Eliminating Fuzzy Duplicates in Crowdsourced Lexical Resources

Yuri Kiselev
Yandex
Yekaterinburg, Russia
yurikiselev@yandex-team.ru

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

Sergey Porshnev
Ural Federal University
Yekaterinburg, Russia
s.v.porshnev@urfu.ru

Abstract

Collaboratively created lexical resources is a trending approach to creating high quality thesauri in a short time span at a remarkably low price. The key idea is to invite non-expert participants to express and share their knowledge with the aim of constructing a resource. However, this approach tends to be noisy and error-prone, thus making data cleansing a highly topical task to perform. In this paper, we study different techniques for synset deduplication including machine- and crowd-based ones. Eventually, we put forward an approach that can solve the deduplication problem fully automatically, with the quality comparable to the expert-based approach.

1 Introduction

A WordNet-like thesaurus is a dictionary of a special type that represents different semantic relations between *synsets*—sets of quasi-synonyms (Miller et al., 1990). It is a crucial resource for addressing such problems as word sense disambiguation, search query extension and many other problems in the fields of natural language processing (NLP) and artificial intelligence (AI). Typical semantic relations represented by thesauri are synonymy, antonomy (primarily for nouns and adjectives), troponymy (for verbs), hypo-/hypernymic relations, and meronymy.

A good linguistic resource should not contain duplicated lexical senses, because duplicates violate the data integrity and complicate addition of semantic relations to the resource. Therefore, removing duplicated synsets from thesauri is an important problem to be addressed, especially in collaboratively created lexical resources like Wiktionary, which is known to suffer this problem (Kiselev et al., 2015). However, deduplication is rather problematic because thesauri may contain fuzzy duplicated synsets composed of different words.

The work, as described in this paper, makes the following contributions: (1) it proposes an automatic approach to synset deduplication, (2) presents a synonymic dictionary-based technique for assessing synset quality, and (3) compares the proposed approach with the crowdsourcing-based one.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 defines the problem of synset duplicates existing in thesauri. Section 4 presents a novel approach to synset deduplication. Section 5 describes the experimental setup. Section 6 shows the obtained results. Section 7 discusses the interesting findings. Section 8 concludes the paper and defines directions for future work.

2 Related Work

One of the most straightforward ways to clear a thesaurus of sense duplicates is to align its entries with another resource of proven quality, e.g. using the OntoClean methodology proposed by Guarino and Welty (2009). Consequently, synsets that will be linked with one synset from another resource represent the same concepts, and should be merged. However, such alignment can be performed only manually. It is also a time-consuming process that requires careful examination of every synset by an expert. Therefore, it is crucial to focus on methods that are either automatic or involve lesser amount of human intervention.

Many studies nowadays aim to evaluate the feasibility of crowdsourcing for various NLP problems. For instance, Snow et al. (2008) showed that non-expert annotators can produce the data whose quality may compete with the expert annotation in such tasks as word sense disambiguation and word similarity estimation (they conducted their study using Amazon Mechanical Turk[1] (AMT), a popular online labor marketplace).

Sagot and Fišer (2012) assumed that semantically related words tend to co-occur in texts. Given such an assumption, they managed to find and eliminate the words that had been added to synsets by mistake. This approach can be used to find sense duplicates, but it requires a large amount of semantic relations to be present in a resource. It should be noted that some resources that contain synsets may not contain any links between them. For instance, Wiktionary represents certain words and relations between them, but it does not explicitly link its synsets.

Sajous et al. (2013) presented a method for semi-automatic enrichment of the Wiktionary-derived synsets. First, they analyzed the contents of Wiktionary and produced new synonymy relations that had not been previously included in the resource. After that, they invited collaborators to manually process the data using a custom Firefox plugin to add missing synonyms to the data.

A similar approach was used by Braslavski et al. (2014) to bootstrap YARN (Yet Another RussNet) project, which aims at creating a large open WordNet-like machine-readable thesaurus for the Russian language by means of crowdsourcing. In this project, a dedicated collaborative synset editing tool was used by the annotators to construct synsets by adding and removing words.

