Crowdsourcing Ontology Lexicons

Bettina Lanser, Christina Unger, Philipp Cimiano
Semantic Computing Group, CITEC, Bielefeld University
Inspiration 1/ Zehlendorfer Damm 201, 33619 Bielefeld, Germany
{blanser, cunger, cimiano}@cit-ec.uni-bielefeld.de

Abstract

In order to make the growing amount of conceptual knowledge available through ontologies and datasets accessible to humans, NLP applications need access to information on how this knowledge can be verbalized in natural language. One way to provide this kind of information are ontology lexicons, which apart from the actual verbalizations in a given target language can provide further, rich linguistic information about them. Compiling such lexicons manually is a very time-consuming task and requires expertise both in Semantic Web technologies and lexicon engineering, as well as a very good knowledge of the target language at hand. In this paper we present an alternative approach to generating ontology lexicons by means of crowdsourcing: We use CrowdFlower to generate a small Japanese ontology lexicon for ten exemplary ontology elements from the DBpedia ontology according to a two-stage workflow, the main underlying idea of which is to turn the task of generating lexicon entries into a translation task; the starting point of this translation task is a manually created English lexicon for DBpedia. Comparison of the results to a manually created Japanese lexicon shows that the presented workflow is a viable option if an English seed lexicon is already available.

Keywords: Ontology lexicalization, crowdsourcing, DBpedia

1. Introduction

1.1. Motivation

As the amount of formalized conceptual knowledge available through ontologies and datasets such as DBpedia (Bizer et al., 2009) grows, there is an increasing need to make this knowledge accessible to humans in an easy and intuitive way. One way to accomplish this is by means of language technology, e.g. in the form of question answering systems, that allows users to access and query repositories of conceptual knowledge through natural language. Of course, systems of this kind do not rely solely on conceptual knowledge, but also need access to lexical information about how the elements described in such repositories may be verbalized in a given language.

Ontology languages support the inclusion of such information to a certain extent, e.g. by means of `rdfs:label` or SKOS properties. However, often the amount and depth of linguistic information offered this way is very limited. In case of DBpedia, for instance, the majority of individuals do not have any language label at all, and while most other ontology elements are assigned an English label, coverage for other languages is very restricted; for example, only around ten percent of DBpedia's classes and properties have a Japanese label, as shown in Table 1. Moreover, even when a label in the language of interest is provided, no further linguistic information is given, such as part-of-speech information, inflectional forms or subcategorization frames. In addition, labels only capture one canonical way of verbalizing an ontology element, but they do not provide lexical variants. For example, for the DBpedia property `spouse` the English label *spouse* is given, but variants such as *wife of*, *husband of* or *to marry* are not covered.

As a result, in order to make resources of conceptual information such as DBpedia accessible to language technology systems, usually further external resources of linguistic knowledge are necessary. One type of resource specifically designed for this task are ontology lexicons (Prévot et al., 2010; McCrae et al., 2011b), which connect ontology elements to possible verbalizations in a given language enriched with various kinds of linguistic information. However, generating ontology lexicons manually is a very time-consuming task and requires expertise in Semantic Web languages and lexicon engineering, as well as knowledge about the domain of the ontology. Furthermore, in order to decide which verbalizations for a given ontology element are appropriate, often the language proficiency of native speakers will be necessary; hence, one either needs to have a very good command of the target language oneself, preferably at native speaker level, or one should at least be able to consult with native speakers, which in case of smaller target languages may pose a problem. Alternatively, ontology lexicons could be induced automatically (Walter et al., 2014) or generated by means of translating an already existing lexicon (McCrae et al., 2011a; Arcan and Buitelaar, 2013); however, those methods have not yet

Language	Individuals	Classes, properties	Total
English	37.72%	99.97%	37.78%
French	9.16%	21.13%	9.17%
German	7.87%	57.63%	7.91%
Italian	7.42%	6.84%	7.42%
Dutch	7.12%	35.62%	7.14%
Spanish	7.00%	7.23%	7.00%
Polish	6.57%	3.07%	6.57%
Portuguese	5.84%	6.22%	5.84%
Russian	5.63%	0.53%	5.62%
Chinese	4.18%	0.31%	4.18%
Japanese	3.69%	10.44%	3.69%
Arabic	2.05%	0.11%	2.05%

Table 1: Percentages of ontology elements for which respective language label is available within the DBpedia ontology[2]

[2]For the sake of brevity, only languages for which labels on individuals are available are given.

reached an accuracy sufficient to produce high-quality lexicons off the shelf.

