Proceedings of the International Multiconference on Computer Science and Information Technology pp. 181–188

ISBN 978-83-60810-14-9
ISSN 1896-7094

Automatic Acquisition of Wordnet Relations by the Morpho-Syntactic Patterns Extracted from the Corpora in Polish

Roman Kurc, Maciej Piasecki
Institute of Applied Informatics, Wrocław University of Technology, Poland
Email: maciej.piasecki@pwr.wroc.pl

***Abstract*—In the paper we present an adaptation of the Espresso algorithm of the extraction of lexical semantic relation to specific requirements of Polish. The introduced changes are of more technical character like the adaptation to the existing Polish language tools, but also we investigate the structure of the patterns that takes into account specific features of Polish as an inflectional language. A new method of the reliability measure computation is proposed. The modified version of the algorithm called Estratto was compared with the more direct reimplementation of Espresso on several corpora of Polish. We tested the influence of different algorithm parameters and different corpora on the received results.**

I. Introduction

STARTING construction of a system from scratch gives much more control over it, but in the case of large, practical systems it usually means that it will never be completed. In current-day software engineering, the component-based architecture and re-usable components became a typical way of construction. In the contemporary Computational Linguistics and Natural Language Engineering a similar role is played by basic language resources and tools. There are attempts to define a basic set of them, e.g. [1][2], or to build architectures supporting their application, e.g. (Clarin). Wordnets[1] built for different languages became commonly applied as the source of linguistic knowledge. The main problem of the basic language resources is that they do not exist for many languages and their construction takes a lot of time and is costly. The construction of the first Polish Wordnet, called *plWordNet* (Polish name: *Słowosieć*) started in the year 2005, and its current version includes 14 677 lexical units (henceforth LUs)—one word or multiword lexemes [4]. In wordnet, LUs which are near synonyms are grouped into synsets, sets of near synonyms, and synsets are linked by lexical relations of several types. In plWordNet, the synset relations can be mapped onto the level of LU relations. One of the most important relations for any wordnet is hypernymy—simplifying, an LU a is the hypernym of the b if b is a kind of a on the basis of their lexical meanings.

The present size of plWordNet is too small for many applications. It could not be larger, since its manual construction was quite expensive. However, knowing that, from the very beginning, we assumed the manual work would be supported by the developed tools for the automatic extraction of instances of lexical semantic relations from corpora. We paid special attention to hypernymy due to its importance, i.e. our aim is to construct a tool acquiring pairs of: hypernym and hyponym from large corpora.

There are two possible paradigms [5]: *pattern-based* and *clustering based* also called *distributional*. The latter results in good recall but there are problems with precision, as its typical product is a *measure of semantic relatedness*, not some lexical semantic relationship itself. For any pair of LUs their level of relatedness can be obtained, but it is very unclear how to perform the identification of an type of relation. In the area of the clustering based paradigm, several works were done for Polish, e.g. [6], [7]. However, the number of works done in the area of pattern-based paradigm is very small, e.g. (Dernowicz, 2007), (Ceglarek & Rutkowski, 2006) the latter one dealing with the machine readable dictionaries, not corpora.

Pattern-based approaches are claimed to express good precision, but very small recall in the case of patterns constructed manually, e.g. [8]. The recall of patterns can be increased by using many or more generic patterns extracted automatically from a corpus, i.e. patterns which have broad coverage but intrinsically low precision. The system Espresso presented by Pantel and Pennacchiotti [5] is so successful an example of such an approach, that it inspired us to adapting this type of approach to Polish.

Our goal was to develop a statistical method of the extraction of lexico-morphosyntactic patterns for the needs of automatic hyperonymy acquisition. Our starting point was the adaptation of the Espresso algorithm to the Polish language, and even more important, to a very limited set of language tools for Polish. In the paper we present the measures introduced in Espresso, elements that should be taken into account when Polish patterns are being extracted and application of Espresso to extracting hypernymy. We also discuss the possibility of acquiring other types of relations. On the basis of the collected experience, an extended version of Espresso is proposed called *Estratto*.

II. Espresso

Espresso is thought to solve the *bootstrapping* problem [9], [10] i.e. learning the structure of the domain, where

[1] A wordnet is an electronic thesaurus of the structure following the main lines of the Princeton WordNet [3]

> “the phenomena and the rules defined in terms of those categories are learned from scratch [...]”

and

> “the specification of a set of rules presupposes a set of categories, but the validity of a set of categories can only be assessed in the light of the utility of the set of rules that they support.”

So both rules and categories must be derived together. In Espresso rules are patterns and categories are semantic relations represented by *instances* defined as pairs of LUs. The algorithm consist of three phases:

- *construction* of patterns on the basis of *instances* of a relation,
- pattern statistical *evaluation*,
- and *extraction* of instances on the basis of positively evaluated patterns.

