Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines

Marta Sabou,* Kalina Bontcheva,† Leon Derczynski,† Arno Scharl*
*MODUL University Vienna
Am Kahlenberg 1, Vienna, Austria
{marta.sabou,arno.scharl}@modul.ac.at
†University of Sheffield
211 Portobello, Sheffield S1 4DP, UK
{K.Bontcheva,L.Derczynski}@dcs.shef.ac.uk

Abstract

Crowdsourcing is an emerging collaborative approach that can be used for the acquisition of annotated corpora and a wide range of other linguistic resources. Although the use of this approach is intensifying in all its key genres (paid-for crowdsourcing, games with a purpose, volunteering-based approaches), the community still lacks a set of best-practice guidelines similar to the annotation best practices for traditional, expert-based corpus acquisition. In this paper we focus on the use of crowdsourcing methods for corpus acquisition and propose a set of best practice guidelines based in our own experiences in this area and an overview of related literature. We also introduce GATE Crowd, a plugin of the GATE platform that relies on these guidelines and offers tool support for using crowdsourcing in a more principled and efficient manner.

Keywords: Crowdsourcing, Human Computation, Corpus Annotation, Guidelines, Survey

1. Introduction

Over the past ten years, Natural Language Processing (NLP) research has been driven forward by a growing volume of annotated corpora, produced by evaluation initiatives such as ACE (ACE, 2004), TAC,[1] SemEval and Senseval,[2] and large annotation projects such as OntoNotes (Hovy et al., 2006). These corpora have been essential for training and domain adaptation of NLP algorithms and their quantitative evaluation, as well as for enabling algorithm comparison and repeatable experimentation. Thanks to these efforts, there are now well-understood best practices in how to create annotations of consistently high quality, by employing, training, and managing groups of linguistic and/or domain experts. This process is referred to as "the science of annotation" (Hovy, 2010).

More recently, the emergence of crowdsourcing platforms (e.g. paid-for marketplaces such as Amazon Mechanical Turk (AMT) and CrowdFlower (CF); games with a purpose; and volunteer-based platforms such as crowdcrafting), coupled with growth in internet connectivity, motivated NLP researchers to experiment with crowdsourcing as a novel, *collaborative* approach for obtaining linguistically annotated corpora. The advantages of crowdsourcing over expert-based annotation have already been discussed elsewhere (Fort et al., 2011; Wang et al., 2012), but in a nutshell, crowdsourcing tends to be cheaper and faster.

There are now a large and continuously growing number of papers, which have used crowdsourcing in order to create annotated data for training and testing a wide range of NLP algorithms, as detailed in Section 2. and listed in Table 1. As the practice of using crowdsourcing for corpus annotation has become more widespread, so has the need for a best practice synthesis, spanning all three crowdsourcing genres and generalising from the specific NLP annotation task reported in individual papers. The meta-review of (Wang et al., 2012) discusses the trade-offs of the three crowdsourcing genres, alongside dimensions such as contributor motivation, setup effort, and human participants. While this review answers some key questions in using crowdsourcing, it does not provide a summary of best practice in how to setup, execute, and manage a complete crowdsourcing annotation project. In this paper we aim to address this gap by putting forward a set of best practice guidelines for crowdsourced corpus acquisition (Section 3.) and introducing GATE Crowd, an extension of the GATE NLP platform that facilitates the creation of crowdsourced tasks based on best practices and their integration into larger NLP processes (Section 4.).

2. Crowdsourcing Approaches

Crowdsourcing paradigms for corpus creation can be placed into one of three categories: mechanised labour, where workers are rewarded financially; games with a purpose, where the task is presented as a game; and altruistic work, relying on goodwill.

Mechanised labour has been used to create corpora that support a broad range of NLP problems (Table 1). Highly popular are NLP problems that are inherently subjective and cannot yet be reliably solved automatically, such as sentiment and opinion mining (Mellebeek et al., 2010), word sense disambiguation (Parent and Eskenazi, 2010), textual entailment (Negri et al., 2011), question answering (Heilman and Smith, 2010). Others create corpora of special resource types such as emails (Lawson et al., 2010), twitter feeds (Finin et al., 2010), augmented and alternative communication texts (Vertanen and Kristensson, 2011).

One advantage of crowdsourcing is "access to foreign markets with native speakers of many rare languages" (Zaidan and Callison-Burch, 2011). This feature is particularly useful for those that work on less-resourced languages such as Arabic (El-Haj et al., 2010) and Urdu (Zaidan and Callison-Burch, 2011). Irvine and Klementiev (2010) demonstrated

[1] www.nist.gov/tac

[2] www.senseval.org

that it is possible to create lexicons between English and 37 out of the 42 low-resource languages they examined.