In this paper we explore a further option, namely making use of crowdsourcing (Howe, 2006; Quinn and Bederson, 2011), which in recent years has already been used for a number of different tasks related both to natural language processing and ontologies (Snow et al., 2008; Ambati and Vogel, 2010; Acosta et al., 2013). To our knowledge, so far there exist no reports on using crowdsourcing specifically for the generation of ontology lexicons, and hence whether ontology lexicons of good quality can be generated this way at acceptable costs is an open question. In the following, we will present an approach to generating a Japanese ontology lexicon for DBpedia by means of crowdsourcing, which can also be applied both to other languages and other ontologies.

2. Methodology

2.1. Overall Workflow

A particular challenge when trying to crowdsource ontology lexicons is finding a task design that is suitable for workers with no knowledge about ontologies or lexical resources: Obviously, it would not make much sense to simply present the workers with an ontology element and then asking them to come up with a good verbalization, or even a whole lexicon entry. Therefore, we instead turn the lexicon generation task into a translation task. The starting point is a seed lexicon in English, in our case a manually created English lexicon for DBpedia (Unger et al., 2013) with over a thousand entries. We ask Japanese crowdsourcing workers to provide Japanese translations of the English verbalizations, with each Japanese translation being understood as a potential verbalization of the ontology element linked to the original English verbalization. As an example, let us assume we are looking for a Japanese verbalization of the property `author`, which in the DBpedia ontology links the classes `Writer` and `Book`.We would search the seed lexicon for entries which reference this property, and among others, we would find an entry containing the verbalization *to write*. Our strategy would then be to ask the crowdsourcing workers for a Japanese translation of the verb *to write*, and each such translation would be treated as a candidate verbalization of `author` for our Japanese ontology lexicon.

There may be English seed verbalizations with more than one possible meaning, or which could be used to verbalize more than one ontology element. For example, *to write* cannot only be used to verbalize the relationship holding between an author and a book, but may link authors to all kinds of written work, and there may be target languages in which these different kinds of relationships are verbalized in different ways. Therefore, we need to ensure that the English verbalizations are understood in the right sense, which can be accomplished by presenting them to the workers embedded in some kind of context. We decided to present the English verbalizations within short sentences, which we automatically generate using the Lemonade tool (Rico and Unger, 2015). Each such sentence is built on the one hand from the English seed verbalization currently looked at and on the other hand from a triple found in the DBpedia dataset that contains the associated ontology element. The way the dataset gets searched for a suitable triple depends on the type of ontology element we are dealing with: When looking for a verbalization of a property, such as `author`, we would search the DBpedia dataset for a triple that contains the respective property as its predicate and include those triple's subject and object in our automatically generated sentence along with the English seed verbalization. Hence, if in our example we were to retrieve the triple `Don_Quixote author Miguel_de_Cervantes` from the DBpedia dataset, the generated sentence would have the form *Miguel de Cervantes wrote Don Quixote*. In contrast, if we were looking for a verbalization of a class, such as e.g. `Book`, we would search the dataset for a triple that links some individual to the respective class by means of the property `rdf:type`, and would use that individual in the generated sentence together with the respective English seed verbalization. Hence, if for the class `Book` we had picked a triple of the form `Don_Quixote rdf:type Book`, and the seed lexicon contained the verbalization *book* for the class `Book`, we would generate a sentence of the form *Don Quixote is a book*. While using real-life sentences from a corpus would also have been an option, we decided against this, as in many cases such sentences tend to be rather long and not always clearly disambiguate the verbalization. In contrast, as they always include entities that are conceptually linked to the ontology element the verbalization is meant to represent, our automatically generated sentences have a high chance of presenting verbalizations in an unambiguous way. Furthermore, the simple structure of these sentences not only makes the translation task easier for the crowdsourcing workers, but will probably allow us to eventually extract the Japanese verbalizations from the translated sentences automatically by means of the M-ATOLL framework (Walter et al., 2014).