Pantel and Pennacchiotti claim [5] that Espresso is characterized by:

- high recall together with a small decrease in precision of extracted instances,
- autonomy of work (weakly supervised algorithm)—only several initial instances of the given relation must be defined at the beginning,
- independence from the size of the used corpus or a domain,
- wide range of relation types that can be extracted.

The small decrease in accuracy results from the application of generic patterns together with specific ones. This balance is achieved by the proposed measure evaluating *reliability* of patterns and instances, explained in Section *Reliability and confidence measures*. *Confidence* of instances extracted by the generic patterns is verified on a large separated corpus. The confidence of an instance originates from its strength of association with reliable patterns and the number of reliable patterns which extract it. Only the best patterns and instances are kept for the following phase of the algorithm

The introduced measure of reliability and confidence reduce the need for manual supervision once Espresso started. The measures are a means of creating rankings of instances and patterns defining the degree to which they express the target relation.

As the system of measure, instances and pattern selection are universal and do not refer to any properties of any particular relation being extracted, Espresso can be applied to a wide range of relation, and was to several, e.g. hypernymy, meronymy, antonymy but also more specific like person—company or person—job title [5].

The characteristics of Espresso inspired us to try to adapt it to Polish. We are going to investigate the key of the algorithm like measures of reliability and confidence, methods of pattern extraction and the usage of verifying corpus for the evaluation of confidence in relation to the characteristic features of Polish.

III. Reliability and confidence measures

The reliability measure that is applied to construct the ranking of patterns and instances is one of the most important elements of the algorithm and is defined for patterns in the following way:

$$r_\pi = \frac{\sum_{i \in I} (\frac{pmi(i,p)}{max_{pmi}} * r_t(i))}{|I|} \tag{1}$$

where p is a pattern, i—an instance, r_t—measure of reliability for instances, pmi—Pointwise Mutual Information, explained below, and $|I|$—the size of the set of instances.

The reliability of instances is defined in a very similar way, but this time the reliability of patterns is utilized in the equation.

PMI measure originates from the Theory of Information and is defined as following:

$$pmi(i,p) = \log \frac{|x,p,y||*,*,*|}{|x,*,y||*,p,*|} \tag{2}$$

where $|x,p,y|$ is the number of occurrences of x and y in contexts matching the pattern p, $x,*,y$—the number of co-occurrences of x and y in the corpus regardless the pattern, etc.

The pmi definition given in [5] does not include the constituent: $|*,*,*|$, i.e. the number of contexts. However, the PMI measure should be usually greater than 0, while the one defined in 2 is not. Moreover, the missing constituent is suggested also by the general definition of PMI:

$$pmi(i,p) = \log \frac{p(I,P)}{p(I)p(P)} \tag{3}$$

Because PMI is significantly greater in the situation in which instances and patterns are not numerous (e.g. the size smaller than 10), PMI is multiplied by a factor proposed in [11].

The measure of confidence of an instance extracted by the generic patterns is based on the application of specific patterns of high reliability to a different validating corpus and is calculated in the following way:

$$S(i) = \sum_{p \in P_R} S_P(i) * \frac{r_\pi(p)}{T} \tag{4}$$

P_R is the set of specific patterns, $S_p = pmi(i,p)$ and T is the sum over the reliability of specific patterns.

It is worth to emphasize that patterns are evaluated not on the basis of instances which were extracted by them, but on the basis of instances that were used to acquire these patterns. Instances are evaluated in a similar way. This is a consequence of the method assumed in Espresso: patterns are not matched to the instances but are induced by the instances.

The intuition behind the measures of reliability and confidence is that patterns which well describe the given relation frequently occur with a large number of confident instances of this relation. The same applies in the opposite way. However, in the case of confidence the difference is that instances extracted by generic patterns will obtain high confidence, if they occur in contexts matched by the specific patterns of good reliability in the validating corpus.

There are two unclear issues in the picture presented above. Firstly, even making a draft calculation, we can check, that reliability is sensitive to possible fluctuations in PMI value. Occurrence of higher PMI values (e.g. originating from small frequencies, even after correction by the discounting factor dependent on the number of occurrences) can cause lower assessment of patterns with balanced ratio of co-occurrence with matched instances in relation to the pattern occurrences and occurrences of instances alone. Such a situation results in the artificially increased value of max_{pmi}. Thus, we would like to look for a measure which would be more insensitive to the problem of the low frequency of pattern matches or instances matched. Secondly, the reliability measure of the best instance or pattern may take the value lower than one even in a situation in which there is a complete match of all patterns and all instances (or the other way round, depending on which we are calculating the reliability). That is why, the reliability propagation to the subsequent phases causes, that new values calculated for patterns on the basis of instances, and vice versa, will be gradually lower according to the size of the set for which the reliability is computed, i.e. patterns or instances. Thus we want to introduce a new measure of reliability, which returns one as the value for the best patterns or instances in every phase.

$$r_\pi(p) = \frac{\sum_{i\in I}(pmi(i,p) * r_t(i)) * d(I,p)}{max_P(\sum_{i\in I}(pmi(i,p) * r_t(i))) * |I|} \qquad (5)$$

where $d(i,p)$ defines how many unique instances the given pattern is associated with.

