Proceedings of the Federated Conference on Computer Science and Information Systems pp. 533–540
DOI: 10.15439/2016F391
ACSIS, Vol. 8. ISSN 2300-5963

Grammatical Case Based IS-A Relation Extraction with Boosting for Polish

Paweł Łoziński, Dariusz Czerski, Mieczysław A. Kłopotek
Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
Email: {pawel.lozinski, dariusz.czerski, mieczyslaw.klopotek}@ipipan.waw.pl

***Abstract*—Pattern-based methods of IS-A relation extraction rely heavily on so called Hearst patterns. These are ways of expressing instance enumerations of a class in natural language. While these lexico-syntactic patterns prove quite useful, they may not capture all taxonomical relations expressed in text. Therefore in this paper we describe a novel method of IS-A relation extraction from patterns, which uses morpho-syntactical annotations along with grammatical case of noun phrases that constitute entities participating in IS-A relation. We also describe a method for increasing the number of extracted relations that we call *pseudo-subclass boosting* which has potential application in any pattern-based relation extraction method. Experiments were conducted on a corpus of about 0.5 billion web documents in Polish language.**

I. Introduction

RELATION extraction is a necessary step of any ontology induction or taxonomy induction task. Typically it takes as input morpho-syntactically annotated text and produces a set of triples (E_1, R, E_2), where E_1 and E_2 are entities and R is a relation in which E_1 and E_2 participate as a pair. In case of ontology induction or information extraction in open domain (as described, e.g., in [1], [2], [3], [4]) no restrictions are imposed on R. There are many types of relations that can be extracted this way, such as *quality*, *part* or *behavior* [5]. In case of taxonomy induction the main interest is in the IS-A (hyponym-hypernym) relation. Approaches to IS-A extraction described in literature rely on evidence from pattern extraction and statistical information (cf. [6], [7], [8]). In methods that are based solely on statistical information it is not uncommon to assume (cf. [7]), that relation extraction is performed only for a predefined list of concepts extracted earlier with a different method (e.g. [9]). Pattern-based methods rely heavily on so called Hearst patterns, first described in [10]. These are ways of expressing instance enumerations of a class in natural language. Typical forms are „c *such as* i1, i2 or i3" or „c, *for example* i1, i2 or i3". Terms extracted with such patterns may serve as input for elaborate taxonomy and ontology construction methods as, e.g., [11]. While these lexico-syntactic patterns prove quite useful, they may not capture all taxonomical relations expressed in text. Therefore in this paper we describe a novel method of IS-A relation extraction from patterns, which uses morpho-syntactical annotations along with grammatical case of noun phrases that constitute entities participating in IS-A relation. As it will be shown in the paper, the method allows for extraction of additional knowledge from text, that is often not expressed with Hearst patterns. The method is unsupervised, as it is based on hand-crafted patterns, dictionary filtering and manually adjusted support level. Precision of this method, understood as the ratio of correct extracted IS-A relations to all extracted relations is estimated using manual scoring of about 110 relations randomly selected from the method's output. Based on an internet corpus of documents, the method produces a big number of IS-A relations. Most of them (roughly 90%) occur only once in the corpus introducing a high level of noise. We show in conducted experiments that even for a slight increase of support (given as a number of occurrences), the estimated precision of this method increases strongly. We also describe a new method for increasing the number of extracted relations for any support level bigger than 1. The method is based on very simple heuristic for detection of hyponymy between class part of extracted relations, thus we call it *pseudo-subclass boosting* (*PSC* in short). It is worth mentioning that this boosting approach can be applied in any pattern-based relation extraction method. Experiments were conducted on a corpus of about 0.5 billion web documents in Polish language crawled in NEKST project (http://www.nekst.pl) and maintained up to date. These include primarily HTML documents, but also other formats found on websites like PDFs and DOCs. In order to process such high volume of data it was implemented using MapReduce framework [12] implemented in Apache Hadoop project (http://hadoop.apache.org) and Hive (http://hive.apache.org). All examples mentioned in the article are real data, taken from working instance of NEKST system.

II. Our approach

It is known that languages that have inflection and free word order are much harder for automatic analysis[1] than, e.g., English. As pointed out in [14, pp. 100], free word order implies non-projective grammar. It is shown in [15] and [16] that dependency parsing for non-projective grammars is NP-hard, apart from a very narrow subclass called edge-factored grammars. This challenge is addressed, among others, by transition-based dependency parsing [17] used in the pre-processing step for the algorithm described in this paper. We argue that inflection in a language is not only a drawback but

The study is cofounded by the European Union from resources of the European Social Fund. Project PO KL „Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

[1]See e.g. [13] for problems with relation mining in German, in which the word order is much less free than in Polish; note that they use an initial lexicon while we do start from scratch when extracting relations.