Games with a purpose (GWAPs) for annotation include *Phratris* (annotating sentences with syntactic dependencies) (Attardi, 2010), *PhraseDetectives* (Poesio et al., 2012) (anaphora annotations), and *Sentiment Quiz* (Scharl et al., 2012) (sentiment). GWAP-based approaches for collecting **speech data** include *VoiceRace* (McGraw et al., 2009), a GWAP+MTurk approach, where participants see a definition on a flashcard and need to guess and speak the corresponding word, which is then transcribed automatically by a speech recognizer; *VoiceScatter* (Gruenstein et al., 2009), where players must connect word sets with their definitions; Freitas et al.'s GWAP (Freitas et al., 2010), where players speak answers to graded questions in different knowledge domains; and *MarsEscape* (Chernova et al., 2010), a two-player game for collecting large-scale data for human-robot interaction.

An early example of leveraging volunteer contributions is *Open Mind Word Expert*, a Web interface that allows volunteers to tag words with their appropriate sense from WordNet in order to collect training data for the Senseval campaigns (Chklovski and Mihalcea, 2002). Also, the MNH ("Translation for all") platform tries to foster the formation of a community through functionalities such as social networking and group definition support (Abekawa et al., 2010). Lastly, crowdcrafting.org is a community platform where NLP-based applications can be deployed.

Notably, volunteer projects that have not been conceived with a primary NLP interest but which delivered results that are useful in solving NLP problems are (i) Wikipedia, (ii) The Open Mind Common Sense project for collecting general world knowledge from volunteers in multiple languages, a key source for the ConceptNet semantic network that can enable various text understanding tasks; (iii) or Freebase a structured, graph-based knowledge repository offering information about almost 22 million entities constructed both by automatic means but also through contributions from thousands of volunteers.

3. Best Practice Guidelines

Conceptually, the process of crowdsourcing language resources can be broken down into four main stages, outlined in Figure 3. and discussed in the following subsections. These stages have been identified based on generalising our experience with crowdsourced corpus acquisition (Rafelsberger and Scharl, 2009; Scharl et al., 2012; Sabou et al., 2013a; Sabou et al., 2013b) and a meta-analysis of other crowdsourcing projects summarized in Table 1.

3.1. Project Definition

The first step is to choose the appropriate crowdsourcing genre, by balancing cost, required completion timescales, and the required annotator skills (Wang et al., 2012). Table 1 lists mostly mechanised labour based works (using either AMT or CF) and one GWAP. Secondly, the chosen NLP problem (e.g. named entity annotation, sentiment lexicon acquisition) needs to be decomposed into a set of simple crowdsourcing tasks, which can be understood and carried out by non-experts with minimal training and compact guidelines. Reward and budget should also be determined as part of the project definition.

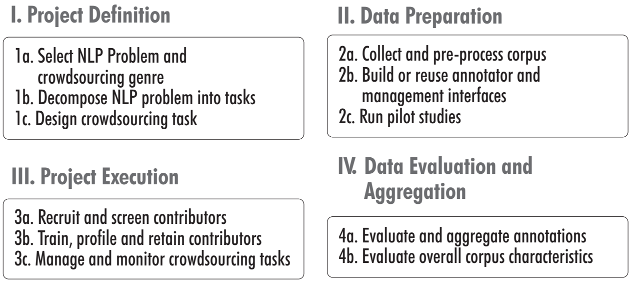
I. Project Definition
1a. Select NLP Problem and crowdsourcing genre
1b. Decompose NLP problem into tasks
1c. Design crowdsourcing task
II. Data Preparation
2a. Collect and pre-process corpus
2b. Build or reuse annotator and management interfaces
2c. Run pilot studies
III. Project Execution
3a. Recruit and screen contributors
3b. Train, profile and retain contributors
3c. Manage and monitor crowdsourcing tasks
IV. Data Evaluation and Aggregation
4a. Evaluate and aggregate annotations
4b. Evaluate overall corpus characteristics

Figure 1: Crowdsourcing Stages

Many NLP problems can be cast as one of a finite range of common task types. For example, given the pattern of *selection task* – where workers are presented with some information and required to select one of a list of possible answers – one can implement word sense disambiguation, sentiment analysis, entity disambiguation, relation typing, and so on. Similarly, a *sequence marking pattern* in which workers highlight items in a sequence can be applied, *inter alia*, to named entity labelling, timex extraction, and actor identification. Determining the common factors in these tasks and using them as templates improves efficiency and makes iterative enhancement of task designs possible. We investigate two such templates in a generic, reusable NLP crowdsourcing architecture described in Section 4.

Keeping tasks simple and intuitive is another important principle, where a simpler design without too much variance tends to lead to better results. Indeed, a simple, undistracting, clean interface helps even more than, for example, switching instructions from L2 to a worker's native language (Khanna et al., 2010). With respect to task scope, annotating one named entity type at a time, albeit more expensive, places lower cognitive load on workers and makes it easier to have brief and yet comprehensive instructions (see e.g., Bontcheva et al. (2014a)). Experience from expert-based annotation (Hovy, 2010) has shown that annotators should not be asked to choose from more than 10, ideally seven, categories. In comparison, crowdsourcing classification tasks often present fewer choices – in most cases between two (binary choice) and five categories, as suggested by the 5th column of Table 1.

