Experiments on the Identification of Predicate-Argument Structure in Polish

Konrad Gołuchowski

Institute of Computer Science, Polish Academy of Science, Warsaw, Poland

Abstract. This paper focuses on automatic methods of extracting a predicate-argument structure in Polish. Two approaches to extract selected aspects of the predicate-argument structure are evaluated. In the first experiment the multi-output version of the Random Forest classifier is used to extract a valency frame for each predicate in a sentence. In the second experiment the Conditional Random Fields classifier is used to find syntactic heads of all arguments realised in a sentence. What is more, the importance of various sources of features is presented, including shallow syntactic parsing, dependency parsing and a verb valency information. Due to the lack of the high-quality syntactic parser, the presented approach does not rely on the deep syntactic information.

Keywords: Argument identification, verb valency, predicate-argument structure.

1 Introduction

Identifying a predicate-argument structure in a sentence is the initial step of many natural language processing tasks, such as Semantic Role Labelling (SRL), Information Extraction and verb clustering. Usual approaches to obtaining such a structure heavily depend on deep parsing. [1,10] show that such information is vital for the identification of argument boundaries. Although multiple efforts have been put into building a syntactic parser for Polish [15,7], none of existing tools gives satisfactory results yet. [2] shows that identification of exact argument boundaries without syntactic parsing for Polish is a difficult task. However, in many NLP tasks finding only selected aspects of a predicate-argument structure (e.g. argument heads) instead of a full structure is useful.

For reasons given above, this paper presents two experiments that try to identify two selected aspects of the predicate-argument structure without the deep parsing information: heads of arguments and for each predicate, types of all its arguments. The dependency parsing is used as an additional source of features for argument types classification.

2 Predicate-Argument Structure

Predicate dependents are usually divided into two groups: arguments – that are predicate-specific, and adjuncts – that can co-occur with almost any predicate.

A. Przepiórkowski and M. Ogrodniczuk (Eds.): PolTAL 2014, LNAI 8686, pp. 185–192, 2014.

This paper focuses only on the first group. Each argument may be labelled with a syntactic type, i.e. a syntactic information on how the argument is realised in a sentence. Argument types used in this paper were taken directly from the Składnica treebank [16,14] (see Sect. 4.1).

The sentence below presents a fairly simple example of the predicate-argument structure. Only one predicate *zjadł* (Eng. *ate*) is present. Its two core arguments, the subject (*subj*) and the nominal phrase in the accusative case (*np(bier)*), as well as an adjunct, have been annotated.

[$_{\text{SUBJ}}$Marek] **zjadł** [$_{\text{NP(BIER)}}$kanapkę] [$_{\text{ADJUNCT}}$wczoraj].
[$_{\text{SUBJ}}$Mark] **ate** [$_{\text{NP(ACC)}}$a sandwich] [$_{\text{ADJUNCT}}$yesterday].

3 Related Work

In the semantic role labelling task, most algorithms for the argument identification start with syntactic parsing and then do a binary classification of each node in a parse tree whether it is an argument or not (see [1]). There are only a few examples that use the shallow syntactic information for the argument identification [2,3,13]. [1,10] both show the necessity of syntactic parsing to obtain argument boundaries in the semantic role labelling task. [6] uses the Markov logic to perform various tasks related to the identification of the predicate-argument structure. The presented Markov-logic-based classifier identifies predicates, arguments and senses at the same time. This approach does not rely on parse trees but uses features based on the dependency parsing. [5] treats assigning valency frame to a predicate as a verb sense disambiguation task. The set of possible verb senses is taken from VALLEX, a Czech lexicon of valency frames. A similar approach is presented in [12], which evaluates an attempt to assign a correct valency frame.

4 Resources and Tools

In subsequent subsections, the most important resources and tools are described.

4.1 Składnica – The Polish Treebank

Składnica is a Polish treebank [16,14]. It consists of 19998 sentences of which 8227 have manually corrected parse trees. This corpus is used both for training classifiers and for their evaluation. To assure that final solutions do not rely on any deep syntactic features, apart from the argument structure, only morphosyntactic features (such as a part-of-speech) were extracted from this corpus.

There are several argument types available in Składnica: 1) subjects, 2) nominal arguments with the case information, 3) prepositional phrases, 4) adjectival arguments, 5) information about reflexive pronoun *się*, 6) various dependent clauses, 7) infinitive arguments with the aspect information and 8) adverbial arguments. Whenever an adverbial argument was realised by a prepositional phrase, it was considered as a prepositional argument.