PMI in the formula (5) is usually modified by the discounting factor, as well.

IV. Patterns

As for the Machine Learning methods the choice of features for objects is a very important decision, so is the choice of pattern structure and pattern language for the pattern-based approaches in acquisition of lexical semantics. The majority of approaches take the scheme proposed by Hearst [12] as their reference point, i.e. patterns being a subset of regular expressions, in which the alphabet includes lemmatized word forms and a set of variables for noun phrases matched as an element of a relation instance. An example for hipernymy could be: NP such as NP1*
NP is a/an NP1

We assumed that patterns are a subset of regular expressions with Kleene closure, but without grouping. The alphabet includes morphological base forms of lexical units. Before presenting the scheme of a pattern for Polish, we need to investigate the characteristic features of Polish, which we considered.

A. Selected aspects of Polish

The vast majority of pattern-based approaches were developed for English. Those approaches base in some extent on the positional, linear syntactic structure of English. It was not clear how one can successfully transform this approach to a language of a significantly diffrent type like Polish.

The basic, unmarked order of a Polish sentence is Subject—Verb—Object, i.e. similarly to English. So, for simple lexico-syntactic patterns based on relative positions of described elements the differences should not be large. However, on the other side, in order to fully explore the potential of pattern-based approach we have to go beyond the analysis of the most simple constructions only. One needs to take into account such phenomena like morphosyntactic agreement of different kinds among word forms and the relaxed order of a sentence, according to which one can use several different orders of a sentence which only slightly changes in meaning. It seems to be reasonable to put more emphasis on the morphological description of pattern elements in terms of scheme introduced in the IPI PAN Corpus of Polish (IPIC) [2]: grammatical class (extended, more fine grained division than Part of Speech) and values of grammatical categories like case, number and gender for nouns and adjectives or aspect and number for verbs. In the case of Polish, the linear positions of LUs in a sentence is not necessarily correlated with their role in a lexical semantic relation when the relation is not symmetrical, e.g. in case of hypernymy most patterns mark hypernym and hyponym by different cases, while their relative positions are changing. Obviously, we can generate many specific patterns for all different combinations, but we can also look for some generalization of a group of patterns on the basis of the morphosyntactic properties.

B. Scheme of patterns

Patterns have a flat structure and describe a sentence as a sequence of word forms or at most groups of word forms. Patterns are not based on any deeper description of the syntactic structure. The alphabet comprises three types of symbols: an empty symbol `*`, *base form* and *matching place*. The empty symbol represents any LU (represented by any of its word form). The base form is a morphological base form of some LU together with the grammatical class, as the same morphological base form can represent more than one LU. A matching place represents all LUs whose morphosyntactic description matches the partial description encoded in the matching place symbol. As grammatical classes of IPIC are too fine grained we introduced a macro collective symbol, e.g. `noun` joining together: *substantives*, *gerunds*, *foreign nominals* and *depreciative nouns*. A matching place is a reduced version of the IPIC morphosyntactic tag, in which only some grammatical categories are specified.

Following [5], there are always two matching places: one at the beginning and one at the end of a pattern. Patterns do not describe the left and right context of a potential instance.

A pattern also encodes the roles of both LUs identified by matching places, e.g.:

`(hypo:subst:nom) jest (hyper:subst:inst)`

—where `jest` is *to be*$_{number=sg,person=3rd}$, `hipo` marks hyponym, and `hiper`—hypernym, `subst`—*substantive*, `nom` and `inst` are case values (all three are from the IPIC descrption)

`(hyper:subst:inst) jest (hypo:subst:nom)`

The described pattern scheme expresses to some extent the characteristic features of Polish. The change of the pattern scheme was the first step leading to an algorithm called Estratto, which is a modification of Espresso that is better suited for an inflectional language like Polish.

V. Induction of patterns and extraction of instances

According to [5], patterns can be inferred by any pattern learning algorithm. In Estratto the generalisation and unification of patterns is based on the longest common substring algorithm. The algorithm is guided by a predefined list of relation specific LUs, e.g. for hypernymy, *być* (*to be*), *stać się* (*to become*), *taki* (*such*), *inny* (*other*), etc.

In Espresso, the inferred patterns are then generalized by replacing all *terminological expressions* (i.e. a subset of noun phrases) by *terminological labels*. Such an approach to generalization is not applicable for Polish, as a required *chunking parser* (chunker) does not exist. Therefore a slightly different method was proposed. Patterns are grouped and then merged with respect to the significant elements of the patterns: specification of matching places (determining properties of morphological similarity to contexts), and words expected to be related in some way to the semantic relation being extracted.

