RICH-CPL: Fact Extraction from Wikipedia-sized Corpora for Morphologically Rich Languages

Sergei Budkov, Kseniya Buraya, Andrey Filchenkov, Ivan Smetannikov, Antonina Puchkovskaia
ITMO University
Saint-Petersburg, Russia
s.a.budkov@gmail.com, {ksburaya,afilchenkov,ismentannikov}@corp.ifmo.ru, artonina@gmail.com

Abstract—**This work deals with never-ending learning approach for fact extraction from unstructured Russian text. It continues the research in the field of pattern learning techniques for morphologically rich free-word-order language. We introduce improvements for CPL-RUS algorithm and choose best initial parameters. We conducted experiments with the extended version, RICH-CPL algorithm on the corpus containing over 1.3 million pages. This paper is shortened version of our paper [7] that includes also new modifications of the proposed methods.**

I. Introduction

Fact extraction from the Internet is a relevant task today, but one major problem is the formalization of knowledge uploaded to the web. Ontologies are one of the best instruments for representing information for IE (information extraction) systems. Ontology is "an explicit specification of a conceptualization" and it can be represented as a structure containing concepts and their relationships [1]. It may also contain a set of axioms that define the relations and constraints on their interpretation [2]. Ontologies allow for the formalization of data to simplify automatic processing, and they can be used in tasks such as information retrieval, text analysis and semantic applications ([3], [4]).

In this work we concentrated on the never-ending learning (NEL) approach for fact extraction from an unstructured text [5]. Using this approach the learner is continuously evolving with time in an autonomous way. One advantage to NEL is the small size of preprocessed data required for the learning process. The never-ending learning system can achieve a high performance level for extracting facts from a large corpus given a small initial ontology.

The first prototype implementation of the NEL approach was done for English and was called NELL (Never-ending language learning) [5]. Since the system showed good results an attempt was made to extend the never-ending learning approach to Portuguese [6]. Experiments showed that applying initial NELL parameters to web-pages written on morphologically rich language would not provide strong results.

The implementation of the approach to considering language rules and morphological features was conducted for Russian [7]. The researchers adapted the CPL algorithm for working with unstructured Russian text. They did experiments for 9 ontology categories using a small dataset of 2.5 million words. The main results of these experiments were that (1) it is possible to adapt CPL for Russian with relatively little effort, (2) morphological constraints are crucial for Russian pattern learning, (3) a small set of manually compiled seed patterns increases CPL accuracy and (4) results vary for different categories. The accuracy of the adapted algorithm after 10 iterations varied from 0.16 to 1.0 for different categories. The maximum average accuracy for Russian was 0.612. Upon completion of the experiments on the small dataset, the best strategy for applying the method to larger datasets was determined and then used in analyzing the entire of the Russian Wikipedia corpus.

The rest of the paper is organized as follows. In Section II, we describe how original algorithm for the Russian language works. In Section III, we describe new CLP component we call RICH-CLP. In Section IV, we describe initial small ontology and 30 patterns we start with to apply NEL with RICH-CLP on the Russian Wikipedia and procedures we use, as well as criteria with which we will test the resulting ontology. In Section V we summarize results of these tests and discuss them in Section VI. Section VII provides a brief review of related work. In Section VIII we summarize results and outline future work.

II. Russian Adapted CPL

CPL (Coupled Pattern Learner) is one of NELL's components. NELL is a system for relationship extraction from unstructured text containing two major parts: a knowledge base (KB) and a set of iterative learners. The system works iteratively in an indefinitely running process. First, the learners try to extract as many candidate facts as possible given a current state of the KB. Next, the KB is updated using the learners' output.

In this work we continue to improve CPL based on the results indicated by [7]. CPL is a free-text extractor that learns contextual patterns for extracting instances of ontology categories. The key idea of CPL is that simultaneous ("coupled") learning of instances and patterns yields higher performance than learning items independently [8].

