The Seventh AAAI Conference on Human
Computation and Crowdsourcing (HCOMP-19)

Progression in a Language Annotation Game with a Purpose

Chris Madge,[1] Juntao Yu,[1] Jon Chamberlain,[2] Udo Kruschwitz,[3] Silviu Paun,[1] Massimo Poesio[1]
[1]Queen Mary University Of London [2]University Of Essex [3]University of Regensburg
{c.j.madge, juntao.yu, s.paun, m.poesio}@qmul.ac.uk, jchamb@essex.ac.uk, Udo.Kruschwitz@ur.de

Abstract

Within traditional games design, incorporating progressive difficulty is considered of fundamental importance. But despite the widespread intuition that progression could have clear benefits in Games-With-A-Purpose (GWAPs)–e.g., for training non-expert annotators to produce more complex judgements– progression is not in fact a prominent feature of GWAPs; and there is even less evidence on its effects. In this work we present an approach to progression in GWAPs that generalizes to different annotation tasks with minimal, if any, dependency on gold annotated data. Using this method we observe a statistically significant increase in accuracy over randomly showing items to annotators.

Introduction

In Human Computation, annotators typically have very mixed ability (Snow et al. 2008). Traditionally, the result of this has been that in both projects based on plain crowdsourcing, and projects based on Games-With-A-Purpose (GWAPs), responses from annotators that fail to pass a periodic assessment against a gold standard (Jurgens and Navigli 2014), or pass an initial test (Downs et al. 2010), are simply disregarded, without attempting to train these annotators to carry out those labelling tasks. This approach is generally complemented by aggregation methods that learn the various annotator abilities based on their agreement (Dawid and Skene 1979; Passonneau and Carpenter 2014; Paun et al. 2018) and task complexity (Carpenter 2008) and use these parameters to weigh an annotator's contributions.

More recently, in the interest of maximising resource utilization, crowdsourcing methods have been proposed to match annotators to specific tasks. Such methods have been found to result in better resource utilization by taking into consideration the workers' specific skills, availability, and cost (Bachrach et al. 2012; Lee, Park, and Park 2014; Basu Roy et al. 2015). Researchers have also come to realize that whereas some human computation tasks only require very simple judgements, in other cases the pool of workers with the required background is restricted. Early GWAPs focused on context-free, decomposable tasks, all of a level of difficulty that was accessible to annotators of all skill levels, such as image labelling (Von Ahn and Dabbish 2004; von Ahn, Liu, and Blum 2006; von Ahn et al. 2007). However, later GWAPs have become increasingly ambitious, and have been used, for instance, for language annotations that require deep linguistic knowledge (Hladká, Mírovský, and Kohout 2011), understanding the context of sentences or sometimes paragraphs (Poesio et al. 2013), carrying out tasks that vary in complexity (Venhuizen et al. 2013) and sometimes require domain specific knowledge (Dumitrache et al. 2013). Such annotation tasks further motivate introducing some progression in the worker's task: starting with easier assignments before progressing to more complex ones when the worker has demonstrated to have acquired enough practice and/or understanding. Yet although many crowdsourcing projects, whether using GWAPs or microtask crowdsourcing, appear to employ some form of progression, we are not aware of any paper in the area proposing some form of progression and demonstrating its benefit. This is the main objective of this paper.

Assigning to workers tasks at the appropriate levels also has benefits that go beyond the optimization of resources. Despite the advertised motivation for participating in crowdsourcing being the financial incentive, studies have shown some evidence that *fun* is one of the leading intrinsic motivators (Hossain 2012) and in some cases, may be even more motivating than money (Kaufmann, Schulze, and Veit 2011); and this is uncontroversially the case for GWAPs (Von Ahn and Dabbish 2008). This provides a further motivation for employing some sort of progression in GWAPs. Ensuring that players have the appropriate level of challenge has been shown to increase motivation (Malone 1981), learning (Hung, Sun, and Yu 2015; Hamari et al. 2016) and enjoyment (Carroll and Thomas 1988; Sweetser and Wyeth 2005). Collectively, these would appear beneficial in recruiting workers, training them to perform complex tasks and retaining them over a long period of time.

Last, but not least, the type of progression explored here is very appropriate for the target players of the particular GWAP used for this study, a language annotation GWAP in which workers are asked to identify noun phrases in text, and whose primary target are players interested in linguistics or in improving their English through playing. Target players can start with simpler types of noun phrases and then progress to more complex ones once they demonstrate to have understood the more basic concepts.

In this paper we present a method for task assignment in GWAPs aiming to present workers with tasks that match their current competence, which is dynamically reassessed possibly leading to progression to more complex tasks. We apply the method to our natural language sequence labelling GWAP, *TileAttack* (Madge et al. 2019), demonstrating that it results in significantly better labelling performance than random assignment of tasks to workers.

