Deliverable D2.3: EHC Framework incl. Workflow Optimisation

MODUL University Vienna

WP 2 (T2.1 - T2.5)

Version History			
Version	**Author**	**Date**	**Comments**
0.1	Michael Föls	01/03/2016	Document structure and initial version
0.2	Arno Scharl	08/03/2016	Major revision
0.3	Michael Föls	10/03/2016	Refine various sections
0.4	Patrick Paroubek	10/03/2016	Active learning section
0.5	Patrick Paroubek	12/03/2016	Experimental workflow optimization part
0.6	Kalina Bontcheva	12/03/2016	NER pre-empting
1.0	Michael Föls	15/03/2016	Final revision and layout
1.1	Arno Scharl	10/08/2016	Minor citation update

Abstract The uComp platform combines the two Human Computation (HC) genres: games with a purpose and mechanized labour. This deliverable presents an overview of the conception, development and usage of the uComp HC Framework. We provide an analysis of both genres and describe how their benefits have been leveraged in the HC-Framework. D2.3 tackles the topics system design, multi-channel task deployment, task management, user profiling, quality control methods and workflow optimization. As such it covers work performed as part of tasks T2.1, T2.2, T2.3, T2.4 and T2.5.

Table of Contents

1 Introduction

This deliverable presents an overview of the conception, development and usage of the Human Computation Framework (HC-Framework) of the uComp research project. It tackles the topics system design, multi-channel task deployment, task management, user profiling, quality control methods and workflow optimization.

The HC-Framework plays a core part in the uComp project. During the first year, we have designed the framework (e.g., define core task types, agree on an API with partners) and started its implementation by extending MOD's existing application framework for social media platforms. The following system features have been implemented: multi-channel deployment and social logins to ensure that games can be played from mobile devices; viral notification system, allowing the possibility to invite friends; a bridge to CrowdFlower, allowing the publication of game elements as Human Intelligence Tasks (HITS) on mechanised labour marketplaces.

In the second year, the API was tested and reviewed within a small team, to ensure the stability of the code, the clarity of the documentation and also the range of the functionality of the API. During the process a small subset of partners used the API to test their use cases. The feedback of the testers was used to address several minor issues and extend the documentation of the API in cases where the initial version was unclear. Since the API of the system is a very central component that cannot be changed easily once it is being used by many people, this process was carefully planned and executed.

To streamline the development process, improve the reliability and ensure system independent deployment of the developed games, the framework was then reworked and implemented as a Docker container. Also a custom Docker registry is used to have a backup-strategy in place and to provide a scaleable architecture for the gaming platform.

At first the game was intended to only use the login capabilities of Facebook, but in order to maximize the number of possible participants (and create a generic login framework for other projects as well), a more comprehensive Single-Sign-On component (using Shibboleth and SimpleSAMLphp) was created. The core SimpleSAMLphp functionality was extended with custom plugins to authenticate via Facebook, Twitter and Google. This should allow all interested players to participate in the game. The authentication framework set the stage for launching two concrete game applications, the Language Quiz and the Climate Challenge.

The crowdsourcing engine and the individual game components were continually refined based on the feedback from beta testers and early adopters. Special emphasis was placed on the interaction design - e.g., the scoring and engagement mechanisms, as well as the flexible support of different task types - e.g., questions where the correct answer will never be known, or questions where the correct answer will be determined at a future point in time (in the former case, the points are based on the mean answer of all players; in the latter case, the points are awarded ex post once the correct answer is known).

Cheating prevention is being addressed through strategies customised to the specifics of knowledge extraction tasks, including (i) economic models to ensure cheaters do no better than break-even; (ii) defensive task design to encourage users to put in genuine efforts to carry out the tasks; and (iii) statistical models for quality and agreement enforcement that identify outliers and compare contributor answers against gold standard data. Task and progress monitoring

during the human computation cycle can also amplify the incentive mechanisms. Monitoring processes can be exposed as REST services through the API, which allows researchers to define alerts or configure automatic behaviour rules.

In the final year, the developed HC-Framework was used through the two games to collect answers from players and solve tasks. Also a Crowdflower promotion mode was created, to allow the promotion of the games through Crowdflower. Since this mode is used to redirect the Crowdflower users to the game, it is possible to acquire players through this method (in contrast to the traditional Crowdflower users who will only stay at the Crowdflower plattform).