The instance extraction phase comes after patterns induction and selection. An instance is a pair $\langle x, y\rangle$ of LUs belonging to the set of instances representing the target semantic relation. Authors of Espresso suggest, that if the algorithm is applied to a small corpus, two methods can be used to enrich the instance set. First each multiword LU in an instance can be simplified according to the head of LU. For example *new record of a criminal conviction* is simplified to *new record* and this to *record*. A new instance is created with a simplified LU and the LU that was in the pair together with the original LU. Second an expansion is made by an instantiating pattern only with one of the LUs: x or y, and searching if it can extract a new instance from additional corpora, for example, for the instance (*dog*, *animal*) and the pattern expressed in the inflectional format used in Estratto:

`(hypo:subst:nom) is a/an (hyper:subst:inst)`

two queries:

`dog is a/an (hyper:subst:inst)`

and

`(hypo:subst:nom) is a/an animal`

are created.

Instances gathered using both of those methods are added to the instance set. However, it worth of noticing that in all experiments described in [5] only one-word LUs are used and the applied corpora are claimed to be large enough to provide statistical evidence.

Generalized patterns, described above, are not classified as *generic* as long as they do not generate ten times more instances than the average number of instances extracted by specific patterns. However high recall results in loosing some of the precision, that is why every instance extracted by a generic pattern is verified. The verification process starts with instantiating all specific patters with the instance in question. Then the instantiated patterns are queried in a validating corpus and the confidence measure is next computed on the basis of collected frequencies. If confidence is above the defined threshold than the tested instance is considered as representing the target relation.

In Espresso, Internet resources were used as a huge validating corpus for instances extracted by generic patterns. Contrary to this, due to several limitations in searching the Internet in Polish (i.g. limited access to the search engine and inflection of the language), we applied a second large corpus, much smaller the aforementioned one, as a validating corpus in Estratto. The necessary condition is that the validating corpus must be similar in its characteristics to the basic one.

The process of the induction of patterns and extraction of instances is controlled by the following set of parameters:

1) the *number of top* k patterns not to be discarded (preserved for the next iterations),
2) the *threshold* for measure of confidence for instances,
3) the *minimum* and *maximum frequency* values for patterns,
4) the *minimum size* of a pattern— all patterns that consist of only matching places and conjunctions are discarded by the assumption,
5) a *filter* on common words in instances and instances that have identical LUs on both positions,
6) the *size* of the validating corpus.

VI. Performance measures

A proper evaluation of the extracted lexical semantic resources is mostly a serious problem, e.g. [13], [14]. However, in the case of lists of instances the situation is simpler: we need to verify how many of them are correct. There are only two possibilities to compare the list with: an existing manually constructed resource, i.e. plWordNet in our case or human judgement. The former will introduce some bias as plWordNet is limited in its size, but gives a possibility of testing the whole set of instances, while the manual evaluation is always laborious.

In both types of comparison we applied the standard measures of *precision* and *recall*, e.g. [15]. The F-measure could not be applied, because of the limitations of recall based on plWordnet, which are discussed later.

Precision is defined in a standard way: $P = \frac{tp}{tp+fp}$ where tp is the number of true positives, i.e. extracted pairs of LUs which are instances of the target relation, fp—false positives.

True positives are patterns or instances (depending on what we are going to measure) that are correct and marked by algorithms as correct and false positives are those that are incorrect but marked by algorithms as correct.

Recall is defined in a standard way too: $R = \frac{tp}{tp+fn}$

However, it is worthy of note that recall is the ratio between instances or patterns that are correctly marked as correct (true positives) and the sum of true positives and all those that are correct but were either marked as incorrect or not extracted at all. The problem is that we certainly cannot treat the limited plWordNet as the exhaustive description of the subsequent relations. Thus, recall in our approach is only the measure of the ratio of rediscovery the plWordNet structure, it is not a recall in relation to all instances or patterns that can be present in the used corpora.

We extracted a ranked list of possible instances which can be sorted in descending order of their reliability. Its values are real numbers and there is no characteristic point below which we can cut off the rest of pairs according to some analytical properties. Thus, instead of pure precision and recall, we prefer to use *cut off precision* and *cut off recall* calculated only in relation to some n first positions on the sorted list of results (instances or patterns).