Related Work

Training and Progression in GWAPs

Whilst historically human computation, particularly in the form of microtask crowdsourcing, focused on unskilled homogeneous tasks, there is now an aspiration to use such methods in more challenging annotation tasks. This is seen to be the future of crowd work (Kittur et al. 2013). However, training is very challenging to design in microtask crowdsourcing. In contrast, games incorporate learning and provide a variety of training mechanisms that can be carried over into GWAPs. For this reason, it has been argued that devising suitable methods for training players is may result in GWAPs surpassing microtask crowdsourcing for complex annotation tasks (Tuite 2014). The dual motivation of progression as a means of training and providing engagement has thus been identified from the very early GWAPs for language resourcing (Lafourcade 2007). This section will look at some methods of training and progression currently used in GWAPs. Whilst all of the progression systems described seem perfectly suitable for the tasks they attempt to address, we discuss the potential positives and negatives of selecting such an approach for a different task.

We refer to the first approach to progression found in the GWAP literature as **switching**. When switching, a system toggles back and forth between players labelling unknown items and being assessed against gold annotated examples. When annotating gold examples the workers are given feedback on the label they chose. As their performance increases, the players see fewer gold examples, and spend more of their time labelling. Such approaches to crowdsourcing could be described as incorporating a form of progression. However, they do not not account for varying difficulty items. Other issues with this approach include the need for gold annotated data, and the reduced resource utilisation of testing a player against a gold, in which time they are not providing labels. The advantage of this approach is that only one player is required at a time, a departure from the original methods (Von Ahn and Dabbish 2008) which can permit for more game-like interfaces (Jurgens and Navigli 2014). We discuss here two prominent examples of what we have referred to as "switching". In the game *PuzzleRacer* (Jurgens and Navigli 2014) players provide annotations tying images with word-senses. They do this by racing through puzzle gates. Each gate has a series of images associated with it for the user to race through. The assessment/gold gates damage the players health when answered incorrect as a means of feedback. The gates through which the player provides a label have no resulting action regardless of if they are answered correctly or not. A model of the confidence of the annotator is held to determine which gate to show. *Quizz* (Ipeirotis and Gabrilovich 2014) is a multiple choice style gamified crowdsourcing system that experimented with recruiting players/workers through targeted advertising rather than the traditional micro-payment approach offered by platforms such as Amazon Mechanical Turk. Quizz users annotate by answering multiple choice questions in a variety of domains. A Markov Decision Process is used to learn which of the two to present to a user next. This system is also designed to optimize retention.

The next method is an example of real progression, that we refer to as **domain agnostic progression**. In the game *Dr. Detective* (Dumitrache et al. 2013), players annotate domain specific named entities in medical texts. *Dr. Detective* models a document's difficulty as the normalized vector of the number of sentences, the number of words, the average sentence length, the number of item types and the readability of the document (using the SMOG measure (Mc Laughlin 1969)). The selection process then involves finding the item with the smallest difficulty increment from all items that have a difficulty greater than or equal to the current item, excluding the current item. The authors mention that they believe computing difficulty based solely on textual metrics was a weakness and that the system would benefit from a domain specific metric of difficulty. A weakness of this approach is the assumption that the readability of the text is linked to the complexity of the task. A very short sentence could incorporate complex linguistic phenomena in a language resourcing task, depending on the nature of the task. A strength of the approach is that it does offer progression and does not require modelling a domain specific measure of complexity for a sentence. No evaluation of the effectiveness of this type of progression was carried out.

ZombiLingo (Fort, Guillaume, and Chastant 2014), a GWAP for annotating the syntactic structure of a text according to Dependency Grammar, uses a **skill-based domain specific progression**. *ZombiLingo* is structured into a number of different tasks for annotating different types of syntactic information. Different skills are required for the different forms of labelling. The initial assessment of the difficulty of an item is based on the type of linguistic phenomena that occurs in that item, and is derived from an automated pre-processing pipeline and the corpus the text comes from. This difficulty continuously evolves based on user responses. A player must undergo separate training for each phenomenon before being allowed to carry out that type of annotation. The strength of this approach is that the notion of difficulty used closely matches the complexity of the actual annotation. This approach however also raises a number of issues, apart from the obvious consideration that the particular measure of difficulty used by *ZombiLingo* could only be used by other GWAPs for dependency structure annotation, but not for other labelling tasks. One issue is that not all labelling tasks are clearly decomposable into a set of separately trainable skills. A second issue is that a reliable automated domain specific system must exist that can be used to identify the skills required to label an item. Such method of inferring complexity may not always exist, particularly if the task is gathering data for a new corpus. And again, no eval-

uation of the effectiveness of this type of progression was carried out. In this work we propose a progression method that we believe applicable to all linguistic labelling tasks.

In conclusion, there have been a variety of approaches taken to incorporating progression into GWAPs. However, as of yet it would seem there is no evaluation on the benefit of applying such mechanics. In this work we carried out an evaluation demonstrating that our proposed progression method results in improved accuracy.

Progression in Game Design

Within the context of traditional games, ensuring that the player is always presented with an appropriate level of challenge is a very active area of research and discussion. Topics that are generally considered fundamental to game design include difficulty scaling (Boutros 2008), user selected difficulty modes (Adams 2008), and dynamic difficulty adjustment (Hunicke 2005).