One obvious challenge with a crowdsourcing task such as this one, where more than one correct answer may exist for a given piece of input data and there is no straightforward way to automatically check the validity of the answers provided by the workers, is quality control. We adopt an approach that is commonly used in translation-related crowdsourcing tasks (Zaidan and Callison-Burch, 2011; Benjamin and Radetzky, 2014) and which involves soliciting multiple translations per English sentence from distinct workers, plus a second crowdsourcing stage in which Japanese workers will be asked to evaluate the translations received in the first stage. Based on these evaluations we can then decide which translations most probably include commonly accepted Japanese verbalizations of ontology elements that should be included in a Japanese ontology lexicon for DBpedia. As an example, assume we have shown the sentence *Miguel de Cervantes wrote Don Quixote* to three separate workers during the translation stage. Worker number one translated the sentence as ミゲル・デ・セルバンテスはドン・キホーテを書いた, the other two only entered gibberish. Distinguishing between the useful answer and the gibberish ones automatically may be extremely difficult, or even impossible. However, during the following evaluation phase filtering out workers who don't

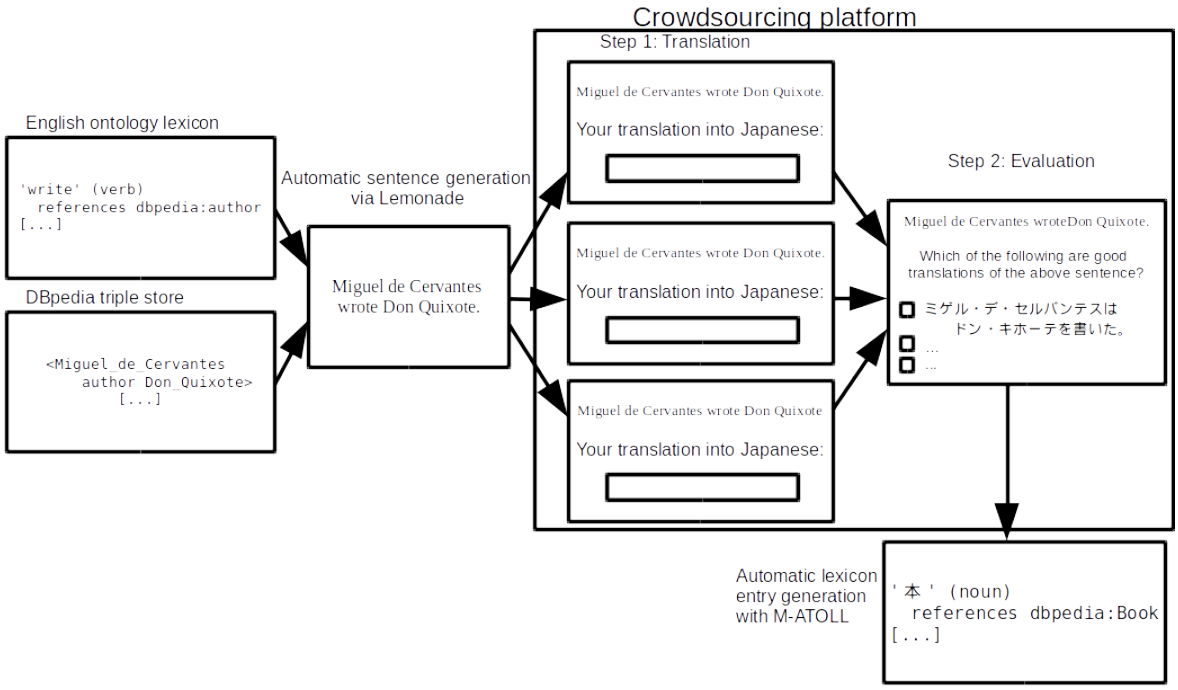

Figure 1: Workflow of our approach for the generation of a Japanese ontology lexicon

provide acceptable evaluations of the translations automatically would be much easier and could e.g. be done through test sentences for which we already know in advance which translations are correct and which are not. Therefore, if in the evaluation stage we received the information that the sentence provided by worker number one is a good translation, while the other two are not, we could be rather confident that this information is correct.

Based on the findings from the evaluation stage, we would then decide that the translation ミゲル・デ・セルバンテスはドン・キホーテを書いた most probably contains a valid Japanese verbalization of the property `author`. We would then have to extract this verbalization — the verb 書く — from the translation and turn it into a complete lexicon entry with further information, e.g. about its part of speech. As mentioned before, in the future it should be possible to do this automatically through M-ATOLL; at the time of writing, however, we still need to perform these steps manually. The overall workflow of our approach is shown in Figure 1.