2 A Comparison of Crowdsourcing Genres

Mechanised labour and games with a purpose are the two most popular human computation (HC) genres, frequently employed to support research activities in fields as diverse as natural language processing, semantic web or databases. Mechanised labour (MLab) is a type of paid-for HC genre, where contributors choose to carry out small tasks (or micro-tasks) and are paid a small amount of money in return (often referred to as micro-payments). Popular platforms for mechanised labour include Amazon's Mechanical Turk (MTurk) and CrowdFlower(CF) which allow requesters to post their micro-tasks in the form of Human Intelligence Tasks (or HITs, or units) to a large population of micro-workers. Games with a purpose(GWAP) enable human contributors to carry out computation tasks as a side effect of playing online games [42]. An example from the area of computational biology is the Phylo game (phylo.cs.mcgill.ca) that disguises the problem of multiple sequence alignment as a puzzle like game [21].

Research projects typically rely on either one or the other of these genres, and therefore *there is a general lack of understanding of how these two genres compare and whether and how they could be used together to offset their respective weaknesses*. For the uComp project, this is a key question to be answered as the planned HC platform aims to span these two genres.

We addressed this question through a series of methods, documented in [32]. In this section we summarize our main finding, and point readers interested in details to the published paper. Our methodology consisted on two instruments: a literature study and an experimental investigation.

As the first part of our investigations, we identified the differences between the two genres, primarily in terms of *cost, speed and result quality*, based on **existing studies in the literature**. The findings illustrated in Table 3 lead to the conclusion that there is a significant complementarity between the two genres, along all key dimensions (cost, speed, quality) and that this fact could be leveraged for building hybrid HC systems that exploit the benefits of both genres simultaneously. For example, complex, interesting tasks could be performed by a dedicated, well-trained player base (on a longer term and virtually for free), while more "boring" tasks that would reduce the motivation of players might be more suitable for execution by intrinsically motivated micro-workers, for a small amount of money.

Starting from these hypotheses above, and as a second methodological step, we aimed to quantify these genre differences through a comparative study that involved performing the same knowledge acquisition task with the Climate Quiz game (see Figure 1) on the one hand and through a similar mechanised labour interface, on the other.

Table 4 sums up our observations when comparing the two HC genres and compares them to the results in [40], a similar study that compares the two genres experimentally albeit on another knowledge acquisition task. Overall, we conclude that the study's findings demonstrate that the two genres are highly complementary, which not only makes them suitable for different types of projects, but also opens new opportunities for building cross-genre human computation solutions that exploit the strengths of both genres simultaneously.

Feature	MLab	GWAP	References
Cost			
Set-up Price	Low(+)	High(-)	[27, 40, 43]
Price per task	Low(-)	None(+)	[27, 40]
Speed			
Set-up Time	Low (+)	High(-)	[27, 40, 43]
Throughput	High(+)	Low(-)	[10]
Throughput predictability	High(+)	Low(-)	[10, 40]
Quality			
Quality	Low(-)	High(+)	[10, 43]
	High(+)	High(+)	[40]
Maintaining motivation	Easy(+)	Difficult(-)	[40]
Incentive to cheat	High(-)	(Mostly) Low (+)	[10, 43]
Task complexity	Low(-)	High(+)	[10]
Importance of task interestingness	Low(+)	High(-)	[40, 46]
Worker diversity	Low(-)	High(+)	[40]
	High(+)	Low(-)	[43]
Other			
Ethical issues	Yes(-)	(Mostly) No(+)	[15]

Table 3: Advantages and disadvantages of mechanised labour and GWAPs.

3 Hybrid-Genre Workflows

Continuing from the conclusions of our comparative study (Section 2), we experimented with the notion of hybrid-genre crowdsourcing workflows aiming to understand (1) if these are feasible and, if yes, (2) to estimate the improvement they bring over single-genre approaches. This part of our work has been documented in a journal paper [30], and therefore this section only provides a summary of the main outcomes, while readers interested in further details are pointed to the article.

3.1 Evaluation of a Single-Genre Approach to Knowledge Creation

To create a baseline of single-genres approaches, we performed an evaluation of the Climate Quiz GWAP and compared the results with those of similar knowledge acquisition games.