4.2 Walenty – The Valency Lexicon

Walenty [9] is the Polish Valency Lexicon that describes possible valency frames for almost 7000 verbs. Valency frame defines for a predicate what arguments in a sentence are expected and how these arguments are realised syntactically.

Some arguments in Walenty frames require certain words to occur inside them (e.g. in case of idiomatic expressions). Presence of such arguments is correlated with predicates, so for the purpose of experiments all such arguments were replaced with appropriate unlexicalized versions.

4.3 IOBBER – The Syntactic Shallow Parser

As mentioned before, reliable deep syntactic parsers are not yet available for Polish. However, tools performing shallow syntactic parsing were developed. One of such tools is IOBBER [11], which annotates a text with three types of syntactic groups: nominal and prepositional groups, verbal groups and adjectival groups. Moreover, IOBBER can find a syntactic head of each group.

4.4 The Polish Dependency Parser

The Polish dependency parser [18] was used to obtain a set of dependency relations between words. Relations between predicates and its dependents are the most interesting as they may indicate both presence of arguments and their type. A detailed description of dependency relations obtained from the Polish dependency parser can be found in [17].

5 Extracting Predicate-Argument Structure

In this paper, two aspects of extracting the predicate-argument structure are considered and presented in subsequent sections.

5.1 Experiment I: Extracting Arguments for Each Predicate

The goal of the first experiment was to obtain types of realised arguments for each predicate in a sentence. To achieve this goal, for each predicate in each sentence a set of features and a set of types of its arguments were extracted. All features were binary and stated that a word in nearest surrounding of the argument had some property (see Table 1). The multi-output version of Random Forest classifier from Scikit toolkit [8] was used because this classifier was able predict multiple arguments for a single input.

One approach to obtaining features for each predicate was to consider a window of neighbouring *chunks* surrounding the predicate. A *chunk* meant either a nominal or an adjectival group returned by the IOBBER or a single word if it was not in any group. This approach makes it possible to find most arguments that are in the nearest neighbourhood. Another approach to obtaining features for each predicate was to take advantage of predicate dependents given by the Polish dependency parser and extract the features for each dependent. Both these approaches, as well as the combined approach, were evaluated.

Table 1. Features used in Experiment I

Type	**Feature**
Morphosyntactic	presence of nouns, pronouns, adjectives with their case presence of conjunctions, complementizer and adverbs with their base forms, presence of noun and verb negation, presence of past participle, presence of reflexive pronoun *się*, presence of questions words (e.g. *what*, *which*, etc.), presence of words describing time (e.g. *yesterday*, *today*), presence of adverbs in base form (e.g. presence of the word *quickly*),
Walenty	all frame elements that can be realised with predicates from the sentence
Dependency parser-based	relation labels of selected dependents
Predicate-based	predicate part-of-speech

5.2 Experiment II: Extracting Argument Heads

The goal of the second experiment was to find which words represent syntactic argument heads in a sentence. Also, types of argument heads were determined. To achieve this goal, each word had the label assigned, either an argument type for an argument head, or not-argument-head label for all other words. In case of arguments that are coordination of phrases, the conjunction is considered as the argument head. Figure 1 presents a sample sentence with argument types labels. The not-argument-head label was denoted as _, the subject as *subj*, the nominal argument in the accusative case as *np(bier)* and the prepositional argument in the genitive case with the preposition *do* as *prepnp(do,dop)*.

Oni	często	chodzili	do	pubu	,	pili	piwo	,	rozmawiali.
They	often	were going	to	a pub	,	drinking	a beer	,	talking.
subj	_	_	*prepnp(do:gen)*	_	_	_	*np(bier)*	_	_

Fig. 1. A sample sentence labelled with the argument heads

For each word in a sentence the following set of properties was extracted: 1) part-of-speech, 2) case (whenever applicable), 3) lemma of selected words (prepositions, complementizers, question words, *nie* and *się*), 4) information whether this word is the head of an IOBBER nominal or an adjectival group, 5) a matching Walenty argument if any, 6) a dependency relation if any. Then all properties in the window starting with two preceding words and one following word were considered as features for labelling a single word. Furthermore, a variant with the larger context (3 preceding and 3 following words) was evaluated.