When designing for challenge in games and looking at how to bring enjoyment, game designers typically look to the theory of **flow** (Csikszentmihalyi 1990; Sweetser and Wyeth 2005). This involves presenting the player with in-game challenges that are commensurate with their increasing skill level to keep the player in the psychological state of "flow"; an enjoyable state of elevated focus and engagement. When the challenge is insufficient, players may become bored; but when the challenge is too great, players may become anxious. Designers try and keep their players in the narrow margin between these two states, known as the **flow channel**. More specifically, they attempt to take a meandering path through the channel (Figure 1) in which the player cycles between feeling the reward of applying their newly acquired skills and the challenge of acquiring new skills to meet the next challenge. In practice, this is often presented in levels in which a player perfects a skill or acquires an ability that makes the level they are currently at easier, shortly before progressing onto a new level where they face new challenges. (Schell 2014)

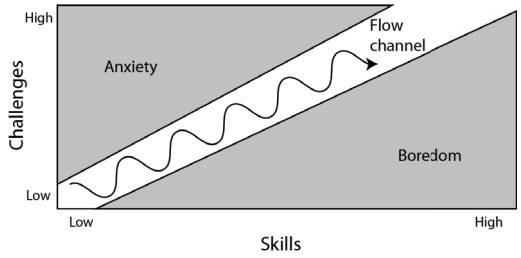

Figure 1: Flow Theory - Wave Channel (Schell 2014)

Training and Progression in Learning Games

Learning games such as *Motion Math: Hungry Fish* (Hung, Sun, and Yu 2015), *Quantum* and *Spunmore* (Hamari et al. 2016), have shown how challenge and flow are important in game-based learning, both directly in terms of the achieved learning outcomes and indirectly in terms of player engagement and satisfaction.

Task Assignment in Crowdsourcing

Crowdsourcing tasks rarely feature progression, or training. However, there have been multiple efforts in crowdsourcing to derive a measure of annotator skill to optimise task distribution and resource utilisation. Such methods often model annotator ability and item difficulty based on inter-annotator agreement (Bachrach et al. 2012; Lee, Park, and Park 2014; Basu Roy et al. 2015).

One such system is the *SmartCrowd* system. SmartCrowd attempts to find the best possible task for a worker based on the worker expertise (the level of knowledge with regards to certain skills), plus other factors such as, wage requirements and the worker acceptance rate. However, having assessed a users ability, SmartCrowd finds the best possible task for that ability and cost. There is no progression. The authors do mention that it would be possible to add skill improvement into the model and discuss the merits of doing so (Basu Roy et al. 2015).

TileAttack

TileAttack[1] (Madge et al. 2019) is a web-based, two-player, blind game in which players are awarded points based on player agreement of the tokens they mark. The visual design of the game is inspired by *Scrabble*, with a tile like visualisation (shown in Figure 2).

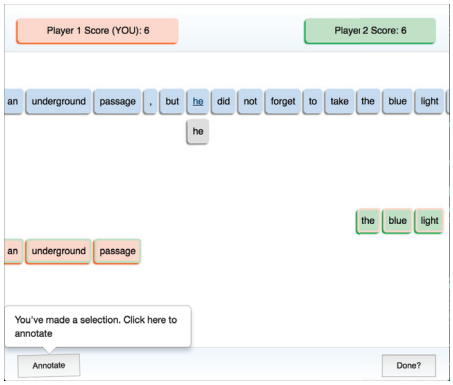

Figure 2: In game screenshot from *TileAttack*

In the game, players perform a text segmentation task which involves marking spans of tokens represented by tiles.

Our approach was to start with a game design that begins from as close as possible to an existing working recipe. We chose a design that is in many respects analogous to *The ESP Game*, but for text annotation. This provides the opportunity to test what lessons learned from games similar to *The ESP Game* still apply with text annotation games, and how, in the domain of text annotation, these lessons can be expanded upon. Like *The ESP Game*, we use the "output-agreement" format for the game, in which two players or

[1] https://tileattack.com

agents are anonymously paired, and must produce the same output, for a given input (Von Ahn and Dabbish 2008).

Gameplay

Following the documentation, but before the game, players are shown a mandatory two round tutorial, shown in Figure 3. In the tutorial players mark two sentences. They are informed of what entities are present in the sentence and how many mentions there are. They can incorrectly mark multiple items, which will be highlighted with a flashing red border, but will only be allowed to proceed once they have discovered all the correct items (shown by the glinting effect). They receive immediate and direct feedback to inform them of their progress.

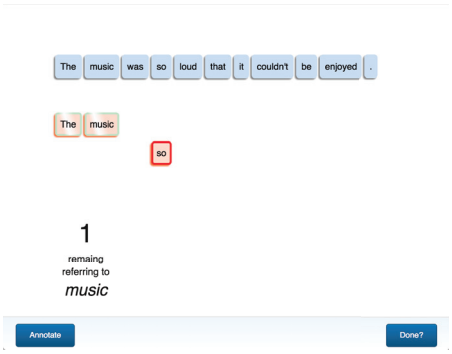

Figure 3: Tutorial screenshot from *TileAttack*

In each game round, the player is shown a single sentence to annotate. The players can choose to select a span from the sentence by simply selecting the start and end token of the item they wish to mark using the blue selection tokens. A preview of their selection is then shown immediately below. To confirm this annotation, they may either click the preview selection or click the *Annotate* button. The annotation is then shown in the player's colour.

