Sense Inventory Alignment Using Lexical Substitutions and Crowdsourcing

Dmitry Ustalov*
*Ural Federal University
Yekaterinburg, Russia
dmitry.ustalov@urfu.ru

Sergey Igushkin†
†ITMO University
Saint Petersburg, Russia
hotkeytlt@gmail.com

***Abstract*—Sense inventory induction is a topical problem of deriving a set of synsets representing concepts using various automatic or human-assisted methods. There might be, and actually are, mistakes in such synsets. In this paper, we are focused on the problem of eliminating potentially duplicate synsets having exactly two words in common as the broader intersection is known to be successfully addressed by heuristics. We exploit the phenomena of lexical substitutions and microtask-based crowdsourcing for aligning the synsets to the individual word senses. We also present an open source mobile application implementing our approach. Our experiments on the Russian language show that the approach scales well and dramatically reduces the number of duplicate synsets in the inventory.**

I. Introduction

A sense inventory is an exhaustive listing of all the senses of every word that an application must be concerned with [1, P. 229]. Popular sources of sense inventories used in practice are the sets of synonyms (synsets) represented in the electronic databases like WordNet or its derivatives, which are the fine-grained lexical resources made by expert lexicographers [2]. Since that not every natural language has a sense inventory of high quality, the researchers are trying to induce it by both using automatic matching and machine learning methods, and human-assisted approaches like crowdsourcing. In contrast to the expensive expert-based approach producing the result of very high quality, both human volunteers and machines tend to make mistakes in recognizing lexical senses.

In our previous study for the Russian language [3], we empirically found that two synsets can be treated as duplicates if they share at least three words, e.g., the synsets $\{\text{car}, \text{machine}, \text{automobile}, \text{auto}\}$ and $\{\text{car}, \text{ride}, \text{automobile}, \text{auto}\}$ are duplicates expressing the equivalent lexical senses. However, such a heuristic fails at narrower intersections, resulting in nonsensically broad synsets. In this study, given the set of words grouped by the set of possibly duplicated synsets, we aim at regrouping these words into the new synsets according to the provided contexts from the sense-distinguished text corpus. For that, we use crowdsourcing in the form of lexical substitution microtasks.

Section II reviews the related work. Section III presents an approach for aligning synsets. Section IV demonstrates the implementations details. Section V describes the experimental setup and the metric used. Section VI shows the results. Section VII discusses the interesting findings. Section VIII concludes the paper and defines directions for future work.

II. Related work

The literature review is dedicated to two aspects. Firstly, we discuss the approaches for sense inventory induction. Secondly, we present a short summary on user interfaces for microtask-based crowdsourcing.

A. Sense inventory induction

The problem of sense inventory induction implies generating a set of concepts or synsets from the text corpus, usually in an unsupervised way. This is achieved computationally by exploiting various properties of word similarity graphs [4] and co-occurence graphs [5]. Various genres of crowdsourcing are also used for addressing this task as in the cases of microtask-based lexical substitutions [6], Wiktionary-based lexical ontology construction [7], and video game-based approach [8]. Kiselev et al. proposed pairwise comparisons for detecting equivalent concepts [3]. However, the latter approach has been found to be difficult for crowd workers, who behaved no better then a heuristic.

B. Microtask interfaces

Worker interfaces for completing microtasks are usually implemented as Web-based interfaces, which were required by early crowdsourcing platforms like Amazon MTurk [9]. The rise of crowdsourcing happened simultaneously with the tremendous adoption of smartphones, which have become available in the developing countries. There, in contrast, a huge attention has been paid to various mobile interfaces for bridging the gap between the crowdsourcing markets and the online labor. The approaches include MTurk task proxying [10], providing a dedicated user interface [11], annotating via SMS [12] or an instant messaging bot [13], and other approaches [14]. Another trending topic is the evaluating the worker interfaces for designing the better ones in such aspects as behaviour-based performance evaluation [15], user attention focusing [16], and accessibility issues for those with special needs [17].

III. Approach

We consider the set of synsets S containing sense duplicates. Our goal is to replace these duplicates with the better composed fine-grained synsets. We propose to perform synset grouping by two common words forming groups (clusters) of synsets G. Having these groups with two common words emitted, we provide each group with the set of usage examples

ISBN 978-952-68397-6-9 (paperback), ISBN 978-952-68397-7-6 (PDF)

obtained from text corpus containing one of these two words per group. Then, we form a set of microtasks to be annotated by the human workers on a crowdsourcing platform.