TABLE I. SUFFIXES IN INSTRUMENTAL CASE FOR POLISH

	masculine	neuter	feminine
singular	-em		-ą
plural	-ami (-mi)		

can also be a great advantage. Typical constructs that express the hypernymy relation explicitly in Polish language are:

$$NP_1^{Nom} \text{ to } NP_2^{Nom}, \tag{1}$$

$$NP_1^{Nom} \text{ jest } NP_2^{Abl}. \tag{2}$$

Both of them are a way of saying NP_1 *is* NP_2 and in both cases noun phrase NP_1 is expressed in nominative. They differ in grammatical case of NP_2, where in the first construct we have nominative and in the second: instrumental. The second pattern has its equivalent for past tense:

$$NP_1^{Nom} \text{ był/była/było } NP_2^{Abl}. \tag{3}$$

Obviously in case of past tense construction it is possible that IS-A relation no longer holds[2]. The problem exists to a lesser extent also in present tense, which for example can be a consequence of outdated web documents. Assessment of correctness with respect to a given point in time is, in our opinion, a research direction of its own, thus it is out of scope of this paper.

As will be shown later, combination of word and grammatical case pattern allows for relation extraction with quite high precision. It is possible partially thanks to the fact that instrumental case in Polish language is *regular* for nouns and has unique suffixes shown in Table I (after [18, pp. 145, 148]). This makes automatic analysis of sentence tokens easy for this case.

We propose a rule-based approach for IS-A relation extraction with the following procedure:

- run each sentence in corpus through POS-tagger and dependency parser,
- select dependency trees with promising structure,
- apply dictionary filtering for the head of NP_2,
- apply a set of construction rules to dependency tree in order to build instance name out of NP_1 and class name out of NP_2,
- apply a set of filtering rules.

This method is additionally extended with a technique that we call *pseudo-subclass boosting* which increases the number of extracted relations.

It is worth noting that *automatic* detection of IS-A patterns is possible. Experiments described in [19] show that handcrafted ontologies like WordNet can be used successfully as a training set for such pattern discovery task. However, our problem setting differs from that research significantly. Apart from the already mentioned inflection challenge and free word order language, our corpus consists of about 11 billion sentences, which is four orders of magnitude more than the Reuters corpus used in [19] and imposes efficiency limitations. On the other hand, the gain in size comes at the price of quality – Internet documents tend to have much more noisy content than printed journal articles. We have no knowledge of any research on IS-A patterns detection in similar setting (that is web-scale), which leads us to first tackle a more realistic problem of extracting IS-A relations with *known* patterns. Nevertheless, this is a task worth trying given experience gained from research reported here.

[2]The relation was valid in the past only

A. POS tagging and dependency parsing

For part-of-speech tagging we use the Apache OpenNLP (http://opennlp.apache.org) tagger trained with Maximum Entropy classifier on NKJP [20] corpus. Additionally, for known words, we optimized the tag disambiguation process by narrowing tags that can be chosen by information taken from the PoliMorf dictionary [21]. For Polish language, whose tagset contains around 1000 tags [22], this simple optimization gives an improvement of tagging in terms of accuracy and processing speed at the same time. To give an example, the word *artykułów* (inflected form of the word *article*) has only two possible tags `subst:pl:gen:m3` and `subst:pl:gen:p3`. Using this knowledge in OpenNLP tagger reduces search space for this word 500 times. Dependency parsing is based on MaltParser framework [23] trained on Polish Dependency Bank that consists of 8030 sentences [24]. To obtain high processing speed (essential for such large volume of text data) the liblinear classification model has been used.

B. Promising dependency tree structure selection

By *promising* structure of a dependency tree we mean one that matches any of the patterns depicted in Figures 1, 2 and 3, where **form**, **dep** and **pos** mean: token form, dependency relation type (as described in [24]) and part-of-speech tag (as described in [20]) respectively.

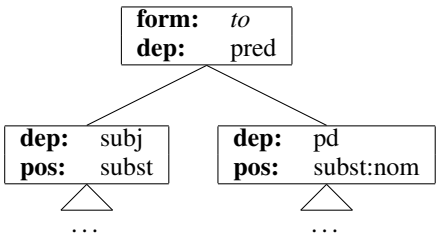

Figure 1. Dependency tree structure for construct (1)

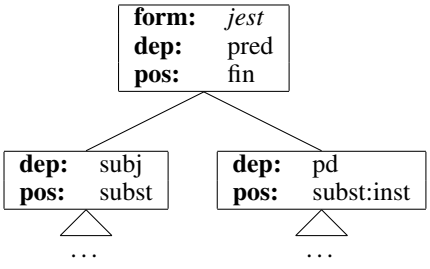

Figure 2. Dependency tree structure for construct (2)

In both nominative and instrumental case, the base structure has a predicate word with outgoing dependency arcs to

two other words with subjective and predicative complement relation type. The difference between structure 1, 2 and 3 is in the grammatical case of the predicative complement and part of speech of the predicate. Our intuition is that selected structures are natural sources of IS-A relation. This claim is supported by the estimated precision obtained in conducted experiments.