When the two players match on a selection, the tiles for the selection in agreement are shown with a glinting effect, in the colour of the player that first annotated the tiles and a border colour of the player that agreed. The players' scores are shown at the top of the screen.

Players receive a single point for marking any item. If a marked item is agreed between the two players, the second player to have marked the item receives the number of points that there are tokens in the selection, and the first player receives double that amount. The player with the greatest number of points at the end of the round wins.

When a player has finished, they click the *Done* button, upon which they will not be able to make any more moves, but will see their opponent's moves. Their opponent is also notified they have finished and invited to click *Done* once they have finished. Once both players have clicked *Done*, the round is finished and both players are shown a round summary screen. This screen shows the moves that both players agreed on, and whether they won or lost the round.

Clicking *Continue* then takes the player to a leaderboard showing wins, losses and the current top fifteen players. From this page they may click the *Next Game* button, to start another round. On the leaderboard, players are also offered the opportunity to sign up.

Opponents

Like all two-player GWAPs, *TileAttack* chooses an artificial agent as opponent for a player if no human opponent is available. An artificial agents is also used in crowdsourcing mode, as is the case with this study. The game uses three different artificial agents as opponents. These are selected in the following order of priority, descending to the next unless the condition is met:

Silver AI Replays the aggregated result of all player games so far - if there are at least 5 games available to aggregate for that item

Replay AI Replays a recorded previous game - if a previous game is available for that item

Pipeline AI Plays the moves from an automated pipeline (modified version of a neural network approach for named entity recognition (Lample et al. 2016)).

The opponent, be it automated, a replay of another player, or an aggregation of previous player annotations, serves as the opportunity for us to use existing judgements about which we are uncertain, to feed back to players.

Progression in *TileAttack*

Worker ability and document complexity

In *TileAttack*, each worker has a linguistic ability level, starting at 0, and the documents to annotate have a readability level. The workers' linguistic level is used to select an item from a document with a matching level.

Progressing to the next level

The progression principle used in the system is that a worker progresses to the next level only after they have provided a sufficient number of high quality annotations at their current level. The key problem to be addressed is how to assess the quality of the annotations in a setting in which we do not necessarily have a gold standard. it is not sufficient to simply assume that once a worker has completed so many items to a certain accuracy they are ready to progress, as the reading levels assigned to the documents do not directly reflect the labelling complexity, and therefore, the detail required to assess the worker's competence (see Table 1 and Figure 4 for relationship between reading levels and labelling complexity).

Instead, the distribution of player accuracies against the aggregation of all worker labels for an item (**silver standard**) is used to assess an item's difficulty. A player is deemed ready to progress to the next level if they have 3 items with an accuracy (F_1) above Q3 of the interquartile range of this distribution.

	# Mentions		Length Mentions	
L	μ (σ)	**min-max**	μ (σ)	**min-max**
0	3.35 (1.44)	1-7	1.84 (1.43)	1-12
1	3.07 (1.72)	1-9	1.93 (1.45)	1-11
2	3.66 (1.53)	1-8	2.19 (1.58)	1-8
3	5.66 (4.51)	1-37	2.81 (4.03)	1-78
4	7.76 (5.02)	1-30	3.60 (4.95)	1-64

Table 1: Document level compared to the average number of mentions per item (#) and the average mention length (in tokens) - from gold annotations

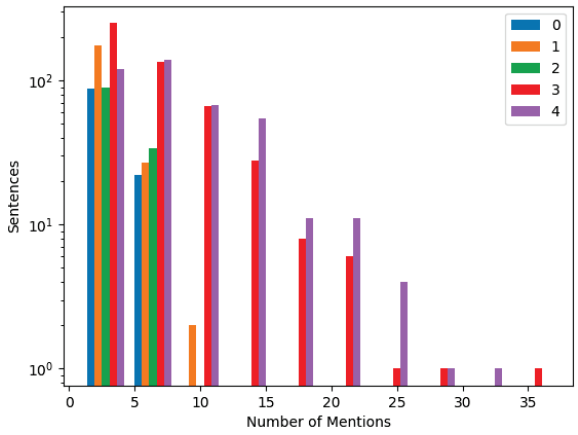

Figure 4: Mention length (tokens) for each level

Aggregation

We can expect the non-expert labelled boundaries to be quite noisy in compared to expert annotations (Snow et al. 2008). To extract "silver standard" annotations from the various non-expert judgements, once a sentence has been annotated 5 or more times, an aggregation step is performed. This step attempts to draw upon the shared wisdom of the annotators as a whole to extract a final judgement. Majority voting assumes equal skill among annotators, an assumption shown to be false in practice (Passonneau and Carpenter 2014). Instead, we use a probabilistic model to capture annotators different levels of ability. More specifically, we use a multi-class version of Dawid & Skene's model (Dawid and Skene 1979) in conjunction with our own method of clustering nested sequence labels (Madge et al. 2019). This method has been found to be at worst comparable to, and at best outperform majority voting in this particular domain (Madge et al. 2019).