To find the best possible labelling of words in a sentence the CRF++, the linear-chain Conditional Random Fields classifier, was used [4].

6 Evaluation

All evaluations were performed using 10-fold cross validation on the corpus extracted from the Składnica treebank. Standard measures of recall, precision and F-measure were used. In subsequent sections recall is understood as a correctly predicted fraction of arguments in the gold standard, precision – as a correctly predicted fraction of all predicted arguments and F-measure – as the harmonic mean of precision and recall. Correctness of the argument prediction is understood differently in each experiment. In the first experiment, the argument was predicted correctly if it was present in the valency frame of the considered predicate. In the second experiment, the predicted argument was considered correct when the argument head was recognised correctly.

6.1 Baseline Algorithms

In the first experiment, the baseline algorithm chooses the most frequent set of arguments for each predicate. For the second experiment – finding heads of arguments – the baseline algorithm chooses for each word the most common label for a concatenation of three features: its part-of-speech, its case (if applicable) and its base form.

6.2 Results of Experiment I: Predicate-Level Evaluation

The first experiment was performed in a few phases. Initially, only morphosyntactic features were extracted for each word in the sentence. In the later phases, features were extracted only for heads of syntactic groups. Next, the features extracted from the Walenty lexicon, as well as the features based on the Polish dependency parser, were added. In the best setting, the Random Forest classifier achieved the F-measure of 85.56% with the recall of 77.18% and the precision of 95.97%. Table 2 shows results achieved by the classifier in various setups. In this experiment the use of the IOBBER groups improves the recall but does not change the precision at all. The largest boost in the precision is obtained when the features based on the dependency parser relations are used.

This task is similar to assigning to a predicate a matching valency frame from the valency lexicon presented in [12]. However, [12] assigns not only arguments realised in a sentence as in experiment presented in Sect. 5.1 but also the full valency frame from VALLEX lexicon, To be able to approximately compare results from these two experiments, the accuracy of finding all arguments for each predicate is reported. The approach presented in this section achieved the accuracy on the level of 65.73% which is a slightly worse result than the accuracy presented in [12] (accuracy achieved by their best setup was 79.86%).

6.3 Results of Experiment II: Evaluation of Argument Head Extraction

In the task of finding argument heads using CRF-based classifier, most words are tagged with no-argument-head label. Including these tags in evaluation scores

Table 2. Results of finding arguments for each predicate in the sentence * – accuracy of finding the full frame for the predicate

Features	Precision (%)	Recall (%)	F (%)	Accuracy* (%)
Baseline	53.40	60.27	56.62	30.65
Window: 5 chunks				
Base features	88.87	74.03	80.77	59.45
+ Only IOBBER group heads	88.83	76.82	82.39	62.59
+ Walenty	89.74	76.57	82.63	63.32
+ Dependency-based relations	92.83	77.35	84.38	64.91
+ Dependency-based chunks	95.97	77.18	85.56	65.86
Window: 7 chunks				
Base features	91.17	73.71	81.51	60.30
+ Only IOBBER group heads	91.06	77.68	84.06	64.85
+ Walenty	92.28	77.13	84.02	65.06
+ Dependency-based labels	95.07	77.03	85.03	65.81
+ Dependency-based chunks	96.03	77.03	85.47	65.73
Dependency-based chunks only				
+ All features	94.00	74.65	83.21	62.22

Table 3. Results of finding heads of arguments * – accuracy of finding all types of arguments in the sentence

Features	Precision (%)	Recall (%)	F (%)	Accuracy* (%)
Baseline	71.31	70.21	70.75	37.68
Base features	85.21	80.90	83.00	57.54
+ IOBBER group head	85.99	84.11	85.05	61.69
+ Walenty	86.70	84.60	85.64	63.47
+ Dependency parser	91.17	88.91	90.02	73.06
+ Larger context	91.25	90.01	91.25	72.93

would lead to over-optimistic results and, in fact, would not reflect the real efficiency of finding argument heads. Therefore, only tags that reflect the actual argument types were user for scores calculations. Table 3 presents the results of this experiment.

Features based on the dependency relations gave the most noticeable improvement. Also, using information about syntactic heads improved the recall considerably. Detailed error analysis of the results showed that dependency relation-based features help to decrease three main sources of errors: misclassification of adverbial arguments, prepositional arguments and nominal arguments in the genitive case.