In our approach, each microtask is a lexical substitution task accomplishing which is possible through checking whether the present words extracted from the synsets in the groups can fully substitute the highlighted word in the present sentence. Our approach has been inspired by the distributional hypothesis, the general idea behind which is the correlation between distributional similarity and meaning similarity [18].

The task layout we use is shown at Fig. 1. In this sense, our approach is similar to the one used for creating TWSI [6], however we do not ask the workers to manually enter the words and use the existing synsets as the input.

Choose all the words which fit the meaning of the sentence.

"A sentence with the highlighted word."

[ ] w_1
[ ] ...
[ ] w_n

Fig. 1. Layout of the crowdsourcing task

A set of the lexical substitution tasks is submitted for human annotation with the overlap of at least five answers per question (checkbox), which is recommended by similar studies on binary tasks [19]. When the annotation process is finished, the answers are aggregated and then for each sentence one synset has been created. Since that it has found to be sufficient, it is now possible to merge the new synsets representing the same concepts by the three common words heuristic [3].

IV. Implementation

Although the approach proposed in Section III can be executed on any crowdsourcing platform, our goal is to invite volunteers for annotating our data rather than the paid crowd workers in the further studies.

A. Architecture

To achieve such a goal, we built a human-machine system depicted at Fig. 2. Here, we expect volunteer workers to use their mobile devices, e.g., smartphones and tablets, to complete the microtasks assigned by the crowdsourcing server hosted and managed by us. Tasks, answers, and worker identifiers are stored in the database. In order to maintain personal data security, the only identifier the worker device shares with our server is the device identifier, which is a unique hexadecimal string.

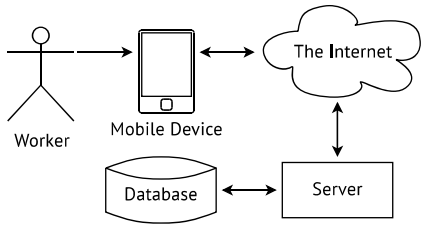

Fig. 2. Architecture of the proposed human-machine system

B. Application

For implementing the mobile application we have chosen the Android operating system as the target platform for now as it accounts 80.7 % of the worldwide smartphone marker according to Gartner [20]. We use Mechanical Tsar, an open source crowdsourcing server, to manage the workers, allocate the microtasks, receive and aggregate the answers [21]. The application operates as a client for the Mechanical Tsar RESTful API that implicitly registers the device identifier on the server, then requests the task, and submits the answer by the worker command. A screenshot of its graphical interface is shown at Fig. 3 demonstrating an allocated task represented with a sentence with the highlighted word, a set of substitution candidates, a button to skip the task, and a button to submit the answer. Additionally, the Internet connection between the application and the server is secured by HTTPS. The source code of the application is open [22], licensed under the Apache License 2.0, and the required Android version is 4.1 or higher.

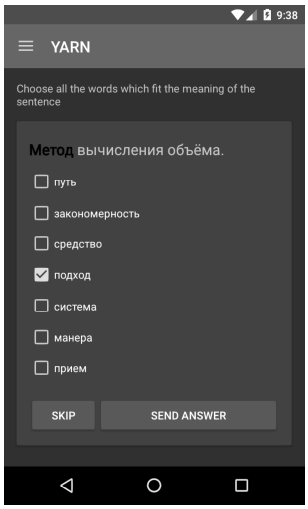

Fig. 3. A screenshot of the mobile microtask-based crowdsourcing application representing the sentence "A method for volume calculation." with the highlighted word "method" provided with the candidate substitutions "way", "regularity", "tool", "approach", "system", "manner", and "technique"

V. Experiments

We conducted two experiments: one pilot experiment on the TurboText microtask platform to study user experience of the application [23] and one massive annotation experiment on Yandex.Toloka to study the robustness of the approach [24].

A. Data preparation

As the target sense inventory we use Yet Another RussNet [25], which is a project aimed at creating a large open electronic thesaurus using both crowdsourcing and automatic methods. We use the word usage examples present in the Russian Wiktionary [7], which were obtained from openly available text corpora and separated by the individual word senses. Then, we use Algorithm 1 to rearrange the synsets S into the groups G, combine their lexical entries into these groups and then produce the microtasks as presented at Fig. 1.