With a translation-related task such as this one, one obvious question is what potential advantages crowdsourcing the task would have over simply hiring a professional translator (Zaidan and Callison-Burch, 2011). On the one hand, that approach would probably be very expensive, and one may assume that crowdsourcing the task will be considerably cheaper. On the other hand, with input from only one person, the variance of the received verbalizations would probably be lower than if possibly a lot of different people provide translations.

2.2. Choice of Crowdsourcing Platform

A lot of crowdsourcing-related research is carried out via Amazon's MechanicalTurk[3]. However, as on Mechanical Turk only workers who are based either in the US or India can be paid in cash[4], well over 90 percent of workers come from either of these countries[5], which would make finding a sufficient amount of workers with a native language other than English or an Indian language rather difficult. Therefore, MechanicalTurk was not a good option for us. We also looked at a number of Japan-based crowdsourcing platforms; however, many of these seemed not very suitable for the crowdsourcing tasks we had in mind, either. On the one hand, some of them are based on an idea of crowdsourcing quite different from the kind of task we wanted to carry out: On some platforms like Lancers[6], companies can find freelancers for rather complex tasks, like designing a corporate design for them, while other platforms like Coconala[7] focus on tasks such as personal consulting. Furthermore, there are a number of platforms where only very specific types of tasks can be carried out, such as e.g. video production on Viibar[8].

We finally chose to work with CrowdFlower[9], which is another global crowdsourcing platform similar to Mechani-

[3] https://www.mturk.com/

[4] https://www.mturk.com/mturk/help?helpPage=worker#how_paid

[5] http://demographics.mturk-tracker.com/#/countries/all

[6] http://www.lancers.jp/

[7] http://coconala.com/

[8] http://viibar.com/

[9] http://www.crowdflower.com/

calTurk. However, in contrast to the latter, CrowdFlower does not have any country-related restrictions on how workers can be paid, and so it seems likely that there is a higher diversity with respect to the countries of origin of the workers; this assumption seems to be backed by a survey done by CrowdFlower itself based on responses of workers to a questionaire, where only 30% in total indicated either the US or India as their country of origin. [10] Furthermore, CrowdFlower's worker interface is available in Japanese, and — again in contrast to Mechanical Turk — workers for a task can be chosen according to their geographical location and their language skills.

However, the platform also has a number of characteristics that are rather disadvantageous with regard to our specific task at hand: First of all, employers on CrowdFlower need to pay for every result submitted by a worker, no matter this result's quality. This is in contrast to what seems to be common practice on most other crowdsourcing platforms, where employers are able to look at and evaluate the received results and decide which of these they actually want to use and pay for. As an incentive to still provide quality work, workers at CrowdFlower are ranked by the platform according to their performance on test questions, which are interspersed among the actual task and which are pre-labeled with known answers provided by the employer. Employers can choose which minimum quality rank workers need to have in order to work on their task, and usually tasks offered only to workers of higher ranks pay better and are more interesting. Obviously, this kind of quality control only works for jobs where test questions can be formulated in the first place, i.e. where at least for certain input data a closed and predictable set of correct results exists. Hence, it is not suitable for translation tasks, as in most cases predicting all possible correct translations to a given seed sentence is simply impossible. We therefore had to make use of a number of alternative quality control mechanisms for the translation task, which will be described in the following.

2.3. Quality Control

As for a task such as ours one cannot really formulate test questions, which form CrowdFlower's main mechanism of quality control, we made use of a number of alternative control mechanisms that are commonly used with translation-related tasks (Irvine and Klementiev, 2010; Zaidan and Callison-Burch, 2011): Generally, a good strategy for discouraging people from cheating is to design one's task in a way that makes cheating as laborious and time-consuming as actually working on the job at hand. Therefore, we showed the English seed sentences — and, in case of the later evaluation stage, the Japanese candidate translations — to the workers in the form of images rather than text, so as to make it more difficult for people to make use of automatic translation services. This measure serves to prevent automatic translations, but it does not help against workers who either just insert gibberish or who sincerely try to work on the task but either did not understand the instructions or for some other reason produce incorrect results. Detecting faulty output of this kind is the main purpose of the second stage of our workflow, in which workers are shown an English seed sentence together with its Japanese candidate translations and are asked to judge the quality of the latter. As mentioned before, for this second kind of crowdsourcing task, one can actually formulate test questions and hence filter out workers who do not submit reliable evaluations automatically. More details about how test questions work on CrowdFlower and how we generated those questions for our task can be found in Section 3.3.