The Experiment

To test the hypothesis that including a progression in *TileAttack*–starting by presenting workers with easier sentences before progressing to more complex ones once we have determined that they could reach a good quality of annotation with simpler documents–results in better accuracy than when presenting sentences in random order we ran an experiment. In the experiment, participants were asked to mark noun phrases.[2]

A between-subjects experiment design was used with two groups. The first group is presented with items from levels at random. The second group uses the *TileAttack* progression mechanism discussed earlier.

Data

In order to get texts at different levels of difficulty, we used a combination of easier texts from English learning collections and 'real,' harder texts from actual coreference corpora. Specifically, the documents at the first three difficulty levels come from the "Read in Easy English" collection available from the FLAX public repository for English learning [3]. The 'real' text include a combination of Wikipedia entries, fiction, and student reports. These are the documents that we would expect to need to annotate for a real NLP corpus, and were considered to be of level 4.

Participants

When evaluating GWAPs, it might be argued that it is best to use organically gathered players as participants, through means such as marketing the game, to stay as true to the natural setting of the application as possible. However, we believe that when testing for accuracy in a between-subjects experiment, as in this study (as opposed to when assessing engagement, retention or recruitment), the best option to nullify as many individual biases as possible is to take a micro-task crowdsourcing approach to player recruitment. Taking this approach and applying minimal filtering (as mentioned above) allows us to gather a large and varied audience of participants in a short time period. We believe the lessons learned should transfer through to an organic player base.

In particular, we use Amazon Mechanical Turk, a platform that remunerates workers on behalf of requesters to carry out small tasks. These tasks are known as *Human Intelligence Tasks* (HITs). A requester can choose from one of several Amazon Mechanical Turk templates to upload data into, or creating a custom integration. They may also specify the number of unique workers to carry out each HIT, and requirements for those workers that include qualifications. These qualifications can be awarded by the requester and serve as a flag to positively or negatively filter workers.

In our implementation, we make use of the *ExternalQuestion API*. This results in *TileAttack* being displayed in a HTML IFrame in the MTurk requester interface as a custom question. Having successfully taken part, workers are awarded an MTurk qualification to track their performance.

Experiment Design

The Amazon Mechanical Turk Workers are shown the game documentation, then taken to the tutorial. They must complete the tutorial before they are allowed to perform the

[2]Specifically, the workers were asked to mark **mentions**, the noun phrases that would be identified by a mention detection system for the use of a relation extraction system or a coreference resolution systems (Lee et al. 2011).

[3]http://flax.nzdl.org/greenstone3/flax

annotation task itself. Having completed the two tutorial rounds they are then asked to annotate three sentences. The core game mechanics, including scores or any evidence of a second player, are removed. The game like interface remains. Having completed the tutorial and three sentences, the participants are then remunerated 0.40 USD for their participation (effectively 0.08 USD/sentence). When accepting future HITs participants are not required to repeat the tutorial but are, instead, asked to annotate five sentences.

Every 5 rounds an assessment round is shown. In this round the annotator's accuracy is assessed against gold annotated data from a separate corpus. The player must score greater than or equal to 30% F_1. If the player fails to stay above this level they are not allowed to continue. This is a low barrier put in place only to remove spammers from the task, not the less capable annotators.

Results

We take two perspectives in our results, a user focused perspective, and an output focused perspective.

The user focused perspective looks at the average accuracy of **player games** in each group. This provides a picture of the effect of the two experiment treatments on the players understanding and ability to perform the tasks. For example, a high standard deviation here, shows a high spread in player ability.

The output focused perspective looks at the accuracy of the annotations on all of the **sentences**, subject to the application of **probabilistic aggregation**. This is the final quality we can expert from the system at that level. Probabilistic aggregation, by design, eliminates some of the less able players annotations. As such, this evaluation perspective alone does not give us a complete picture of the impact of the experiment treatment on the players.

Worker Focused Perspective

We ran an experiment with 149 workers in the *progression group* playing 3,875 games, and 156 workers in the *random group* playing 5,669 games. Both groups show the typical Zipfian distributions in terms of contribution (Figures 5 and 6). We excluded any contributions from workers that did not play at least 3 games.

Table 2 shows the average precision, recall and F_1 at the different levels for the two groups of random and progressive difficulty respectively. In levels 3 and 4, where the tasks are more difficult, we see a significant difference when evaluating the players games against the gold standard. The groups that have been delivered tasks progressively in line with their ability score much higher. This is particularly evident with recall.

Figure 7 shows a box plot of recall for levels 2-4 - those for which there is statistical significance (see Table 3). On the whole, the *progression* group has a tighter distribution, with a lower standard deviation than the *random* group, in the more challenging levels. This is also visible in Table 2, particularly in the precision.