Finally, we used the following evaluation measures:

1) *Cut off precision based on plWordNet* - this measure marks as correct only those instances and patterns that were found both in plWordNet and an additional list provided a priori by human judge. It is worth to consider that the limited size of plWordNet can influence precision negatively because some LUs are not present yet or although included into plWordNet, still not connected. This precision is computed for each element on the list of instances.
2) *Precision based on human judgement* is evaluated according to a randomly drawn sample from the list of instances. This evaluation measure was used only for the first group of experiments (ref. Experiments). The error level of sample was 3% and the confidence level was 95%.
3) *Recall based on plWordNet* is evaluated at the set of word pairs generated form plWordNet. However this measure does not describe the recall from corpora

VII. Experimental setup

The experiments were performed on three datasets corpora:

a) IPIC [2] including about 254 millions of tokens, is not balanced but contains texts of different genres: literature, poetry, newspapers, legal texts and stenographic records from parliament, and scientific texts,
b) 100 millions tokens from Rzeczpospolita [16]—Polish newspaper (henceforth RC)
c) and a corpus of large text documents collected from the Internet, texts including larger numbers of spelling errors and duplicates were semi-automatically filtered out (LC), LC includes about 220 millions of tokens.

We tested several configurations of systems during the experiments, namely:

a) **ESP-** —Espresso without generic patterns,

TABLE I
Influence of the extended reliability measure and changes in the form of patterns

	Precision levels (plWN)	Hum. eval	Recall plWN	**Instances**
ESP-	36%/50%/75%	39%	27%	3982/903/14
ESP-nm	37%/50%/75%	47%	26%	3784/774/96
ESP+	32%/50%/75%		27%	4221/613/11
EST-	52%/52%/75%	54%	18%	1628/1628/169
EST-nm	16%/50%/75%	59%	18%	1775/1598/120

b) **ESP-nm**—Espresso without generic patterns, but with the extended reliability measure 5,
c) **ESP+** Espresso with generic patterns,
d) **EST-** Estratto without generic patterns, exploiting specific features of Polish,
e) **EST-nm** Estratto without generic patterns, exploiting specific features of Polish language and the extended reliability measures 5,
f) **EST+nm** same as VII but using generic patterns.

If not stated otherwise the threshold for confidence is 1.0 for all ESP systems and 2.6 for EST. The *number of top* k patterns was set to $k = 2 + I$, where I is the number of the present iteration. The number of iterations was set to four. In those experiments whose results are presented we focused only on the hypo/hypernymy relation and we selected as the main corpus, on which we performed experiments compared in tables, was IPIC.

VIII. Experiments

Research on Espresso and Estratto can be divided into three groups. The first one includes experiments, that were designed to analyse the influence of the proposed extended reliability measure 5 and of the form (i.e. if they are improved for using selected aspects of Polish or no) of patterns for the **ESP-**, **ESP-mn** and **EST-**, **EST-nm**. The results are shown in Table I, where values in the column labelled "Precision level" refer to the number of instances in the last column e.g. in the first row **ESP-** extracted 3982 instances with precision of 36%, 903 instances with precision 50% and so on. The column labelled “Hum. eval” refers to the evaluation of the results made by one of the authors.

On the basis of the results of the first group of experiments, Table I, one can conclude, that the use of the original reliability measure 1 results in extraction of more instances. The overall cut-off precision based on plWordNet for 4000 of instances is around 35%. On the other hand, the cut-off precision evaluated for EST suggested, that EST performs worse. However plWN is relatively small, that is why, the evalutation can be misleading. Therefore one of the authors performed manual evaluation. This additional evaluation showed, that in fact the plWN might be used only for a very rough estimation of the precision. The results of the manual evaluation suggest also, that the use of the new measure increases the precision of **ESP-/EST-**. In each case ESP-nm vs. **EST-nm** is better. During experiments it was also observed, that the value of original

TABLE II
DEPENDENCY OF THE ALGORITHMS ON THE VALUES OF PARAMETERS.

	Precision levels (plWN)	Recall plWN	Instances
EST-nm:th1.0			
EST-nm:th2.6	16%/50%/75%	18%	1775/1598/120
EST-nm:th5.2	**47%/50%/75%**	**20%**	**1907/1736/117**
EST+nm:patt4iter4	27%/50%/75%	27%	4372/1537/86
EST+nm:patt8iter4	32%/50%/75%	24%	3999/1521/47
EST+nm:patt4iter6	17%/50%/75%	29%	7265/1505/83
EST+nm:patt8iter6	30%/50%/75%	27%	4210/1485/58
EST+nm:PMI	**27%/50%/75%**	**27%**	**4187/1505/86**
EST+nm:Tscore	6%/-	25%	8934/-/-
EST+nm:Zscore	34%/50%/75%	26%	3563/1419/59

reliability (1) decreases very fast and after 6th iteration it is far below 10–12. This is a reason for the drop of newly extracted instances. Applying the extended reliability (5) allows to avoid that problem. Recall based on plWN is comparable, and depends on the number of extracted instances. Another matter of concern is the scheme of the patterns adjusted for Polish. It is clear that the application of the adjusted patterns produces better precision **EST-** and **EST-nm** in comparison to **ESP-** and **ESP-nm**. However the recall is decreased.