Finally, a further important measure of quality control concerns the choice of workers: CrowdFlower groups workers into three groups according to their previous performance on test questions, and we only allowed workers of the highest quality group to work on our tasks, which is what CrowdFlower advises for tasks one cannot formulate test questions for.[11] Furthermore, for the translation task we experimented with different settings for the country of origin and language skills of allowed workers, as will be described in more detail in Section 3.2.

3. Test Stage

3.1. Overview

In order to gain some first experiences and to check whether our approach is feasible, we first conducted a test stage in which only for a small number of ontology elements Japanese verbalizations should be found: We started with ten ontology elements, the types of which were chosen so as to roughly mirror the overall distribution of element types (classes, object properties, datatype properties) in the DBpedia ontology. Furthermore, for each element type we chose one element with many (≤ 4) and one with some (2–3) verbalizations in the seed lexicon, plus one or two elements with only one verbalization. The actual ontology elements and their verbalizations from the English seed lexicon can be found in Table 2.

3.2. Translation Stage

For each of the 25 verbalizations shown in Table 2 we automatically generated three example sentences in the manner described in Section 2.1., and for each such sentence we asked for translations by three separate workers, totalling $3 * 3 * 25 = 225$ data rows to be retrieved. As mentioned above, we experimented with different settings for the countries of origin and language skills of allowed workers, and also tried out whether the language the task instructions are given in has any effect on the workers' performance.

One of the challenges of crowdsourcing is finding the right amount of payment for the workers that on the one hand provides an incentive for people to work on the task but on the other hand does not make it too attractive for cheaters. As we were not sure what kind of payment would be appropriate, we always started at a payment of one cent per sentence, and slightly increased the payment every time no one had worked on the task for a longer period of time (at

[10] https://success.crowdflower.com/hc/en-us/articles/202703345-Crowd-Demographics

[11] https://success.crowdflower.com/hc/en-us/articles/201855969-Guide-To-Running-Surveys

Element type	URI	Verbalizations
Classes	PowerStation	generating station
		power station
		power plant
		generating plant
		electricity station
	Star	star
		sun
	Artist	artist
Object properties	parent	child
		father
		daughter
		son
		parent
		mother
	occupation	occupation
		to work
	colourName	color
Datatype properties	numberOfStudents	student population
		to have an enrollment of
		enrollment
		to serve
	weight	to weigh
		weight
	budget	budget
	yearOfConstruction	constructed

Table 2: Exemplary ontology elements and verbalizations used in the test stage

least 24 hours). In order to estimate the quality of the translations, we randomly picked five translations from each worker and looked at whether they contained obvious semantic or grammatical errors.

In our first run, we only required the allowed workers to have passed CrowdFlower's Japanese proficiency test, but did not impose any restrictions on the workers' country of origin. Furthermore, the task instructions were given in English, as workers should not only have a good understanding of Japanese but also at least basic English skills for this task in order to understand the English seed sentences. This run was finished very fast within only four hours, so we did not have to raise the payment of one cent per sentence. However, the overall results we retrieved in this run were of very poor quality, as there were many contributors who either only entered gibberish or gave grammatically incorrect word-by-word translations of the English sentences. A possible explanation would be that many people may cheat on CrowdFlower's language proficiency tests and pass them even though they do not speak the respective language at all: As mentioned before, the English sentences are presented to the workers as images, so for someone with a sufficient knowledge of both Japanese and English simply translating a sentence themselves should actually be less work than presumably entering it into some automated translation system by hand, or even looking up every single word on its own.

As a result, we did a second run of the translation task, which only differed from the first one in that now only workers from Japan were allowed to work on it. This time, results were significantly better, and there were no obvious word-by-word translations or workers who entered gibberish. However, at around one month this run also took much longer to finish. As can be seen in Table 3, while we already achieved a completion of nearly fifty percent at a payment of only one cent per sentence, in the end we had to raise payment to up to eight cent per sentence in order to also receive results for the last pending microtasks. It should be noted here that each worker was only allowed to provide translations for at most half of all English seed sentences; this setting was chosen so as to achieve higher variance, as otherwise it may have been possible for only three separate workers to complete the whole task. Working with a different setting here may of course have resulted in the task getting completed in a shorter time and at a lower maximum cost, as people who were satisfied with a lower payment per sentence may have worked on a larger number of sentences then.

