

# Valence extraction using EM selection and co-occurrence matrices

Łukasz Dębowski<sup>†</sup>

*Instytut Podstaw Informatyki PAN*

*J.K. Ordona 21, 01-237 Warszawa, Poland*

**Abstract.** This paper discusses two new procedures for extracting verb valences from raw texts, with an application to the Polish language. The first novel technique, the EM selection algorithm, performs unsupervised disambiguation of valence frame forests, obtained by applying a non-probabilistic deep grammar parser and some post-processing to the text. The second new idea concerns filtering of incorrect frames detected in the parsed text and is motivated by an observation that verbs which take similar arguments tend to have similar frames. This phenomenon is described in terms of newly introduced co-occurrence matrices. Using co-occurrence matrices, we split filtering into two steps. The list of valid arguments is first determined for each verb, whereas the pattern according to which the arguments are combined into frames is computed in the following stage. Our best extracted dictionary reaches an *F*-score of 45%, compared to an *F*-score of 39% for the standard frame-based BHT filtering.

**Keywords:** verb valence extraction, EM algorithm, co-occurrence matrices, Polish language

## 1. Introduction

The aim of this paper is to explore two new techniques for verb valence extraction from raw texts, as applied to the Polish language. The methods are novel compared to the standard framework (Brent, 1993; Manning, 1993; Ersan and Charniak, 1995; Briscoe and Carroll, 1997) and motivated in part by resources available for this language and in part by certain linguistic observations.

The task of valence extraction for Polish invites novel approaches indeed. Although there is no treebank for this language on which a probabilistic parser can be trained, a few interesting resources are available. Firstly, the non-probabilistic parser Świgr (Woliński, 2004; Woliński, 2005) provides an efficient implementation of the large formal grammar of Polish by Świdziński (1992). Secondly, three detailed valence dictionaries have been compiled by formal linguists (Polański, 1992; Świdziński, 1994; Bańko, 2000). Those dictionaries are potentially useful as a gold standard in automatic valence extraction but two of them, Polański and Bańko, are printed on paper in several volumes, whereas Świdziński's dictionary, though rather small, is available electronically. The text file by Świdziński lists about 1000 verbal entries whereas 6000 entries can be found in COMLEX, a detailed syntactic dictionary of English (Macleod et al., 1994).

The information provided by Polish valence dictionaries is of comparable complexity to information available in COMLEX. Verbs in the dictionary entries select for nominal (NP) and prepositional (PP) phrases in specific morphological cases (7 distinct cases and many more prepositions). Valence frames may contain the reflexive marker *się* and certain adjuncts (e.g., adverbs) but not necessarily a subject, which also contributes to the combinatorial explosion. For instance, Świdziński (1994) provides 329 frame types for the 201 test verbs described later in Section 4. The most frequent frame among them, {np(nom), np(acc)}, is valid for 124 test verbs and there are 183 hapax frames.

<sup>†</sup> The author is presently on leave for Centrum Wiskunde & Informatica, Science Park 123, NL-1098 XG Amsterdam, the Netherlands. E: [debowski@cwi.nl](mailto:debowski@cwi.nl), T: +31 20 592 4193, F: +31 20 592 4312.



Such lack of computational data is a strong incentive to develop automatic valence extraction as efficiently as possible. Thus we have devised two procedures. The first one, called the EM selection algorithm, performs unsupervised selection of alternative valence frames. These frames were obtained for sentences in a corpus by applying the parser Świgr and some post-processing. In this way, we cope with the lack of a probabilistic parser and of a treebank.

The EM selection procedure, to our knowledge described here for the first time, assumes that the disambiguated alternatives are highly repeatable atomic entities. The procedure does not rely on what formal objects the alternatives are but it only takes their frequencies into account. Thus, the EM selection looks like an interesting baseline algorithm for many unsupervised disambiguation problems, e.g. part-of-speech tagging (Kupiec, 1992; Merialdo, 1994). Computationally, the algorithm is far simpler than the inside-outside algorithm for probabilistic grammars (Chi and Geman, 1998), which also instantiates the expectation-maximization scheme and is used for treebank and valence acquisition (Briscoe and Carroll, 1997; Carroll and Rooth, 1998).

The second novel technique concerns filtering of incorrect valence frames detected in the parsed text. Despite a large number of distinct frames occurring in the available Polish valence dictionaries, verbs which take similar arguments tend to have similar frames. This phenomenon was surveyed in particular by Dębowski and Woliński (2007) and their observations are reported here in more detail in Section 2. The cited authors proposed that sets of verbal frames be described in terms of argument lists, which strongly depend on a verb, and pairwise combination rules for arguments, called co-occurrence matrices, which are largely independent of a verb.

In this article, we recall this formalism and propose an analogous two-stage approach to filtering incorrect frames. The list of arguments is filtered for each verb initially and then the co-occurrence matrices are processed. In both steps we use filtering methods that resemble those used so far for whole frames. We will show that verbal frames are easier to extract when decomposed into simpler entities than when treated as atomic objects. The qualitative analysis of errors is also easier to perform.

Verb valence frames have been learned as atomic entities in all previous valence extraction experiments (see also: Sarkar and Zeman, 2000; Przepiórkowski and Fast, 2005; Fast and Przepiórkowski, 2005; Chesley and Salmon-Alt, 2006) although recent research exploits certain correlations among the verb meanings, diathesis, and sub-categorization (McCarthy, 2001; Korhonen, 2002, Chapter 4; Lapata and Brew, 2004; Schulte im Walde, 2006). This line of computational experiment is more and more inspired by formal research in semantic classes of arguments, verbs, and frame alternations, cf. Levin (1993) and Baker and Ruppenhofer (2002).

Our unorthodox less resource- and theory-intensive approach to decomposing valence frames stems from an independent insight into their distribution and structure, built on the preliminary valence extraction experiment for Polish by Przepiórkowski and Fast (2005). In that experiment, the  $F$ -score of the automatically extracted dictionary reached about 40%, whereas the  $F$ -score of two gold-standard dictionaries by Polański (1992) and Bańko (2000) compared with each other equalled 65%. This apparently low agreement between manually compiled dictionaries and the lack of explicit information about semantic classes inspired us to seek other patterns in valence frames and to develop an alternative extraction scheme.

The experiment described in this paper differs from both of the works by Przepiórkowski and Fast in several aspects. Firstly, we explore whether it is better

to filter frames in two steps or in one step as done previously. Secondly, we extract all kinds of arguments occurring in the gold-standard dictionaries, whereas only non-subject NPs and PPs were considered in the two previous works. Thirdly, we compare our extracted dictionaries with three gold-standard dictionaries simultaneously and investigate types of errors. Fourthly, we use the Świgr parser of Świdziński’s grammar and the EM algorithm to parse raw texts, whereas Przepiórkowski and Fast applied a very simple regular grammar of 18 rules. We analyze fewer texts but we analyze them more thoroughly, which means higher precision but not necessarily lower recall. The final difference is that our test set covers twice as many verbs (201 lemmas) as considered by Przepiórkowski and Fast.

The frame-based binomial hypothesis test (BHT, Brent, 1993) is assumed in this work as a baseline against which our new ideas of filtering are compared, since it gave the best results according to Fast and Przepiórkowski (2005). The authors reapplied several known frame filtering methods: the BHT, the log-likelihood ratio test (LLR) (Gorrell, 1999; Sarkar and Zeman, 2000), and the maximum likelihood threshold (MLE) (Korhonen, 2002). Applying the one-stage BHT to our data, we obtain 26% recall and 75% precision ( $F = 39\%$ ). To compare, the dictionary obtained by applying the novel two-stage filtering of frames to the same counts of parses exhibits 32% recall and 60% precision ( $F = 42\%$ ). The set-theoretic union of both dictionaries combines their strengths and features  $F = 45\%$ . These statistics relate to extracting whole frames, whereas Przepiórkowski and Fast obtained similar values for the simpler task of extracting only NPs and PPs. We find our results to be an encouraging signal that similarities of frame valence frame sets should be exploited across different verbs as much as possible, and also in an algorithmic way. The method introduced here allows various extensions and modifications.

The rest of this article describes our experiment in more detail. In Section 2, a brief introduction to co-occurrence matrices is provided; Section 3 presents the verb valence extraction procedure; the obtained dictionary is analyzed in Section 4; Section 5 contains the conclusion. Three appendices follow the article. Appendix A gives additional details for the co-occurrence matrix formalism; Appendix B describes the initial corpus parsing; Appendix C introduces the EM selection algorithm.

## 2. The formalism of co-occurrence matrices

Let us introduce the new description of valence frames which is applied to valence extraction in this paper. To begin with a more usual formal concept, consider a prototypical entry from our gold-standard valence dictionary. It consists of the set of valence frames

$$\mathbf{F}(\textit{przyłapać}) = \left\{ \begin{array}{l} \{\textit{np}(\textit{nom}), \textit{np}(\textit{acc})\}, \\ \{\textit{np}(\textit{nom}), \textit{np}(\textit{acc}), \textit{na}+\textit{np}(\textit{loc})\}, \\ \{\textit{np}(\textit{nom}), \textit{sie}, \textit{na}+\textit{np}(\textit{loc})\} \end{array} \right\} \quad (1)$$

for the verb *przyłapać* (= *to catch somebody red-handed*). The symbol *sie* denotes the reflexive marker *się* and *na+np(loc)* is a prepositional phrase with preposition *na* (= *on*), which requires a nominal phrase in the locative case. The notations for cases are as in the IPI PAN Corpus tagset: **nom**(inative), **gen**(itive), **dat**(ive), **acc**(usative), **inst**(rumental), **loc**(ative), and **voc**(ative), cf. Przepiórkowski and Woliński (2003) or

<http://korpus.pl/>. For simplicity, it is assumed that no argument type can be repeated in a single valence frame. This restriction can be overcome by assigning unique identifiers to repetitions.