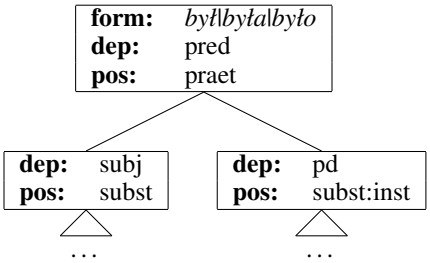

Figure 3. Dependency tree structure for construct (3)

Figure 4 illustrates an example of sentence that matches pattern 2, parsed with our dependency parser and printed in CoNLL [25] format. It is worth noting that in this case the part-of-speech tagger made an error in assigning a case to the adjective *myśliwski* (hunting), where instrumental instead of locative should appear. This may happen because singular masculine adjective suffixes for instrumental (as noted in [18, p. 160]) are not unique as with nouns. That's why in our analysis we focus only on the grammatical case of the head of noun phrase and assume the same case for its dependent adjective tokens. This assumption is justified by the fact, that for Polish language agreement exists between noun and adjective in a noun phrase [26, p. 174]. POS tag in the example is repeated twice because CoNLL format specifies CPOSTAG and POSTAG allowing for coarse-grained and fine-grained part-of-speech tagsets which are the same for Polish language. The following steps illustrate how pattern 2 applies to the example sentence from figure 4:

- find a root word of the sentence (*jest* in our case), and check its dependency relation (must be *pred*) and a POS tag (must be *fin*),
- if the root word has two descendants, then test if:
  - its left descendant (*golden*) has correct dependency relation (must be *subj*) and a POS tag (must be *subst*),
  - its right descendant (*pies*) has correct dependency relation (must be *pd*) and a POS tag (must be *subst:inst*),
- if all requirements are fulfilled, the sentence is moved to the phase of dictionary filtering (section II-C) and instance and class name construction (section II-D).

Given a sentence whose dependency tree matches one of above-mentioned patterns, we construct NP_1 from its left sub-tree and NP_2 from its right sub-tree. Head (or root) of left and right sub-tree will be denoted N_1^H and N_2^H respectively.

C. Dictionary filtering for the head of NP_2

Preliminary experiments showed that many of sentences matching constructs (1) and (2) contain very general, ambiguous nouns in NP_2 like *problem*, *aspect*, *element* or *outcome*. Those nouns cannot be considered proper classes in the sense of IS-A relation, rather they are catch-all phrases used to express various thoughts about what is contained in NP_1.

We eliminated those nouns by manually evaluating a random sample of about 1000 experiment results and creating a dictionary of such meaningless „classes". In this step of our extraction procedure we filter extractions with this dictionary. This process was repeated in three iterations. Size of the dictionary started with 95 catch-all phrases increased by 50, and 20 reaching the level of about 170.

D. Construction rules for NP_1 and NP_2

We construct both instance name (from NP_1) and class name (from NP_2) out of lemmatized tokens. The first step is to serialize tokens present in both dependency sub-trees with operators *leftOffspring* and *rightOffspring*, which operate as follows:

1) put all nodes of dependency sub-tree in a list L,
2) sort L by CoNLL token id descending (for *leftOffspring* operator)/ascending (for *rightOffspring* operator),
3) find index i_H of sub-tree head in L,
4) create sub-list L' from i_H to the first occurrence of interpunction or end of L,
5) in case of *leftOffspring*: sort L' by CoNLL token id ascending,
6) concatenate lemmas of tokens in L' and return.

Computational complexity of this algorithm is $O(n)$, where n is the sentence length. Actual sorting of tokens in case of steps 2. and 5. is not necessary and was introduced to simplify the description[3]. Boundaries detection of instance name is quite simple because it is typically directly defined by left sub-tree of all considered dependency structures (Figures 1, 2 and 3). Therefore it is constructed as concatenation:

$$leftOffspring(N_1^H) + N_1^H + rightOffspring(N_1^H)$$

Creation of class name is more complicated as it is often preceded by degrees of comparison and followed by the rest of the sentence which may be loosely coupled with the class itself. Consider the following sentences:

> Trójmorski Wierch jest jedyną polską górą, z której spływają wody aż do trzech mórz.

(*Trójmorski Wierch is the only Polish mountain, from which waters flow to as many as three seas.*)

> Korona norweska to waluta oznaczana międzynarodowym kodem – NOK.