CPL analyzes relations of the following type: (category word; instance extracting pattern; instance word). The learning process can be divided into two steps:

- **Instance extraction**. To extract new instances, the system finds a co-occurrence of the category word with a pattern from the trusted list before identifying the instance word. If both the category and instance words satisfy the conditions of the pattern, then the found word is added to the pool of candidate instances for the current iteration. When all sentences are processed, the candidate instance evaluation begins, after which the most reliable instances are added to the set of trusted instances;

- **Pattern extraction**. To extract new patterns, the system finds a co-occurrence of the category word with one of its trusted instances. The sequence of words between category and instance are identified as a candidate pattern. When all candidate patterns are collected, the most reliable patterns are added to the trusted set.

In the first version of adapted method, named CPL-RUS, the researchers presented extra morphological restrictions that increased the accuracy of the algorithm for Russian text. They conducted experiments to determine the dependence of method on sizes of initial sets and continue to improve CPL-RUS to increase the quality of extracted facts.

III. RICH-CPL

A. Strategies for expanding the trusted sets

During the first step of each iteration, the algorithm collects instances and patterns in sets of candidates for each category from the KB. The second step is creating trusted sets by filtering sets of candidates. In this paper two filtering strategies were considered: THRESHOLD-SUPPORT and THRESHOLD-N. Both strategies are based on *Support* metric which is calculated using the following formulas:

$$Support_c^{(t)}(i) = \frac{\sum_{p \in \text{TruPat}_c^{(t-1)}} Count_c(i,p)}{Count_c(i)}$$

for instances and

$$Support_c^{(t)}(p) = \frac{\sum_{i \in \text{TruInst}_c^{(t-1)}} Count_c(i,p)}{Count_c(p)}$$

for patterns, where i is an instance word, p is a pattern, $Count_c(i,p)$ is the number of cases when i and c match as arguments of p in the corpus related to category c (joint occurrence), $Count_c(x)$ is the total number of matches of x in the corpus related to category c, $TruInst$ is a set of trusted instances, $TruPat$ is a set of trusted patterns, and (t) is an iteration.

THRESHOLD-SUPPORT strategy is based on assumption that the probability of mistake on the first iteration is very low because of human-collected initial ontology and initial patterns set. During the first iteration the algorithm calculates the minimal support for each iteration as a minimum value of *Support* metric for objects of category. Then for each iteration the algorithm will add all objects to the trusted sets where *Support* is greater than minimal support for its category.

THRESHOLD-N strategy using fixed N for each category from KB. For each iteration the candidate sets are sorted by *Support* metric. The algorithm adds first N objects with highest *Support* to the trusted sets.

Experiments were conducted for three independent categories on thematic texts from Wikipedia. For each category RICH-CPL ran for ten iterations with THRESHOLD-SUPPORT strategy and then with THRESHOLD-N strategy for N=50. Experiments results can be found in section 4.3.

B. Joint occurrences

To filter candidate lists Support metric was used. This metric has one flaw: if a really rare object that exists only once in text is processed the Support metric will be 1 (maximum value). Some times it can help to find rare but correct instance or pattern but usually this objects decreases the quality of the algorithm. To solve this problem we added extra filtering by minimal count of joint occurrences. Joint occurrence is when the pattern or instances occurs in text with its category word. We conducted experiments to determine the optimal N for this extra filtering. Experiments and results can be found in sections 4.2 and 4.3.

C. Subpatterns and entity containers

During algorithm execution sometimes there were patterns in KB which were concatenations of other patterns and instances of the same category. More often such patterns decrease the quality of the algorithm for future iterations because of occupying positions in trusted sets without extracting new instances for several iterations. RICH-CPL was improved by adding detection of sub-patterns in complex patterns. A subpattern is usually more general pattern for the same category. Examples of complex patterns and the sub-patterns they were replaced with can be found in Table III-C.

TABLE I. EXAMPLES OF COMPLEX PATTERNS AND DETECTED SUBPATTERNS.

