spreyer, K.: Improving Data-Driven Dependency Parsing Using Large-Scale LFG Grammars. In: Proceedings of ACL-IJCNLP (Conference Short Papers). pp. 37–40 (2009)
19. Padó, S., Lapata, M.: Cross-linguistic Projection of Role-semantic Information. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. pp. 859–866 (2005)
20. Pęzik, P., Ogrodniczuk, M., Przepiórkowski, A.: 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. pp. 511–515 (2011)
21. Smith, D.A., Eisner, J.: Parser Adaptation and Projection with Quasi-Synchronous Grammar Features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. pp. 822–831 (2009)
22. Spitkovsky, V.I., Alshawi, H., Jurafsky, D., Manning, C.D.: Viterbi Training Improves Unsupervised Dependency Parsing. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning. pp. 9–17 (2010)
23. Steinberger, R., Eisele, A., Klocek, S., Pilos, S., Schlüter, P.: DGT-TM: A freely Available Translation Memory in 22 Languages. In: Proceedings of the 8th International Conference on Language Resources and Evaluation. pp. 454–459 (2012)
24. Tiedemann, Jörg: Parallel Data, Tools and Interfaces in OPUS. In: Proceedings of the 8th International Conference on Language Resources and Evaluation. pp. 2214–2218 (2012)
25. Wróblewska, A.: Polish Dependency Bank. Linguistic Issues in Language Technology 7(1), 1–15 (2012)
26. Wróblewska, A., Woliński, M.: Preliminary Experiments in Polish Dependency Parsing. In: Security and Intelligent Information Systems: International Joint Conference (SIIS 2011), Revised Selected Papers. Lecture Notes in Computer Science, vol. 7053, pp. 279–292. Springer-Verlag (2012)
27. Yarowsky, D., Ngai, G.: Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In: Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics. pp. 200–207 (2001)