The most recognized crowdsourcing workflow is the Find-Fix-Verify pattern proposed by Bernstein et al. and used in Soylent, a Microsoft Word plugin that submits human intelligence tasks to AMT for rephrasing and improving the original text (Bernstein et al., 2010). As the name implies, the workflow includes the three stages: 1) in the *Find* stage crowd workers find the text area that can be shortened without changing the meaning, 2) in the *Fix* stage the workers propose improvements for these text areas, and 3) in the *Verify* stage the workers select the *worst* proposed fixes.

Inspired by this pattern, Ustalov and Kiselev (2015) presented the Add-Remove-Confirm workflow for improving synset quality. Similarly, it contains three stages: 1) in the *Add* stage workers choose the words to be added to a synset from a given list of candidates, 2) in the *Remove* stage the workers choose the words that should be removed from a synset, 3) in the *Confirm* stage the workers choose which synset is better—the initial one or the fixed one.

3 Problem

In our study, we focus on the synsets represented in a WordNet-like thesaurus. Hence, we regard a thesaurus as a set of synsets S, where every synset $s \in S$ consists of different words and represents some sense or concept.

In lexical resources created by expert lexicographers, synsets usually correspond to different meanings, so synset duplicates never arise. Unfortunately, it is not true for the resources created by non-expert users, e.g. through the use of crowdsourcing. One approach to synset creation would be to combine manually constructed synsets with synsets that are imported from open resources. Obviously, it is going to lead to the situation where there is a plenty of synsets representing identical concepts. The crowdsourcing approach to synset creation is also prone to this drawback, as the crowd is likely to create duplicate synsets.

The following example from the Russian Wiktionary[2] shows that it contains synsets with identical meanings. For example, the synset {стоматолог (*stomatologist*), дантист (*dentist*), зубной врач (*"tooth doctor"*)} and the synset {дантист (*dentist*), стоматолог (*stomatologist*)} definitely describe the same concept "a person qualified to treat the diseases and conditions that affect the teeth". Hence, such synsets should be combined, yet they both are present in the Russian Wiktionary. Note that in this example the second synset is a full subset of the first one; however, it is possible that two synsets may intersect only partly while sharing the same meaning.

For a native speaker, it is relatively easy to detect whether two synsets share the same meanings. So, the detection may be done by non-experts via crowdsourcing. However, the key problem here is how to retrieve the pairs of synsets that presumably represent identical concepts. In the next sec-

[1] https://www.mturk.com/mturk/welcome

[2] https://ru.wiktionary.org/

tion, we propose a simple, yet effective approach.

4 Approach

Suppose the word w has several meanings. According to Miller et al. (1990), it is usually enough to provide one synonym for every meaning of w to a native speaker of a language to be able to distinguish the meanings from each other (provided that the speaker is familiar with the corresponding concepts). This phenomenon is widely exploited by explanatory dictionaries. It is also utilized in some thesauri which assume that a synset itself is enough to deduce its meaning, therefore definitions of synsets may be omitted.

Hence, we formulate the meaning deduplication problem as follows. Given a pair of different synsets $s_1 \in S$ and $s_2 \in S$, we treat them as *duplicates* if they share exactly two words:

$$\exists s_1 \in S, s_2 \in S : s_1 \neq s_2 \wedge |s1 \cap s2| = 2.$$

Obviously, this is a strong criterion that may be violated, so we propose the following two-stage workflow for synset deduplication.

Filtering. In this stage, the possible duplicates are retrieved using the above described criterion resulting in the set of synset pairs (s_1, s_2) for further validation.

Voting. In this stage, the obtained synset pairs are subject to manual verification. The pairs voted as equivalent are combined.

The assessment required in the Voting stage may be provided by expert lexicographers; in crowdsourced resources, the contributors may be invited not only to add the new data, but also to increase the quality of the created data and to deduplicate it.