Complex pattern	Detected subpattern	Category
arg1 as Singapore and arg2 arg1 как Сингапур и arg2	arg1 as arg2 arg1 как arg2	Country Страна
arg1: herring, arg2 arg1: сельдь, arg2	arg1: arg2 arg1: arg2	Fish Рыба
arg1 – football and arg2 arg1 – футбол и arg2	arg1 – arg2 arg1 – arg2	Sport Спорт
arg1, for example sambo, arg2 arg1, например самбо, arg2	arg1, for example arg2 arg1, например arg2	Sport Спорт

To avoid possible problems with the loss of some instances detection was added of containers of instances. Containers helps to detect instances in enumerations. RICH-CPL works with containers of instances of the following type: ”$arg(,|and)arg(,|and)arg...(and)arg$”.

D. Complex entities

In this work processing complex instances was included. We work with instances of the following formats: (adjective + noun) or (noun + adjective) and we added a list to instance entry in KB containing all additional words it can be used with. During the instances extraction step we looked first broader then pattern to find additional words if they exist. During the patterns extraction step we used instances with each additional word from its entry to extract more general patterns.

IV. INCREASING ONTOLOGY WITH RICH-CLP USING THE RUSSIAN WIKIPEDIA

A. Initial ontology and set of patterns

RICH-CPL requires an initial ontology and set of initial patterns. [5] showed that given an initial ontology that contains 10–20 seeds for each category as an input, NELL can achieve a high performance level for extracting facts and relations from a large corpus. [6] made a conclusion that in order to extend the NELL approach to a new language, it is necessary to prepare a new seed ontology and contextual patterns that depend on the language rules.

Initial ontology for adapted algorithm can be represented as a table containing three columns: Category name, Seed instances and Numbers of initial patterns (indexes of patterns from set of initial patterns which are suitable for category). The set of seed patterns can be represented as a table containing eight columns: Id of pattern, String presentation of pattern, case of arg1, case of arg2, number of arg1, number of arg2, part of speech of arg1 and part of speech of arg2. To simplify the notation in KB we used Pymorphy2 notation to denote grammemes.

We created an initial ontology with 12 categories with 10–15 seed instances for each and set of seed patterns containing 39 objects. Our ontology contains previously used parts of translated ontology [6]. Our set of seed patterns contains patterns from previous CPL-RUS experiments [7] and other patterns [9]. Preprocessed data can be found in Tables II and III, IV.

B. Experiment design

The first experiments were designed to choose the best filtering strategy. We ran experiments for three independent categories on thematic texts from Wikipedia. For each category the RICH-CPL ran for ten iterations with Threshold-Support strategy and then with Threshold-N strategy for $N = 50$. We used the initial ontology and seed patterns presented by [7].

In this work, we conducted all experiments for all categories on each part of text corpus independently. Then we collected all extracted instances and manually annotate them as correct or incorrect. Then for each category c, we evaluated precision using the following formula:

$$Precision(c) = \frac{CorrInst(c)}{AllInst(c)},$$

where $CorrInst(c)$ is the number of correct instances extracted for category c, and $AllInst(c)$ is the whole number of instances, that were extracted by RICH-CPL for category c.

The next experiment sought to choose best N N for extra filtering by joint occurrences. In these experiments we used the initial ontology and patterns set described in 4.1. We conducted the experiments for three categories using a randomly selected $1/10$ of Russian Wikipedia corpus. We tested $N = 2$ and $N = 3$.

The most important experiments in this work are testing RICH-CPL on the Wikipedia corpus, which contained about 1.3 million pages. All the structural features of Wikipedia were eliminated from the texts, and the text corpus was divided into four parts.

We used Threshold-N strategy for filtering candidate sets with $N = 500$ for instances and patterns as well. An algorithm run for ten iterations (or less) for each part of text corpus after that we performed a final filtering using minimal support value 0.1.