There are two subtleties which concern our implementation of notation (1) and are worth exposing to avoid possible confusion later:

- (i) We treat the reflexive marker *się* as an ordinary verb argument rather than as a part of the verb lemma. The frames for a verb without *się* are merged with the frames of its possible counterpart with *się* into one entry, unlike the traditional linguistic analysis applied in Polish valence dictionaries. This affects all our counts of verb entries in the following work. However, we do not combine entries for corresponding perfective and imperfective verbs, which often take the same frames and occur in almost complementary pairs, cf. Młynarczyk (2004).
- (ii) A valence frame may lack the subject **np(nom)**. According to the analysis applied in Polish dictionaries, this lack is a counterpart of the English expletive *it* and it differs syntactically to the dropped subject (denoted always as **np(nom)** in the valence frame for a sentence). If a sentence lacks an overt subject, such a subject can or cannot be inserted depending on the verb. Certain verbs do not subcategorize for subject at all, e.g. *trzeba* (= *should*) or *brakować* (= *lack*). Several other verbs often occur without the subject but allow it in certain uses, such as *padać* (= *fall/rain*). The valences of the second class of verbs are particularly hard to extract automatically since Polish is a pro-drop language.

Summarising our remarks, there are many specific verbs such that **sie** or **np(nom)** (a) must appear in all their frames, (b) cannot appear in any frames, or (c) may be present or omitted, affecting the occurrence of other arguments. Similar interactions involving the reflexive marker and the subject have been studied in valence acquisition for other languages (Mayol et al., 2005; Surdeanu et al., 2008).

Dębowski and Woliński (2007) proposed an approximate description of complex interactions within the frame set  $\mathbf{F}(v)$  in terms of three simpler objects: the set of possible arguments  $\mathbf{L}(v)$ , the set of required arguments  $\mathbf{E}(v) \subset \mathbf{L}(v)$ , and the argument co-occurrence matrix  $\mathbf{M}(v) : \mathbf{L}(v) \times \mathbf{L}(v) \rightarrow \{\leftarrow, \rightarrow, \leftrightarrow, \times, \perp\}$ . The definitions of the first two objects correspond to the following naming convention. An argument is possible for  $v$  if it appears in at least one frame and it is called required for  $v$  if it occurs in all frames. Thus we have

$$\mathbf{L}(v) := \bigcup_{f \in \mathbf{F}(v)} f, \quad \mathbf{E}(v) := \bigcap_{f \in \mathbf{F}(v)} f. \quad (2)$$

For instance,

$$\begin{aligned} \mathbf{L}(\textit{przylapać}) &= \{\textit{np(nom)}, \textit{np(acc)}, \textit{sie}, \textit{na+np(loc)}\}, \\ \mathbf{E}(\textit{przylapać}) &= \{\textit{np(nom)}\}. \end{aligned}$$

To define the co-occurrence matrix, let us denote the set of verb frames which contain an argument type  $a$  as  $\langle a \rangle := \{f \in \mathbf{F}(v) \mid a \in f\}$ . Next, we will introduce five implicitly

verb-dependent relations:

$$\begin{aligned}
a \times b &\iff \langle a \rangle \cap \langle b \rangle = \emptyset && (a \text{ excludes } b), \\
a \leftrightarrow b &\iff \langle a \rangle = \langle b \rangle && (a \text{ and } b \text{ co-occur}), \\
a \rightarrow b &\iff \langle a \rangle = \langle a \rangle \cap \langle b \rangle \neq \langle b \rangle && (a \text{ implies } b), \\
a \leftarrow b &\iff \langle a \rangle \neq \langle a \rangle \cap \langle b \rangle = \langle b \rangle && (b \text{ implies } a), \\
a \perp b &\iff \langle a \rangle \cap \langle b \rangle \notin \{\langle a \rangle, \langle b \rangle, \emptyset\} && (a \text{ and } b \text{ are independent}).
\end{aligned}$$

Then the cells of matrix  $\mathbf{M}(v)$  are defined via the equivalence

$$\mathbf{M}(v)_{ab} := R \iff a R b \quad (3)$$

for the verb arguments  $a, b \in \mathbf{L}(v)$ . The symbol  $\perp$  that denotes “formal” independence was chosen intentionally to resemble the symbol  $\perp\!\!\!\perp$ , which is usually applied to denote probabilistic independence.

For the discussed example we obtain:

$\mathbf{M}(\textit{przylapać})$	np(nom)	np(acc)	sie	na+np(loc)
np(nom)	$\leftrightarrow$	$\leftarrow$	$\leftarrow$	$\leftarrow$
np(acc)	$\rightarrow$	$\leftrightarrow$	$\times$	$\perp$
sie	$\rightarrow$	$\times$	$\leftrightarrow$	$\rightarrow$
na+np(loc)	$\rightarrow$	$\perp$	$\leftarrow$	$\leftrightarrow$

This unconventional approach to describing verb valences appears quite robust. For example, consider an observed agreement score (cf. Artstein and Poesio, 2008) of the co-occurrence matrix cells taken for the triples  $(a, b, v)$  appearing simultaneously in two compared dictionaries. Formally this agreement score equals

$$A_o := \frac{|\{(a, b, v) \in T \mid \mathbf{M}_1(v)_{ab} = \mathbf{M}_2(v)_{ab}\}|}{|T|}, \quad (4)$$

where  $\{\mathbf{M}_i(v) \mid v \in V_i\}$ ,  $i = 1, 2$ , are the two compared collections of co-occurrence matrices and

$$T = \{(a, b, v) \mid v \in V_1 \cap V_2, a, b \in \mathbf{L}_1(v) \cap \mathbf{L}_2(v)\}$$

is the appropriate subset of triples  $(a, b, v)$ . The agreement scores (4) for the dictionaries of Polański, Świdziński, and Bańko range from 86% to 89%, cf. Dębowski and Woliński (2007).

Dębowski and Woliński noticed also that the values of the matrix cells  $\mathbf{M}(v)_{ab}$  for fixed arguments  $a$  and  $b$  tend not to depend on the verb  $v$ . The latter fact appears favourable for automatic valence extraction. We may learn objects  $\mathbf{L}(v)$ ,  $\mathbf{E}(v)$ , and  $\mathbf{M}(v)$  separately with much higher accuracy and restore the set of frames  $\mathbf{F}(v)$  from these by approximation. For example, consider the maximal set  $\bar{\mathbf{F}}(v) \subset 2^{\mathbf{L}(v)}$  of frames that contain all required arguments in  $\mathbf{E}(v)$  and induce the co-occurrence matrix  $\mathbf{M}(v)$ . Precisely,

$$\bar{\mathbf{F}}(v) := \left\{ f \in 2^{\mathbf{L}(v)} \left| \begin{array}{l} \forall a \in \mathbf{E}(v) \ a \in f, \\ \forall a, b \in \mathbf{L}(v) \ \phi(f, \mathbf{M}(v), a, b) \end{array} \right. \right\}, \quad (5)$$

where

$$\phi(f, \mu, a, b) := \begin{cases} \neg(a \in f \wedge b \in f), & \mu_{ab} = \times, \\ a \in f \iff b \in f, & \mu_{ab} = \leftrightarrow, \\ a \in f \implies b \in f, & \mu_{ab} = \rightarrow, \\ a \in f \impliedby b \in f, & \mu_{ab} = \leftarrow, \\ \text{true}, & \mu_{ab} = \perp. \end{cases}$$

It is easy to see that  $\bar{\mathbf{F}}(v) \supset \mathbf{F}(v)$ . We have  $\bar{\mathbf{F}}(v) \neq \mathbf{F}(v)$  for some verbs, as shown in Subsection 4.3. In our application, however, the number of frames introduced by using  $\bar{\mathbf{F}}(v)$  rather than  $\mathbf{F}(v)$  is small, see the last paragraph of Subsection 4.3.  $\bar{\mathbf{F}}(v)$  may be used conveniently also for syntactic parsing of sentences. Typically, a grammar parser checks whether a hypothetical frame  $f$  of the parsed sentence belongs to the set  $\mathbf{F}(v)$ , defined by a valence dictionary linked to the parser. If  $\bar{\mathbf{F}}(v)$  rather than  $\mathbf{F}(v)$  is used for parsing, which enlarges the set of accepted sentences, then there is no need to compute  $\bar{\mathbf{F}}(v)$  in order to check whether  $f \in \bar{\mathbf{F}}(v)$ . The parser can use a valence dictionary which is stored just as the triple  $(\mathbf{L}(v), \mathbf{E}(v), \mathbf{M}(v))$ . In our application, however, the reconstructed set  $\bar{\mathbf{F}}(v)$  is needed explicitly for dictionary evaluation. Thus we provide an efficient procedure to compute  $\bar{\mathbf{F}}(v)$  in Appendix A.

### 3. The adjusted extraction procedure

#### 3.1. OVERVIEW

Our valence extraction procedure consists of four distinct subtasks.

**Deep non-probabilistic parsing of corpus data:** The first task was parsing a part of the IPI PAN Corpus of Polish to obtain a bank of reduced parse forests, which represent alternative valence frames for elementary clauses suggested by Świdziński’s grammar. The details of this procedure are described in Appendix B.

The obtained bank included 510 743 clauses which were decorated with reduced parse forests like the following two examples (correct reduced parses marked with a ‘+’):

```
'Kto zastąpi piekarza?'
(= 'Who will replace the baker?')
+zastąpić :np:acc: :np:nom:
zastąpić :np:gen: :np:nom:
'Nie płakał na podium.'
(= 'He did not cry on the podium.')
płakać :np:nom: :prepn:na:acc:
+płakać :np:nom: :prepn:na:loc:
```

Reduced parses are intended to be the alternative valence frames for a clause plus the lemma of the verb. In contrast to full parses of sentences, reduced parses are highly repeatable in the corpus data. Thus, unsupervised learning can be used to find approximate counts of correct parses in the reduced parse forests and to select the best description for a given sentence on the basis of its frequency in the whole bank.