Algorithm 1 Grouping$(S) \to G$

$W \leftarrow \emptyset$ {words mapping to sets of synsets}
for all $s \in S$ **do**
  for all $w \in s$ **do**
    if $w \notin W$ **then**
      $W[w] \leftarrow \emptyset$
    end if
    $W[w] \leftarrow W[w] \cup \{s\}$
  end for
end for
$G \leftarrow \emptyset$ {sets of words mapping to sets of synsets}
for all $s_1 \in S$ **do**
  for all $s_2 \in \bigcup_{w \in s_1} W[w]$ **do**
    $I \leftarrow s_1 \cap s_2$ {synset intersection}
    if $|I| \neq 2$ **then**
      continue
    end if
    if $I \notin G$ **then**
      $G[I] \leftarrow \emptyset$
    end if
    $G[I] \leftarrow G[I] \cup \{s_1, s_2\}$
  end for
end for
return G

Given a set of synsets, Algorithm 1 produces clusters of synsets having exactly two words in common. First, iterating over the given synsets and their words, for each encountered word it collects the synsets containing this word. Then, using this information, for each synset it just finds the synsets having at least one common word with it and filters those having exactly two words in common with it into clusters marked by the two common words. The synsets inside a cluster can be treated as suspicious for duplication and are then used for task generation. The running time of the algorithm is $O(|S| \cdot k \cdot m^2)$, where $k = \max_{w' \in \{w : s \in S \wedge w \in s\}} \left|\{s : s \in S \wedge w' \in s\}\right|$, $m = \max_{s \in S} |s|$.

As the result, we obtained 17 348 groups uniting 27 137 synsets. For this study, we have randomly chosen a subset of 424 groups uniting 875 synsets. Then, for each group we assigned the usage examples corresponging to each lexical sense represented in the Wiktionary, one sentence per sense, one sense per group (Fig. 4).

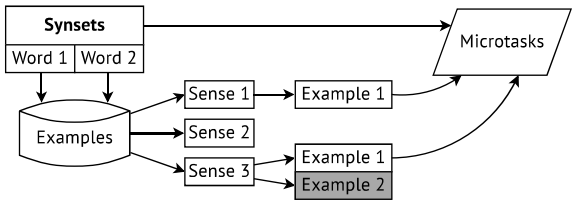

Fig. 4. Approach for microtask generation

Therefore, for the experiments we ran 205 microtasks on TurboText and 2408 microtasks on Toloka. It should be noted that when a group contained more than seven words, we split it into subgroups independently annotated as separate microtasks. These subgroups are identified by the common tag and then integrated back when processing the results.

B. Evaluation metric

As the gold standard for our task in Russian is not available under the compatible license [25], we decided to ask an expert to evaluate the quality of our synsets according to the following ordinal scale [26]:

"0" the synset does not make any sense, e.g., {sky, watch, gun},

"1" the synset has extraneous irrelevant lexemes along with the correctly established synonyms, e.g., {road, path, route, asphalt},

"2" the synset is composed of synonyms, but has missing the relevant words, e.g., {automobile, auto, motorcar} missing "car", and, possibly, "machine",

"3" the synset is good in general, e.g., {calamity, catastrophe, disaster, tragedy, cataclysm}.

In order to study whether the workers are confused with the lexical sense represented in the sentence, the expert additionally flagged the irrelevant resulting synsets as *Confused* and the properly constructed ones as *Not Confused*.

C. Pilot experiment on TurboText

The goal of the pilot experiment was to study the user experience of the Android application and to receive feedback from the workers available on a popular copywriting-focused platform. Prior to the experiment on TurboText we published the application described in Section IV to Google Play [27]. The workers were asked to install the application to their compatible mobile devices and then complete all the 205 available microtasks as shown at Fig. 3. Thus, we managed to provide each microtask with the answers submitted by five different workers. As the result, we received $5 \times 205 = 1025$ answers, which cost us approximately $4.

D. Massive experiment on Toloka

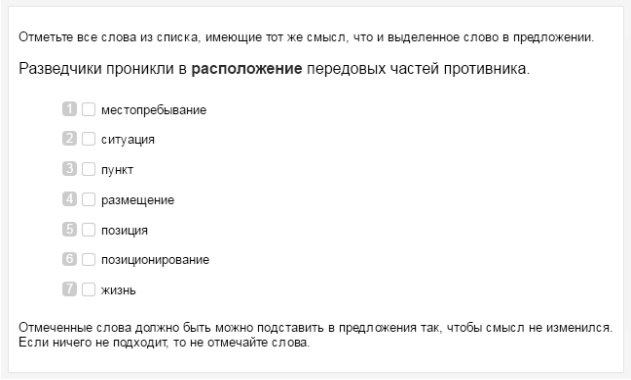

Fig. 5. A screenshot of the Toloka worker interface demonstrating the sentence "The scouts have infiltrated the location of the forward units of the enemy." provided with the candidate substitutions "residency", "situation", "point", "allocation", "position", "positioning", and "life"