As depicted in Figure 1, Climate Quiz invites Facebook users and their online friends to evaluate whether two concepts presented by the system are related (e.g. *environmental activism*, *activism*), and which label is the most appropriate to describe this relation (e.g. $is\ a\ sub-category\ of$). The system controls the types of relations between concept pairs, focusing both on generic ($is\ a\ sub-category\ of$, $is\ identical\ to$, $is\ the\ opposite\ of$) and on domain-specific

	Study Observations		Thaler et al. [40]	
Feature	**CrowdFlower**	**Climate Quiz**	**MTurk**	**OntoPronto**
Cost				
Set-up Price	$450	$9,000	est. $4,500	est. $22,500
Price per unit	$0.183	$0	$0.74	$0
Speed				
Set-up Time	2 days	2 months	1 month	5 months
Throughput	243	180	-	-
Throughput predictability	within hours	completion difficult to estimate	-	-
Quality				
Precision	CF1= 59%	72%	99%	97%
	CF2= 75%	72%		
Maintaining motivation	no effort to recruit micro-workers	significant effort for recruiting players	easy (financial)	difficult
Task complexity	similar	similar	similar	similar
Importance of task interesting	micro-workers solve all tasks	players skip many tasks	-	-
Worker diversity	83	648	16	270

Table 4: Comparison of mechanised labour and games based on our observations and [40].

(*opposes*, *supports*, *threatens*, *influences*, *works on/with*) relations. Two further relations, *other* and *is not related to* were added for cases not covered by the previous eight relations. The game's interface allows players to switch the position of the two concepts or to skip ambiguous pairs.

Climate Quiz acts as a game with a purpose with the main aim of collecting knowledge assets to support an ontology learning algorithm [45]. A human-machine workflow is therefore established as depicted in Figure 2. The "machine" part of the workflow is the ontology learning algorithm that extracts terms from unstructured and structured data sources. The term pairs that are most likely related based on the algorithm's input data sources are subsequently sent to Climate Quiz, where the human element of the workflow assigns relations to these pairs. These relations are fed back into the algorithm which uses them to perfect the learned ontology and to derive new term pairs that should be connected.

Evaluation Results: Based on the evaluation of the game (see details in [30]), we conclude that while Climate Quiz has attracted a significant number of players (the highest number of all knowledge acquisition games) and managed to build a 50+ core community of players (as opposed to only 10 in PhraseDetectives), it has achieved only medium average lifetime play (ALP) values. Additionally, its throughput was the lowest of all games and so was the agreement of the game results with the gold standard dataset.

The evaluation also revealed the high difficulty of the task which we assume to be the

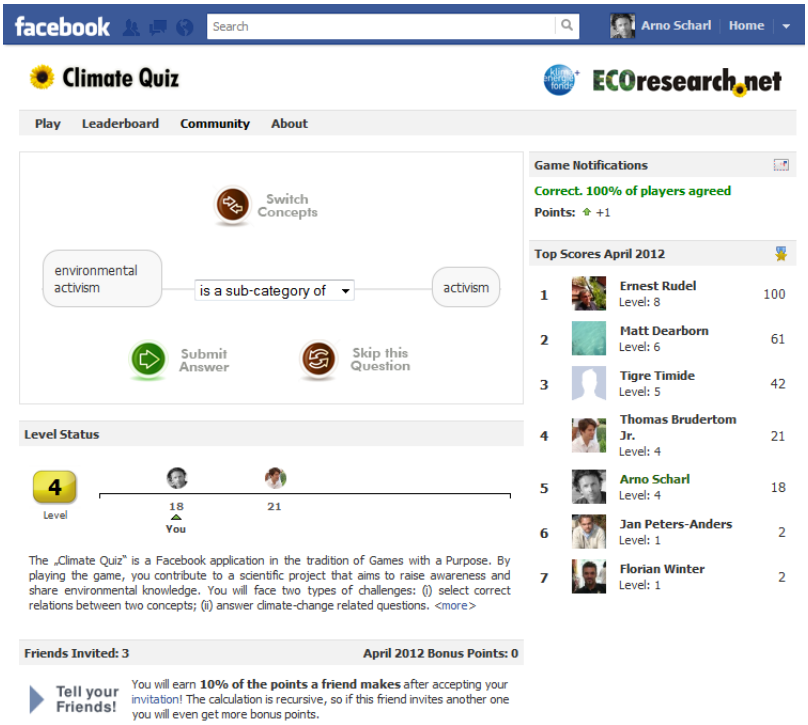


Figure 1: The Climate Quiz Interface.