Experiments from the second group were performed only for **EST+nm** and **EST-nm**, using suggested measure, and this group was aimed at determining the influence of the algorithm parameters on the result. The following dependencies were investigated:

i) influence of the confidence threshold on the precision of instances achieved within subsequent iterations,

ii) influence of the number of the top k patterns on the stability of the algorithm and the precision of instances,

iii) dependency on the filtering infrequent and very frequent patterns and instances.

iv) influence of the number of initial instances (seeds) on the induced patterns, and then the influence of the ration between instances and patterns inducted by them,

v) a way in which different statistical similarity measures used in reliability calculation change the precision of the results.

In the case of **i)** it seems, that best results are achieved, when the threshold is higher, see Table II. However one must keep a balanced ratio between chosen instances and new patterns. If there is a small number of instances, there is no statistical evidence to induce proper patterns and EST/ESP crawls picking almost random patterns. That leads to the decrease in precision.

Considering **ii)**, on the basis of the obtained numbers, it can be noticed, that using a smaller number of the k-top patterns results in higher precision. This is due to the stability of a model, in which semantic relations are generated by a small group of elite patterns. An interesting idea would be to use a dynamic k-top factor. Should it be more strict at the begging, the more stable set of patterns would be indicated. Then in the subsequent iterations the k-top would grow faster, and as a result more correct patterns could extract instances from corpus in the next phase.

The data for **iii)** are not presented in Table II. However the experiments have shown, that infrequent patterns (occurring less than four times) should be filtered before generalization, because they introduce additional noise, which causes good patterns to be evaluated as worse.

Initial seeds, the point **iv)**, are meant to generate a skeleton of a model of the lexical semantic relation. If the number of seeds is not enough high, the best extracted patterns can be random. Of course, one could collect a small number of seeds, that would indicate only expected patterns. However that would require a precise analysis of the corpus, that would be used for instance extraction. That is pointless, because using more seeds one can acquire the same patterns with less effort.

In the case of **v)**, the data shows, that PMI is better than Z-score and T-score as the measure of similarity in the extraction of lexical semantic relations. T-score results are especially disappointing, and that might be due to the fact of the insufficient statistical evidence (the algorithm very often accepted instances occurring only once).

The third and the last group of experiments was prepared to check the ability of EST and ESP to use a different corpus and extract other relations than hypo/hypernymy. Performed experiments showed that both algorithms: EST and ESP can be applied to different corpora successfully, however it seems, that each time the corpus is changed, a new confidence threshold must be discovered by some method For IPIC the threshold value was 2.6 but in the case of RC we found 0.9 as working fine. Tests performed on the LC corpus appeared to be unsuccessful. But this is a rather special case, as most of the text in LC are written in literary style, so the language expressions are more complex. Moreover, one should expect less defining sentences than in utility texts. It seems that this kind of corpus requires more powerful patterns to catch some syntactic dependencies. The other problem, namely the application of EST to different relation types appeared to be only partially successful. Tests on meronymy ended with a rather poor result, i.e. the estimated precision was lower than 30%. There are at least three main reasons for this failure. Firstly, the expressive power of patterns is too low and some important morpho-syntactic dependencies are missed. Secondly, meronymy is indeed a set of quite varied sub-relations. That is why, it could be reasonable to try to extract each sub-relation separately. Thirdly, the trials were done only on one corpus. On the other hand, initial experiments on extracting antonymy (but only for adjectives) gave promising results. The human-judged cut-off precision reached 39%. Both meronymy and antonymy will be further investigated.

IX. EXAMPLES

Below we present examples of instances (hyponym; hypernym) extracted by the **ESP-** algorithm from IPIC:

szkoła(*school*); *instytucja*(*institution*)
maszyna(*machine*); *urządzenie*(*mechanism*)
wychowawca(*tutor*); *pracownik*(*employee*)
kombatant(*combatant*); *osoba*(*person*)
bank(*bank*); *instytucja*(*institution*)
pociąg(*train*); *pojazd*(*vehicle*)
telewizja(*television*); *medium*(*medium*)
prasa(*press*); *medium*(*mass media*)
szpital(*hospital*); *placówka*(*establishment*)
czynsz(*rent*); *opłata*(*payment*)
grunt(*land*); *nieruchomość*(*real estate*)
Wisła(*Wisła*); *rzeka*(*river*)
świadectwo(diploma); *dokument*(*document*) *opłata*(*payment*); *należność*(*charge*)
ryba(*fish*); *zwierzę*(*animal*)
Włochy(*Italy*); *kraj*(*country*)
jezioro(*lake*); *zbiornik*(*reservoir*)
jarmark(*fair*); *impreza* (*entertainment*)
piwo(*beer*); *artykuł*(*comestible*)
zasiłek(*dole*); *świadczenie*(*welfare, benefit*)
powódź(*flood*); *klęska*(*disaster*)
paszport(*passport*); *dokument*(*document*)