(*Norwegian krone is a currency marked with the international code – NOK.*)

In the first example, the word *jedyną* (the only) cannot be considered as part of class name. Likewise, anything that comes after word *waluta* (currency) in the second example is merely a description of Norwegian krone, not part of a class name. To address such issues construction rules for class name simply omit the output of *leftOffspring* operator and truncate

[3]It unifies the procedure for left and right part of the sentence.

1	Golden	golden	subst	subst	sg:nom:m3	3	subj
2	retriever	retriever	subst	subst	sg:nom:m2	1	app
3	jest	być	fin	fin	sg:ter:imperf	0	pred
4	psem	pies	subst	subst	sg:inst:m2	3	pd
5	myśliwskim	myśliwski	adj	adj	sg:loc:m3:pos	4	adjunct
6	.	.	interp	interp	_	3	punct

Figure 4. Tree pattern match example in CoNLL format for the Polish sentence "Golden retriever jest psem myśliwskim" (*Golden retriever is a hunting dog*). Note that the parser did not produce fully correct dependency tree (e.g. "Golden" is tagged as noun and linked directly with "jest"). This does not affect our extraction process.

rightOffspring output: it is iterated from left to right only as long as the tokens have POS tag from set {adj, subst, ger} *and* dependency type from set {adjunct, app, conjunct, obj}. So the class name results from concatenating:

$$N_2^H + truncate(rightOffspring(N_2^H))$$

This forces extraction of shorter phrases, which increases the probability of observing a given instance-class pair more than once. As we show in section III, this highly influences the precision of the method. Extraction results for above examples are: *Trójmorski Wierch* `IS-A` *góra* [*Trójmorski Wierch IS-A mountain*] and *Korona norweska* `IS-A` *waluta* [*Norvegian crown IS-A currency*], while from such sentence:

> Narodowy Bank Belgijski jest bankiem centralnym od 1850 roku.[4]

we acquire *Narodowy Bank Belgijski* `IS-A` *bank centralny* [*Belgian National Bank IS-A central bank*].

E. Final filtering rules

It is common that NP_1 contains reference to earlier parts of text. Two types of such reference can be distinguished:

1) explicit:
   > Ten wikipedysta jest numizmatykiem.[5]
2) implicit:
   > Pisarka jest członkiem Związku Pisarzy Białorusi.[6]

In both cases NP_1 typically contains a class of referenced entity, not the entity itself which leads to erroneous extractions. As long as this reference is explicit, we filter such cases with a dictionary of referencing words (pronouns and textual references like *above-mentioned*). The case where reference is implicit is much harder, and at this point left for further research, as described later in section VI.

F. Pseudo-subclass (PSC) boosting

Our experiments showed that the number of extracted relations drops significantly with increase of support level t. To compensate this loss we designed a boosting method that is based on the following intuition: if *I* `IS-A` *C* and *I* `IS-A` *C'* are extracted relations and *C* is a substring of *C'*, then there is high chance that *C'* is a way of describing *I* more precisely than *C*, i.e., *C'* is a pseudo-subclass of *C*. If so, we can boost our confidence in the fact that *I* `IS-A` *C* is properly extracted. To give an example:

> Kraków to najchętniej odwiedzane miasto przez turystów w Polsce. Kraków – dawna stolica Polaków jest miastem magicznym.[7]

Above two sentences allow for boosting confidence in extraction *Kraków* IS-A *miasto* (*Cracow* IS-A *city*). From the first sentence we get the relation *Kraków* IS-A *miasto* and from the second *Kraków* IS-A *miasto magiczne* (*Cracow* IS-A *magic city*). As "miasto magiczne" is a superstring of "miasto", the second sentence supports the first extracted relation. In general, to detect class/pseudo-subclass matches for each extraction *R* = *I* `IS-A` *C* we generate a list *L* of

- prefix lists of tokens from *C*,
- suffix lists of tokens from *C* that don't include leading adjectives.

In Map phase of MapReduce job, we emit the pair *(I, C)* with *R*'s occurrence count and pairs *(I, c)* (with the same count) for each $c \in L$. Reduce phase aggregates our data by matched pairs and here we acquire knowledge about pseudo-subclasses' occurrence count and type of constructs they were discovered in. Figure 5 illustrates a more elaborate case of pseudo-subclass boosting. Each numbered row represents a relation *mukowiscydoza* IS-A ... extracted from text. Row 13 is an example of suffix list boosting with *wieloukładowa* being an adjective removed at the stage of creating list *L*. Rows 2-12 boost relation *mukowiscydoza* IS-A *choroba*, additionally rows 4-7 boost *mukowiscydoza* IS-A *choroba genetyczna*, etc.