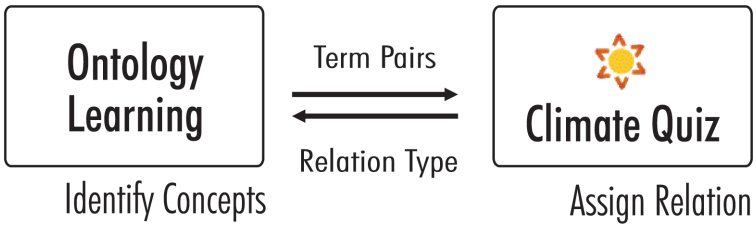


Figure 2: Human-machine workflow involving Climate Quiz and an ontology learning algorithm.

main cause of players playing the game for short intervals only (hence the average ALP) and providing results that have a low quality when compared to other games (although, in line with the quality provided by paid annotators). More specifically, we distinguish two core problematic issues that lead to the limitations of the game.

1. Firstly, the game is fed *noisy input data*, generated automatically by the ontology learning algorithm and containing terms that are ambiguous, obscure or do not make sense at all. A severe negative effect is that such confusing terms frustrate players and reduce the enjoyment of the game, which is the main motivational factor Climate Quiz relies on. Therefore, frustrated players play less (lower ALP) and are likely to lose motivation and leave the game, thus preventing the game from maintaining a stable community over long periods of time and jeopardizing its long-term success. Noisy input data also leads to wasting precious game resources (i.e., players' time and effort) on obscure terms and

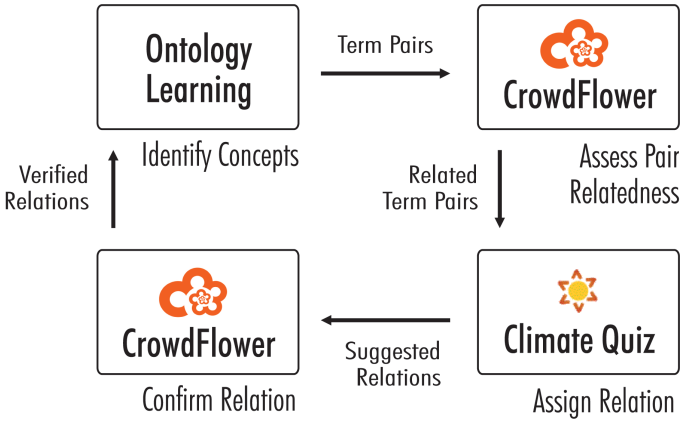


Figure 3: Hybrid-genre workflow.

inherently to low performance as disagreement tends to be high on these ambiguous pairs.

2. Secondly, the game *loses good quality output data* As discussed in [30], the high number of relations to choose from and their semantic overlap often lead to cases when a pair of terms can be correctly related with multiple relations (e.g., *threatens* and *influences*), Climate Quiz, however, derives a single relation between any input pair using a majority voting based mechanism causing that less popular but still correct relations are excluded from the final result set. For example, the game assigned the relation *works on/with* to the pair (*green industry*, *clean energy products*) as the most popular one, and therefore did not include the relation *supports*, which was the second most popular relation voted by the players and can be considered a correct relation.

3.2 Hybrid-Genre Crowdsourcing Workflows

As a way to mitigate the problematic issues discussed above, we propose a workflow that combines two different crowdsourcing genres in order to leverage their complementary strengths, hence the term hybrid-genre workflow. In our context, and considering the pros and cons of crowdsourcing genres discussed in Section 2, we assign simple (and boring) tasks to microworkers and keep more complex (but interesting) tasks for game players thus ensuring game enjoyment and reinforcing players' intrinsic motivation. Therefore, our workflow is novel compared to the existing workflow types (described in Section 2.3 of [30]), which either relied on a single crowdsourcing genre (most frequently mechanised labour) or combined machine and human computation. Concretely, our workflow has three stages (see Figure 3).

1. **Stage 1: Judge Pair Relatedness.** This stage addresses the problem of noisy input data by asking CrowdFlower workers to check which pairs of terms extracted by the ontology learning algorithm might be related before feeding these into the game. Acting similarly as the "Find" phase of the Soylent workflow [7], this stage detects the problem instances worth investigating and therefore reduces the ambiguity of the input data. We hypothesize that this will lead to several positive effects such as (i) a more enjoyable

game resulting in higher player motivation and retention as well as (ii) higher quality game results in terms of better agreement with the gold standard.

2. **Stage 2: Assign Relation.** Climate Quiz is used in this stage to solve the complex problem of assigning one of ten relations between term pairs resulting from Stage 1. As such it corresponds to the "Fix" phase of the Soylent workflow which solves the problem instances identified in the previous Find phase.