5 Experiments

Since task submission to Amazon Mechanical Turk requires a U.S. billing address, this solution is not accessible to users from other countries. Although there are many other crowdsourcing platforms, e.g. CrowdFlower, Microworkers, Prolific Academic, etc., yet the proportion of Russian speakers on such platforms is still low (Pavlick et al., 2014).

Given the fact that our workers are native Russian speakers, we decided to use the open source crowdsourcing engine Mechanical Tsar[3], which is designed for rapid deployment of mechanized labor workflows (Ustalov, 2015). Inspired by the similar annotation study conducted by Snow et al. (2008), we used the default configuration, i.e. the majority voting strategy for answer aggregation, the fixed answer number per task strategy for task allocation, and the no worker ranking. The workers were invited from VK, Facebook and Twitter via a short-term open call for participation posted by us.

5.1 Stage "Filtering"

We used two different electronic thesauri for the experiments. The first one was chosen from among crowdsourced lexical resouces. Selecting between the Russian Wiktionary and YARN, we settled on the latter because it comprises one and half time more synsets, and it is easier to parse because YARN[4] synsets are available in the CSV format.

We were also interested in applying the described approach to a resource created by expert lexicographers. The current situation with electronic thesauri for the Russian language is that there is only one resource that is large enough and is available for study. This resource is RuThes-lite[5], a publicly available version of the RuThes linguistic ontology, which has been developing for many years (Loukachevitch, 2011).

We retrieved 210 presumably duplicated synsets from each resource—70 synsets with exactly two common words, 70 synsets with three, and 70 synset with four or more common words. Such a stratification is motivated by the interest in analyzing how the number of shared words correlates with their meanings.

By randomly sampling pairs of possibly duplicated synsets from YARN, we concluded that the proposed criterion for synset equivalence is very robust. It appears that for YARN this approach may be used even without the Voting stage. Thus, we decided to study whether the manual annotation does increase the quality of synset deduplication. In order to do this, we selected synsets from YARN as follows.

Since synsets in YARN are not always accompanied by sense definitions, we asked an expert to

[3] http://mtsar.nlpub.org/

[4] http://russianword.net/yarn-synsets.csv

[5] http://www.labinform.ru/pub/ruthes/

manually align the selected synsets with an expert-built lexical resource. We chose the Babenko dictionary (2011) (hereinafter referred to as BAB) as an expert-built lexical resource because it is a relatively recent dictionary with a wide language coverage. As a result of the alignment, each YARN synset s was provided with a corresponding synset s_{BAB} defined by a sense definition d.

5.2 Stage "Voting"

The goal of the Voting stage is to choose true equivalents among the prepared presumably equivalent synset. The input of this stage is a pair of synsets (s_1, s_2) from a resource, and a worker is to determine if the synsets share the same meaning (Figure 1).

Do the following synsets have the same meanings: "s_1" and "s_2"?
[ ] Yes
[ ] No

Figure 1: Task format for Voting stage (the original text was in Russian).

6 Results

6.1 Quality metrics

We use precision and recall to measure the quality of synsets in a thesaurus S. Precision $P(s)$ of a synset $s \in S$ is the fraction of the synset words with the meaning represented by s, compared to all the words in the language representing the meaning of the synset $\mathcal{L}(s)$.

$$P(s) = \frac{|s \cap \mathcal{L}(s)|}{|s|} \qquad (1)$$

Recall $R(s)$ of a synset s is the fraction of all words S in the language that have the meaning that s represents.

$$R(s) = \frac{|s \cap \mathcal{L}(s)|}{|\mathcal{L}(s)|} \qquad (2)$$

As may be easily noticed, it is impossible to precisely calculate the measure of synset recall $R(s)$, since the whole set of words that can correspond to a particular meaning is unknown. In order to estimate $\mathcal{L}(\cdot)$, we used the data retrieved at the Filtering stage. We combined the YARN synsets in each pair (s_1, s_2) into a new synset s. Then, we provided the resulting synset s with a corresponding definition d from the BAB and asked the same expert as in the Filtering stage to remove words from s, which do not correspond to the definition d. The fixed synsets s' were then combined with the corresponding synsets s_{BAB}. These combined synsets were used as the gold standard synsets s_{GS} for concepts, as we considered that such synsets contained all the words representing the concepts.