TABLE II. Initial ontology

	Category	Seed instances
1	Профессия Profession	Менеджер, Биолог, Химик, Пилот, Министр, Нотариус, Охранник, Полицейский, Диетолог, Окулист, Астроном, Физик, Учитель, Кондитер Manager, Biologist, Chemist, Pilot, Minister, Notary, Guard, Police officer, Nutritionist, Oculist, Astronomer, Physicist, Teacher, Confectioner
2	Болезнь Illness	Простуда, Грипп, Рак, Кашель, Спид, Анемия, Алкалоз, Ангина, Кандидоз, Кариес, Клонорхоз Colds, Flu, Cancer, Cough, Speed, Anemia, Alkalosis, Angina, Candidiasis, Caries, Clonorchiasis
3	Язык Language	Русский, Английский, Французский, Испанский, Немецкий, Иврит, Португальский, Украинский, Итальянский, Китайский Russian, English, French, Spanish, German, Hebrew, Portuguese, Ukrainian, Italian, Chinese
4	Овощ Vegetable	Огурец, Помидор, Морковь, Репа, Сельдерей, Авокадо, Капуста, Картофель, Лук, Спаржа Cucumber, Tomato, Carrot, Turnip, Celery, Avocado, Cabbage, Potatoes, Onions, Asparagus
5	Спорт Sport	Футбол, Баскетбол, Теннис, Волейбол, Керлинг, Гольф, Бильярд, Скалолазание, Бейсбол, Серфинг Football, Basketball, Tennis, Volleyball, Curling, Golf, Billiards, Rock-climbing, Baseball, Surfing
6	Автомобиль Automobile	Форд, Хонда, Ниссан, Тойота, Сааб, Мазда, Камаз, Жигули, Ваз, Волга Ford, Honda, Nissan, Toyota, Saab, Mazda, Kamaz, Zhiguli, Vaz, Volga
7	Фрукт Fruit	Апельсин, Персик, Лимон, Киви, Ананас, Яблоко, Мандарин, Банан, Груша, Виноград Orange, Peach, Lemon, Kiwi, Pineapple, Apple, Mandarin, Banana, Pear, Grapes
8	Животное Animal	Муравей, Краб, Бронтозавр, Сова, Пчела, Птица, Червяк, Улитка, Белка, Кит, Оса, Угорь, Жираф, Носорог, Акула Ant, Crab, Brontosaurus, Owl, Bee, Bird, Worm, Snail, Squirrel, Whale, Wasp, Eel, Giraffe, Rhinoceros, Shark
9	Птица Bird	Малиновка, Дрозд, Кардинал, Иволга, Гусь, Утка, Лебедь, Фламинго, Пеликан, Пингвин, Индюк, Курица, Дятел, Сойка, Ласточка Robin, Thrush, Cardinal, Oriole, Goose, Duck, Swan, Flamingo, Pelican, Penguin, Turkey, Chicken, Woodpecker, Jay, Swallow
10	Рыба Fish	Акула, Анчоус, Бас, Рыболов, Треска, Пикша, Лосось, Сельдь, Сом, Палтус, Щука, Осетр, Тунец Shark, Anchovy, Bass, Fisherman, Cod, Haddock, Salmon, Herring, Catfish, Halibut, Pike, Sturgeon, Tuna
11	Страна Country	Канада, Бразилия, Ирак, Россия, Япония, Китай, Испания, Португалия, Голландия Canada, Brazil, Iraq, Russia, Japan, China, Spain, Portugal, Holland
12	Напиток Drink	Кофе, Сок, Молоко, Пиво, Эль, Вино, Пепси, Чай, Шампанское, Мартини, Кисель, Компот Coffee, Juice, Milk, Beer, Ale, Wine, Pepsi, Tea, Champagne, Martini, Kissel, Compote

The final experiments aimed to calculate recall. We used randomly selected 150,000 pages from Wikipedia and ran experiments for six categories. First RICH-CPL was trained on 90 percents of corpus and then we measures recall on remaining 10 percents. We calculate recall by following formula:

$$Recall = \frac{Number\ of\ instances\ found\ for\ category}{Number\ of\ all\ instances\ for\ category\ in\ text}$$

We calculated recall on each iteration for all categories independently.