When longer documents are being annotated, one needs to decide whether to put the entire document into one task, to split it up into smaller parts – one per crowdsouring task (e.g. paragraphs or sentences), or to avoid including in the corpus any documents above a certain size. For many NLP problems, a sentence provides sufficient context, however, this is not always the case. For example, Poesio et al. (2012) annotated Wikipedia articles and books from the Gutenberg project. Their game splits larger texts into paragraphs, which each becomes a separate unit. This introduced a problem in cases of long-distance anaphora, where the antecedent is not present in the current paragraph and hence cannot be selected by the game player. In general, given the limited time which contributors spend on each

Approach	Annotation of	Project Definition				Data Prep.	Project Execution (Annottion)						Aggregation&Evaluation		
		Genre	Workers/task	Nr. of categories	Reward Amount	Prototype	Recruitment	Training	Screen	Profile	Retain	In task Quality	Contribution Evaluation	Aggregation	Resource Evaluation
(Finin et al., 2010)	NEs in Tweets	CF, AMT	2	4	$0.05	Y	MLP	GS, instr.	-	Y	-	GS	Prof.	MV	-
(Voyer et al., 2010)	NEs	CF	5	2	-	-	MLP	instr	-	-	-	GS	-	CF	IAA, Task
(Lawson et al., 2010)	NEs in Emails	AMT*	4, 6, 7	3	$0.01+ bonus	Y	MLP	instr	-	-	Bonus	-	MV	MV	Task
(Yetisgen-Yildiz et al., 2010)	Medical NEs	AMT*	4	3	$0.01-0.05 +bonus	Y	MLP	instr	-	-	Comm., Bonus	-	-	MV	IAA, F-score
(Rosenthal et al., 2010)	PrepPhrase Attachment	AMT	3	3	$0.04	Y	MLP	-	LOC	-	-	-	-	MV	Precision
(Jha et al., 2010)	PP Attach.	AMT	5	varies	$0.04	Y	MLP	instr	-	-	-	-	-	MV	IAA, Rec
(Snow et al., 2008)	Affect, Wrd Sim.	AMT	10	range	-	-	MLP	instr	-	Y	-	-	-	Avg	IAA, Task
	Event&TE, WSD	AMT	10	2, 3	-	-	MLP	instr	-	Y	-	-	-	MV	IAA
(Yano et al., 2010)	Bias in polit. blogs	AMT	5	3, 6	$0.02, $0.04	-	MLP	instr	LOC, AR90	-	-	-		MV	IAA
(Mellebeek et al., 2010	Polarity	AMT	3	3, 11 range	$0.02	Y	MLP	instr	COMP	-	-	-	-	MV	IAA Task
(Laws et al., 2011)	NEs&Affect	AMT*	2+n	-/2	$0.01	-	MLP	-	-	Y	-	AL	MV	MV	Task
(Sayeed et al., 2011)	Opinion	CF	3	4	$0.12	Y	MLP	instr GS	LOC	Y	-	GS	Prof.	MV	IAA F-score
(Hong and Baker, 2011)	Word Senses	AMT* CF*	10	4-5	$0.15	Y	MLP	instr	COMP, LOC,AR75	-	-	GS	-	MV	Acc.
(Rafelsberger et al., 2009)	Opinions	GWAP	7+5	5	-	-	SN	-	-	-	Levels, Boards	-	MV	Avg.	-

Table 1: Crowdsourcing corpus collection. Abbreviations: MLP(*)= mechanised labour platform (own interface); ALTR=altruistic crowdsourcing; SN=social network; GS= gold standard; LOC=geo-location based screening; ARn=average reliability based screening; COMP=competency test based screening; MV=Majority Vote; AL=Active Learning

task, the length of text to be annotated/read needs to be kept reasonably short, without compromising accuracy.

Determining when to reward contributors and the value of the reward (in game points or money) have an influence over the time-completion of the task and the quality of the gathered data. In terms of *what is rewarded*, most straightforwardly, in mechanized labour, workers are paid for each correctly completed task, and can be refused payment if they are discovered cheating. If some of the answers are known a-priori, then answers that agree with the Gold Standard are rewarded through comparative scoring. The CrowdFlower platform (Biewald, 2012) automatically mixes gold units in each task and the recommended amount is 20% gold data per task. Providing the crowdworkers with the option to comment on gold data annotations is also a useful motivational mechanism. Otherwise, a common strategy is to award the answers on which most contributors agree. In the latter case, however, a higher number of judgements per task need to be collected (typically between seven and ten), in order to minimise the effect of cheating.