III. Experiments

Experiments were conducted on a corpus of about 0.5 billion web documents in Polish language with roughly 11 billion sentences. Tables II, III and V present the results of passing the entire collection through the algorithm described in section II.

Method evaluation was conducted for four levels of the value of t, which, as earlier described, is the minimal IS-A relation occurrence count acceptance threshold. Precision evaluation was based on manual scoring of about 110 randomly selected relations from given experiment's results. Estimated precision was calculated by the formula 4.

$$\hat{Pr} = \frac{TP}{TP + FP} \tag{4}$$

[4]Belgian National Bank is the central bank since 1850.
[5]This wikipedian is a numismatist.
[6]The writer is a member of Union of Belarus Writers.
[7]Cracow is the most visited city by tourists in Poland. Cracow – the former capital of the Poles is a magical city.

```
mukowiscydoza (cystic fibrosis) IS-A
 1. choroba (disease)
 2.   choroba dziedziczna (hereditary disease)
 3.   choroba genetyczna (genetic disease)
 4.     choroba genetyczna ludzi rasy białej
        (genetic disease of white race people)
 5.     choroba genetyczna ogólnoustrojowa (systemic genetic disease)
 6.     choroba genetyczna rasy białej (genetic disease of white race)
 7.     choroba genetyczna układu pokarmowego
        (genetic disease of the digestive system)
 8.   choroba monogenowa (monogenic disease)
 9.   choroba nieuleczalna (incurable disease)
10.   choroba przewlekła (chronic disease)
11.   choroba wielonarządowa (multiorgan disease)
12.   choroba wieloukładowa (multisystem disease)
13.   wieloukładowa choroba (multisystem disease)
14.     wieloukładowa choroba monogenowa
        (multisystem monogenic disease)
15. przyczyna wykonywania (cause of performing)
16.   przyczyna wykonywania przeszczepu płuca
      (cause of performing lung transplant)
17. schorzenie (disease - synonym)
18.   schorzenie genetyczne (genetic disease - synonym)
```

Figure 5. Tree representation of pseudo-subclass boosting.

where TP is the number of relations scored as correct and FP is the number of relations scored as erroneous. Note that we cannot compute other traditional measures as accuracy, recall or F-measure. This is due to the fact, that in Open Relation Extraction setting the number of false negatives (relations incorrectly left out in the extraction process) is not known.

Tables II, III and IV show results of these experiments. Column *nom* contains number of unique IS-A relations extracted only from nominative construct, *inst* is the number of unique relations only from instrumental constructs, *nom∩inst* refers to count of relations extracted from nominatives and instrumentals. Table III refers to the number of relations that were additionally accepted only thanks to pseudo-subclass boosting which helped to observe a given relation more than t times or with both grammar cases.

Total number of extracted IS-A relations, for either nominative or instrumental construction, is slightly above 4 milion (table II). Increase of support level results in drop of accepted relations (up to 1 order of magnitude between consecutive levels). Final count of relations (for $t = 4$) does not exceed 90000, which is almost 2 orders of magnitude lower than the total.

Pseudo-subclass boosting method allows to extract around 86000 more relations at support level 2. Nominal number of additional relations decreases for higher support levels, but increases in terms of relative gain (as shown in the last column of table III).

Estimated precision of our method is 61% at the lowest support level, and achieves 87% for level 4 (table IV). Increasing the number of accepted relations with pseudo-subclass boosting comes at the cost of lower estimated precision. At support level 2 this loss is 1%, but for 3 and 4 jumps to several percent. Estimated precision of our method, equipped with pseudo-subclass boosting, increases with the increase of t, saturating at the level of about 80%. Table IV contains also estimated precision of our implementation of Hearst patterns which is substantially lower (from 14% to 29%).

Experiments were performed on a cluster of 70 machines with total of 980 CPU cores and 4.375TB of RAM. Total processing time of raw web documents: lemmatization, POS tagging, dependency parsing and IS-A relation extraction was under 24 hours.

IV. Relation to Hearst patterns

In order to compare our method with the most popular approach, we implemented Hearst patterns extraction algorithm as follows:

- Detect enumeration phrase R (one of „*taki jak*", „*taki jak na przykład*", „*taki jak np.*" which are special cases of phrase "*such as*" in English) in a sentence, based on lexical constructions proposed in [10].
- Check if words from R to the end of the sentence form a comma separated list of phrases (with the last element optionally separated by conjunction: „*i*" or „*oraz*"). The list is assumed to represent instances of a class.
- Detect the class name in words left to R with a Conditional Random Field model [27]. Words in this part of sentence are labeled with either „1" or „0". The sequence of „1" nearest to R is assumed to represent the class. The model was trained on manually annotated set of around 600 sentences. Its precision calculated on 10-fold cross validation is 93.89%.