TABLE III. Initial patterns set, part i

	Pattern	arg1, case	arg2, case	arg1, num	arg2, num	arg1, pos	arg2, pos
1	arg1, такие как arg2 arg1, such as arg2	nomn	nomn	plur	all	noun	noun
2	arg2 являются arg1 arg2 are arg1	ablt	nomn	all	all	noun	noun
3	arg2 относятся к arg1 arg2 refer to arg1	datv	nomn	all	all	adjf	noun
4	arg2 относятся к arg1 arg2 refer to arg1	datv	nomn	all	all	noun	noun
5	такие arg1, как arg2 such arg1, as arg2	nomn	nomn	plur	all	noun	noun
6	таких arg1, как arg2 such arg1, as arg2	gent	nomn	plur	all	noun	noun
7	таким arg1, как arg2 such arg1, as arg2	datv	datv	plur	all	noun	noun
8	таким arg1, как arg2 such arg1, as arg2	datv	nomn	plur	all	noun	noun
9	arg1, таких как arg2 arg1, such as arg1	gent	nomn	plur	all	noun	noun
10	arg1, таким как arg2 arg1, such as arg1	datv	nomn	plur	all	noun	noun
11	arg2, а также другие arg1 arg2, as well as arg1	nomn	nomn	plur	all	noun	noun
12	arg2, также как и другие arg1 arg2, as well as other arg1	nomn	nomn	plur	all	noun	noun
13	arg2, и другие arg1 arg2, and other arg1	nomn	nomn	plur	all	noun	noun
14	arg2, а также другим arg1 arg2, and also oher arg1	datv	datv	plur	all	noun	noun
15	arg2, также как и другим arg1 arg2 as well as the other arg1	datv	datv	plur	all	noun	noun
16	arg2, и другим arg1 arg2 and other arg1	datv	datv	plur	all	noun	noun
17	arg2, а также других arg1 arg2, and also other arg1	gent	gent	plur	all	noun	noun
18	arg2, также как и других arg1 arg2, as well as other arg1	gent	gent	plur	all	noun	noun
19	arg2, и других arg1 arg2 and other arg1	gent	gent	plur	all	noun	noun
20	виды arg1, как arg2 types arg1, as arg2	gent	nomn	plur	all	noun	noun

TABLE IV. Initial patterns set, part II

20	виды arg1, как arg2 types arg1, as arg2	gent	nomn	plur	all	noun	noun
21	типы arg1, как arg2 arg1 types, like arg2	gent	nomn	plur	all	noun	noun
22	формы arg1, как arg2 forms arg1, as arg2	gent	nomn	plur	all	noun	noun
23	разновидности arg1, как arg2 varieties of arg1, like arg2	gent	nomn	plur	all	noun	noun
24	сорта arg1, как arg2 varieties arg1, as arg2	gent	nomn	plur	all	noun	noun
25	arg2 — вид arg1 arg2 — form of arg 1	gent	nomn	all	all	noun	noun
26	arg2 — тип arg1 arg2 — type of arg1	gent	nomn	all	all	noun	noun
27	arg2 — форма arg1 arg2 — form of arg1	gent	nomn	all	all	noun	noun
28	arg2 — разновидность arg1 arg2 — kind of arg1	gent	nomn	all	all	noun	noun
29	arg2 — сорт arg1 arg2 — sort of arg1	gent	nomn	all	all	noun	noun
30	виды arg1, как arg2 types arg1, as arg2	gent	nomn	plur	all	adjf	noun
31	типы arg1, как arg2 types arg1, like arg2	gent	nomn	plur	all	adjf	noun
32	формы arg1, как arg2 forms arg1, as arg2	gent	nomn	plur	all	adjf	noun
33	разновидности arg1, как arg2 varieties of arg1, like arg2	gent	nomn	plur	all	adjf	noun
34	сорта arg1, как arg2 varieties arg1, like arg2	gent	nomn	plur	all	adjf	noun
35	arg2 — вид arg1 arg2 — form of arg1	gent	nomn	all	all	adjf	noun
36	arg2 — тип arg1 arg2 — type of arg1	gent	nomn	all	all	adjf	noun
37	arg2 — форма arg1 arg2 — form of arg1	gent	nomn	all	all	adjf	noun
38	arg2 — разновидность arg1 arg2 — kind of arg1	gent	nomn	all	all	adjf	noun
39	arg2 — сорт arg1 arg2 — sort of arg1	gent	nomn	all	all	adjf	noun