The goal of the massive experiment was to study the robustness of the proposed approach and the performance of the workers available on Toloka during its private beta testing. Our lexical substitution task (Fig. 5) has attracted 291

TABLE I. GRADE DISTRIBUTION ON TURBOTEXT

Grade	Confused	Not Confused	Total
0	2	1	3
1	5	8	13
2	4	8	12
3	0	1	1
Total	11	18	29

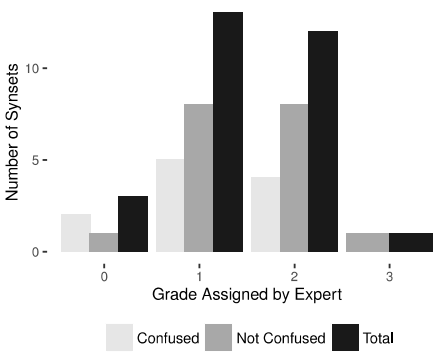

Fig. 6. Quality assessment of the experiment on TurboText

TABLE II. GRADE DISTRIBUTION ON TOLOKA

Grade	Confused	Not Confused	Total
0	41	30	71
1	38	44	82
2	84	112	196
3	3	17	20
Total	166	203	369

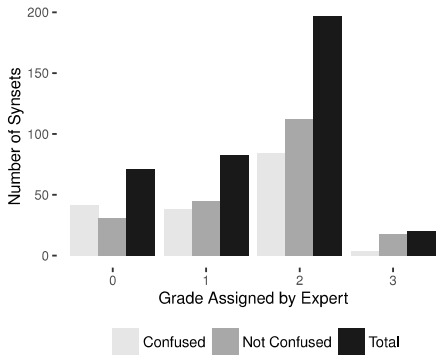

Fig. 7. Quality assessment of the experiment on Toloka

different workers, 251 of whose submitted 775 batches per 16 microtasks. As in the pilot experiment, we used the overlap of 5 workers per task. The time limit has been set to 30 minutes per batch. As the result, we received $5 \times 2480 = 12\,400$ answers within a little over an hour (1 hour 13 minutes), which cost us approximately $12.

VI. RESULTS

We aggregated the annotation results using the majority voting heuristic and then ask an expert to assess the output as described in Section V.

A. Results for TurboText

Having the TurboText experiment completed, we obtained 29 synsets created using our mobile application (Table I). Despite the number of the resulting synsets is not large, we needed to study whether the system (Fig. 2) is working properly, because unlike Toloka, TurboText makes it possible to send private messages to the workers for collecting the feedback. Interestingly, most synsets created on TurboText received the grade "1".

B. Results for Toloka

Having the Toloka experiment completed on the initial 875 synsets, we obtained 369 synsets, most of which have the "2" grade assigned (Table II, Fig. 7). We also conducted a 4-sample χ^2 test for equality of proportions without continuity correction to study whether all the synsets have the same true proportion of the confused ones. As the p-value of $p = 0.0062 < 0.05$ implies, at least in one grade the true proportion is different. It is interpreted as the number of the confused synsets does neither proportionally increase nor reduce as the number of the synsets increases per grade.

VII. DISCUSSION

Here, we discuss some of the interesting findings obtained when analyzing the experimental results.

A. Error analysis

Surprisingly, we found that people often fail to recognize the lexical sense given the contextual usage example! We suppose it was caused by a cognitive bias when the workers do not see any word matching the highlighted one, they begin to select the words they think may form a synset irrelevantly to the context exemplified by the present sentence. We believe this problem can be solved using "gold" questions for training the workers.

We also studied the obtained data and found that every grade but "3" has its own typical pattern of mistake:

- "0" the words are picked randomly (otherwise we can not explain these results rationally so far),
- "1" the synsets contained hypernyms and hyponyms or have a close yet irrelevant word included (we observed the similar behaviour in our another study on improving the synsets [28]),
- "2" the synsets are too coarse due to lack of the relevant words among the available variants.