Out of the three workers who delivered clearly low-quality results, two worked on the task only after payment per sentence had been raised to eight cent. While the low overall number of workers does not allow one to make any definite statements here, this may be seen as a sign that at around this amount of payment there is a threshold at which this kind of task also becomes attractive for cheaters.

We wanted to know if the language the task instructions are given in has any significant effect on the quality of the results; therefore, we started a third run which only differed from the secod one in that the instructions were now given in Japanese. However, we stopped this final run at around

Cent/sentence	Total number of workers	Workers with low-quality contributions	Tasks completed
1	4	0	48.89%
4	6	1	86.67%
6	6	1	88%
7	7	1	93.33%
8	10	3	100%

Table 3: Results from second run of translation stage

	First run	Second run
Language of instructions	Japanese	English
Total number of workers	46	29
... who passed test questions	10	8
Low-quality contributors	2	3
Duration	7 days	3 days
Maximum cent/sentence	1	1

Table 4: Basic data of the first and second run of the evaluation stage

72 percent completion, as it turned out that all workers who had contributed to this run so far had also contributed to the second one, delivering basically the same quality, and we assumed that also for the remaining microtasks the results would most probably not differ that much from those from the second run. In the following steps, we only considered the data from the second run.

3.3. Evaluation Stage

When we wanted to start the evaluation stage of our test run, it turned out CrowdFlower had removed Japan from the list of countries that can be used to filter which workers are allowed to work on one's task, and it had also deactivated the option to narrow down allowed workers to only those who speak Japanese. When we contacted CrowdFlower's support about this, we were told they had removed these options due to the small number of Japanese workers currently available through the site, and that they may get activated again some time in the future should CrowdFlower find a way to provide a larger Japanese workforce to the employers who use their platform. This change means that we may not be able to use CrowdFlower for future crowdsourcing experiments. However, we decided to at least try to finish our current test run on the platform.

In each microtask of this stage the workers were shown one English seed sentence together with the three translations we had received for it in the translation stage, and they were asked to mark all translations that they considered correct. For each seed sentence, we elicited evaluations of its translations by three separate workers; as a result, the number of microtasks in this stage was the same as in the preceding stage (225). In addition, we uploaded twenty test questions. In CrowdFlower these need to have the same structure as the actual microtasks; hence, each such test question consisted of an English sentence and three Japanese sentences for which we knew in advance which of them constituted correct translations of the English sentence and which did not, and that we pre-labeled accordingly. On the one hand, CrowdFlower uses these questions to test workers before the actual task starts in so-called quiz mode, where workers need to answer a certain amount of test questions — five in our case — before they can actually work on the task. On the other hand, also during the task itself a certain amount of the microtasks shown to the workers — in our setup twenty percent — are actually test questions. Workers need to answer a certain percentage of test questions correctly throughout the job — eighty percent in our case — or else CrowdFlower will keep them from working on further microtasks. We generated our test questions from English seed sentences we had already used in the translation stage. Each such sentence we combined with clearly correct translations from the preceding translation stage and/or randomly chosen translations of other sentences. Furthermore, for some test questions we added manual translations of our own in which we had included grammatical errors.

In total, we did two separate runs of the evaluation stage, one with Japanese instructions and one with English ones, to test again if the language of instructions has any effect on the received results. Some basic data for both runs are shown in Table 4. To check if the test questions are efficient at filtering out workers with low-quality contributions, we looked at a number of normal microtasks for which at least for some of the translations it is clear whether they are correct or not, and looked at how the workers who passed the test questions performed on these. In many respects, the two runs proceeded very similarly: Both took considerably shorter than the runs of the preceding translation stage, and both were also considerably cheaper. Also in both cases, a large number of people attempted to work on the task, but failed at the test qestions. Workers who feel that the test questions are incorrect or unfair can give feedback to the employer; in three cases we received feedback that lead us to deactivate the respective test questions, as at closer inspection of these we had the impression that the criticism was justified. However, the vast amount of workers who did not pass the test questions did not give any feedback at all, and the performance of many of these looked as if they had simply answered randomly. Hence, one insight from these test runs seems to be that a lot of people will try to work on tasks they are clearly not competent for, and in case of the first run whose instructions they do not even understand. Furthermore, it looks as if the test questions are efficient at filtering out people who do not provide acceptable results. Given the small overall amount of people who actually passed the test questions, it is difficult to make any definite statements about the differences between the two runs with respect to this group. For the second run the amount of people who passed the test is slightly lower, while the number of people out of this group who still delivered low-quality contributions is slightly higher.