Figure 8 and Table 3 shows the difference in F_1 accuracy between the *random* and *progression* groups across the lev-

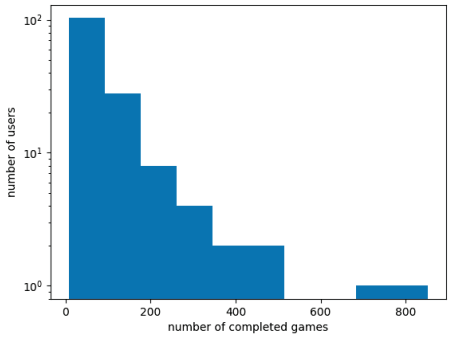

Figure 5: Distribution of worker contribution in the *progression* group

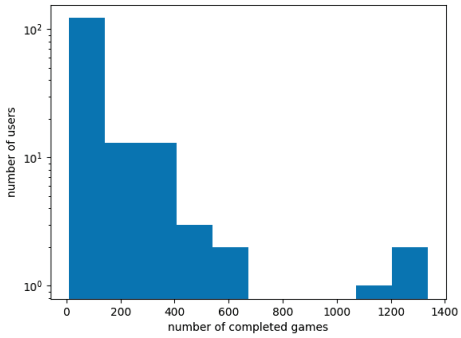

Figure 6: Distribution of worker contribution in the *random* group

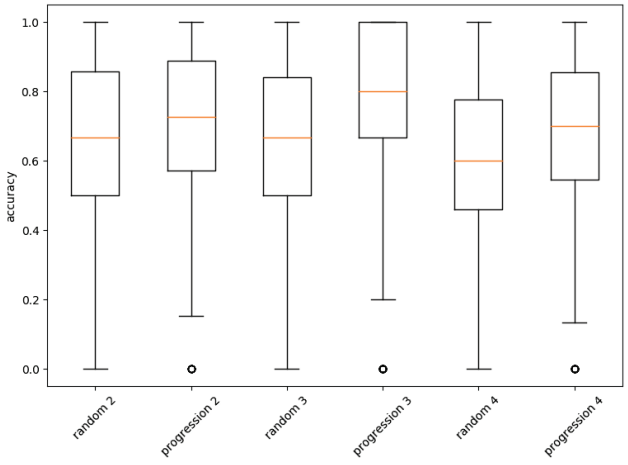

Figure 7: Player Game F_1 on levels 2-4 for *random* and *progression* groups against gold standard

	Random Group				Progression Group			
L	**# games**	**Precision** μ (σ)	**Recall** μ (σ)	F_1 μ (σ)	**# games**	**Precision** μ (σ)	**Recall** μ (σ)	F_1 μ (σ)
0	1059	73.8 (0.266)	85.4 (0.212)	76.3 (0.217)	623	69.0 (0.300)	76.7 (0.256)	68.7 (0.253)
1	1289	69.4 (0.288)	86.4 (0.222)	73.5 (0.238)	592	69.8 (0.317)	78.1 (0.268)	70.2 (0.270)
2	1184	64.5 (0.279)	78.4 (0.244)	67.2 (0.230)	505	71.7 (0.263)	75.2 (0.239)	71.0 (0.223)
3	1424	64.7 (0.284)	74.0 (0.265)	65.7 (0.245)	1337	83.9 (0.220)	75.1 (0.241)	77.3 (0.210)
4	713	62.9 (0.273)	66.1 (0.265)	61.1 (0.237)	818	78.9 (0.235)	64.2 (0.258)	68.5 (0.227)

Table 2: Accuracy for worker games - *random* vs. *progression* groups **exact boundary evaluation** (rounded to 1 dp)

els. Mann-Whitney U test is used to test for statistical significance. Whilst the *random* group appears to outperform the *progression* group in the lower levels (0 and 1), this difference is not statistically significant. (We hypothesize that this non-significant difference may be due to the fact that in the progression group only inexperienced players ever tackle those sentences, whereas in the random group the players tackling level 0 sentences might do so as their first sentence, or their last, after gaining much experience.) For sentences at levels 2-3, however, the *progression* group outperforms the *random* group by a large margin, particularly in level 3 (11.56%), and this difference is statistically significant.

L	**Random** F_1 μ (σ)	**Progression** F_1 μ (σ)	**Difference**	**P-Value**
0	76.3 (0.217)	68.7 (0.253)	-7.58	1.000
1	73.5 (0.238)	70.2 (0.270)	-3.32	0.973
2	67.2 (0.230)	71.0 (0.223)	+3.79	0.001
3	65.7 (0.245)	77.3 (0.210)	+11.56	0.000
4	61.1 (0.237)	68.5 (0.227)	+7.39	0.000

Table 3: F_1 for worker games - *random* vs. *progression* groups with Mann-Whitney U test **exact boundary evaluation** (rounded to 1 dp)