6.2 Example of Quality Calculation

Consider the following example in order to better understand the described process of data preparation and the further evaluations. Let say that YARN contains synset s_1=\{*think*, *opine*, *suppose*, *sleep*\} and synset s_2=\{*think*, *suppose*, *reckon*\}, and BAB contains synset s_{BAB}=\{*think*, *opine*, *suppose*, *imagine*\} with definition d "expect, believe, or suppose" ($|s_1 \cap s_2| = |\{think, suppose\}| = 2$ and $|s_1 \cap s_{BAB}| = |\{think, opine, suppose\}| = 3$). Assume that the expert aligned s_1 and s_{BAB} in the Filtering stage. In that case the expert would be provided with synset $s = s_1 \cup s_2$=\{*think*, *opine*, *suppose*, *sleep*, *reckon*\} and definition d from BAB. After fixing this synset s (by removing the wrong word *sleep*), it will be combined with the corresponding synset s_{BAB}. So the synset that will be further treated as the gold standard for this concept is s_{GS}=\{*think*, *opine*, *suppose*, *imagine*, *reckon*\}. This set will be used as $\mathcal{L}$ for calculating (1) and (2) (for the corresponding s_1 and s_{BAB}, $\mathcal{L}(s_1) = \mathcal{L}(s_{BAB})$). According to this,

$$P(s_1) = \frac{|s_1 \cap \mathcal{L}(s_1)|}{|s_1|} = \frac{3}{4} = 0.75,$$

$$R(s_{BAB}) = \frac{|s_{BAB} \cap \mathcal{L}(s_{BAB})|}{|\mathcal{L}(s_{BAB})|} = \frac{4}{5} = 0.8.$$

Note that in the proposed evaluation method, precision P of any synset from BAB s_{BAB} is 1.0.

6.3 Quality Assessment

The procedure described in Section 6.1 allowed us to calculate the suggested quality measures for the resources (Table 1). The *BAB* row is calculated for 210 synsets from the Babenko dictionary, the YARN, *aligned* row—for 210 synsets s_1 from YARN that were aligned with the BAB by the expert, and the YARN, *machine*—for the automatically merged all 210 presumably equivalent synsets (s_1, s_2) of YARN.

The F_1-measure for YARN is expectedly lower than for the BAB, yet, after a simple merging of

Table 1: Synset quality.

	Avg P	Avg R	Avg F_1
BAB	1.000	0.661	0.796
YARN, *aligned*	0.901	0.634	0.744
YARN, *machine*	0.840	0.774	0.805

the presumably equivalent synsets, its average F_1-measure became higher than for the BAB. However, this result was due to the significant increase in the recall, while the precision dropped.

To investigate how people's participation can improve the quality of automatic merging, we conducted a crowdsourcing experiment. Every task (Figure 1) was annotated by at least three different workers. The decision about merging was made by majority voting. Table 2 shows the share of synsets that the workers decided to merge.

Table 2: Crowdsourcing synset deduplication.

# of common words	2	3	4+
YARN	${}^{61}/_{70}$	${}^{64}/_{70}$	${}^{68}/_{70}$
RuThes-lite	${}^{25}/_{70}$	${}^{40}/_{70}$	${}^{51}/_{70}$

Quite expectedly, the two analyzed lexical resources proved very different. Our equivalence criterion worked only in one third of the cases for RuThes-lite. And even the stronger version of the criterion (the one considering synsets that share 4+ words as sense duplicates) was true only in $\frac{2}{3}$ cases according to the annotators. However, for YARN the criterion proved to be rather robust, so that it can be applied without crowd checking, provided that the results of the merging will be verified by a moderator of the resource.