3.4. Results

Prior to our test run we had manually compiled a Japanese gold standard lexicon for the ten exemplary ontology elements shown in Table 2. Table 5 shows the results of comparing this gold standard against a number of differ-

# Votes \ # Translations	≥ 1 (=all)			≥ 2			≥ 3		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-Score
≥ 0 (=all)	0.705	**0.77**	0.736	0.852	0.688	0.761	**1.0**	0.595	0.746
Votes from first run of evaluation stage (Japanese instructions)									
≥ 1	0.721	0.77	0.744	0.87	0.688	0.768	1.0	0.595	0.746
≥ 2	0.83	0.737	0.78	0.885	0.688	0.774	1.0	0.595	0.746
≥ 3	0.876	0.713	**0.786**	0.893	0.676	0.769	1.0	0.583	0.736
≥ 4	0.988	0.643	0.779	0.988	0.643	0.779	1.0	0.55	0.709
Votes from second run of evaluation stage (English instructions)									
≥ 1	0.793	0.77	0.781	0.861	0.688	0.764	1.0	0.595	0.746
≥ 2	0.815	0.691	0.747	0.885	0.643	0.744	1.0	0.562	0.719
≥ 3	0.88	0.68	0.767	0.938	0.643	0.762	1.0	0.562	0.719
≥ 4	0.988	0.609	0.753	0.988	0.609	0.753	1.0	0.562	0.719

Table 5: Precision, recall and F-score of different subsets of the crowdsourced lexicon in comparison to the gold standard.

ent subsets of the candidate verbalizations retrieved during the translation stage of the test run, formed according to a) the number of translations a given candidate verbalization occured in, and b) the number of upvotes a verbalization received during one of the runs of the evaluation stage. This way one can see if and how the evaluation stage of our workflow can actually improve the quality of the resulting lexicon, or if a simple majority decision based on the number of translations a candidate verbalization occurs in would be sufficient to generate lexicons of good quality. As can be seen in the table, one can reach a precision of 1.0 even through majority decision alone, meaning that all verbalizations in the respective subset of the crowdsourced lexicon are also found in the gold standard. However, the data also shows that applying majority decision leads to a clear drop in recall. While for larger datasets this effect may be not as strong, it is actually to be expected if the gold standard also contains verbalizations less commonly used that may not occur that often in the retrieved translations. If one wants to keep such more uncommon but still correct verbalizations in the crowdsourced lexicon, filtering candidate verbalizations based on the results of the evaluation stage alone may be a better option. Accordingly, for our test run the overall best result in terms of F-score is reached by only filtering based on votes from the first run of the evaluation stage.

Out of the 49 verbalizations given in the gold standard, eleven do not occur in the result set of our test run at all; therefore, even for the whole set of candidate verbalizations from the translation stage (# Translations ≥ 1,# Votes ≥ 0) recall is only at 0.77. One possible reason may be the influence the English seed sentences have on the syntactic constructions — and therefore parts-of-speech — and semantic level of granularity workers will use in their translations. For example, for the property `yearOfConstruction` the Japanese gold standard, among other entries, contains the construction （に）完成する, which is a rather general term that could be translated as *to complete [in]*. However, the English gold standard we worked with contained the more specific *constructed [in]* as the only verbalization of `yearOfConstruction`. Accordingly, workers were only shown sentences of the form *X was constructed in [year] Y* for this ontology element, and we only retrieved Japanese verbalizations at around the same level of semantic specificity.

In most cases, filtering based on the votes from the first run of the evaluation stage yields better recall and better F-scores, though not always better precision, than filtering based on the votes from the second run. However, overall the quality of the votes from both runs seems to be very similar, which would back our finding from the translation stage that the language the task instructions are given in does not have much effect on the quality of the retrieved results.

The English seed lexicon contains 1,217 entries in total, which — given the settings of our test run — would amount to 10,953 microtasks for both the translation and evaluation stage. Hence, at a payment of one cent per microtask, carrying out the evaluation stage for the whole lexicon would cost 109.53 dollars. The translation stage would cost something between that same amount and 876.24 dollar in case we have to pay eight cent for every translation.