B. Answer aggregation

In this study, we used majority voting without any weighting to aggregate the multiple answers provided for each microtask. However, it is reasonable to compare the present results to the results aggregated using adaptive crowdsourcing algorithms implemented in Mechanical Tsar [21]. These algorithms model worker expertise and task difficulty, which hopefully will increase the result quality.

C. Application reception

During the TurboText study, we contacted each worker and personally asked them to give feedback for their user experience. All the workers were comfortable with the user interface and the pertinency of the given lexical substitutions task (which contradicts the performance evaluation). Some workers voluntarily have given our application a five-star rating on Google Play (we had not asked for it neither implicitly nor explicitly).

All the workers complained that the candidate words contained a significant proportion of rude words, but we treat this incident as the imperfectness of the input data. Having studied the data thoroughly, we found that the corresponding groups had the coarse word synonyms of "thing" and other abstract concepts.

Amusingly, both the workers and the used telemetry system reported that the application crashed when every task is completed by the worker and there is nothing to allocate. At the moment, this bug has been fixed [22].

D. Practical considerations

According to the evaluation results, there is a lot of room for improvements to the proposed approach. The main obstacles we faced were the fuzziness of the input data and the lack of the crowd worker training.

Nevertheless, the present approach makes it possible to significantly reduce the number of synsets to be considered, e.g., only 369 synsets out of the initial 875 left—it is about the half. This makes the quality control procedures easier for the humans.

We plan to extend and adopt this microtask-based approach for the further development of Yet Another RussNet, because the currently used wiki-based synset assembly workflow was actually the source of the duplicates problem [25].

VIII. Conclusion

In the present paper we proposed an approach for improving the sense inventory by aligning the synsets to the set of sentences using lexical substitutions and crowdsourcing. Although we conducted all the experiments focusing on Russian as the target language, we believe our study can easily be used for other languages and datasets.

We see several directions for future work:

- lowering the rate of confused synsets by introducing an additional automatic method or crowdsourcing task for matching the meaning of the produced synsets,
- providing the crowd workers with preliminarily annotated ("gold") questions to train them how to complete the task successfully,
- using post-hoc statistical quality control methods for weighting the judgements of the crowd workers,
- employing the crowd work to assist or even replace the expert when analyzing the experimental results,
- annotating the whole Yet Another RussNet electronic thesaurus to make it a better lexical resource,
- combining the present approach with the microtask-based approach for establishing the semantic relations between the induced synsets [29],
- exploiting word embeddings and other approaches for distributional semantics, especially those providing the disambiguated senses [30].

The deliverables of this study, including the source code of experiment analysis programs under the MIT license, as well as the input, intermediate and output data under the CC BY-SA 3.0 license, are available for download on http://ustalov.imm.uran.ru/pub/lexsub-ismw.tar.gz.

Acknowledgment

The reported study was funded by RFBR according to the research project no. 16-37-00354 mol_a "Adaptive Crowdsourcing Methods for Linguistic Resources". This work was supported by the Russian Foundation for the Humanities project no. 13-04-12020 "New Open Electronic Thesaurus for Russian" and project no. 16-04-12019 "RussNet and YARN thesauri integration". The authors are grateful to Andrew Krizhanovsky who provided the Russian Wiktionary dump in machine-readable format, to George Agapov who participated in preparing the experimental setup, and to Mikhail Mukhin for fruitful discussions on the present paper.

References

[1] K. Allan, *Concise Encyclopedia of Semantics*. Oxford, UK: Elsevier Science, 2009.

[2] C. Fellbaum, "Large-scale Lexicography in the Digital Age", *International Journal of Lexicography*, vol.27(4), Sep. 2014, pp. 378-395.

[3] Y. Kiselev, D. Ustalov, S. Porshnev, "Eliminating Fuzzy Duplicates in Crowdsourced Lexical Resources", *in Proceedings of the 8th Global WordNet Conference, GWC 2016*, Jan. 2016, pp. 161-167.

[4] P. Pantel, D. Lin, "Discovering Word Senses from Text", *in Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul. 2002, pp. 613-619.

[5] C. Biemann, "Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems", *in Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, Jun. 2006, pp. 73-80.

[6] C. Biemann, "Creating a system for lexical substitutions from scratch using crowdsourcing", *in Language Resources and Evaluation*, vol.47(1), Mar. 2012, pp. 97-122.

[7] A.A. Krizhanovsky, A.V. Smirnov, "An approach to automated construction of a general-purpose lexical ontology based on Wiktionary", *Journal of Computer and Systems Sciences International*, vol.52(2), Mar. 2013, pp. 215-225.