**EM disambiguation of reduced parse forests:** In the second subtask, the reduced parse forests in the bank were indeed disambiguated to single valence frames per

clause. It is a standard approach to disambiguate full parse forests with a probabilistic context-free grammar (PCFG). However, reformulating Świdziński’s metamorphosis grammar as a pure CFG and the subsequent unsupervised (for the lack of a treebank) PCFG training would take too much work for our purposes. Thus we have disambiguated reduced parse forests by means of the EM selection algorithm introduced in Appendix C. Let  $A_i$  be the set of reduced parse trees for the  $i$ -th sentence in the bank,  $i = 1, 2, \dots, M$ . We set the initial  $p_j^{(1)} = 1$  and applied the iteration (11)–(12) from Appendix C until  $n = 10$ . Then one of the shortest parses with the largest conditional probability  $p_{ji}^{(n)}$  was sampled at random.

Just to investigate the quality of this disambiguation, we prepared a test set of 190 sentences with the correct reduced parses indicated manually. Since the output of our disambiguation procedure is stochastic and the test set was small, we performed 500 Monte Carlo simulations on the whole test set. Our procedure chose the correct reduced parse for 72.6% sentences on average. Increasing the number of the EM iterations to  $n = 20$  did not improve this result. As a comparison, sampling simply a parse  $j$  with the largest  $p_{ji}^{(n)}$  yielded an accuracy of 72.4%, sampling a parse with the minimal length was accurate in 57.5% cases, whereas blind sampling (assuming equidistribution) achieved 46.9%. The difference between 72.6% and 72.4% is not significant but, given that it does not spoil our results, we prefer using shorter parses.

**Computing the preliminary dictionary from parses:** Once the reduced parse forests in the bank had been disambiguated, a frequency table of the disambiguated reduced parses was computed. This will be referred to as the preliminary valence dictionary. The entries in this dictionary looked like this:

```
'przyłapać' => {
  'np(acc),np(gen),np(nom)' => 1,
+ 'na+np(loc),np(nom),sie' => 1,
  'na+np(loc),np(gen),np(nom)' => 1,
+ 'np(acc),np(nom)' => 4,
  'adv,np(nom)' => 1,
+ 'na+np(loc),np(acc),np(nom)' => 3
}
```

The numbers are the obtained reduced parse frequencies, whereas the correct valence frames are marked with a '+', cf. (1). Notice that the counts for each parse are low. We chose a low frequency verb for this example to make it short. Another natural method to obtain a preliminary dictionary was to use  $Mp_j^{(n)}$  coefficients as the frequencies of frames. This method yields final results that are 1% worse than for the dictionary based on the frequency table.

**Filtering of the preliminary dictionary:** The preliminary dictionary contains many incorrect frames, which are due to parsing or disambiguation errors. In the last subtask, we filtered this dictionary using supervised learning, as done commonly in related work.

For example, the BHT filtering by Brent (1993) is as follows. Let  $c(v, f)$  be the count of reduced parses in the preliminary dictionary that contain both verb  $v$  and valence frame  $f$ . Denote the frequency of verb  $v$  as  $c(v) = \sum_f c(v, f)$ . Frame  $f$  is retained in

the set of valence frames  $\mathbf{F}(v)$  if and only if

$$\sum_{n=c(v,f)}^{c(v)} \binom{c(v)}{n} p_f^n (1 - p_f)^{c(v)-n} \leq \alpha, \quad (6)$$

where  $\alpha = 0.05$  is the usual significance level and  $p_f$  is a frequency threshold. The parameter  $p_f$  is selected as a value for which the classification rule (6) yields the minimal error rate against the training dictionary. In the idealized language of statistical hypothesis testing,  $p_f$  equals the empirical relative frequency of frame  $f$  for the verbs that *do not select* for  $f$  according to the ideal dictionary.

We have used the BHT as the baseline, against which we have tested a new procedure of frame filtering. The new procedure applied the co-occurrence matrices presented in Section 2. It was as follows:

1. Compute  $\mathbf{L}(v)$  and  $\mathbf{E}(v)$  via Equation (2) from the sets of valence frames  $\mathbf{F}(v)$  given by the preliminary dictionary.
2. Correct  $\mathbf{L}(v)$  and  $\mathbf{E}(v)$  using the training dictionary.
3. Reconstruct  $\mathbf{F}(v)$  given the new  $\mathbf{L}(v)$  and  $\mathbf{E}(v)$ . This reconstruction is defined as the substitution  $\mathbf{F}(v) \leftarrow \{(f \cup \mathbf{E}(v)) \cap \mathbf{L}(v) \mid f \in \mathbf{F}(v)\}$ .
4. Compute  $\mathbf{M}(v)$  from  $\mathbf{F}(v)$  via Equation (3).
5. Correct  $\mathbf{M}(v)$  using the training dictionary.
6. Reconstruct  $\mathbf{F}(v)$  given the new  $\mathbf{M}(v)$ . This reconstruction consists of substitution  $\mathbf{F}(v) \leftarrow \bar{\mathbf{F}}(v)$ , where  $\bar{\mathbf{F}}(v)$  is defined in Equation (5) and computed via the procedure described in Appendix A.
7. Output  $\mathbf{F}(v)$  as the valence of verb  $v$ .

Steps 2. and 5. are described in Subsections 3.2 and 3.3 respectively.

In our experiment, the training dictionary consisted of valence frames for 832 verbs from the dictionary of Świdziński (1994). It contained all verbs in Świdziński's dictionary except those included in the test set introduced in Section 4.

### 3.2. FILTERING OF THE ARGUMENT SETS

For simplicity of computation, the correction of argument sets  $\mathbf{L}(v)$  and  $\mathbf{E}(v)$  was done by setting thresholds for the frequency of arguments as in the maximum likelihood thresholding test for frames (MLE) proposed by Korhonen (2002). Thus a possible argument  $a$  for verb  $v$  was retained if it accounted for a certain proportion of the verb's frames in the corpus. Namely,  $a$  was kept in  $\mathbf{L}(v)$  if and only if

$$c(v, a) \geq p_a c(v) + 1, \quad (7)$$

where  $c(v)$  is the frequency of reduced parses in the preliminary dictionary that contain  $v$ , as in (6), and  $c(v, a)$  is the frequency of parses that contain both  $v$  and  $a$ . Parameter  $p_a$  was evaluated as dependent on the argument but independent of the verb. The optimal  $p_a$  was selected as a value for which the classification rule (7) yielded the minimal error rate against the training dictionary.



The difference between the BHT and the MLE is negligible if the count of the verb  $c(v)$  and the frequency threshold  $p_a$  are big enough. This condition is not always satisfied in our application but we preferred MLE for its computational simplicity and its lack of need to choose an appropriate significance level  $\alpha$ . In a preceding subexperiment, we had also tried out the more general model  $c(v, a) \geq p_a c(v) + t_a$  instead of (7), where  $t_a$  was left to vary. Since  $t_a = 1$  was learned for the vast majority of  $a$ 's then we set constant  $t_a = 1$  for all verb arguments later.

Since the same error rate could be obtained for many different values of  $p_a$ , we applied a discrete minimization procedure to avoid overtraining and excessive searching. Firstly, the resolution level  $N := 10$  was initialized. In the following loop, we checked the error rate for each  $p_a := n/N$ ,  $n = 0, 1, \dots, N$ . The number of distinct  $p_a$ 's yielding the minimal error rate was determined and called the degeneration  $D(N)$ . For  $D(N) < 10$ , the loop was repeated with  $N := 10N$ . In the other case, the optimal  $p_a$  was returned as the median of the  $D(N)$  distinct values that allowed the minimal error rate. Selecting the median was inspired by the maximum-margin hyperplanes used in support vector machines to minimize overtraining (Vapnik, 1995).

Similar supervised learning was used to determine whether a given argument is strictly compulsory for a verb. By symmetry, an argument  $a$  that was found possible with verb  $v$  was considered as required unless it was rare enough. Namely,  $a \in \mathbf{L}(v)$  was included in the new  $\mathbf{E}(v)$  unless

$$c(v) - c(v, a) \geq p_{\neg a} c(v) + 1, \quad (8)$$

where  $p_{\neg a}$  was another parameter, estimated analogously to  $p_a$ .

### 3.3. CORRECTION OF THE CO-OCCURRENCE MATRICES

Once we had corrected the argument sets in the preliminary dictionary, the respective co-occurrence matrices still contained some errors when compared with the training dictionary. However, the number of those errors was relatively small and it was not so trivial to propose an efficient scheme for their correction.

A possible approach to such correction is to develop statistical tests with clear null hypotheses that would detect structural zeroes in contingency tables

	$a \notin f$	$a \in f$
$b \notin f$	$N - N_a - N_b + N_{ab}$	$N_a - N_{ab}$
$b \in f$	$N_b - N_{ab}$	$N_{ab}$

where  $N = |\mathbf{F}(v)|$ ,  $N_a = |\langle a \rangle|$ ,  $N_b = |\langle b \rangle|$ , and  $N_{ab} = |\langle a \rangle \cap \langle b \rangle|$  are appropriate counts of frames. Relations  $\leftarrow$ ,  $\rightarrow$ ,  $\leftrightarrow$ , and  $\times$  correspond to particular configurations of structural zeroes in these tables.

Constructing structural zero detection tests appeared to be difficult under the common-sense requirement that the application of these tests cannot diminish the agreement score (4) between the corrected dictionary and the training dictionary. We have experimented with several such schemes but they did not pass the aforementioned criterion empirically. Eventually, we have discovered successful correction methods which rely on the fact that values of matrix cells for fixed arguments tend not to depend on a verb, see Section 2.