Determining *how much to award* is another challenging issue as award quantity influences critical parameters such as the task completion time and the quality of the obtained data. Launching a pilot job helps determine the average time per task in the case of paid work; this in turn enables one to be sure that at least the minimum wage is being paid to workers, many of whom rely on crowd work as a primary source of income (Fort and Sagot, 2010). Current approaches listed in Table 1 mostly offer 0.01 - 0.05$ per task. Some variance has been seen in the relation between result quality and reward (Poesio et al., 2014). While some initial reports found that high rewards attracted noise that was detrimental to quality (Mason and Watts, 2010), more recent research was unable to repeat this finding, reporting that increased reward only affected the number of workers attracted to a job (and thus sped up its overall completion) but not the quality (Aker et al., 2012).

3.2. Data Preparation

In this stage, user interfaces need to be designed and the data collected and prepared. Interface design can be a major task, especially in the case of games with a purpose. Data processing may involve preliminary annotation with an automated tool or filtering objectionable content.

Automatic pre-processing of source data can speed up corpus creation, although it can also introduce bias and annotation errors (Hovy, 2010). NLP infrastructures are often used for bootstrapping manual annotation and iterative development of NLP applications (a prototype is developed and to annotate a set of documents for human annotators to correct). The corrected annotations generated by the crowd can then be used to improve the application, and the process is repeated. This technique was originally developed for low-level NLP tasks such as POS tagging, where it is known to improve annotation speed considerably (Fort and Sagot, 2010), and it also works well for higher-level annotation (e.g., patent annotation, bio-informatics ontologies, named entities, events).

We distinguish two kinds of user interface tool. *Acquisition interfaces* are designed for and used by the non-expert human contributors to carry out crowdsourcing tasks. *Management interfaces* are required by the person running the crowdsourcing project, in order to allow them to monitor progress, assess quality, and manage contributors.

Acquisition interfaces used for solving, primarily, classification and content generation problems have been successfully created within Mechanical Turk and CrowdFlower, as listed in the third column of Table 1 (approaches denoted with "*" build their own interfaces and make use of the MLPs only as a means to recruit workers). The interface in such cases is based on the widgets supported by the chosen platform. However, careful consideration must be paid to designing the tasks in a defensive manner (i.e., *defensive task design*), which reduces cheating. Enter type interfaces used for solving content generation problems can be cheated by providing random texts as input. To reduce the risk of text being copied from online sources or from one task to another, Vertanen and Kristensson (2011) prevent users from pasting text and allow only typing. A technique that is more specific to translation is to display the source sentence as an image rather than text, thus preventing contributors from simply copying the text into an automatic translator, e.g. (Zaidan and Callison-Burch, 2011). Select type interfaces are often easy to cheat on, as cheaters can easily provide random selections. Laws et al. (2011) have found that their interface based on simple radio button selection have attracted high amount of spam, driving down the overall classification accuracy to only 55%. Kittur et al. (2008) designed a task where workers had to rate the quality of a Wikipedia article against a set of criteria and obtained 48% invalid responses. While many techniques exist to filter out invalid responses after the completion of the task, it is preferable to prevent cheating in the first place. Interface design plays a key role here. Both Laws et al. (2011) and Kittur et al. (2008) have extended their interfaces with *explicitly verifiable questions* which force the users to process the content and also signal to the workers that their answers are being scrutinized. This seemingly simple technique has increased classification accuracy for (Laws et al., 2011) to 75% and reduced the percent of invalid responses to only 2.5% for (Kittur et al., 2008).

Management Interfaces support NLP researchers in monitoring the status of their tasks and in fine-tuning the task details including the selection and screening of contributors. Game and volunteer based projects must build these interfaces from scratch, for example, Poesio et al. (2012) report on the extenssive management interfaces they built to support PhraseDetectives. CrowdFlower and MTurk offer some of this functionality already. For example, CrowdFlower supports requesters through the life-cycle of the crowdsourcing process including acquisition interface design (Edit page), data and gold standards data management (Data and Gold pages), calibration of key parameters such as the number of units per page/HIT, judgements required per units and pay for unit based on the desired completion time and accuracy (Calibration page), overview of the job's progress and overall status during the process itself (Dashboard), detailed analysis of the workers that have contributed to the job including their trust level, accuracy (in relation to a supplied gold standard) and accuracy history.

There are proven benefits to performing a *small scale pilot* for testing the task definition, for ensuring that the appropriate task granularity and annotator instructions (e.g., (Vertanen and Kristensson, 2011; Feng et al., 2009)) are chosen and for fine-tuning the key parameters of the crowdsourcing task (e.g., payment, size). Indeed, several of the approaches listed in Table 1 make use of an initial prototype system to fine-tune their crowdsourcing process. Note that, piloting requires that the complete application is in place, and therefore it is performed in the "Preparation" rather than "Project definition" step. If a pilot is not successful however, the project definition step would need to be revisited.