Table V shows the number of extracted Hearst patterns and overlap between this method and our approach (percentage

TABLE II. NUMBER OF EXTRACTED RELATIONS FOR DIFFERENT VALUES OF MANUALLY ADJUSTED ACCEPTANCE SUPPORT LEVELS t. NUMBER OF RELATIONS EXTRACTED ARE GIVEN IN COLUMNS: "**NOM**" FOR NOMINATIVE CONSTRUCT AND "**INST**" FOR INSTRUMENTAL CONSTRUCTS. COLUMN "**NOM∩INST**" CONTAINS THE NUMBER OF RELATIONS EXTRACTED WITH BOTH NOMINATIVE AND INSTRUMENTAL CONSTRUCTS.

	nom	**inst**	**nom∩inst**	**total**
$t = 1$	1647500	2380021	39865	4027521
$t = 2$	138877	264764	9895	403641
$t = 3$	52430	100320	4938	152750
$t = 4$	29210	55232	3154	84442

TABLE III. NUMBER OF ADDITIONAL RELATIONS EXTRACTED THANKS TO PSEUDO-SUBCLASS BOOSTING (FOR DIFFERENT VALUES OF SUPPORT LEVEL t). COLUMN "**NOM**" CONTAINS RESULTS FOR NOMINATIVE CONSTRUCT AND "**INST**" FOR INSTRUMENTAL CONSTRUCTS. COLUMN "**NOM∩INST**" CONTAINS THE NUMBER OF ADDITIONAL RELATIONS EXTRACTED WITH BOTH NOMINATIVE AND INSTRUMENTAL CONSTRUCTS.

	nom	**inst**	**nom∩inst**	**total**	**PSC gain**
$t = 1$	0	0	0	0	0%
$t = 2$	24335	61244	2931	85579	21.20%
$t = 3$	13122	38004	2116	51126	33.47%
$t = 4$	8726	26702	1521	35428	41.95%

TABLE IV. ESTIMATED PRECISION ($\hat{Pr}$ – SEE EQUATION 4) OF EXTRACTION FOR DIFFERENT ACCEPTANCE SUPPORT LEVELS. "PSC" STANDS FOR PSEUDO-SUBCLASS BOOSTING. OUR APPROACH IS MARKED WITH "**NOM∩INST**", WHILE "**HRST**" STANDS FOR HEARST PATTERNS.

t	**nom+inst** (no PSC)	**nom+inst** (with PSC)	**hrst**
1	0.61	0.61	0.47
2	0.71	0.72	0.56
3	0.87	0.79	0.58
4	0.87	0.81	0.62

values in brackets are calculated relative to the number of Hearst patterns-based extractions). The overlap varies from 0.57% to 1.02% for nominative scheme and from 1.19% to 2.65% for instrumental. Relations detected in all three methods constitute from 0.25% to 0.58% of relations extracted with the basic method. This suggests that our method allows for extraction of new relations, not expressed in language constructs described by Hearst, with even higher precision.

V. DISCUSSION

Experiments lead to interesting conclusions. Firstly, there is little intersection between IS-A relations extracted by the three methods: Hearst traditional method and our methods, one based on nominative, the other based on instrumental case. The IS-A relation space seems too sparse for such methods to produce overlapping results. Nominative construction produces less relations than instrumental, which presumably is a consequence of the fact that this construct is only applicable for present tense. Decrease in total extractions count is much bigger going from support level 1 to 2 (9.98 times) than when in other cases ($2 \rightarrow 3$: ~2.64 times, $3 \rightarrow 4$: ~1.81 times). It can be connected to the natural model of language, where distribution of word frequencies has power law probability distribution [28]. There is a lot of particular, domain specific taxonomical information that is infrequent in textual resources accessible on the Internet. On the other hand more common knowledge that can be found multiple times in text is substantially less frequent.

Of course pseudo-subclasses don't give any boost when $t = 1$ and do not affect precision, because we simply accept everything that passes the final filtering rules. In other cases PSC increases the number of extractions significantly (the higher t the better), although not as much as to eliminate the effect of increased t. This boosting method is very beneficial for support level 2 as it increases extractions count by 23% with no observable loss in precision (see Table IV). For $t = 3$ and $t = 4$ the gain in extractions count comes at the price of significantly lower precision.