V. Results

A. Results on choosing best strategy

On determining the best strategy experiments results can be found in Table V. Results shows that using the Threshold-N strategy gives a better balance of number of extracted instances and precision. Also number of extracted instances distributed more evenly. This fact excludes possible error accumulation in case of the really small support will be chosen during the first iteration while using Threshold-Support strategy. We used the Threshold-N strategy in the final version of RICH-CPL algorithm.

TABLE V. Experiments results on best strategy

Category	Strategy	Precision / Number of instances		
	Minimal support	0,1	0,5	1
Bird	Threshold-Support	0,56 / 97	1 / 9	1 / 5
	Threshod-N	0,83 / 55	0,84 / 55	0,86 / 22
Fish	Threshold-Support	0,2 / 90	0,6 / 8	0,8 / 5
	Threshod-N	0,6 / 55	0,7 / 10	0,71 / 7
Mammal	Threshold-Support	0,6 / 70	0,9 / 19	0,91 / 12
	Threshod-N	0,94 / 52	1 / 27	1 / 16
Average	Threshold-Support	0,45/ **85**	0,83 / 12	**0,9** / 7
	Threshod-N	**0,79** / 54	**0,85** / **31**	0,86 / **15**

B. Results on choosing best N for extra filtering

On choosing the best N for extra filtering experiment results can be found in Table VI. Results shows that using $N = 3$ is excessive because it decreases the number of extracted facts and precision. We used $N = 2$ for extra filtering in the final version of RICH-CPL.

TABLE VI. Experiments results on choosing N for extra filtering

Category	N	Precision / Number of instances		
	Minimal support	0,1	0,5	1
Bird	N=3	0,53 / 13	1 / 6	1 / 6
	N=2	0,65 / 26	0,85 / 7	1 / 6
Sport	N=3	0,9 / 52	0,95 / 20	0,9 / 11
	N=2	0,83 / 55	0,76 / 30	1 / 13
Fish	N=3	0,75 / 20	0,75 / 8	0,8 / 5
	N=2	0,75 / 36	0,9 / 11	1 / 5
Average	N=3	0,73 / 28	**0,9** / 11	0,9 / 7
	N=2	**0,74 / 39**	0,84 / **16**	**1 / 8**

C. Results on testing RICH-CPL on Wikipedia corpus

On testing RICH-CPL on Wikipedia corpus this section demonstrates results of running RICH-CPL on the Russian Wikipedia. Results can be found in Table VII.

TABLE VII. Results on testing RICH-CPL on Wikipedia corpus

Category		Number of extracted facts	Precision
1	Profession	80	96%
2	Illness	533	65%
3	Language	453	78%
4	Vegetable	77	65%
5	Sport	475	63%
6	Automobile	24	54%
7	Fruit	74	62%
8	Animal	1145	83%
9	Bird	299	68%
10	Fish	310	83%
11	Country	222	43%
12	Drink	92	68%
Average		315	69%

D. Results on calculating recall

On calculating recall one of the advantages of NEL is that recall metric is tends to 100% while texts size tends to infinity. But because of the impossibility of achieving infinity we conducted experiments on finite texts. Results can be found in Table VIII and on Fig. 1.

TABLE VIII. Results on calculating recall

Iteration	1	2	3	4	5	6	7	8
Category								
Country	0,092	0,101	0,814	0,824	0,861	0,861	0,861	**0,861**
Language	0,084	0,084	0,65	0,668	0,674	0,674	0,674	**0,674**
Illness	0,17	0,17	0,2	0,517	0,689	0,689	0,758	**0,793**
Animal	0,2	0,2	0,4	0,43	0,53	0,73	0,8	**0,8**
Bird	0,22	0,31	0,31	0,36	0,45	0,59	0,63	**0,681**
Fish	0,26	0,26	0,26	0,42	0,52	0,63	0,684	**0,737**