3. **Stage 3: Check Relation Correctness.** This stage asks workers to assess the correctness of the relations assigned in stage 2 above (similarly to Soylent's "Verify" stage). As such, it should further increase the quality of the game's output but also extend it with potentially correct but rejected relations thus alleviating the problem of losing good quality output data.

By **evaluating** the precision of the results, the introduction of Stage 1 already provided an improvement of 4% over the precision obtainable by Climate Quiz alone (76% up from 72%). Stage 3 further raised the precision of the task to 78%. Therefore, the hybrid workflow, could, in principle, lead to a **6% increase in the quality of results obtainable by single-genre approaches**. Additional benefits, which were not explicitly evaluated during our experiments but need to be verified in the future include: **reducing task execution times** (as a significant number of obscure pairs are quickly filtered out by CF in Stage 1), **improving the quality of input data** and as a result providing a **more positive player experience**, which will presumably lead to **longer average lifetime play** and more contributions.

In the rest of the deliverable we focus on issues related to composing tasks and outsourcing them to diverse platforms as part of hybrid-genre workflows.

4 Composition Model

In this section we provide a detailed description of the composition model in terms of its inputs and outputs. Figure 4 depicts graphically the main elements of the composition model, which are explained in detail next:

Input from Requester:

- J is a job provided by the requester, which contains N tasks;
- JPT is the number of judgments requested for each task T of job J;
- t_{dl} is the deadline for finishing a job (that is obtaining all $N * JPT$ judgements from the platform);
- $Q = \{Q_m, Q_{tv}\}$ is a tuple specifying the output quality expectations of the requester, in terms of a quality measure (Q_m) and its expected threshold value (Q_{tv}). The quality measurement can be performed in various ways, for example, (1) as the number of judgements expected per task; (2) by computing the average agreement of workers for a task (e.g., all tasks with agreement lower than a threshold value Q_{tv} should be kept in the system and more judgments should be collected for them); (3) as the agreement of the collected judgements with a gold standard, if available; (4) as the agreement with objectively verifiable questions included in the HIT. Defining and implementing quality control measures is part of task T2.4, to start in M13 of the project.
- M monetary resources as a sum of available money;
- $W = \{(W_{ch}, W_v)\}$ - a collection of objectively measurable worker constraints and their values e.g., geo location, skills, previous accuracy (on similar task types), etc;
- $mode = \{quiz, crowdflower, hybrid\}$ - the requester can choose the type of crowdsourcing genres to be used for his task; for now, we envision three possibilities: using only GWAPs ($quiz$), using only CF ($crowdflower$ - in this case, the uComp framework will act as a wrapper to CF), using hybrid-genres approaches, in which case the uComp framework dynamically decides, based on priorities, which part of the tasks is solved via games and which via CF. Current implementation covers the first two modes.

Input available in the uComp platform:

- $U = \{(U_{ch}, U_v)\}$ - a collection of objectively measurable user characteristics and their values e.g., geo location, skills, previous accuracy (on similar task types), etc;
- TT - task types (for now only classification).

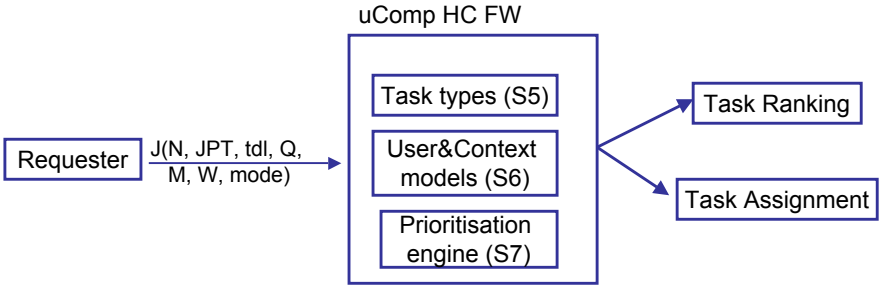


Figure 4: Overview of the composition model.

Expected Output:

1. **Ranking of HC tasks by priority**: $R = \{(T_k, P_{T_k,J_i,t_1})\}$, where each task k of each current job i, T_k, is assigned at time point t_1 a priority value P_{T_k,J_i,t_1}. The tasks with the highest priority values should be resolved first. A set of initial prioritisation algorithms are described in Section 7.