Analysis of false-positive extractions reveal several types of errors made by this method:

1) Implicit reference – which leads to errors like
   - *autor* `IS-A` *dyrektor jednostki* (*author* `IS-A` *director of the unit*),
   - *sobota* `IS-A` *dzień koncertu głównego* (*Saturday* `IS-A` *main concert day*).
2) Wrong decision about phrase begin/ending point[8]:
   - *trening funkcjonalny* `IS-A` *rodzaj (... czego?)* (*functional training* `IS-A` *kind (... of what?)*),
   - ~~*zdecydowana większość*~~ *kandydatów do Parlamentu* `IS-A` *członek określonej partii politycznej* (~~vast majority of~~ *candidates to Parliament* `IS-A` *member of a particular political party*).
3) Ever growing dictionary mentioned in section II-C. After each iteration of catch-all phrases eliminations new such phrases emerge in result samples. Above-mentioned experiments revealed such false-positive classes as: *result*, *an essential element* and *something amazing*. The number of such phrases decreased in each dictionary-construction iteration, which allows us to assume that this set is relatively small. Nonetheless, we are aware that manual construction of this set doesn't take evolution of the language's vocabulary into account.

VI. FUTURE WORK

Plans for future development include dealing with issues detected in above-mentioned experiments. The problem of detecting implicit references to earlier parts of text is known in natural language processing as coreference resolution and

[8]Missing parts are added in brackets, unwanted parts are striked out.

TABLE V. NUMBER OF RELATIONS EXTRACTED WITH HEARST PATTERNS FOR DIFFERENT VALUES OF MANUALLY ADJUSTED ACCEPTANCE SUPPORT LEVELS t.

	hrst	**nom∩hrst**	**inst∩hrst**	**nom∩inst∩hrst**
$t = 1$	4007927	23044 (0.57%)	47953 (1.19%)	10222 (0.25%)
$t = 2$	781419	6492 (0.83%)	15567 (1.99%)	3434 (0.44%)
$t = 3$	356873	3488 (0.98%)	8728 (2.45%)	1899 (0.53%)
$t = 4$	224200	2295 (1.02%)	5939 (2.65%)	1298 (0.58%)

constitutes an independent field of research as described in [29, p. 614] or specifically for Polish: [30]. It is planned to adapt selected coreference resolution methods to our BigData environment and verify their effectiveness in increasing precision of our extraction method.

We plan to achieve better detection of phrase begin/ending points by replacing construction rules described in section II-D with Conditional Random Field classifier trained on sentences scored in our experiment with manually annotated proper phrase boundaries. Creating of such golden standard set of sentences with IS-A relations is of course more time consuming than the approach proposed in this paper. In case of Hearst patterns it turned out to be a necessity. Sentences with Hearst-like enumerations contain more complicated dependency structures which are harder to parse correctly.

Better catch-all phrases elimination can be done as a post-processing step. Membership in these classes should be uniformly distributed over instances and subclasses in the taxonomy, so there should be no significant correlation between membership in these classes and proper classes. Filtering methods based on such correlation will be investigated.

Taking into account the number of filtered out IS-A relations (starting from support level 2) it is worthwhile to consider development of other ways of assessing their correctness. The support level criterion (frequency based) effectively increases quality of extracted information, but at the same time significantly reduces its quantity. It would be interesting to choose one of the most popular classification methods (ea. Support Vector Machine or Random Forest classifier) and check its ability to learn a more sophisticated filtering criterion of incorrect IS-A relations. The feature space for this classification problem could be much richer than simple information about occurrence frequency. One can use more sophisticated characteristics of IS-A relation like for example: size of class and instance phrase (count in number of words), type of sources (nominative, instrumental), popularity of instance and class phrase independently (expressed in number of occurrences among all extracted IS-A relations).

It would be also interesting to compare precision of Hearst patterns implemented with pseudo-subclass boosting.

VII. CONCLUSIONS

This paper presents a novel method of IS-A relation extraction from patterns for Polish that is different from so popular Hearst patterns and is applicable in inflected languages with free word order. Thanks to this method we were able to extract knowledge that may not be expressed in enumeration constructs defined by Hearst. Additionally, a method for boosting relation extractions count is introduced. As mentioned at the beginning, thanks to its simplicity it has potential application in any pattern-based IS-A relation extraction method. As experiments showed, the algorithm achieves satisfactory precision [9] (although there is still room for improvement) and is capable of generating high number of taxonomical relations. This makes it a valuable input source of data for any taxonomy induction task.

It is needless to say that experiments described in this paper do not provide a full statistical overview of millions of IS-A relations extracted from the corpus of Polish Internet documents. We focus on an assessment of precision of the proposed IS-A relation extraction method. In-depth statistical analysis of such a dataset is desirable and remains as a task to be accomplished in the next publication devoted to the research path outlined in the previous section.

REFERENCES

[1] H. Poon and P. Domingos, "Unsupervised ontology induction from text," in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 296–305.

[2] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1535–1545.

[3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the Web," in Proceedings of the 20th International Joint Conference on Artifical Intelligence, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2670–2676.

[4] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation," in Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One, ser. IJCAI'11. AAAI Press, 2011, pp. 3–10.