7 Conclusions and Future Work

This paper tackles the problem of finding two aspects of predicate-argument structure without the use of deep syntactic parsing. Two experiments were presented, as well as the impact of the various features types. All experiments gave reasonable results. Especially, in the experiment of finding argument heads, presented method achieved satisfying results, recognising correctly all arguments in 73% of sentences. Although the dependency parser outputs the predicate dependents, the first experiment showed that using a window around the predicate increases the recall of detected arguments.

In future work, it will be vital to merge both experiments into a single one, in order to obtain the predicate-argument structure, i.e. both argument heads and their governing predicates. Additional work may be necessary to improve the recall of finding valency frame for the predicate. Moreover, introducing semantic-based features (e.g. WordNet synset) should improve the distinction between adjuncts and arguments.

Acknowledgments. The paper is co-founded by the European Union from financial resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications".

References

1. Gildea, D., Palmer, M.: The necessity of parsing for predicate argument recognition. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 239–246. Association for Computational Linguistics (2002)
2. Gołuchowski, K., Przepiórkowski, A.: Semantic role labelling without deep syntactic parsing. In: Isahara, H., Kanzaki, K. (eds.) JapTAL 2012. LNCS (LNAI), vol. 7614, pp. 192–197. Springer, Heidelberg (2012)
3. Hacioglu, K., Pradhan, S., Ward, W., Martin, J.H., Jurafsky, D.: Semantic Role Labeling by Tagging Syntactic Chunks. In: Proceedings of CoNLL 2004, pp. 110–113 (2004)
4. Kudo, T.: CRF++: Yet another CRF toolkit (2005), Software available at `http://crfpp.sourceforge.net`
5. Lopatková, M., Bojar, O., Semecký, J., Benešová, V., Žabokrtský, Z.: Valency lexicon of czech verbs vallex: Recent experiments with frame disambiguation. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 99–106. Springer, Heidelberg (2005)
6. Meza-Ruiz, I., Riedel, S.: Jointly identifying predicates, arguments and senses using markov logic. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 155–163. Association for Computational Linguistics (2009)
7. Patejuk, A., Przepiórkowski, A.: Towards an LFG parser for Polish: An exercise in parasitic grammar development. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, pp. 3849–3852. ELRA, Istanbul (2012)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)

9. Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., Świdziński, M.: Walenty: Towards a comprehensive valence dictionary of Polish. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, pp. 2785–2792. ELRA, Reykjavík (2014), http://www.lrec-conf.org/proceedings/lrec2014/index.html
10. Punyakanok, V., Roth, D., Yih, W.T.: The importance of syntactic parsing and inference in semantic role labeling. Computational Linguistics 34(2), 257–287 (2008)
11. Radziszewski, A., Pawlaczek, A.: Large-scale experiments with NP chunking of Polish. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 143–149. Springer, Heidelberg (2012)
12. Semeckỳ, J.: Verb valency frames disambiguation. The Prague Bulletin of Mathematical Linguistics 88, 31–52 (2007)
13. Sun, W., Sui, Z., Wang, M., Wang, X.: Chinese semantic role labeling with shallow parsing. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 3, pp. 1475–1483. Association for Computational Linguistics, Stroudsburg (2009)
14. Świdziński, M., Woliński, M.: Towards a bank of constituent parse trees for Polish. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS (LNAI), vol. 6231, pp. 197–204. Springer, Heidelberg (2010)
15. Woliński, M.: An efficient implementation of a large grammar of polish. Archives of Control Sciences 15, 481–488 (2005)
16. Woliński, M.: Dendrarium – an open source tool for treebank building. In: Kłopotek, M.A., Marciniak, M., Mykowiecka, A., Penczek, W., Wierzchoń, S.T. (eds.) Proceedings of IIS 2010, pp. 193–204. Wydawnictwo Akademii Podlaskiej (2010)
17. Wróblewska, A.: Polish dependency bank. Linguistic Issues in Language Technology 7(1) (2012), http://elanguage.net/journals/index.php/lilt/article/view/2684
18. Wróblewska, A., Woliński, M.: Preliminary experiments in Polish dependency parsing. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) SIIS 2011. LNCS, vol. 7053, pp. 279–292. Springer, Heidelberg (2012)