In this paper we compare three such correction methods. Let us denote the value of a cell  $\mathbf{M}(v)_{ab}$  after Step 4 as  $S$ . On the other hand, let  $R$  be the most frequent

relation for arguments  $a$  and  $b$  given by the training dictionary across different verbs. We considered the following correction schemes:

- (A)  $\mathbf{M}(v)_{ab}$  is left unchanged (the baseline):  $\mathbf{M}(v)_{ab} \leftarrow S$ .
- (B)  $\mathbf{M}(v)_{ab}$  becomes verb-independent:  $\mathbf{M}(v)_{ab} \leftarrow R$ .
- (C) We use the most prevalent value only if there is enough evidence for a verb-independent interaction:

$$\mathbf{M}(v)_{ab} \leftarrow \begin{cases} R, & C(a R b) \geq p_{S \Rightarrow R} C(a, b) + t_{S \Rightarrow R}, \\ S, & \text{else,} \end{cases} \quad (9)$$

where  $C(a R b)$  is the number of verbs for which  $a R b$  is satisfied and  $C(a, b)$  is the number of verbs that take both  $a$  and  $b$ ; both numbers relate to the training dictionary. Coefficients  $p_{S \Rightarrow R}$  and  $t_{S \Rightarrow R}$  are selected as the values for which rule (9) returns the maximal agreement score (4) against the training dictionary.

There were only a few relation pairs  $S \Rightarrow R$  for which method (C) performed substitutions  $\mathbf{M}(v)_{ab} \leftarrow R$  when applied to our data. These were:  $\leftarrow \Rightarrow \times$ ,  $\rightarrow \Rightarrow \times$ ,  $\perp \Rightarrow \leftarrow$ ,  $\perp \Rightarrow \rightarrow$ , and  $\perp \Rightarrow \times$ . Unlike the case of argument filtering, the optimal  $t_{S \Rightarrow R}$  was equal to 1 only for one relation pair, namely  $\perp \Rightarrow \times$ . The evaluation of methods (A), (B) and (C) against an appropriate test set is presented in Section 4.3.

## 4. Evaluation of the dictionary

### 4.1. OVERVIEW

Having applied the procedures described in Section 3, we obtained an automatically extracted valence dictionary that included 5443 verb entries after Step 6, which is five times more than in Świdziński (1994). As mentioned in the previous section, all parameters were trained on frame sets provided by Świdziński (1994) for 832 verbs. In contrast, the valence frames in our test set were simultaneously given by Świdziński (1994), Bańko (2000), and Polański (1992) for 201 verbs different from the training verbs. Except for 5 verbs missing in Polański and one missing in Bańko, each verb in the test set was described by all dictionaries and we kept track of which dictionary contributed which frame.

We preferred to compare the automatically extracted dictionary with three reference dictionaries at once to sort out possible mistakes in them. In particular, the majority voting (MV) of the three dictionaries was also considered. The verbs for the test set were selected by hand for the following reasons: Firstly, each reference dictionary contained a different set of verbs in its full version. Secondly, entries from the dictionaries by Bańko and Polański had to be typed into the computer manually and interpreted by an expert since these authors often described arguments abstractly, like the “adverbial of time/direction/cause/degree”, rather than as NPs, PPs or adverbs. Thirdly, verbs taking rare arguments were intentionally overrepresented in our test set. Although we could not enlarge or alter the test set easily to perform reasonable  $n$ -fold cross-validation, the variation of scores can be seen by comparing different automatically extracted

Table I. The evaluation of argument filtering.

POSSIBLE	$p_a$	P	GSP	FN	FP	E
np(nom)	0.06	199	201	2	0	2
np(acc)	0.08	126	142	25	9	34
sie	0.08	71	96	29	4	33
np(dat)	0.02	65	80	26	11	37
np(inst)	0.04	39	61	31	9	40
ZE	0.13	26	54	30	2	32
adv	0.18	56	46	23	33	56
do+np(gen)	0.07	25	46	25	4	29
na+np(acc)	0.06	17	41	25	1	26
PZ	0.06	3	31	28	0	28
w+np(loc)	0.34	1	30	30	1	31
z+np(inst)	0.08	8	28	20	0	20
BY	0.14	4	28	26	2	28
inf	0.1	14	27	13	0	13
np(gen)	0.31	8	24	17	1	18
z+np(gen)	0.08	7	23	19	3	22
w+np(acc)	0.06	8	19	14	3	17
o+np(loc)	0.03	11	19	8	0	8
za+np(acc)	0.03	3	17	15	1	16
od+np(gen)	0.1	2	17	15	0	15
o+np(acc)	0.01	13	16	6	3	9
adj(nom)	0.77	1	3	2	0	2
NOT REQUIRED	$p_{\neg a}$	P	GSP	FN	FP	E
np(nom)	0.54	3	19	19	3	22
np(acc)	0.24	174	174	10	10	20
sie	0.12	186	188	5	3	8
do+np(gen)	0.04	201	199	0	2	2
inf	0.13	199	199	0	0	0
np(dat)	0.02	201	199	0	2	2

dictionaries with different gold-standard dictionaries. We find this more informative for future research than the standard cross-validation.

The evaluation is divided into three parts. We analyze some specific errors of our two-stage approach, each stage assessed separately. In the following, we relate our results to previous research.

#### 4.2. ANALYSIS OF THE ARGUMENT FILTERING

Table I presents the results for parameters  $p_a$  and  $p_{\neg a}$  tested solely on Świdziński (1994) for the 201 test verbs. The notations in the column titles are: P – the number of positive outcomes in the automatically extracted dictionary after Step 3 of dictionary filtering (one outcome is one verb taking the argument), GSP – the number of gold-standard

positive outcomes in Świdziński ( $GSP = P - FP + FN$ ),  $FN$  – the number of false negatives,  $FP$  – the number of false positives, and  $E$  – the number of errors ( $E = FN + FP$ ). We have  $0 \leq FN, FP \leq GSP, P, E \leq 201$ . The notations for certain arguments in the table rows are: **sie** – the reflexive marker *sie*, **x+np(y)** – the prepositional phrase introduced by preposition  $x$  requiring a noun in case  $y$ , **ZE** – the clause introduced by *że* (= *that*), **PZ** – the clause introduced by *czy* (= *whether*), and **BY** – the clause introduced by *żeby* (= *so as to*).

Although the overall precision of single argument extraction is high (it reaches 89%, see the (verb, argument) scores in Table II below), all numerical values for this task depend heavily on the type of extracted argument. The case of frequency thresholds  $p_a$ , being in the range of 0.02–0.77, is notable. These thresholds are higher for arguments that can be used as NP modifiers, e.g. **adj(nom)** and **np(gen)**, or verbal adjuncts, e.g. **adv** and **w+np(loc)**. In general, the errors concentrate on low-frequency arguments. That occurs probably because the frequency of tokens coming from parsing errors does not depend systematically on the argument type. Thus this frequency dominates the frequency of tokens coming from well parsed sentences for low-frequency types. Except for the extraction of a direct object **np(acc)** and adverbial phrase **adv**, gold-standard positive outcomes ( $GSP$ ) outnumber the positive ones ( $P$ ). Put differently, false positives ( $FP$ ) are fewer than false negatives ( $FN$ )—although the learning objective was set to minimize the error rate ( $E = FP + FN$ ). The same phenomenon appears in Brent (1993).

We have also noticed that the extracted valences are better for less frequent verbs. We can see several reasons for this. Firstly, there are more types of infrequent verbs than of frequent ones, so thresholds  $p_a$  get more adjusted to the behaviour of less frequent verbs. Secondly, the description of infrequent verb valences given by the training dictionary is less detailed. In particular, the gold-standard dictionary fails to cover less frequent arguments that are harder to extract. Unfortunately, the small size of our training and test data does not enable efficient exploration of how thresholds  $p_a$  could depend on the frequency of the verb. According to Table I, about half of the argument types were acknowledged in the test data for just a few verbs.

The arguments that we found particularly hard to extract are the adverbs (**adv**), with inequality  $P > GSP$ , and a group of arguments with  $P$  much smaller than  $GSP$ . The latter include several adjunct-like prepositional phrases (e.g., **w+np(loc)**, *w* means *in*), certain clauses (**PZ** and **BY**), and the possible lack of subject **np(nom)** (= non-required **np(nom)**), which corresponds roughly to the English expletive *it*. The inequality  $P > GSP$  for adverbs probably reflects their inconsistent recognition as verb arguments in the gold standard.

The climbing of clitics and objects was another important problem that we came across when we studied concrete false positives. Namely, some arguments of the Polish infinitive phrase required by a finite verb can be placed anywhere in the sentence. In contrast to Romance languages, this phenomenon concerns not only clitics. Unfortunately, Świdziński’s grammar does not model either object or clitic climbing and this could have caused the following FPs:

- 4 of 9 outcomes for **np(acc)**: *kazać* (= *order*), *móc* (= *may*), *musieć* (= *must*), *starać (sie)* (= *make efforts*),
- 3 of 11 outcomes for **np(dat)**: *móc*, *pragnąć* (= *desire/wish*), *starać (sie)*.

There were no FPs that could be attributed to the climbing of the reflexive marker *się*, although this clitic climbs most often. For no clear reason, the optimal threshold  $p_a$  for *się* was much higher for the training dictionary than for the test dictionary.

These three frequent arguments also featured relatively many FPs that were due to omissions in the test dictionary:

- 1 of 9 outcomes for np(acc): *skarżyć* (= *accuse*),
- all outcomes for *się*: *pogorszyć* (= *make worse*), *przyzwyczajać* (= *get used*), *wylewać* (= *pour out*), *związać* (= *bind*),
- 6 of 11 outcomes for np(dat): *ciec* (= *flow*), *dostosować* (= *adjust*), *drżeć* (= *thrill*), *dźwigać* (= *carry*), *ratować* (= *save*), *wsadzić* (= *put into*).