[5] E. Barbu, "Property type distribution in wordnet, corpora and wikipedia," Expert Systems with Applications, vol. 42, no. 7, 2015, pp. 3501 – 3507.

[6] W. Wu, H. Li, H. Wang, and K. Zhu, "Probase: A probabilistic taxonomy for text understanding," in ACM International Conference on Management of Data (SIGMOD), May 2012.

[7] T. Fountain and M. Lapata, "Taxonomy induction using hierarchical random graphs," in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012, pp. 466–476.

[8] P. Cimiano, A. Hotho, and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis." J. Artif. Intell. Res.(JAIR), vol. 24, 2005, pp. 305–339.

[9] P. Szwed, "Concepts extraction from unstructured Polish texts: A rule based approach," in Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on, Sept 2015, pp. 355–364.

[10] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in Proceedings of the 14th Conference on Computational Linguistics - Volume 2, ser. COLING '92. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 539–545.

[9] 60-80% precision seems to be achieved by other researchers too, see e.g. [31] Figure 4 or [32] table 5.

[11] Z. Kozareva, "Simple, Fast and Accurate Taxonomy Learning," in Text Mining. Springer International Publishing, 2014, pp. 41–62.

[12] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, Jan. 2008, pp. 107–113. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[13] F. Xu, D. Kurz, J. Piskorski, and S. Schmeier, "Term extraction and mining of term relations from unrestricted texts in the financial domain," in Proceedings of the 5th International Conference on Business Information Systems, Poznan, Poland, 2002.

[14] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "Maltparser: A language-independent system for data-driven dependency parsing," Natural Language Engineering, vol. 13, no. 02, 2007, pp. 95–135.

[15] R. McDonald and F. Pereira, "Online learning of approximate dependency parsing algorithms," in In Proc. of EACL, 2006, pp. 81–88.

[16] R. McDonald and G. Satta, "On the complexity of non-projective data-driven dependency parsing," in Proceedings of the 10th International Conference on Parsing Technologies, ser. IWPT '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 121–132.

[17] M. Kuhlmann and J. Nivre, "Transition-based techniques for non-projective dependency parsing," Northern European Journal of Language Technology, vol. 2, no. 1, 2010, pp. 1–19.

[18] A. Nagórko, Zarys gramatyki polskiej. Warszawa: Wydawnictwo Naukowe PWN, 2007.

[19] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," in Advances in Neural Information Processing Systems (NIPS 2004), November 2004.

[20] A. Przepiórkowski, M. Bańko, R. L. Górski, and B. Lewandowska-Tomaszczyk, Eds., Narodowy Korpus Języka Polskiego. Warszawa: Wydawnictwo Naukowe PWN, 2012.

[21] M. Woliński, M. Miłkowski, M. Ogrodniczuk, and A. Przepiórkowski, "Polimorf: a (not so) new open morphological dictionary for Polish," in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), N. Calzolari (Conference Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

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

[23] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "Maltparser: A language-independent system for data-driven dependency parsing," Natural Language Engineering, vol. 13, 6 2007, pp. 95–135.

[24] A. Wróblewska, "Polish Dependency Bank," Linguistic Issues in Language Technology, vol. 7, no. 2, 2012.

[25] S. Buchholz and E. Marsi, "Conll-x shared task on multilingual dependency parsing," in Proceedings of the Tenth Conference on Computational Natural Language Learning, ser. CoNLL-X '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 149–164.

[26] Z. Saloni and M. Świdziński, Składnia współczesnego języka polskiego. Warszawa: Wydawnictwo Naukowe PWN, 2011.

[27] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in Proceedings of the Eighteenth International Conference on Machine Learning, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[28] C. D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. Cambridge, MA, USA: MIT Press, 1999.

[29] A. Clark, C. Fox, and S. Lappin, The Handbook of Computational Linguistics and Natural Language Processing. Wiley-Blackwell, 2010.

[30] M. Ogrodniczuk, A. Wójcicka, K. Głowińska, and M. Kopeć, "Detection of nested mentions for coreference resolution in Polish," in Advances in Natural Language Processing: Proceedings of the 9th International Conference on NLP, PolTAL 2014, Warsaw, Poland, September 17–19, 2014, ser. Lecture Notes in Artificial Intelligence, A. Przepiórkowski and M. Ogrodniczuk, Eds. Heidelberg: Springer International Publishing, 2014, vol. 8686, pp. 270–277.

[31] P.-M. Ryu and K.-S. Choi, "Automatic acquisition of ranked is-a relation from unstructured text," 2007.

[32] D. Ravichandran, P. Pantel, and E. Hovy, "The Terascale Challenge," in Proceedings of KDD Workshop on Mining for and from the Semantic Web (MSW-04), 2004, pp. 1–11.