2. **Assignment of HC tasks to the appropriate platform/contributor**: each task k of each job i, T_k, is assigned to an appropriate platform Pf_m and a suitable user u_n, namely: $A = \{(T_k, Pf_m, u_n)\}$. Note that the development of matching algorithms between user profiles and task requirements is envisioned as taking part during the third stage of T2.3, and therefore will not be discussed in this deliverable.

5 HC Task Types in uComp

The uComp framework supports a wide range of HC task types, including:

Classification tasks through which contributors select one or more values from a set of given values. Several variants of this task are discussed in Section 5.1.

Opinion Poll tasks obtain survey-like results from players.

Prediction tasks ask the player to share their opinion about a future condition.

Pledges ask participants for feedback on recommendations for reducing energy consumption and making sustainable lifestyle choice. They can declare if they have already adopted the recommendation, rate the environmental impact of this action, and share accepted pledges via their social media channels.

From the tasks types above, classification style tasks tend to be the most frequently used. Furthermore the other three task types are limited to the Climate Challenge as they present a customized user interface that can not be easily reproduced in Crowdflower. This means only variations of the classifcation task can be mapped to Crowdflower. Therefore, these types of tasks are discussed in more detail in the next section (Section 5.1). Additionally, we present the API-level mapping between the uComp classification tasks and CrowdFlower HITs in Section 5.2 and summarize the implementation details of the uComp to CrowdFlower bridge, a software component that allows automatic crowdsourcing of uComp tasks through the CrowdFlower mechanised labour marketplace, thus enabling the construction of hybrid-genre workflows.

5.1 Hybrid Crowdsourcing Tasks

Classification style tasks allow users to select one (or more) values from a given set of values. Depending on the type of the values as well as the size of the value set, we distinguish two major categories of classification tasks.

Selecting between multiple categories - in this case, the set of values consists of a set of discrete values. When the number of these values is two, the classification problem becomes a *Binary Choice* problem.

Selecting from a range of values - in this case, users select a point in a continuous interval of values.

To exemplify these tasks, we provide examples from the seminal work of Snow [39]. For each task type, we detail the input/output data and provide a screenshot of the uComp interface as well as a set of uComp API parameters that would need to be instantiated to create such a task. Identifying and detailed these HC tasks was performed with the entire consortium (lead by MOD/WU and relying on feedback from USFD/LIMSI) and helped in defining the uComp

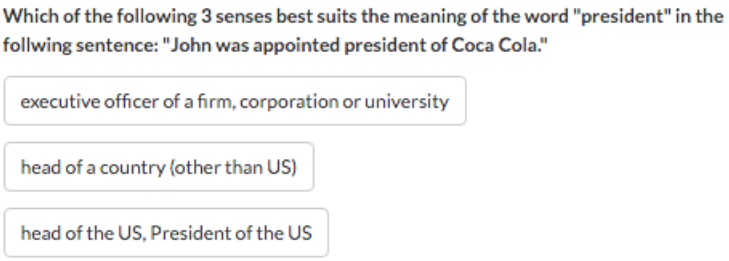


Figure 5: Interface for the classification task .

API in a way that meets the expectations of all partners[1].

Task Title: TC1: Multiple Categories

Task Description Snow et al perform a word sense disambiguation (WSD) task where the workers are presented with a text snippet containing the word *president* and they have to decide which one of three possible word senses is the most appropriate to describe the word *president* in that text (see section 4.5 of [39]).

Task Interface See Figure 5

Task Input Data A set of sentences containing the word *president*:

- John was appointed president at Coca Cola company
- President Obama loves apples.
- The President of Austria is the federal head of state of Austria.

Task Output Data

- the individual category choices of each worker, e.g., President Obama loves Apples, 2, 3, 3 (meaning that senses 2, 3, 3 were chosen in the case of this sentence).
- an aggregated value based on the chosen aggregation method (majority vote, weighted vote considering worker trust). Some aggregation methods might also return a confidence value. e.g., President Obama loves Apples, 3, 66%

API Call

- General
  - Title: Selecting a word sense for the word president.
  - Category: Classification
  - Instruction: Which of the following 3 senses.
- Specific

[1]The documentation of the uComp API can be found at `http://soc.ecoresearch.net/facebook/election2008/ucomp-quiz-beta/api/v1/documentation/`

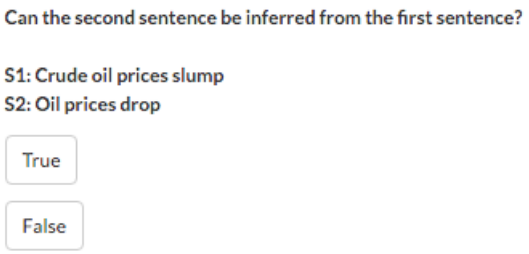


Figure 6: Interface for the binary choice task.