As we can see, almost all FPs for these arguments are connected either to clitic and object climbing or to omissions in the test set. There is room for substantial improvement both in the initial corpus parsing and in the test dictionaries.

### 4.3. EVALUATION OF THE CO-OCCURRENCE MATRIX ADJUSTMENT

We obtained the following agreement scores for the three methods of co-occurrence matrix adjustment defined in Section 3.3:

	agreement score
method (A) — no adjustment (baseline)	77%
method (B) — verb-independent matrices	80%
method (C) — a combination of those	83%

The scores are statistics (4) computed on the 201 test verbs for the dictionary of Świdziński (1994) and the preliminary dictionary processed until Step 6. Method (C) gave the best results so it is the only method considered subsequently.

In more detail, Table II presents scores for all manually compiled dictionaries and the automatically extracted dictionary at several stages of filtering: AE is the preliminary dictionary, AE-A is the dictionary after correcting the argument sets (Step 3), AE-C is the one where co-occurrence matrices were corrected using method (C) (Step 6), and AE-F is the baseline filtered only with the frame-based binomial hypothesis test (6). We have constructed several dictionaries derived from these, such as set-theoretic unions, intersections, or majority voting, but present only the best result—the AE-C+F, which is the union of frames from the two-stage filtered AE-C and the one-stage filtered AE-F. The displayed MV is the majority voting of Bańko, Polański, and Świdziński, which are denoted as Bań., Pol., and Świ.

Each cell of two triangular sections of Table II presents the number of pairs, (verb, frame) or (verb, argument), that appear simultaneously in two dictionaries specified by the row and column titles counted for the 201 test verbs. The displayed recall, precision, and  $F$ -score were computed against the MV dictionary. Recall and precision against other dictionaries can be computed from the numbers given in the triangular sections.

Although a large variation of precision and recall can be observed in Table II, the  $F$ -scores do not vary so much. Assuming the  $F$ -score as an objective to be maximized,

the two-stage filtering is better than the frame-based BHT. Namely, we have  $F = 42\%$  for the AE-C whereas  $F = 39\%$  for the AE-F, the scores referring to pairs (verb, frame). The set-theoretic union of both dictionaries, AE-C+F, exhibits even a larger  $F = 45\%$ . In the case of not displayed dictionaries, we have observed the following triples of recall/precision/ $F$ -score: (a) 20%/81%/32% for the intersection of AE-A, AE-C, and AE-F, (b) 33%/61%/43% for their majority voting, (c) 39%/45%/42% for their union, and (d) 39%/46%/42% for the union of just AE-A and AE-F.

The precision of both AE-C and AE-F with respect to the MV is equal to or higher than that of manually edited dictionaries, whether we look at single arguments or at frames. A word of caution is in order, however. Very high precision against the MV test dictionary, provided the recall is sufficient, is a desirable feature of the automatically extracted dictionary. The converse should be expected for the contributing sources of the MV dictionary. These should be favoured for presenting frames not occurring in other sources provided all frames are true. Formally, the contributing sources should feature very high recall and relatively lower precision against their MV aggregate. Exactly this can be observed in Table II.

In general, through the correction of co-occurrence matrices in Step 5 and the frame reconstruction (5), more frames are deleted from the AE-A dictionary than added. The AE-A contains 338 pairs (verb, frame) which do not appear in the obtained AE-C dictionary, whereas only 13 such pairs from the AE-C are missing in the AE-A. The sets of pairs (verb, argument) are almost the same for both dictionaries.

A problem that is buried in the apparently good-looking statistics is the actual shape of co-occurrence matrices in the AE-C dictionary. In Step 5 of dictionary filtering, many matrix cells are reset as independent of the verb. This affects verbs such as *dziwić* (= *surprise/wonder*). The correct set of frames for this verb is close to

$$\mathbf{F}(\textit{dziwić}) = \left\{ \begin{array}{l} \{\textit{np}(\textit{nom}), \textit{np}(\textit{acc})\}, \\ \{\textit{ZE}, \textit{np}(\textit{acc})\}, \\ \{\textit{np}(\textit{nom}), \textit{sie}\}, \\ \{\textit{np}(\textit{nom}), \textit{sie}, \textit{np}(\textit{dat})\}, \\ \{\textit{np}(\textit{nom}), \textit{sie}, \textit{ZE}\} \end{array} \right\}. \quad (10)$$

The subordinate clause **ZE** excludes subject **np(nom)** when *sie* is missing but it excludes direct object **np(acc)** when *sie* is present (for there is a reflexive diathesis, *dziwić sie=be surprised*).

The reconstruction (5) does not recover the frame set (10) properly for two reasons. Firstly, clause **ZE** excludes **np(acc)** and implies **np(nom)** for the majority of verbs. Secondly, the co-occurrence matrix formalism cannot model any pairwise exclusion that is conditioned on the absence or presence of another argument. However, we suppose that such an argument interaction is very rare and this deficiency is not so important en masse.

#### 4.4. COMPARISON WITH PREVIOUS RESEARCH

The scores reported in the literature of verb valence extraction are so varied that fast conclusions should not be drawn from just a single figure. For example, Brent (1993) achieved 60% recall and 96% precision in the unsupervised approach. This was done for English and for a very small set of extracted valence frames (the set counted only 6 distinct frames). English-based researchers that evaluated their extracted valence dictionaries against more complex test dictionaries reported the following pairs

Table II. The comparison of all dictionaries.

(verb, frame)	AE	AE-A	AE-C	AE-C+F	AE-F	Bań.	Pol.	Świ.	MV
AE	7877								
AE-A	848	983							
AE-C	587	645	658						
AE-C+F	675	674	658	746					
AE-F	413	354	325	413	413				
Bań.	857	494	418	469	311	1660			
Pol.	699	415	359	400	275	778	1536		
Świ.	697	409	363	406	294	766	778	1374	
MV	701	444	394	441	311	992	1004	992	1218
recall	0.58	0.36	0.32	0.36	0.26	0.81	0.82	0.81	
precision	0.09	0.45	0.60	0.59	0.75	0.6	0.65	0.72	
F	0.16	0.40	0.42	0.45	0.39	0.69	0.73	0.76	

  

(verb, argument)	AE	AE-A	AE-C	AE-C+F	AE-F	Bań.	Pol.	Świ.	MV
AE	4051								
AE-A	687	687							
AE-C	674	674	674						
AE-C+F	735	680	674	735					
AE-F	582	527	521	582	582				
Bań.	1093	611	603	639	524	1342			
Pol.	1033	593	586	623	520	966	1336		
Świ.	988	589	581	618	521	907	963	1265	
MV	1007	608	600	638	530	1066	1122	1063	1222
recall	0.82	0.50	0.49	0.52	0.43	0.87	0.92	0.87	
precision	0.25	0.89	0.89	0.87	0.91	0.79	0.84	0.84	
F	0.38	0.64	0.63	0.65	0.58	0.83	0.88	0.85	

of recall/precision: 36%/66% (Briscoe and Carroll, 1997) against the COMLEX and ANLT dictionaries, 43%/90% (Manning, 1993) against *The Oxford Advanced Learner's Dictionary*, and 75%/79% (Carroll and Rooth, 1998) against the same dictionary.

Other factors matter as well. Korhonen (2002, page 77) demonstrated that the results depend strongly on the filtering method: BHT gives 56%/50%, LLR — 48%/42%, MLE — 58%/75%, no filtering — 84%/24%, all methods being frame-based and applied to the same English data. For Czech, a close relative of Polish, Sarkar and Zeman (2000) found the recall/precision pair 74%/88% but these were evaluated against a manually annotated sample of texts rather than against a gold-standard valence dictionary. Moreover, Sarkar and Zeman acquired valence frames from a manually disambiguated treebank rather than from raw data, so automatic parsing did not contribute to the overall error rate.

The closest work to ours is Fast and Przepiórkowski (2005), who regarded their own work as preliminary. They also processed only a small part of the 250-million-word IPI PAN Corpus. Approximately 12 million running words were parsed but sentence

parsing was done with a simple 18-rule regular grammar rather than with Świdziński’s grammar. Moreover, the dictionary filtering was done according to several frame-based methods discussed in the literature and the reference dictionary used was only a small part of Świdziński (1994)—100 verbs for a training set and another 100 verbs for a test set. In contrast to our experiment, Fast and Przepiórkowski extracted only non-subject NPs and PPs. They ignored subjects, `np(nom)`, since almost all verbs subcategorize for them. The best score in the complete frame extraction they reported was 48% recall and 49% precision ( $F = 48\%$ ), which was obtained for the supervised version of the binomial hypothesis test (6).

So as to come closer to the experimental setup of Fast and Przepiórkowski, we reapplied all frame filtering schemes to the case when only non-subject NPs and PPs were retained in the preliminary dictionary AE and the three manually edited dictionaries. The statistics are provided in Table III. Under these conditions our two-stage filtering method added to the frame-based BHT is better again than any of these methods separately;  $F = 57\%$  for the AE-C+F vs.  $F = 53\%$  for both the AE-F and AE-C. The AE-C+F is not only better than the AE-F and AE-C with respect to  $F$ -score but it also contains 15% to 38% more frames. Much higher precision of all these dictionaries than reported by Fast and Przepiórkowski (2005) may be attributed to the deep sentence parsing with Świgr and the EM disambiguation. The best recall remains almost the same (47%) for the AE-C+F dictionary, although we extracted valences from a four fold smaller amount of text.