For example, Negri and Mehdad (2010) devote 10 days and $100 to experiment with different methodologies for determining the optimal approach for collecting translations which includes gold-units, verification cycles, worker filtering mechanisms to achieve the right balance between cost, time and quality. More generally, McCreadie et al. (2012) advocate an iterative methodology where crowdsourcing tasks are submitted in multiple batches allowing continuous improvements to the task based on worker feedback and result quality. In our experience, an iterative methodology also offers protection from data loss and other problems that may occur during long-running crowdsourcing projects.

3.3. Project Execution

This is the main phase of each crowdsourcing project, which on smaller paid-for crowdsourcing projects can be completed sometimes in a matter of minutes (Snow et al., 2008), or run for many months or years, when games or volunteers are used for collecting large datasets.

It consists of three kinds of tasks: task workflow and management, contributor management (including profiling and retention), and quality control. Choices that need to be made include whether the entire corpus is to be annotated multiple times to allow for a reconciliation and verification step (higher quality, but higher costs) or whether it is sufficient to have only two or three annotators per document as long as they agree.

The decentralised nature of crowdsourcing and, potential relative lack of reusable workflow definition, task management and quality assurance interfaces can make this project stage rather challenging. Additional challenges exist in handling individual contributors, which involves training, screening, profiling, retaining and dealing with worker disputes; and in quality control during annotation.

Attracting and retaining a large number of contributors is key to the success of any crowdsourcing system. Therefore, a core challenge of all crowdsourcing approaches is how to motivate contributors to participate. This issue has been analyzed extensively in recent surveys, e.g. Doan et al. (2011) focus on two different aspects when considering motivation, which they refer to as the challenge of *"How to recruit and retain users?"*.

Contributor recruitment consists in a set of primarily advertising activities to attract contributors to the crowdsourcing project. Note that our view on recruitment differs from that of Doan et al. as we primarily look at mechanisms to attract contributors and are agnostic to the motivational aspect additionally considered by Doan. While these two issues are inherently related, we choose to examine them in separation for more clarity.

Most NLP projects recruit their contributors from marketplaces that offer a large and varied worker base who monitor newly posted tasks (see Table 1, column "Recruitment"). The idea of a portal bundling multiple crowdsourcing projects is also used, to lesser extent, in the GWAP area, where gwap.com publishes a collection of games built by von Ahn and his team, while OntoGame bundles together games for the semantic web area.

Another strategy is the use of multi-channel advertisements for attracting users. Chamberlain et al. (2009) advertised their game through a wealth of channels including local and national press, science websites, blogs, bookmarking websites, gaming forums, and social networking sites.

Filtering workers prior to the task (based on e.g. prior performance, geographic origin, and initial training) is important to improve quality. Extensive screening can however lead to slower task completion, so filtering through task-design is preferred to filtering through crowd characteristics. Although a worker's prior acceptance rate is one of the key filtering mechanisms offered by Mechanical Turk and CrowdFlower, sometimes this type of screening cannot be used on its own reliably and needs to be complemented with other filters such as geographic location.

Munro et al. (2010) describe a set of methods for assessing linguistic attentiveness prior to the actual task. These involve showing language constructs that are typically acquired late by first-language learners or stacked pronominal possessive constructions (e.g., John's sister's friend's car) and asking workers to select a correct paraphrase thereof. These techniques not only help identify workers that have a sufficient command of English, but also prompt for higher attentiveness during the task.

Screening activities are feasible when using crowdsourcing marketplaces where (at least some) characteristics of the workers are known - and indeed, some of the approaches we surveyed in Table 1 employ location (LOC), prior performance (AR) and competence based (COMP) screening (see column "Screening"). GWAPs and altruistic crowdsourcing projects do not usually have this opportunity since most often their user community of not a-priory known. A common approach here is to use *training* mechanisms in order to make sure that the contributor qualifies. Many projects embed positive (and/or negative) gold standard examples within their tasks to determine the general quality of data provided by each worker. For example, CrowdFlower offers immediate feedback to workers when they complete a "gold" unit, thus effectively training them. In general, training mechanisms using instructions and gold standard data are well-spread (see the "Training" in Table 1).

3.4. Data Evaluation and Aggregation

In this phase, the challenge lies in evaluating and aggregating the multiple contributor inputs into a complete linguistic resource, and in assessing the resulting overall quality.

This stage is required in order to make acquisition tasks reproducible and therefore scalable, and to ensure good corpus quality. Sub-tasks include tracking worker performance over time, worker agreement, and converting contributed judgments into a consistent set of annotations (see Section 4. on the latter). Some tools are available for judging worker accuracy to help smooth this process (Hovy et al., 2013). As shown in Table 1, contributor aggregation primarily relies on majority voting or average computation based algorithms, while the evaluation of the resulting corpus is usually performed by computing inter annotator agreement (IAA) within crowd-workers and/or with a baseline resource provided by an expert; by task-centric evaluation as well as by Precision, Recall and F-measure.