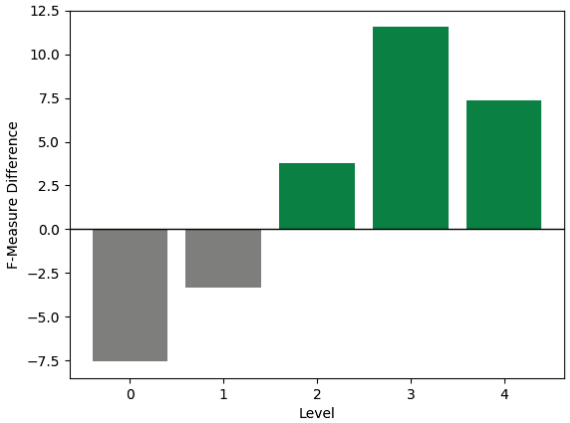

Figure 8: F_1 difference between *random* and *progression* groups

Sentence Focused Perspective

Next, we compared the two groups with respect to the quality of the annotation of a sentence as a function of that sentence's complexity. For this analysis we considered only items (sentences) with at least 3 games played. This leaves us with **688 items** for the *random* group and **657 items** for the *progression* group. We take at most the first 5 games for each item. No worker plays a game on a single item more than once. A probabilistic aggregation method is used (the very same used as part of determining an items difficulty. The results suggest that both groups do best on the easiest items, at level 0, but as item difficulty increases, the accuracy begins to decrease. However, the *progression* group is far more resistant to the increase in difficulty. At the start, the *random* group does slightly better, just as in the case of the worker-centered evaluation; and again, we hypothesize that this may be due to fact that some of the random players may have been playing for a long time and gained some expertise before getting to the sentences at level 0, whereas the progression players would have all been beginners at level 0. (Figure 9).

Figure 9 shows the F_1 of aggregation at the respective levels for items labelled by both groups. As one might expect, with items labelled by the *random* worker group, as the difficulty increases throughout the levels, the accuracy decreases. However, in the items labelled by the *progression* group, whilst the accuracy of the items decreases for the first two levels in line with the increasing difficulty, the remaining levels are are far more resilient to the increasing difficulty.

Discussion and Conclusions

In this paper, we presented a method for introducing progression in a text-labelling GWAP that we believe should apply to any text-labelling task that supports aggregation and varies in difficulty, but cannot be broken down into easily identifiable distinct skills. We use general, domain agnostic readability levels for identifying item difficulty, but our assessment of player ability is based on agreement against aggregation. We demonstrated this approach with a sequence labelling task of identifying candidate mentions and evaluated against randomly assigning items to players. The approach was tested via micro-task crowdsourcing in order to controlling the between-participants nature of the study, and nullify the individual biases present with organic players by gathering a much larger audience.

	Random Group				Progression Group			
L	**# items**	**Precision**	**Recall**	F_1	**# items**	**Precision**	**Recall**	F_1
0	55	90.3	86.4	88.3	55	88.5	87.5	88.0
1	103	85.2	85.5	85.4	102	86.7	85.6	86.1
2	62	82.8	78.4	80.5	62	83.7	79.3	81.4
3	256	79.9	75.3	77.5	240	90.1	80.6	85.1
4	212	78.5	66.8	72.2	198	90.8	74.9	82.1
all	688	80.5	73.1	76.6	657	89.5	79.0	83.9

Table 4: Accuracy at levels

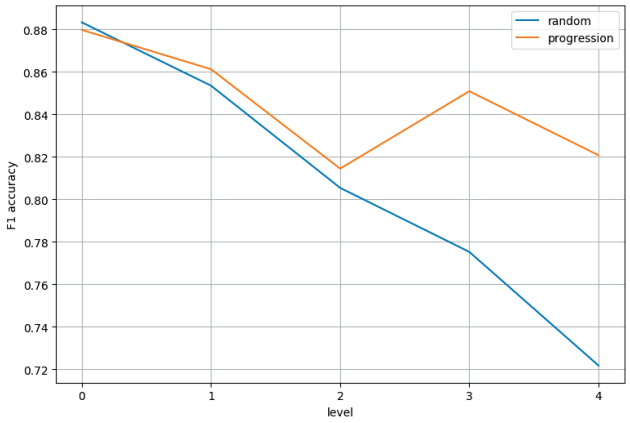

Figure 9: F_1 of probabilistic aggregation of annotations on items for *random* and *progression* groups

Our results demonstrate noticeable benefits to applying this strategy. On average, workers with the progression treatment perform considerably better on more difficult items than those who play randomly (all with a high statistical significance).

There is a similar picture with the resulting output of the system. The aggregation of the labels provided by the progression group are much more resistant to the increasing difficulty than those provided by the random group.

Data from the experiment has been made available on the *TileAttack* website [4].

Acknowledgements

This research was supported in part by the EPSRC CDT in Intelligent Games and Game Intelligence (IGGI), EP/L015846/1; in part by the DALI project, ERC Grant 695662.

References

Adams, E. 2008. The designer's notebook: Difficulty modes and dynamic difficulty adjustment.

Bachrach, Y.; Graepel, T.; Minka, T.; and Guiver, J. 2012. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*.

Basu Roy, S.; Lykourentzou, I.; Thirumuruganathan, S.; Amer-Yahia, S.; and Das, G. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal—The International Journal on Very Large Data Bases* 24(4):467–491.

Boutros, D. 2008. Difficulty is difficult: Designing for hard modes in games.

Carpenter, B. 2008. *Multilevel bayesian models of categorical data annotation. Available at http://lingpipe-blog.com/lingpipe-white-papers*.