[8] D. Jurgens, R. Navigli, "It's All Fun and Games until Someone Annotates: Video Games with a Purpose for Linguistic Annotation", *Transactions of the Association for Computational Linguistics*, vol.2, Oct. 2014, pp. 449-464.

[9] Amazon Mechanical Turk - Welcome, Web: https://www.mturk.com/mturk/welcome.

[10] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, M. Corner, "mCrowd: A Platform for Mobile Crowdsourcing", *in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Nov. 2009, pp. 347-348.

[11] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, B. Hartmann, "MobileWorks: A Mobile Crowdsourcing Platform for Workers at the Bottom of the Pyramid", *in Human Computation: Papers from the 2011 AAAI Workshop*, Aug. 2011, pp. 121-123.

[12] A. Gupta, W. Thies, E. Cutrell, R. Balakrishnan, “mClerk: Enabling Mobile Crowdsourcing in Developing Regions”, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, May 2012, pp. 1843-1852.

[13] D. Ustalov, “Teleboyarin—Mechanized Labor for Telegram”, *in Proceedings of the AINL-ISMW FRUCT*, Nov. 2015, pp. 195-197.

[14] Y. Wang, X. Jia, Q. Jin, J. Ma, “Mobile crowdsourcing: framework, challenges, and solutions”, *Concurrency and Computation: Practice and Experience*, Feb. 2016.

[15] S. Komarov, K. Reinecke, K.Z. Gajos, “Crowdsourcing Performance Evaluations of User Interfaces”, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Apr. 2013, pp. 206-216.

[16] S. Rothwell, S. Carter, A. Elshenawy, D. Braga, “Job Complexity and User Attention in Crowdsourcing Microtasks”, *in Proceedings of the Crowdsourcing Breakthroughs for Language Technology Applications Workshop*, Nov. 2015, pp. 20-25.

[17] K. Zyskowski, M.R. Morris, J.P. Bigham, M.L. Gray, S.K. Kane, “Accessible Crowdwork?: Understanding the Value in and Challenge of Microtask Employment for People with Disabilities”, *in Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, Mar. 2015, pp. 1682-1693.

[18] M. Sahlgren, “The distributional hypothesis”, *Italian Journal of Linguistics*, vol.20(1), Oct. 2008, pp. 33-53.

[19] R. Snow, B. O’Connor, D. Jurafsky, A.Y. Ng, “Cheap and Fast—but is It Good?: Evaluating Non-expert Annotations for Natural Language Tasks”, *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Oct. 2008, pp. 254-263.

[20] Gartner Says Worldwide Smartphone Sales Grew 9.7 Percent in Fourth Quarter of 2015, Web: http://www.gartner.com/newsroom/id/3215217.

[21] D.A. Ustalov, “A Crowdsourcing Engine for Mechanized Labor”, *Proceedings of the Institute for System Programming*, vol.27(3), Jul. 2015, pp. 351-364.

[22] russianwordnet/yarn-android: Android client for Yet Another RussNet, Web: https://github.com/russianwordnet/yarn-android.

[23] TurboText, a convenient copywriting market, Web: http://www.turbotext.ru/.

[24] Yandex.Toloka, Web: https://toloka.yandex.com/.

[25] P. Braslavski, D. Ustalov, M. Mukhin, Y. Kiselev, “YARN: Spinning-in-Progress”, *in Proceedings of the 8th Global WordNet Conference, GWC 2016*, Jan. 2016, pp. 58-65.

[26] Y.A. Kiselev, S.V. Porshnev, M.Y. Mukhin, “Current Status of Russian Electronic Thesauri: Quality, Completeness and Availability”, *Programmnaya Ingeneria*, vol.6, Jun. 2015, pp. 34-40.

[27] YARN - Android Apps on Google Play, Web: https://play.google.com/store/apps/details?id=net.russianword.android.

[28] D. Ustalov, Y. Kiselev, “Add-Remove-Confirm: Crowdsourcing Synset Cleansing”, *in Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*, Oct. 2015, pp. 143-147.

[29] D. Ustalov, “Crowdsourcing Synset Relations with Genus-Species-Match”, *in Proceedings of the AINL-ISMW FRUCT*, Nov. 2015, pp. 118-124.

[30] S. Faralli, A. Panchenko, C. Biemann, S.P. Ponzetto, “Linked disambiguated distributional semantic setworks”, Proc. ISWC 2016, in press.