3.5. Legal and Ethical Issues

The use of crowdsourcing raises, among others, the following three issues of legal and ethical nature, which have so far not received sufficient attention, including: how to properly acknowledge contributions; how to ensure contributor privacy and wellbeing; and how to deal with consent and licensing issues.

No clear guidelines exist for the first issue, how to properly acknowledge crowd contributions. Some volunteer projects (e.g., FoldIt, Phylo) already include contributors in the authors' list (Cooper et al., 2010; Kawrykow et al., 2012).

The second issue is contributor privacy and well-being. Paid-for marketplaces (e.g. MTurk) go some way towards addressing worker privacy, although these are far from sufficient and certainly fall short with respect to protecting workers from exploitation, e.g. having basic payment protection (Fort et al., 2011). The use of mechanised labour (MTurk in particular) raises a number of workers' rights issues: low wages (below $2 per hour), lack of protection, and legal implications of using MTurk for longer term projects. We recommend at the least conducting a pilot task to see how long jobs take to complete, and ensuring that average pay exceeds the local minimum wage.

The third issue is licensing and consent, i.e. making it clear to the human contributors that by carrying out these tasks they are contributing knowledge for scientific purposes and that they agree to a well-defined license for sharing and using their work. Typically, open licenses such as Creative Commons are used and tend to be prominently stated in volunteer-based projects/platforms (Abekawa et al., 2010). In contrast, GWAPs tend to mostly emphasize the scientific purpose of the game, while many fail to state explicitly the distribution license for the crowdsourced data. In our view, this lack of explicit consent to licensing could potentially allow the exploitation of crowdsourced resources in a way which their contributors could find objectionable (e.g. not share a new, GWAP-annotated corpus freely with the community). Similarly, almost one third of psychology reviews on MTurk post no informed consent information at all (Behrend et al., 2011).

4. The GATE Crowdsourcing plugin

We relied on these best practice guidelines during the development of GATE Crowd, an open-source plugin for the GATE NLP platform (Cunningham et al., 2013) which offers crowdsourcing support to the platform's users (Bontcheva et al., 2014b). The plugin contains reusable task definitions and crowdsourcing workflow templates which can be used by researchers to commission the crowdsourcing of annotated corpora directly from within GATE's graphical user interface, as well as pre-process the data automatically with relevant GATE linguistic analysers, prior to crowdsourcing. Once all parameters are configured, the new GATE crowdsourcing plugin generates the respective crowdsourcing tasks automatically, which are then deployed on the chosen platform (e.g. CrowdFlower). On completion, the collected multiple judgements are imported back into GATE and the original documents are enriched with the crowdsourced information, modelled as multiple annotation layers (one per contributor). GATE's existing tools for calculating inter-annotator agreement and corpus analysis can then be used to gain further insights into the quality of the collected information.

In the first step, task name, instructions, and classification choices are provided, in a UI configuration dialog. For some categorisation NLP annotation tasks (e.g. classifying sentiment in tweets into positive, negative, and neutral), fixed categories are sufficient. In others, where the available category choices depend on the text that is being classified (e.g. the possible disambiguations of Paris are different from those of London), choices are defined through annotations on each of the classification targets. In this case, the UI generator then takes these annotations as a parameter and automatically creates the different category choices, specific to each crowdsourcing unit.

In sequential selection, sub-units are defined in the UI configuration – tokens, for this example. The annotators are instructed to click on all words that constitute the desired sequence (the annotation guidelines are given as a parameter during the automatic user interface generation).

Since the text may not contain a sequence to be annotated, we also generate an explicit confirmation checkbox. This forces annotators to declare that they have made the selection or there is nothing to be selected in this text.

The GATE Crowdsourcing plugin is available for download now via the developer versions at http://www.gate.ac.uk, and is bundled with GATE v8 due in 2014.

5. Conclusions

Annotation science and reusable best practice guidelines have evolved in response to the need for harnessing collective intelligence for the creation of large, high-quality language resources. While crowdsourcing is increasingly regarded as a novel collaborative approach to scale up LR acquisition in an affordable manner, researchers have mostly used this paradigm to acquire small- to medium-sized corpora. The novel contribution of this paper lies in defining a set of best practice guidelines for crowdsourcing, as the first step towards enabling repeatable acquisition of large-scale, high quality LRs, through the implementation of the necessary infrastructural support within the GATE open source language engineering platform.

A remaining challenge for crowdsourcing projects is that the cost to define a single annotation project can outweigh the benefits. Future work should address this by providing a generic crowdsourcing infrastructure which transparently combines different crowdsourcing genres (i.e. marketplaces, GWAPs, and volunteers). Such an infrastructure should help with sharing meta-information, including contributor profiles, annotator capabilities, past work, and history from previously completed projects. Solving this challenge could help prevent annotator bias and minimise human oversight required, by implementing more sophisticated crowd-based annotation workflows, coupled with in-built control mechanisms. Such infrastructure will also need to implement reusable, automated methods for quality control and aggregation and make use of the emerging reusable task definitions and workflow patterns. The reward is increased scalability and quality from the almost limitless power of the crowd.