## 5. Conclusion

Two new ideas for valence extraction have been proposed and applied to Polish language data in this paper. Firstly, we have introduced a two-step scheme for filtering incorrect frames. The list of valid arguments was determined for each verb first and then a method of combining arguments into frames was found. The two-stage induction was motivated by an observation that the argument combination rules, such as co-occurrence matrices, are largely independent of the verb. We suppose that this observation is not language-specific and the co-occurrence matrix formalism can be easily tailored to improve verb valence extraction for many other languages and special datasets (also subdomain corpora and subdomain valence dictionaries). The second new idea is a simple EM selection algorithm, which is a natural baseline method for unsupervised disambiguation tasks such as choosing the correct valence frame for a sentence. In our application it helped high-precision valence extraction without a large treebank or a probabilistic parser.

Although the proposed frame filtering technique needs further work to address the drawbacks noticed in Subsection 4.3 and to improve the overall performance, the present results are encouraging and suggest that two-step frame filtering is worth developing. In future work, experiments can be conducted using various schemes of decomposing the information contained in the sets of valence frames and, due to the scale of the task, this decomposition should be done to a large extent in an algorithmic way. The straightforward idea to explore is to express the verb valence information in terms of  $n$ -ary rather than binary relations among verbs and verb arguments, where  $n > 2$ . Subsequently, one can investigate the analogous learning problem and propose a frame-set reconstruction scheme for the  $n$ -ary relations. Are ternary relations sufficient to describe the valence frame sets? We disbelieve that relations of irreducibly large arities



Table III. The case of source dictionaries restricted to non-subject NPs and PPs.

(verb, frame)	AE	AE-A	AE-C	AE-C+F	AE-F	Bań.	Pol.	Świ.	MV
AE	3746								
AE-A	695	713							
AE-C	533	539	544						
AE-C+F	615	585	544	626					
AE-F	453	417	371	453	453				
Bań.	827	481	407	463	377	1255			
Pol.	693	426	367	412	338	684	1128		
Świ.	645	422	368	413	346	662	661	939	
MV	694	455	395	446	372	820	819	797	955
recall	0.73	0.48	0.41	0.47	0.39	0.86	0.86	0.83	
precision	0.19	0.64	0.73	0.71	0.82	0.65	0.73	0.85	
F	0.30	0.55	0.53	0.57	0.53	0.74	0.79	0.84	

  

(verb, argument)	AE	AE-A	AE-C	AE-C+F	AE-F	Bań.	Pol.	Świ.	MV
AE	2364								
AE-A	392	392							
AE-C	385	385	385						
AE-C+F	415	388	385	415					
AE-F	354	327	324	354	354				
Bań.	717	353	349	369	322	881			
Pol.	659	333	330	346	306	603	813		
Świ.	585	323	319	334	296	547	567	715	
MV	633	346	342	360	317	665	685	629	747
recall	0.85	0.46	0.46	0.48	0.42	0.89	0.92	0.84	
precision	0.27	0.88	0.89	0.87	0.90	0.75	0.84	0.88	
F	0.41	0.60	0.61	0.62	0.57	0.81	0.88	0.86	

appear in human language lexicons since, for example, Halford et al. (1998) observed that human capacity for processing random  $n$ -ary relations depends strongly on the relation arity.

Knowing algebraic constraints on the verb argument combinations is important also for language resource maintenance. Because our test dictionaries do not list valid argument combinations extensively, many false positive frames in the two-stage corrected dictionary were in fact truly positive. Thus, it is advisable to correct gold-standard dictionaries themselves, for example using a modification of the reconstruction (5). However, prior to resetting the gold-standard in this way, it must be certain that the reconstruction process does not introduce linguistically implausible frames. Also for this reason, the effective complexity of verb-argument and argument-argument relations in natural language should be investigated thoroughly from a more mathematical point of view.

## Appendix

### A. A faster reconstruction of the frame set

Although there is no need to compute  $\bar{\mathbf{F}}(v)$  defined in (5) to verify condition  $f \in \bar{\mathbf{F}}(v)$  for a given  $f$ , the reconstruction  $\bar{\mathbf{F}}(v)$  can be computed efficiently if needed for other purposes. A naive solution suggested by formula (5) is to search through all elements of the power set  $2^{\mathbf{L}(v)}$  and to check for each independently whether it is an element of  $\bar{\mathbf{F}}(v)$ . However, we can do it faster by applying some dynamic programming.

Firstly, let us enumerate the elements of  $\mathbf{L}(v) = \{b_1, b_2, \dots, b_N\}$ . In the following, we will compute the chain of sets  $A_0, A_1, \dots, A_N$  where  $A_n = \{(B_n \cap f, B_n \setminus f) \mid f \in \bar{\mathbf{F}}(v)\}$  and  $B_n = \{b_1, b_2, \dots, b_n\}$ .

In fact, there is an iteration for this chain:

$$A_0 = \{(\emptyset, \emptyset)\},$$

$$A_n = \left\{ (f \cup \{b_n\}, g) \left| \begin{array}{l} (f, g) \in A_{n-1}, \\ \forall a \in f \mathbf{M}(v)_{b_n a} \neq \times, \\ \forall a \in g \mathbf{M}(v)_{b_n a} \neq \leftrightarrow, \\ \forall a \in g \mathbf{M}(v)_{b_n a} \neq \leftarrow \end{array} \right. \right\}$$

$$\cup \left\{ (f, g \cup \{b_n\}) \left| \begin{array}{l} (f, g) \in A_{n-1}, \\ \{b_n\} \notin \mathbf{E}(v), \\ \forall a \in f \mathbf{M}(v)_{b_n a} \neq \leftrightarrow, \\ \forall a \in f \mathbf{M}(v)_{b_n a} \neq \leftarrow \end{array} \right. \right\}.$$

Once the set  $A_N = \{(f, \mathbf{L}(v) \setminus f) \mid f \in \bar{\mathbf{F}}(v)\}$  is computed,  $\bar{\mathbf{F}}(v)$  can be read off easily.

### B. Parsing of the IPI PAN Corpus

The input of the valence extraction experiment discussed in this paper came from the 250-million-word IPI PAN Corpus of Polish (<http://korpus.pl/>). The original automatic part-of-speech annotation of the text was removed, since it contained too many errors, and the sentences from the corpus were analyzed using the Świgr parser (Woliński, 2004, 2005), see also <http://nlp.ipipan.waw.pl/~wolinski/swigra/>. Technically, Świgr utilizes two distinct language resources: (1) Morfeusz—a dictionary of inflected words (a.k.a. a morphological analyzer) programmed by Woliński (2006) on the basis of about 20,000 stemming rules compiled by Tokarski (1993), and (2) GFJP—the formal grammar of Polish written by Świdziński (1992). Świdziński’s grammar is a DCG-like grammar, close to the format of the metamorphosis grammar by Colmerauer (1978). It counts 461 rules and examples of its parse trees can be found in Woliński (2004). For the sake of this project, Świgr used a fake valence dictionary that allowed any verb to take none or one NP in the nominative (the subject) and any combination of other arguments.

Only a small subset of sentences was actually selected to be parsed with Świgr. The following selection criteria were applied to the whole 250-million-word IPI PAN Corpus:

1. The selected sentence had to contain a word recognized by Morfeusz as a verb and the verb had to occur  $\geq 396$  times in the corpus. (396 is the lowest corpus frequency

of a verb from the test set described in Section 4. The threshold was introduced to speed up parsing without loss of empirical coverage for any verb in the test set. The selected sentence might contain another less frequent verb if it was a compound sentence.)

2. The selected sentence could not be longer than 15 words. (We supposed that the EM selection would find it difficult to select the correct parse for longer sentences.)
3. Maximally 5000 sentences were selected per recognized verb. (We supposed that a frame which was used less than once per one 5000 verb occurrences would not be considered in the gold-standard dictionaries.)

In this way, a subset of 1 011 991 sentences (8 727 441 running words) was chosen. They were all fed to Świgr’s input but less than half (0.48 million sentences) were parsed successfully within a preset time of 1 minute per sentence. Detailed statistics are given in Table IV below. All mentioned thresholds were introduced in advance to compute only the most useful parse forests in the pre-estimated total time of a few months. It was the first experiment ever in which Świgr was applied to more than several hundred sentences. The parsing actually took 2 months on a single PC station.

Not all information contained in the obtained parse forests was relevant for valence acquisition. Full parses were subsequently reduced to valence frames plus verbs, as in the first displayed example in Section 3. First of all, the parse forests for compound sentences were split into separate parse forests for elementary clauses. Then each parse tree was reduced to a string that identifies only the top-most phrases. To decrease the amount of noise in the subsequent EM selection and to speed up computation, we decided to skip 10% of clauses that had the largest number of reduced parses. As a result, we only retained clauses which had  $\leq 40$  reduced parses.

To improve the EM selection, we also deleted parses that contained certain syntactically idiosyncratic words—mostly indefinite pronouns *to* (= *this*), *co* (= *what*), and *nic* (= *nothing*)—or highly improbable morphological word interpretations (like the second interpretation for *albo* = 1. the conjunction *or*; 2. the vocative singular of the noun *alb*—a kind of liturgical vestment). The stop list of improbable interpretations consisted of 646 word interpretations which never occurred in the SFPW Corpus but were possible interpretations of the most common words according to Morfeusz. The SFPW Corpus is a manually POS tagged 0.5-million-word corpus prepared for the frequency dictionary of 1960s Polish (Kurcz et al., 1990), which was actually commenced in the 1960s but not published until 1990.

Our format of reduced parses approximates the format of valence frames in Świdziński (1994), so it diverges from the format proposed by Przepiórkowski (2006). To convert a parse in Przepiórkowski’s format into ours, the transformations must be performed as follows:

1. Add the dropped personal subject or the impersonal subject expressed by the ambiguous reflexive marker *się* when their presence is implied by the verb form.
2. Remove one nominal phrase in the genitive for negated verbs. (An attempt to treat the genitive of negation.)
3. Transform several frequent adjuncts expressed by nominal phrases.