Examples of patterns extracted by **ESP-** from IPIC and used in the extraction of the above instances are presented below:

```
occ=31 rel=0.26803 (hypo:subst:nom) być
(hyper:subst:inst)
(hypo:subst:nom) is/are (hyper:subst:inst)

occ=20 rel=0.222222 (hypo:subst:nom) i
inny (hyper:subst:nom)
(hypo:subst:nom) and other
(hyper:subst:nom)

occ=26 rel=0.103449 (hypo:subst:inst) a
inny (hyper:base:inst)
(hypo:subst:inst) but other
(hyper:base:inst)

occ=15 rel=0.0684905 (hypo:subst:inst)
przypominać (hyper:subst:acc)
(hypo:subst:inst) resemble
(hyper:subst:acc)

occ=41 rel=0.0263854 (hypo:subst:loc) i
w inny (hper:subst:loc)
(hypo:subst:loc) and in other
(hper:subst:loc)

occ=86 rel=0.00708506 (hypo:subst:nom)
stać się (hyper:subst:inst)
(hypo:subst:nom) become (hyper:subst:inst)

occ=88 rel=0.0060688 (hypo:subst:acc)
interp który być (hyper:subst:inst)
(hypo:subst:acc) interp which is
(hyper:subst:inst)
```

X. Conclusions and further work

In the paper we presented a partially successful application of the *Espresso* algorithm [5] to Polish. The modified version of the algorithm was called *Estratto*. Experiments showed that the reliability measure proposed by Pantel and Pennacchoti [5] works usually well as a ranking measure for the extraction of lexical semantic relations. However the plWN-based precision of the Espresso/Estratto algorithm is lower when measured on Polish corpora than the precision reported in [5]. This might be due to a slightly different approach to precision evaluation, which was performed partially on the basis of plWordNet (of a limited size) and combined next with a limited manual evaluation. On the other hand the results of the manual evaluation are similar to the results reported in [5]. Results obtained for different measures of similarity as the basis of the reliability suggest that PMI gives the best results for the given test suit.

The adjustment of the pattern structure to the characteristic features of Polish improved the precision in comparison to patterns using only word forms and Parts of Speech as features.

The extended version of Espresso—namely Estratto showed to be successful in extracting hypernyny and antonymy from the IPI PAN Corpus [2] and the Rzeczpospolita corpus [16]. Unfortunately attempts to extract meronymy did not bring positive results.

During experiments we tested several parameters that have a significant influence on the algorithm. The most important of them appeared to be: the *number of seed instances*, the *confidence threshold* and the *number of the k-top patterns* preserved between the subsequent iterations. The number of seed instances should be more than 10. The confidence threshold depends strongly on a corpus, e.g. for IPIC the best found value was about 0.3. Each time the algorithm is applied to a new corpora both seed instances and the measure of confidence must be reset. The number of the k-top patterns should be low i.e. about two. Such a number results in a stable representation of the semantic relation, i.e. by the means of the set of patterns. However, it is still unclear, how to explore patterns, that seem to be correct and are close to the top. Those patterns usually disappear in next iterations and that means that some instances are also excluded from final results.

Espresso/Estratto is an intrinsically weakly supervised algorithm, although the preparation of seeds and setting the initial values of parameters might require even some initial runs of Espresso/Estratto or browsing the corpus.

Additionally it turned out, that in order to maintain a stable representation of relations, the appropriate ratio between patterns and instances must be kept. The ratio was estimated during experiments and equals for patterns vs. instances: 1:15/20. If there are less instances, the algorithm becomes unstable. Using more instances results in a longer time of computation.

An interesting result of experiments is the observation of the “intensifying” patterns. Such patterns do not represent any particular semantic relation and when applied alone they extract instances belonging to relations of multiple types. However when the intensifying patterns are combined with regular ones they deliver additional statistical evidence to correct but infrequent instances and as a result rise the precision of the algorithm, e.g.,

`(hypo/holo:subst:nom) w (hyper/mero:subst:inst)`

where *w* means *in*.