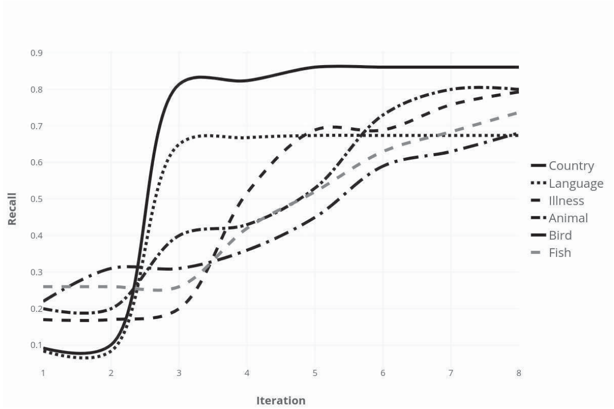

Fig. 1. Dependence of RICH-CPL recall on iteration

VI. Discussion

Since this work continues to improve and expand CPL-RUS presented in [7] we will mostly compare our results with them. In this work we added extra filtering by joint occurrences and chose the best N for it; detection of sub-patterns and containers of instances; processing of complex instances; and chose best filtering strategy. By adding all these features to CPL-RUS we got a new version of an adapted algorithm named RICH-CPL.

CPL-RUS experiments concentrated on thematic texts and a really small initial ontology and patterns set to get the base results. We expanded the initial data depending on the language rules. We used the whole Russian Wikipedia corpus without structural features to get more independent results. We also presented recall metric for RICH-CPL.

RICH-CPL shows precision from 43% to 96% depends on category after processing Wikipedia corpus. The average precision of fact in KB was 69%. CPL- RUS showed precision from 16% to 100% with average 61% while processing less texts for less categories.

CPL-RUS showed a better result for processing texts written in morphologically rich language presented in [6]. RICH-CPL indicated better results as well. The results for the Portuguese version of CPL are presented separately for 5, 10, 15, 20 iterations of the algorithm. Since we did not run more than 10 iterations of CPL for each category, the most valuable result of comparison of two CPL realizations is to choose the accuracy of 10-iterations of the Portuguese CPL. The results of the average accuracy for the Portuguese CPL varied from 0.04 to 0.95 [6].

Comparison with original CPL algorithm applied to English text [5] will be not valid because of big difference in languages and processed texts. Original CPL showed precision from 20% to 100% with average 78%.

RICH-CPL showed recall from 67% to 86% depends on category. Unfortunately there aren't any recall metrics for other CPL realizations.

VII. Related Work

In this paper we introduced improvements and experiments results for RICH- CPL algorithm which is the adaptation of CPL algorithm from NELL system. More general introduction to NELL and its predecessors can be found in [5]. There are more systems which goal is facts extraction from unstructured text written in natural language e.g. OLLIE [12], Snowball [11] and Sofie [10].

This work continues the research in the field of pattern learning techniques for the Russian language [7]. We tested our method on the Wikipedia corpus as it was done earlier by [13], who applied a number of machine learning techniques to automatic relation extraction from the Russian Wikipedia but their method depends on the specific structure of the resource. There is research on Russian in fields of IE [14], building of linguistic resources [15], [16] and taxonomic relations extraction from text [17], 2010, [9] .

Our method uses bootstrapping for ontology learning from unstructured text which has been applied, for example, by [18]. We use coupled learning of both instances and patterns because though these two tasks are traditionally broken down into separate components. This is a rather artificial division leading to over-simplification and error propagation from the earlier tasks to the later steps [19]. RICH-CPL uses knowledge base to extract relations. This approach has been previously proposed as a distant supervision approach by, among others, [20] and [21].

VIII. Conclusion

In this work, we introduced a new RICH-CPL algorithm chose best initial parameters. We conducted experiments with it on the corpus containing over 1.3 million pages. We presented the recall metric for the adapted method.

This work leaves room for further experiments. We plan to run CPL on much bigger datasets to make a valid comparison with original work, conduct experiments to stabilize precision for all categories and conduct experiments to learn out the strategy for dynamic selection of N depends on category.