This conclusion agreed with the quality estimates of the merging performed according to human annotations (Table 3). The first row (YARN, *machine*) corresponds to the automatic merge of all 210 synsets repeats the row of Table 1 with the same name, and the second row (YARN, *crowd*) corresponds to the selective merge performed according to the human judgements. So, $61+64+68$ synset pairs (s_1, s_2) were merged (Table 2), and the 17 remained synsets we left as they were (s_1).

Table 3: YARN synset deduplication.

	Avg P	Avg R	Avg F_1
YARN, *machine*	0.840	0.774	0.805
YARN, *crowd*	0.852	0.764	0.805

7 Discussion

The F_1-measure shows no change after applying the Voting stage, yet the precision increases by 0.012 while the recall drops by 0.01. Despite the fact that the overall quality is constant regardless of the human annotations, it still presents an interesting finding, since people increase the precision of the merging. This is important because it allows to compensate, at least partially, for the reduction in the precision against the original synsets caused by the automatic merge. (Table 3).

It is also of interest that YARN contains 24.8 thousand synsets that presumably have a duplicate (58% of the synsets with two or more words), while the Russian Wiktionary has 13.2 thousand (40%), and RuThes-lite has only 6.3 thousand (28%). We may therefore conclude that the proposed approach should mainly be applied to resources that a priori are known to contain duplicate synsets rather than to improve the quality of expert-built resources.

7.1 Synset Ambiguity

The analysis of the results of the experiments and the annotations provided by our expert showed that in some cases it is almost impossible to derive a meaning from a synset. For instance, just a couple of synonyms is not enough to distinguish the meaning "a woman thought to have *evil* magic powers" from "a woman who uses magic or sorcery" (the latter definition does not imply an "*evil*" woman, which can be not obvious from a synonymy row).

Another example of such ambiguity are the concepts corresponding to "a bed *with* a back" and "a bed *without* a back". Given only a synset, it is barely possible to discern this shade of meaning and distinguish any of these two concepts from the more common one (simply "a bed"). With this observation in mind, we suggest that the authors of the wordnets for which the meanings of synsets are optional should take it into account and include definitions for vague concepts.

7.2 Pairwise Annotation

Special attention should be given to the performance of the crowd workers. In our experiment, 25 workers provided 1262 answers to 420 pairwise comparison tasks (Figure 1). The workers repeatedly reported that the tasks were time consuming due to data inconsistency. Suppose that synset sizes are n_1 and n_2 correspondingly, and an annotator spends $O(n_1 + n_2)$ time to make a decision. Hence, even in the simplest case (Table 4) an annotator will perform $4 + 4 = 8$ operations per pair, which is inconvenient.

Table 4: Average synset sizes.

# of common words	**2**	**3**	**4+**
YARN	4.2	4.6	5.5
RuThes-lite	4.3	5.0	5.8

Further studies should avoid pairwise comparison in problems involving contextual or domain knowledge for making a decision by annotators. However, it still may be useful in various visual recognition tasks, especially when the workers are provided with an observable hint (Deng et al., 2013). We should also note that this outcome agrees well with the study conducted by Wang et al. (2012), when cluster-based task generation led to lower time spent rather than in pair-based tasks.

7.3 Agreement & Issues

We have analyzed all the cases when all the three workers gave the same answer to the task (Table 5). For YARN , the number of cases when all the workers agreed rises with the number of common words in synsets. This is quite expected considering that sharing more common words makes it more obvious that the synsets have common senses. However, we do not observe the same in RuThes-lite.

Table 5: # of merge decisions made unanimously.