We observed a problem with the number of instances collected by the **ESP+**/**EST+** versions of the algorithms. This number is comparable to the number of instances extracted by **ESP-**/**EST-** while one would expect it to be much higher. This might be a result of the characteristic features of the corpus, namely IPIC, used in the experiments or of the size of the validating corpus. This problem might be partially solved by the use of Google as a validating corpus. Unfortunately, in contrast to English, Polish LUs have multiple word forms. As a result queries issued to Google will have to be more complicated. The other reason might be the limited expressive power of patterns. The expressive power of the patterns is the element of the algorithm that should be investigated. The extended structure of patterns still seems to miss some lexico-semantic dependencies, especially in stylistic reach text. The experiments on extracting hypernymy from the corpus LC, mostly consisting of text in literary style, was unsuccessful. The first step towards strengthening patterns is to take into account possible agreements in elements of the patterns that match the instances. The patterns used in EST are very strict about grammatical categories e.g.,

`(hypo:subst:gen) i inny (hyper:subst:gen)`

(two nouns in the genitive case) is treated as a completely different pattern from:

`(hypo:subst:inst) i inny (hyper:subst:inst)`

which matches two nouns in the instrumentative case. It seems to be helpful to allow merging of such patterns into e.g., the form of:

`(hypo:subst:case1) i inny (hyper:subst:case2)`

where `case1 = case2`. On the basis of the results for **ESP-** and **EST-**, where in **ESP-** there are no such strict constraints, one can expect the increase in recall. The other way, much more complicated, is to enrich the pattern representation, so that additional syntactic information (at least about nominal phrases) could be used.

Natural extension of present representation of instances for Polish is the introduction of multiword lexical units(LUs). Due to their present form it is sometimes possible to obtain instances consisting of two similar words, e.g., for *word office* and *post office* the resulting instance would be (office, office), as only single words are matched.

The list of acquired instances cannot be easily imported to plWordNet. This is due to the fact that the list is a flat structure. Such a representation cannot indicate, which wordnet classes an instance belongs to and what is the distance between the LU in the instance. This problem has been already addressed by Pantel and Pennacchiotti [5].

ACKNOWLEDGMENT

Work financed by the Polish Ministry of Education and Science, project No. 3 T11C 018 29.

REFERENCES

[1] V. Mapelli and K. Choukri, “Report on a (minimal) set of LRs to be made available for as many languages as possible, and map of the actual gaps,” ENABLER project, Internal report Deliverable 5.1, 2003. [Online]. Available: http://www.elda.org/blark/fichiers/report.doc

[2] A. Przepiórkowski, *The IPI PAN Corpus: Preliminary version*. Warsaw: Institute of Computer Science, Polish Academy of Sciences, 2004.

[3] C. Fellbaum, Ed., *WordNet—An Electronic Lexical Database*. The MIT Press, 1998.

[4] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, “Words, concepts and relations in the construction of Polish WordNet,” in *Proceedings of the Global WordNet Conference, Seged, Hungary January 22–25 2008*, A. Tanâcs, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177.

[5] P. Pantel and M. Pennacchiotti, “Espresso: Leveraging generic patterns for automatically harvesting semantic relations.” ACL, 2006, pp. 113–120. [Online]. Available: http://www.aclweb.org/anthology/P/P06/P06-1015

[6] M. Piasecki, S. Szpakowicz, and B. Broda, “Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns,” in *Proc. Text, Speech and Dialog 2007 Conference*, ser. LNAI, vol. 4629. Springer, 2007.

[7] M. P. Bartosz Broda, Magdalena Derwojedowa and S. Szpakowicz, “Corpus-based semantic relatedness for the construction of polish wordnet,” in *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[8] M. A. Hearst, *Automated Discovery of WordNet Relations*. The MIT Press, 1998.

[9] S. Pinker, *Language Learnability and Language Development*. Cambridge, MA: Harvard University Press, 1984.

[10] S. Finch and N. Chater, “Bootstrapping syntactic categories using statistical methods,” in *Background and Experiments in Machine Learning of Natural Language*, W. Daelemans and D. Powers, Eds. Tilburg University: Institute for Language Technology and AI, 1992, pp. 229–235.

[11] P. Pantel and D. Ravichandran, “Automatically labeling semantic classes,” in *HLT-NAACL 2004: Main Proceedings*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 – May 7 2004, pp. 321–328. [Online]. Available: http://acl.ldc.upenn.edu/N/N04/N04-1041.pdf

[12] M. A. Hearst, “Automatic acquisition of hyponyms from large text corpora.” in *Proceeedings of COLING-92*. Nantes, France: The Association for Computer Linguistics, 1992, pp. 539–545.

[13] T. Zesch and I. Gurevych, “Automatically creating datasets for measures of semantic relatedness,” in *Proceedings of the Workshop on Linguistic Distances*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 16–24. [Online]. Available: http://www.aclweb.org/anthology/W/W06/W06-1104

[14] M. Piasecki, S. Szpakowicz, and B. Broda, “Extended similarity test for the evaluation of semantic similarity functions,” in *Proceedings of the 3rd Language and Technology Conference, October 5–7, 2007, Poznań, Poland*, Z. Vetulani, Ed. Poznań: Wydawnictwo Poznańskie Sp. z o.o., 2007, pp. 104–108.

[15] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 2001.

[16] “Korpus rzeczpospolitej,” [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita.

[17] ACL 2006, Ed., *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. The Association for Computer Linguistics, 2006.