6. Acknowledgements

This work is part of the uComp project (www.ucomp.eu), which receives the funding support of EPSRC EP/K017896/1, FWF 1097-N23, and ANR-12-CHRI-0003-03, in the framework of the CHIST-ERA ERA-NET.

7. References

Abekawa, T., Utiyama, M., Sumita, E., and Kageura, K. (2010). Community-based Construction of Draft and Final Translation Corpus through a Translation Hosting Site Minna no Hon'yaku (MNH). In *Proc. LREC*.

ACE, (2004). *Annotation Guidelines for Event Detection and Characterization (EDC)*, Feb. Available at http://www.ldc.upenn.edu/Projects/ACE/.

Aker, A., El-Haj, M., Albakour, M.-D., and Kruschwitz, U. (2012). Assessing crowdsourcing quality through objective tasks. In *Proc. LREC*, pages 1456–1461.

Attardi, G. (2010). Phratris – A Phrase Annotation Game. In *INSEMTIVES Game Idea Challenge*.

Behrend, T., Sharek, D., Meade, A., and Wiebe, E. (2011). The viability of crowdsourcing for survey research. *Behav. Res.*, 43(3).

Biewald, L. (2012). Massive multiplayer human computation for fun, money, and survival. In *Current Trends in Web Engineering*, pages 171–176. Springer.

Bontcheva, K., Derczynski, L., and Roberts, I. (2014a). Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*. Springer.

Bontcheva, K., Roberts, I., and Derczynski, L. (2014b). The GATE Crowdsourcing Plugin: Crowdsourcing Annotated Corpora Made Easy. In *Proc. EACL*.

Callison-Burch, C. and Dredze, M., editors. (2010). *Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Chamberlain, J., Poesio, M., and Kruschwizt, U. (2009). A new life for a dead parrot: Incentive structures in the Phrase Detectives game. In *Proc. of the Webcentives Workshop*.

Chernova, S., Orkin, J., and Breazeal, C. (2010). Crowdsourcing HRI through Online Multiplayer Games. In *Dialog with Robots: Papers from the AAAI Fall Symposium (FS-10-05)*.

Chklovski, T. and Mihalcea, R. (2002). Building a Sense Tagged Corpus with Open Mind Word Expert. In *Proc. of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popovic, Z., and players, F. (2010). Predicting protein structures with a multiplayer online game. *Nature*, 466(7307).

Cunningham, H., Tablan, V., Roberts, A., and Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS computational biology*, 9(2):e1002854.

Doan, A., Ramakrishnan, R., and Halevy, A. Y. (2011). Crowdsourcing Systems on the World-Wide Web. *Commun. ACM*, 54(4), April.

El-Haj, M., Kruschwitz, U., and Fox, C. (2010). Using Mechanical Turk to Create a Corpus of Arabic Summaries. In *Proc. LREC*.

Feng, D., Besana, S., and Zajac, R. (2009). Acquiring High Quality Non-Expert Knowledge from On-Demand Workforce. In *Proc. of The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating Named Entities in Twitter Data with Crowdsourcing. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Fort, K. and Sagot, B. (2010). Influence of Pre-annotation on POS-tagged Corpus Development. In *Proc. of the Fourth Linguistic Annotation Workshop*.

Fort, K., Adda, G., and Cohen, K. (2011). Amazon Mechanical Turk: Gold Mine or Coal Mine? *Computational Linguistics*, 37(2):413 –420.

Freitas, J., Calado, A., Braga, D., Silva, P., and Dias, M. (2010). Crowdsourcing platform for large-scale speech data collection. *Proceedings of FALA, Vigo*.

Gruenstein, E., Mcgraw, I., and Sutherl, A. (2009). A Self-Transcribing Speech Corpus: Collecting Continuous Speech with an Online Educational Game. In *Proc. of The Speech and Language Technology in Education (SLaTE) Workshop*.

Heilman, M. and Smith, N. A. (2010). Rating Computer-Generated Questions with Mechanical Turk. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Hong, J. and Baker, C. F. (2011). How Good is the Crowd at "real" WSD? In *Proc. of the 5th Linguistic Annotation Workshop*.

Hovy, E., Marcus, M. P., Palmer, M., Ramshaw, L. A., and Weischedel, R. M. (2006). OntoNotes: The 90% Solution. In *Proc. NAACL*.

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning Whom to trust with MACE. In *Proc. of NAACL-HLT*, pages 1120–1130.

Hovy, E. (2010). Annotation. In *Tutorial Abstracts of ACL*.

Irvine, A. and Klementiev, A. (2010). Using Mechanical Turk to Annotate Lexicons for Less Commonly Used Languages. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Jha, M., Andreas, J., Thadani, K., Rosenthal, S., and McKeown, K. (2010). Corpus Creation for New Genres: A Crowdsourced Approach to PP Attachment. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., Zarour, E., and players, P. (2012). Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment. *PLoS ONE*, 7(3):e31362.