  - Default Categories: (1, "executive officer of a firm, corporation or university"), (2, "head of a country (other than US)"), (3, "head of the US, President of the US")
  - Multiple Choice: false
  - Judgements per unit: 10
  - User Characteristics (location, languages, trust): (loc, US), (lang, EN), (trust, 80)
  - Aggregation: majority vote
  - Task deadline: 30.11.2013

- Data
  - John was appointed president at Coca Cola company, [cat-1, cat-2, cat-3]
  - President Obama loves apples.
  - The President of Austria is the federal head of state of Austria.

Task Title: (TC2) Binary Choice

Task Description Snow et al (see Section 4.3 of [39]) ask workers to judge whether a sentence can be inferred logically from the other. The workers are presented with two sentences (this is the variable data) and can choose True or False depending on whether the second sentence can be inferred from the first one (i.e., it is an entailment of the first text).

Task Interface See Figure 6

Task Input Data A set of sentence pairs:

- ("Crude oil prices slump", "Oil prices drop")
- ("The government announced that it plans to raise oil prices", "Oil prices drop")

Task Output Data

- the individual category choices of each worker, e.g., e.g., ("Crude oil prices slump", "Oil prices drop"), 1, 1, 2
- an aggregated value based on the chosen aggregation method (majority vote, weighted vote considering worker trust). Some aggregation methods might also return a confidence value. e.g., ("Crude oil prices slump", "Oil prices drop"), 1, 66%

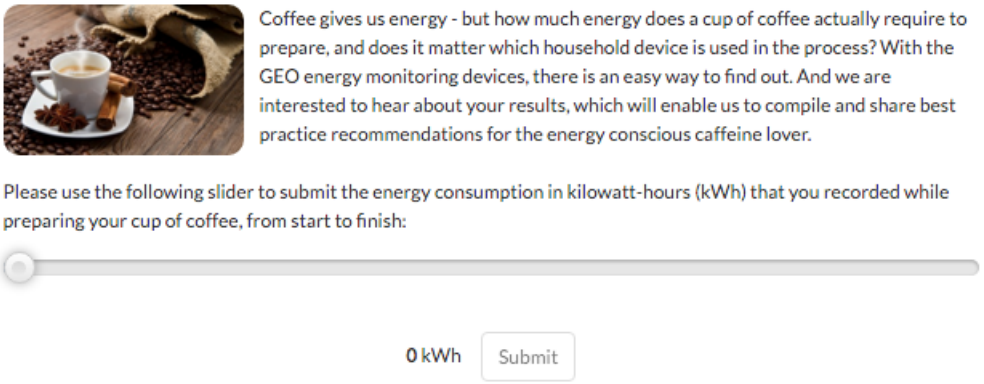


Figure 7: Interface for the simple slider task.

API Call

- General
  - Title: Judging textual entailment.
  - Category: Classification
  - Instruction: Can the second sentence be inferred from the first one?
- Specific
  - Default Categories: (1, "True"), (2, "False")
  - Multiple Choice: false
  - Judgements per unit: 10
  - User Characteristics (location, languages, trust): (loc, US), (lang, EN), (trust, 80)
  - Aggregation: majority vote
  - Task deadline: 30.11.2013
- Data
  - ("Crude oil prices slump", "Oil prices drop")
  - ("The government announced that it plans to raise oil prices", "Oil prices drop")

Task Title: (TC3-1) Simple Slider

Task Description Snow et al (see Section 4.2 of [39]) describe a task for judging word similarity where workers are presented with pairs of words (e.g., boy, lad, noon, string) and for each pair their have to specify how related the words are on a range of [0,10]

Task Interface See Figure 7

Task Input Data a set of term pairs:

- (boy, lad)
- (noon, string)
- (cow, grass)

Task Output Data

- the individual score selected by each worker, e.g., (boy, lad), 8, 9, 10
- an aggregated value based on the chosen aggregation method. In the case of the slider average fits better than majority vote. No confidence value (unless it is a majority vote weighted based on worker performance). e.g., (boy, lad), 9