Carroll, J. M., and Thomas, J. M. 1988. FUN. *ACM SIGCHI Bulletin* 19(3):21–24.

Csikszentmihalyi, M. 1990. Flow: The psychology of optimal performance.

Dawid, A. P., and Skene, A. M. 1979. Maximum Likelihood Estimation of observer error-rates using the EM algorithm. *Applied Statistics* 28(1):20–28.

Downs, J. S.; Holbrook, M. B.; Sheng, S.; and Cranor, L. F. 2010. Are Your Participants Gaming the System? Screening Mechanical Turk Workers. 4.

Dumitrache, A.; Aroyo, L.; Welty, C.; Sips, R.-J.; Levas, A.; et al. 2013. Dr. Detective: combining gamification techniques and crowdsourcing to create a gold standard for the medical domain.

Fort, K.; Guillaume, B.; and Chastant, H. 2014. Creating zombilingo, a game with a purpose for dependency syntax annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, 2–6. ACM.

Hamari, J.; Shernoff, D. J.; Rowe, E.; Coller, B.; Asbell-Clarke, J.; and Edwards, T. 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior* 54:170–179.

Hladká, B.; Mírovský, J.; and Kohout, J. 2011. An attractive game with the document: (im)possible? *The Prague Bulletin of Mathematical Linguistics* 96(1).

Hossain, M. 2012. Crowdsourcing: Activities, incentives and users' motivations to participate. In *2012 International Conference on Innovation Management and Technology Research*, 501–506. Malacca, Malaysia: IEEE.

Hung, C.-Y.; Sun, J. C.-Y.; and Yu, P.-T. 2015. The benefits of a challenge: student motivation and flow experience

[4] https://tileattack.com/data

in tablet-pc-game-based learning. *Interactive Learning Environments* 23(2):172–190.

Hunicke, R. 2005. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 429–433. ACM.

Ipeirotis, P. G., and Gabrilovich, E. 2014. Quizz: targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World wide web - WWW '14*, 143–154. Seoul, Korea: ACM Press.

Jurgens, D., and Navigli, R. 2014. It's all fun and games until someone annotates: Video games with a purpose for linguistic annotation. *TACL* 2:449–464.

Kaufmann, N.; Schulze, T.; and Veit, D. 2011. More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk. 12.

Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*, 1301. San Antonio, Texas, USA: ACM Press.

Lafourcade, M. 2007. Making people play for lexical acquisition with the jeuxdemots prototype. In *SNLP'07: 7th international symposium on natural language processing*, 7.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. *arXiv:1603.01360 [cs]*. arXiv: 1603.01360.

Lee, H.; Peirsman, Y.; Chang, A.; Chambers, N.; Surdeanu, M.; and Jurafsky, D. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 28–34. Association for Computational Linguistics.

Lee, S.; Park, S.; and Park, S. 2014. A quality enhancement of crowdsourcing based on quality evaluation and user-level task assignment framework. In *Big Data and Smart Computing (BIGCOMP), 2014 International Conference on*, 60–65. IEEE.

Madge, C.; Yu, J.; Chamberlain, J.; Kruschwitz, U.; Paun, S.; and Poesio, M. 2019. Crowdsourcing and Aggregating Nested Markable Annotations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 797–807. Florence, Italy: Association for Computational Linguistics.

Malone, T. W. 1981. Toward a theory of intrinsically motivating instruction. *Cognitive science* 5(4):333–369.

Mc Laughlin, G. H. 1969. Smog grading-a new readability formula. *Journal of reading* 12(8):639–646.

Passonneau, R. J., and Carpenter, B. 2014. The benefits of a model of annotation. *Transactions of the ACL* 2:311–326.

Paun, S.; Carpenter, B.; Chamberlain, J.; Hovy, D.; Kruschwitz, U.; and Poesio, M. 2018. Bayesian annotation methods for NLP: An evaluation. *Transactions of the ACL* 6:571–585.

Poesio, M.; Chamberlain, J.; Kruschwitz, U.; Robaldo, L.; and Ducceschi, L. 2013. *Phrase Detectives*: Utilizing collective intelligence for internet-scale language resource creation. *ACM TiiS* 3(1):3.

Schell, J. 2014. *The Art of Game Design: A book of lenses*. AK Peters/CRC Press.

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*, 254–263. Association for Computational Linguistics.

Sweetser, P., and Wyeth, P. 2005. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3(3):3–3.

Tuite, K. 2014. GWAPs: Games with a Problem. 7.

Venhuizen, N.; Evang, K.; Basile, V.; and Bos, J. 2013. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*.

Von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *SIGCHI*, 319–326. ACM.

Von Ahn, L., and Dabbish, L. 2008. Designing games with a purpose. *Communications of the ACM* 51(8):58–67.

von Ahn, L.; Ginosar, S.; Kedia, M.; and Blum, M. 2007. Improving Image Search with PHETCH. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, IV–1209–IV–1212. Honolulu, HI: IEEE.

von Ahn, L.; Liu, R.; and Blum, M. 2006. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, 55. Montréal, Québec, Canada: ACM Press.