Khanna, S., Ratan, A., Davis, J., and Thies, W. (2010). Evaluating and improving the usability of Mechanical Turk for low-income workers in India. In *Proceedings of the first ACM symposium on Computing for Development*. ACM.

Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing User Studies with Mechanical Turk. In *Proc. of the 26th Conference on Human Factors in Computing Systems*.

Laws, F., Scheible, C., and Schütze, H. (2011). Active Learning with Amazon Mechanical Turk. In *Proc. EMNLP*.

Lawson, N., Eustice, K., Perkowitz, M., and Yetisgen-Yildiz, M. (2010). Annotating Large Email Datasets for Named Entity Recognition with Mechanical Turk. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Mason, W. and Watts, D. J. (2010). Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108.

McCreadie, R., Macdonald, C., and Ounis, I. (2012). Identifying Top News Using Crowdsourcing. *Information Retrieval*. 10.1007/s10791-012-9186-z.

McGraw, I., Gruenstein, A., and Sutherland, A. (2009). A Self-Labeling Speech Corpus: Collecting Spoken Words with an Online Educational Game. In *Proc. of INTERSPEECH*.

Mellebeek, B., Benavent, F., Grivolla, J., Codina, J., Costa-jussà, M. R., and Banchs, R. (2010). Opinion Mining of Spanish Customer Comments with Non-Expert Annotations on Mechanical Turk. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Munro, R., Bethard, S., Kuperman, V., Lai, V. T., Melnick, R., Potts, C., Schnoebelen, T., and Tily, H. (2010). Crowdsourcing and Language Studies: The New Generation of Linguistic Data. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Negri, M. and Mehdad, Y. (2010). Creating a Bi-lingual Entailment Corpus through Translations with Mechanical Turk : 100 for a 10-day Rush. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., and Marchetti, A. (2011). Divide and Conquer: Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora. In *Proc. EMNLP*.

Parent, G. and Eskenazi, M. (2010). Clustering Dictionary Definitions Using Amazon Mechanical Turk. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Poesio, M., Kruschwitz, U., Chamberlain, J., Robaldo, L., and Ducceschi, L. (2012). Phrase Detectives: Utilizing Collective Intelligence for Internet-Scale Language Resource Creation. *Transactions on Interactive Intelligent Systems*.

Poesio, M., Chamberlain, J., and Kruschwitz, U. (2014). Crowdsourcing. In *Handbook of Linguistic Annotation*. Springer.

Rafelsberger, W. and Scharl, A. (2009). Games with a Purpose for Social Networking Platforms. In *Proc. ACM conference on Hypertext and Hypermedia*.

Rosenthal, S., Lipovsky, W., McKeown, K., Thadani, K., and Andreas, J. (2010). Towards Semi-Automated Annotation for Prepositional Phrase Attachment. In *Proc. LREC*.

Sabou, M., Bontcheva, K., Scharl, A., and Föls, M. (2013a). Games with a Purpose or Mechanised Labour? A Comparative Study. In *Proc. International Conference on Knowledge Management and Knowledge Technologies*.

Sabou, M., Scharl, A., and Föls, M. (2013b). Crowdsourced Knowledge Acquisition: Towards Hybrid-Genre Workflows. *International Journal on Semantic Web and Information Systems*, 9(3).

Sayeed, A. B., Rusk, B., Petrov, M., Nguyen, H. C., Meyer, T. J., and Weinberg, A. (2011). Crowdsourcing syntactic relatedness judgements for opinion mining in the study of information technology adoption. In *Proc. of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH '11)*.

Scharl, A., Sabou, M., Gindl, S., Rafelsberger, W., and Weichselbraun, A. (2012). Leveraging the wisdom of the crowds for the acquisition of multilingual language resources. In *Proc. LREC*.

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 *Proc. EMNLP*.

Vertanen, K. and Kristensson, P. O. (2011). The Imagination of Crowds: Conversational AAC Language Modeling using Crowdsourcing and Large Data Sources. In *Proc. EMNLP*.

Voyer, R., Nygaard, V., Fitzgerald, W., and Copperman, H. (2010). A Hybrid Model for Annotating Named Entity Training Corpora. In *Proc. of the Fourth Linguistic Annotation Workshop (LAW IV '10)*.

Wang, A., Hoang, C., and Kan, M. Y. (2012). Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation*.

Yano, T., Resnik, P., and Smith, N. A. (2010). Shedding (a Thousand Points of) Light on Biased Language. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Yetisgen-Yildiz, M., Solti, I., Xia, F., and Halgrim, S. R. (2010). Preliminary Experience with Amazon's Mechanical Turk for Annotating Medical Named Entities. In Callison-Burch and Dredze (Callison-Burch and Dredze, 2010).

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing Translation: Professional Quality from Non-Professionals. In *Proc. ACL*.