4. Outlook

The scale of this first test run is enough to show that crowdsourcing ontology lexicons in general and our workflow presented above in particular are basically feasible. Still, a number of questions that occurred during our work, e.g. concerning the appropriate amount of payment during the translation stage or the effect the language of instruction has, would require more data to answer with confidence. Therefore, further tests at a larger scale — e.g. based on a larger number of exemplary ontology elements — would be due. However, given the current changes on CrowdFlower described in Section 3.3., we would most probably have to look for a new crowdsourcing platform: While the evaluation stage of our current test run was quite successful, both the outcome of our first run of the translation stage and the large amount of people who attempted to work on the evaluation stage but failed at the test questions show that attempting to re-run the translation stage on CrowdFlower would not make much sense as long as we can no longer

narrow down the set of allowed workers. One possible alternative would be to switch to a Japanese crowdsourcing platform such as Yahoo! Crowdsourcing[12].

5. Conclusion

We presented a two-stage workflow for crowdsourcing ontology lexicons through a translation task, which we tested by generating a Japanese lexicon for DBpedia for ten exemplary ontology elements. Comparison of this lexicon to a manually created one shows that in particular for smaller languages, where it may be difficult to find people with sufficient language competency otherwise, generating ontology lexicons this way is a viable option. Further tests could be necessary to figure out the optimal settings for our workflow.

6. Acknowledgements

This work was supported by the Cluster of Excellence Cognitive Interaction Technology CITEC (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

7. Bibliographical References

Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., and Lehmann, J. (2013). Crowdsourcing linked data quality assessment. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*, pages 260–276.

Ambati, V. and Vogel, S. (2010). Can crowds build parallel corpora for machine translation systems? In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 62–65, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arcan, M. and Buitelaar, P. (2013). Ontology label translation. In Lucy Vanderwende, et al., editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 40–46. The Association for Computational Linguistics.

Benjamin, M. and Radetzky, P. (2014). Multilingual Lexicography with a Focus on Less-Resourced Languages: Data Mining, Expert Input, Crowdsourcing, and Gamification. In *9th edition of the Language Resources and Evaluation Conference*.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, September.

Howe, J. (2006). The rise of crowdsourcing. *Wired Magazine*, 14(6), 06.

Irvine, A. and Klementiev, A. (2010). Using mechanical turk to annotate lexicons for less commonly used languages. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 108–113, Stroudsburg, PA, USA. Association for Computational Linguistics.

McCrae, J., Espinoza, M., Montiel-Ponsoda, E., Aguado-de Cea, G., and Cimiano, P. (2011a). Combining statistical and semantic approaches to the translation of ontologies and taxonomies. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, SSST-5, pages 116–125, Stroudsburg, PA, USA. Association for Computational Linguistics.

McCrae, J., Spohr, D., and Cimiano, P. (2011b). Linking Lexical Resources and Ontologies on the Semantic Web with lemon. In *Proceedings of the 8th Extended Semantic Web Conference (ESWC)*.

Prévot, C.-R. H. L., Calzolari, N., Gangemi, A., Lenci, A., and Oltramari, A., (2010). *Ontology and the lexicon: a multi-disciplinary perspective (introduction)*. Studies in Natural Language Processing. Cambridge University Press, April.

Quinn, A. J. and Bederson, B. B. (2011). Human computation: A survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1403–1412, New York, NY, USA. ACM.

Rico, M. and Unger, C. (2015). Lemonade: A web assistant for creating and debugging ontology lexica. In Chris Biemann, et al., editors, *Natural Language Processing and Information Systems - 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015 Passau, Germany, June 17-19, 2015 Proceedings*, volume 9103 of *Lecture Notes in Computer Science*, pages 448–452. Springer.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (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.

Unger, C., McCrae, J., Walter, S., Winter, S., and Cimiano, P. (2013). A lemon lexicon for dbpedia. Proceedings of 1st International Workshop on NLP and DBpedia, co-located with the 12th International Semantic Web Conference (ISWC 2013), October 21-25, Sydney, Australia. CEUR Workshop Proceedings.

Walter, S., Unger, C., and Cimiano, P., (2014). *M-ATOLL: A Framework for the Lexicalization of Ontologies in Multiple Languages*, volume 8796 of *The Semantic Web – ISWC 2014*, pages 472–486. Springer International Publishing.

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1220–1229, Stroudsburg, PA, USA. Association for Computational Linguistics.

[12] `http://crowdsourcing.yahoo.co.jp/`