Table IV. Sizes of the processed parts of the IPI PAN Corpus.

	sentences/clauses	words
sentences sent to Świgras's input	1 011 991 sentences	8 727 441
sentences successfully parsed with Świgras	481 039 sentences	3 421 863
sentences with $\leq 40$ parses split into clauses	569 307 clauses	3 149 391
the final bank of reduced parse forests	510 743 clauses	2 795 357

4. Skip the parse if it contains pronouns *to* (= *this*), *co* (= *what*), and *nic* (= *nothing*). (Instead of converting these pronouns into regular nominal phrases.)
5. Remove lemmas from non-verbal phrases and sort phrases in alphabetic order.

The resulting bank of reduced parse forests included 510 743 clauses with one or more proposed valence frames. We parsed successfully only 3.4 million running words of the whole 250-million-word IPI PAN Corpus—four times less than the 12 million words parsed by Fast and Przepiórkowski (2005). However, our superior results in the valence extraction task indicate that skipping a fraction of available empirical data is a good idea if the remaining data can be processed more thoroughly and the skipped portion does not provide different efficiently usable information.

### C. The EM selection algorithm

Consider the following abstract statistical task. Let  $Z_1, Z_2, \dots, Z_M$ , with  $Z_i : \Omega \rightarrow J$ , be a sequence of discrete random variables and let  $Y_1, Y_2, \dots, Y_M$  be a random sample of sets, where each set  $Y_i : \Omega \rightarrow 2^J \setminus \emptyset$  contains the actual value of  $Z_i$ , i.e.,  $Z_i \in Y_i$ . The objective is to guess the conditional distribution of  $Z_i$  given an event  $(Y_i = A_i)_{i=1}^M$ ,  $A_i \subset J$ . In particular, we would like to know the conditionally most likely values of  $Z_i$ . The exact distribution of  $Y_i$  is not known and unfeasible to estimate if we treat the values of  $Y_i$  as atomic entities. We have to solve the task via some rationally motivated assumptions.

Our heuristic solution was iteration

$$p_{ji}^{(n)} = \begin{cases} p_j^{(n)} / \sum_{j' \in A_i} p_{ji'}^{(n)}, & j \in A_i, \\ 0, & \text{else,} \end{cases} \quad (11)$$

$$p_j^{(n+1)} = \frac{1}{M} \sum_{i=1}^M p_{ji}^{(n)}, \quad (12)$$

with  $p_j^{(1)} = 1$ . We observed that coefficients  $p_{ji}^{(n)}$  converge to a value that can be plausibly identified with the conditional probability  $P(Z_i = j | Y_i = A_i)$ .

Possible applications of iteration (11)–(12), which we call the EM selection algorithm, cover unsupervised disambiguation tasks where the number of different values of  $Y_i$  is very large but the internal ambiguity rate (i.e., the typical cardinality  $|Y_i|$ ) is rather small and the alternative choices within  $Y_i$  (i.e., the values of  $Z_i$ ) are highly repeatable. There may be many applications of this kind in NLP and bioinformatics. To our knowledge, however, we present the first rigorous treatment of this particular selection problem.

In this appendix, we will show that the EM selection algorithm belongs to the class of expectation-maximization (EM) algorithms. For this reason, our algorithm resembles many instances of EM used in NLP, such as the Baum-Welch algorithm for hidden Markov models (Baum, 1972) or linear interpolation (Jelinek, 1997). However, normalization (11), which is done over varying sets  $A_i$ —unlike the typical case of linear interpolation, is the singular feature of EM selection. The local maxima of the respective likelihood function also form a convex set, so there is no need to care much for initializing the iteration (11)–(12), unlike e.g. the Baum-Welch algorithm.

To begin with, we recall the universal scheme of EM (Dempster et al., 1977; Neal and Hinton, 1999). Let  $P(Y|\theta)$  be a likelihood function, where  $Y$  is an observed variable and  $\theta$  is an unknown parameter. For the observed value  $Y$ , the maximum likelihood estimator of  $\theta$  is

$$\theta_{\text{MLE}} = \arg \max_{\theta} P(Y|\theta).$$

When the direct maximization is impossible, we may consider a latent discrete variable  $Z$  and function

$$Q(\theta', \theta'') = \sum_z P(Z = z|Y, \theta') \log P(Z = z, Y|\theta''),$$

which is a kind of cross entropy function. The EM algorithm consists of setting an initial parameter value  $\theta_1$  and iterating

$$\theta_{n+1} = \arg \max_{\theta} Q(\theta_n, \theta) \quad (13)$$

until a sufficient convergence of  $\theta_n$  is achieved. It is a general fact that  $P(Y|\theta_{n+1}) \geq P(Y|\theta_n)$  but EM is worth considering only if maximization (13) is easy.

Having outlined the general EM algorithm, we come back to the selection problem. The observed variable is  $Y = (Y_1, Y_2, \dots, Y_M)$ , the latent one is  $Z = (Z_1, Z_2, \dots, Z_M)$ , whereas the parameter seems to be  $\theta_n = (p_j^{(n)})_{j \in J}$ . The appropriate likelihood function remains to be determined. We may suppose from the problem statement that it factorizes into  $P(Z, Y|\theta) = \prod_i P(Z_i, Y_i|\theta)$ . Hence  $Q(\theta', \theta'')$  takes the form

$$Q(\theta', \theta'') = \sum_i \sum_j P(Z_i = j|Y_i = A_i, \theta') \log P(Z_i = j, Y_i = A_i|\theta'').$$

Assume now

$$P(Y_i = A|Z_i = j, \theta) = \begin{cases} g(A), & j \in A, \\ 0, & \text{else,} \end{cases} \quad (14)$$

$$P(Z_i = j|\theta) = p_j \quad (15)$$

for  $\theta = (p_j)_{j \in J}$  and a parameter-free function  $g(\cdot)$  satisfying

$$\sum_{A \in 2^J} \mathbf{1}_{\{j \in A\}} g(A) = 1, \quad \forall j \in J, \quad (16)$$

where

$$\mathbf{1}_{\{\phi\}} = \begin{cases} 1, & \phi \text{ is true,} \\ 0, & \text{else.} \end{cases}$$

For example, let  $g(A) = q^{|A|-1}(1-q)^{|J|-|A|}$ , where  $|A|$  stands for the cardinality of set  $A$  and  $0 \leq q \leq 1$  is a fixed number not incorporated into  $\theta$ . Then the cardinalities of sets  $Y_i$  are binomially distributed, i.e.,  $P(|Y_i|-1|\theta) \sim B(|J|-1, q)$ . This particular form of  $g(A)$ , however, is not necessary to satisfy (16).

The model (14)–(15) is quite speculative. In the main part of this article, we need to model the probability distribution of the reduced parse forest  $Y_i$  under the assumption that the correct parse  $Z_i$  is an arbitrary element of  $Y_i$ . In particular, we have to imagine what  $P(Y_i = A|Z_i = j, \theta)$  is like if  $j$  is a semantically implausible parse. We circumvent the difficulty by saying in (14) that this quantity is the same as if  $j$  were the correct parse.

Assumption (14) leads to an EM algorithm which does not depend on the specific choice of function  $g(\cdot)$ . Therefore the algorithm is rather generic. In fact, (14) assures that  $P(Y_i = A_i|\theta) = g(A_i)P(Z_i \in A_i|\theta)$  and

$$P(Z_i = j|Y_i = A_i, \theta) = P(Z_i = j|Z_i \in A_i, \theta). \quad (17)$$

In consequence, iteration (13) is equivalent to

$$0 = \frac{\partial}{\partial p_j} \left[ Q(\theta_n, \theta) - \lambda \left( \sum_{j' \in J} p_{j'} - 1 \right) \right] \Big|_{\theta=\theta_{n+1}} = \frac{\sum_{i=1}^M p_{ji}^{(n)}}{p_j^{(n+1)}} - \lambda, \quad (18)$$

where  $p_{ji}^{(n)} = P(Z_i = j|Z_i \in A_i, \theta_n)$  is given exactly by (11).

If the Lagrange multiplier  $\lambda$  is assigned the value that satisfies constraint  $\sum_{j \in J} p_j = 1$  then equation (18) simplifies to (12). Hence it becomes straightforward that iteration (11)–(12) maximizes locally the log-likelihood

$$L(\theta) := \log P((Y_i = A_i)_{i=1}^M | \theta) = \log \left[ \prod_{i=1}^M \frac{P(Z_i \in A_i | \theta)}{g(A_i)} \right], \quad (19)$$

or simply  $L^{(n+1)} \geq L^{(n)}$  for

$$L^{(n)} := L(\theta_n) + \sum_{i=1}^M \log g(A_i) = \sum_{i=1}^M \log \left[ \sum_{j \in A_i} p_j^{(n)} \right], \quad n \geq 2.$$

Moreover, there is no need to care for the initialization of iteration (11)–(12) since the local maxima of function (19) form a convex set  $\mathcal{M}$ , i.e.,  $\theta, \theta' \in \mathcal{M} \implies q\theta + (1-q)\theta' \in \mathcal{M}$  for  $0 \leq q \leq 1$ . Hence that function is, of course, constant on  $\mathcal{M}$ . To show this, observe that the domain of log-likelihood (19) is a convex compact set  $\mathcal{P} = \{\theta : \sum_j p_j = 1, p_j \geq 0\}$ . The second derivative of  $L$  reads

$$L_{jj'}(\theta) := \frac{\partial^2 L(\theta)}{\partial p_j \partial p_{j'}} = - \sum_{i=1}^M \frac{\mathbf{1}_{\{j \in A_i\}} \mathbf{1}_{\{j' \in A_i\}}}{\left( \sum_{j'' \in A_i} p_{j''} \right)^2}.$$

Since matrix  $\{L_{jj'}\}$  is negative definite, i.e.,  $\sum_{j,j'} a_j L_{jj'}(\theta) a_{j'} \leq 0$ , function  $L$  is concave. As a general fact, a continuous function  $L$  achieves its supremum on a compact set  $\mathcal{P}$  (Rudin, 1974, Theorem 2.10). If additionally  $L$  is concave and its domain  $\mathcal{P}$  is convex then the local maxima of  $L$  form a convex set  $\mathcal{M} \subset \mathcal{P}$ , where  $L$  is constant and achieves its supremum (Boyd and Vandenberghe, 2004, Section 4.2.2).