# of common words	**2**	**3**	**4+**
YARN	$^{32}/_{70}$	$^{47}/_{70}$	$^{57}/_{70}$
RuThes-lite	$^{36}/_{70}$	$^{35}/_{70}$	$^{32}/_{70}$

Manual analyses of the data from RuThes-lite showed that its authors tend to discriminate meanings of synsets with common words by means of only one word, e.g. using a hyponym for a concept in one set and a corresponding hypernym in another. It is enough to emphasize the difference in meanings, but workers may find it problematic to detect the only pair of words that defines the difference in the pair of synsets. This task may become even more complicated in large synsets, as they grow in size along with the increase in the number of common words in them (Table 4).

8 Conclusion

In this study, we presented an automated approach to synset deduplication. The results were obtained from expert labels and annotations provided by crowd work. At least three different annotations per every synset pair from two different resources (YARN and RuThes-lite) were used. The approach allows to significantly increase the synset quality in crowdsourcing lexical resources. Participation of people does not notably affect the average synset quality, though the precision slightly increases when people are involved.

The results showed that two synonyms are not sufficient for defining a meaning, but three words usually give a satisfactory result. So, it is three words that should be used as a threshold value for merging duplicate synsets when using the proposed deduplication approach in a fully automatic mode. Our results, including the crowd answers and the produced gold standard, are available[6] under the terms of Creative Commons Attribution-ShareAlike 3.0 license.

As a possible future direction, we may suggest using more sophisticated similarity measures to select a threshold for fully automatic merging of synsets. Another possible way to improve the approach is to detect not just pairs, but clusters of synsets. This is hardly possible in resources that are manually crafted by a team of experts, but it is definitely worth exploring for crowdsourcing resources.

Acknowledgments

This work is supported by the Russian Foundation for the Humanities, Project No. 13-04-12020 “New Open Electronic Thesaurus for Russian”. The authors are grateful to Yulia Badryzlova for proofreading the text, and to Alisa Porshneva for labeling synsets. The authors would also like to thank all those who participated in the crowdsourced experiment.

[6] `http://ustalov.imm.uran.ru/pub/duplicates-gwc.tar.gz`

References

Ljudmila G. Babenko, editor. 2011. *Dictionary of synonyms of the Russian Language*. AST: Astrel, Moscow, Russia.

Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 313–322, New York, NY, USA. ACM.

Pavel Braslavski, Dmitry Ustalov, and Mikhail Yu. Mukhin. 2014. A Spinning Wheel for YARN: User Interface for a Crowdsourced Thesaurus. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 101–104, Gothenburg, Sweden. Association for Computational Linguistics.

Jia Deng, Jonathan Krause, and Li Fei-Fei. 2013. Fine-Grained Crowdsourcing for Fine-Grained Recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 580–587.

Nicola Guarino and Christopher A. Welty. 2009. An Overview of OntoClean. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 201–220. Springer Berlin Heidelberg.

Yuri Kiselev, Andrew Krizhanovsky, Pavel Braslavski, et al. 2015. Russian Lexicographic Landscape: a Tale of 12 Dictionaries. In *Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue"*, volume 1, pages 254–271. RGGU, Moscow.

Natalia V. Loukachevitch. 2011. *Thesauri in information retrieval tasks*. Moscow University Press, Moscow, Russia.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database. *Lexicography*, 3:235–244.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The Language Demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2:79–92.

Benoît Sagot and Darja Fišer. 2012. Cleaning noisy wordnets. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Franck Sajous, Emmanuel Navarro, Bruno Gaume, Laurent Prévot, and Yannick Chudy. 2013. Semi-Automatic Enrichment of Crowdsourced Synonymy Networks: The WISIGOTH System Applied to Wiktionary. *Language Resources and Evaluation*, 47(1):63–96.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. 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*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dmitry Ustalov and Yuri Kiselev. 2015. Add-Remove-Confirm: Crowdsourcing Synset Cleansing. In *Application of Information and Communication Technologies (AICT), 2015 IEEE 9th International Conference on*, pages 143–147. IEEE.

Dmitry Ustalov. 2015. A Crowdsourcing Engine for Mechanized Labor. *Proceedings of the Institute for System Programming*, 27(3):351–364.

Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *Proc. VLDB Endow.*, 5(11):1483–1494.