We will also continue working on implementing other modules of NELL system. Another line of research is to run CPL on top of syntactic annotation; in principle, this should increase precision though some amount of errors might be introduced by syntax parser itself.

Future work will be directed on stabilization of precision for all categories. We also plan to expand the initial ontology and text corpus to original CPL data [5] to make a valid comparison. Also current results show some problems because of choosing a fixed N for the Threshold-N strategy. We plan to conduct experiments to learn out the strategy for dynamic selection of N depends on category. To this day, our work concentrated on adapting and improving CPL without concern for other NELL modules. In addition to the final improvement of RICH-CPL we are going to adapt other learners of the NELL system to work with Russian language.

Acknowledgment

The authors would like to thank Lidia Pivovarova for useful conversation.

This work was financially supported by the Government of the Russian Federation (Grant 08-08).

References

[1] Gruber, Thomas R., *Toward principles for the design of ontologies used for knowledge sharing?* International journal of human-computer studies 43.5-6 (1995): 907-928.

[2] Guarino, Nicola, ed., *Formal ontology in information systems*, Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy. Vol. 46. IOS press, 1998.

[3] Albertsen, Thomas, and Eva Blomqvist, *Describing ontology applications*. European Semantic Web Conference. Springer, Berlin, Heidelberg, 2007.

[4] Staab, Steffen, and Rudi Studer, eds, *Handbook on ontologies*. Springer Science & Business Media, 2010.

[5] Carlson, Andrew, et al., *Toward an architecture for never-ending language learning*. AAAI. Vol. 5. 2010.

[6] Duarte, Maisa C., and Estevam R. Hruschka, *How to read the web in portuguese using the never-ending language learner's principles.* Intelligent Systems Design and Applications (ISDA), 2014 14th International Conference on. IEEE, 2014.

[7] Buraya, Kseniya, et al., *Toward Never Ending Language Learning for Morphologically Rich Languages.* Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing. Association for Computational Linguistics, 2017.

[8] Carlson, Andrew, et al., *Coupled semi-supervised learning for information extraction*. Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010.

[9] Sabirova, Kristina, and Artem Lukanin., *Automatic Extraction of Hypernyms and Hyponyms from Russian Texts*. AIST (Supplement). 2014.

[10] Suchanek, Fabian M., Mauro Sozio, and Gerhard Weikum., *SOFIE: a self-organizing framework for information extraction*. Proceedings of the 18th international conference on World wide web. ACM, 2009.

[11] Agichtein, Eugene, and Luis Gravano., *Agichtein, Eugene, and Luis Gravano*. Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000.

[12] Schmitz, Michael, et al., *Open language learning for information extraction*. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012. 2014.

[13] Kuznetsov, Artem, Pavel Braslavski, and Vladimir Ivanov., *Family Matters: Company Relations Extraction from Wikipedia*. International Conference on Knowledge Engineering and the Semantic Web. Springer, Cham, 2016.

[14] Starostin, A. S., et al., *FactRuEval 2016: evaluation of named entity recognition and fact extraction systems for Russian*. (2016).

[15] Loukachevitch, Natalia, and Boris Dobrov., *RuThes linguistic ontology vs. Russian wordnets*. Proceedings of the Seventh Global Wordnet Conference. 2014.

[16] Braslavski, Pavel, et al., *Yarn: Spinning-in-progress*. (2016): 58-65.

[17] Bocharov, Victor, et al., *Ontological parsing of encyclopedia information*. nternational Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg, 2010.

[18] Liu, Wei, et al., *Semi-automatic ontology extension using spreading activation*. Journal of Universal Knowledge Management 1 (2005): 50-58.

[19] Li, Qi, and Heng Ji., *Incremental joint extraction of entity mentions and relations*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2014.

[20] Mintz, Mike, et al, *Distant supervision for relation extraction without labeled data*. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. Association for Computational Linguistics, 2009.

[21] LRiedel, Sebastian, et al., *Relation extraction with matrix factorization and universal schemas*. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013.