## Acknowledgements

Grateful acknowledgements are due to Marcin Woliński for his help in using Świgr, to Witold Kieraś for retyping samples of the test dictionaries, and to Marek Świdziński for offering the source file of his valence dictionary. The author thanks also Adam Przepiórkowski, Jan Mielniczuk, Laurence Cantrill, and the anonymous reviewers for many helpful comments concerning the composition of this article. The work was supported by the Polish State Research Project, 3 T11C 003 28, *Automatyczna ekstrakcja wiedzy lingwistycznej z dużego korpusu języka polskiego*.

## References

- Artstein, R. and M. Poesio: 2008, 'Inter-coder agreement for computational linguistics'. *Computational Linguistics* **34**, 555–596.
- Baker, C. F. and J. Ruppenhofer: 2002, 'FrameNet's Frames vs. Levin's Verb Classes'. In: *Proceedings of the 28th Annual Meeting of the Berkeley Linguistics Society*, pp. 27–38.
- Bańko, M. (ed.): 2000, *Inny słownik języka polskiego*. Warszawa: Wydawnictwo Naukowe PWN.
- Baum, L. E.: 1972, 'Inequality and Associated Maximization Technique In Statistical Estimation of Probabilistic Functions of Markov processes'. *Inequalities* **3**, 1–8.
- Bennett, E. M., R. Alpert, and A. C. Goldstein: 1954, 'Communications through limited questioning'. *Public Opinion Quarterly* **18**(3), 303–308.
- Boyd, S. and L. Vandenberghe: 2004, *Convex Optimization*. Cambridge: Cambridge University Press.
- Brent, M. R.: 1993, 'From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax'. *Computational Linguistics* **19**, 243–262.
- Briscoe, T. and J. Carroll: 1997, 'Automatic Extraction of Subcategorization from Corpora'. In: *Proceedings of the 5th ACL Conference on Applied Natural Language Processing, Washington, DC*. Morgan Kaufmann, pp. 356–363.
- Carroll, G. and M. Rooth: 1998, 'Valence Induction with a Head-Lexicalized PCFG'. In: *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung, no. 4, vol. 3*. pp. 25–54.
- Chesley, P. and S. Salmon-Alt: 2006, 'Automatic extraction of subcategorization frames for French'. In: *Proceedings of the Language Resources and Evaluation Conference, LREC 2006, Genua, Italy*.
- Chi, Z. and S. Geman: 1998, 'Estimation of probabilistic context-free grammars'. *Computational Linguistics* **24**, 299–305.
- Colmerauer, A.: 1978, 'Metamorphosis grammar'. In: *Natural Language Communication with Computers*, Lecture Notes in Computer Science 63. New York: Springer, pp. 133–189.
- Dempster, A. P., N. M. Laird, and D. B. Rubin: 1977, 'Maximum Likelihood from Incomplete Data via the EM algorithm'. *Journal of the Royal Statistical Society, series B* **39**, 185–197.
- Dębowski, Ł. and M. Woliński: 2007, 'Argument co-occurrence matrix as a description of verb valence'. In: Z. Vetulani (ed.): *Proceedings of the 3rd Language & Technology Conference, October 5-7, 2007, Poznań, Poland*. pp. 260–264.
- Ersan, M. and E. Charniak: 1995, 'A statistical syntactic disambiguation program and what it learns'. In: S. Wermter, E. Riloff, and G. Scheler (eds.): *Learning for Natural Language Processing*. New York: Springer, pp. 146–159.
- Fast, J. and A. Przepiórkowski: 2005, 'Automatic Extraction of Polish Verb Subcategorization: An Evaluation of Common Statistics'. In: Z. Vetulani (ed.): *Proceedings of the 2nd Language & Technology Conference, Poznań, Poland, April 21-23, 2005*. pp. 191–195.
- Gorrell, G.: 1999, 'Acquiring Subcategorisation from Textual Corpora'. M. Phil. dissertation, University of Cambridge.
- Halford, G. S., W. H. Wilson, and W. Phillips: 1998, 'Processing capacity defined by relational complexity: Implications for comparative, developmental and cognitive psychology'. *Behavioral Brain Sciences* **21**(6), 803–864.
- Jelinek, F.: 1997, *Statistical Methods for Speech Recognition*. Cambridge, MA: The MIT Press.
- Korhonen, A.: 2002, 'Subcategorization Acquisition'. Ph. D. dissertation, University of Cambridge.

- Kupiec, J.: 1992, 'Robust part-of-speech tagging using a hidden Markov model'. *Computer Speech and Language* **6**, 225–242.
- Kurcz, I., A. Lewicki, J. Sambor, and J. Woronczak: 1990, *Słownik frekwencyjny polszczyzny współczesnej*. Kraków: Instytut Języka Polskiego PAN.
- Lapata, M. and C. Brew: 2004, 'Verb Class Disambiguation using Informative Priors'. *Computational Linguistics* **30**, 45–73.
- Levin, B.: 1993, *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago and London: The University of Chicago Press.
- Macleod, C., R. Grishman, and A. Meyers: 1994, 'Creating a Common Syntactic Dictionary of English'. In: *SNLR: International Workshop on Sharable Natural Language Resources, Nara, August, 1994*.
- Manning, C.: 1993, 'Automatic acquisition of a large subcategorization dictionary from corpora'. In: *Proceedings of the 31st Annual Meeting of the ACL, Columbus, Ohio*. pp. 235–242.
- Mayol, L., G. Boleda, and T. Badia: 2005, 'Automatic acquisition of syntactic verb classes with basic resources'. *Language Resources and Evaluation* **39**, 295–312.
- McCarthy, D.: 2001, 'Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences'. Ph.D. thesis, University of Sussex.
- Merialdo, B.: 1994, 'Tagging English text with a probabilistic model'. *Computational Linguistics* **20**, 155–171.
- Młynarczyk, A. K.: 2004, 'Aspectual Pairing in Polish'. Ph.D. thesis, Universiteit Utrecht.
- Neal, R. and G. Hinton: 1999, 'A view of the EM algorithm that justifies incremental, sparse, and other variants'. In: M. I. Jordan (ed.): *Learning in Graphical Models*. Cambridge, MA: The MIT Press, pp. 355–368.
- Polański, K. (ed.): 1980–1992, *Słownik syntaktyczno-generatywny czasowników polskich*. Wrocław: Zakład Narodowy im. Ossolińskich / Kraków: Instytut Języka Polskiego PAN.
- Przepiórkowski, A.: 2006, 'What to acquire from corpora in automatic valence acquisition'. In: V. Koseska-Toszewa and R. Roszko (eds.): *Semantyka a konfrontacja językowa (3)*. Warszawa: Slawistyczny Ośrodek Wydawniczy PAN.
- Przepiórkowski, A. and J. Fast: 2005, 'Baseline Experiments in the Extraction of Polish Valence Frames'. In: M. A. Kłopotek, S. T. Wierchoń, and K. Trojanowski (eds.): *Intelligent Information Processing and Web Mining*. New York: Springer, pp. 511–520.
- Przepiórkowski, A. and M. Woliński: 2003, 'A Flexemic Tagset for Polish'. In: *Proceedings of Morphological Processing of Slavic Languages, EACL 2003*. pp. 33–40.
- Rudin, W.: 1974, *Real and complex analysis*. New York: McGraw-Hill.
- Sarkar, A. and D. Zeman: 2000, 'Automatic Extraction of Subcategorization Frames for Czech'. In: *Proceedings of the 18th International Conference on Computational Linguistics, COLING 2000, Saarbrücken, Germany*. pp. 691–698.
- Schulte im Walde, S.: 2006, 'Experiments on the Automatic Induction of German Semantic Verb Classes'. *Computational Linguistics* **32**, 159–194.
- Surdeanu, M., R. Morante, and L. Màrquez: 2008, 'Analysis of Joint Inference Strategies for the Semantic Role Labeling of Spanish and Catalan'. In: *Proceedings of the Computational Linguistics and Intelligent Text Processing 9th International Conference, CICLing 2008*. pp. 206–218.
- Świdziński, M.: 1992, *Gramatyka formalna języka polskiego*. Warszawa: Wydawnictwa Uniwersytetu Warszawskiego.
- Świdziński, M.: 1994, 'Syntactic Dictionary of Polish Verbs'. Warszawa: Uniwersytet Warszawski / Amsterdam: Universiteit van Amsterdam.
- Tokarski, J.: 1993, *Schematyczny indeks a tergo polskich form wyrazowych*. Warszawa: Wydawnictwo Naukowe PWN.
- Vapnik, V. N.: 1995, *The Nature of Statistical Learning Theory*. New York: Springer.
- Woliński, M.: 2004, 'Komputerowa weryfikacja gramatyki Świdzińskiego'. Ph.D. thesis, Instytut Podstaw Informatyki PAN, Warszawa.
- Woliński, M.: 2005, 'An efficient implementation of a large grammar of Polish'. *Archives of Control Sciences* **15(LI)**, **3**, 251–258.
- Woliński, M.: 2006, 'Morfeusz—a Practical Tool for the Morphological Analysis of Polish'. In: M. A. Kłopotek, S. T. Wierchoń, and K. Trojanowski (eds.): *Intelligent Information Processing and Web Mining*. New York: Springer, pp. 503–512.