API Call

- General
  - Title: Judging word relatedness.
  - Category: Classification - Simple Slider
  - Instruction: On a scale from 0 (totally unrelated) to 10 (same), how related are the two terms in the following term pair?
- Specific
  - Slider name: Relatedness
  - Slider Range: (0,10)
  - Judgements per unit: 10
  - User Characteristics (location, languages, trust): (loc, US), (lang, EN), (trust, 80)
  - Aggregation: average
  - Task deadline: 30.11.2013
- Data
  - (boy, lad)
  - (noon, string)
  - (cow, grass)

5.2 Integration of uComp HC Tasks and the CrowdFlower Platform

In order to realize hybrid-genre workflows, it is important that the platform can translate game tasks into HITs on mechanised labour platforms. This task is foreseen by point (d) of T2.2: " translation of certain tasks with lower skill requirements into Human Intelligence Tasks (HITs) to be carried out through mechanised labour through marketplaces such as MTurk and CrowdFlower."

5.2.1 Mapping of Application Programming Interfaces (APIs)

Therefore a mapping between the components of a Classification tasks and the API of a concrete platform must be clarified. We chose CrowdFlower (CF) as the concrete platform given our earlier experience with it, its reliability, its access to a large number of third-party crowdsourcing marketplaces (including Amazon Mechanical Turk - AMT) and its availability to requestors without a US bank account (a constraint imposed by AMT). Table 18 sums up this mapping.

uComp API	**CF API**
task_title	Job title
task_type: "Classification"	cml:checkboxes
task_description	cml:checkboxes (parameter "label" for all checkboxes)
ts_default_categories	cml:checkboxes (parameter "label" for each checkbox)
ts_multiple_choice	Create different cml templates for different task modes (e.g. multiple or single choice questions)
Judgements per Unit	Jobs/ judgments_per_unit
Worker characteristics	Jobs allows specifying a desired channels (e.g., amt) and excluding/including certain countries (included_countries)
Aggregation	Various strategies described at https://crowdflower.com/docs/cml/#aggregation
task_duration	CF support only for Premium accounts

Table 18: Mapping between uComp and CrowdFlower APIs.

5.2.2 Implementation Details of the uComp to CF Bridge

The uComp API[2] was built using PHP. A uComp to CF bridge was built, which allows pushing requests to CrowdFlower(CF). This component was implemented using an Open Source PHP library, which matches commands to corresponding HTTP requests and allows to easily access the CF API[3].

It is necessary for the requester to manually specify the crowdsourcing genres used for solving his task: one can either send a job and its units to the quiz and let players solve the tasks ($mode = quiz$), or he can opt for directly sending the tasks to CF ($mode = crowdflower$). By creating a job via the uComp API and setting the mode parameter to $crowdflower$, the corresponding CF job creation call is issued. No complex parameter transformations are required given the straightforward mapping between the two APIs (see Table 18) which is handled by the uComp platform: since the uComp API was designed with the CF API in mind, they are very similar and therefore it is possible to just adapt a few parameters and use the uComp API to send jobs to CF, upload units and receive results. When a job is created on CF the id of the created job is returned.

[2] `http://soc.ecoresearch.net/facebook/election2008/ucomp-quiz-beta/api/v1/documentation/`
[3] `https://github.com/supertom/php-crowdflower`

Data to the newly created job is sent as a CSV file via the uComp API. If the mode parameter is set to *crowdflower*, the uploaded CSV file will be slightly modified: Headers are added and if the user has specified to include gold standard in the CSV file, the last column of the file will be marked as gold data. In addition a CML template will be created that uses the newly created headers as placeholders for the data. If the user chooses to send data to CF all information about the job will be also stored in the uComp database in order to provide all functionality of the API and to create an automatic logging function of the activities.

The pausing and resuming of a job, as well as the retrieving of results in CSV format are straightforward and require just the *jobid* as a parameter.

6 User and Context Models

The content of this section covers the work envisioned by task *T2.3: User Profiling and Contextualisation*. In this section, we will report on: (i) an overview of the information that is covered by the user and context models, based on what is collected by the uComp framework and a range of derived information which will be computed; and (ii) a list of ontologies that could be re-used for modeling this information.

6.1 Content of the User and Context Models

Table 21 sums up the various user and context information provided by the uComp platform and CF. The uComp information is gathered through the social account (Facebook, Google, Twitter) API from the information made public by the user, and which he consents to share with the uComp games when he first installs them as one of his social applications. We have divided the available data into various categories, including a digital profile, personal data, context information, skills, community data and game/session related data. From this table, it is apparent that through the uComp platform we can access much richer game play data that allows building complex models of user behavior over time. In addition, the uComp platform will monitor skill/performance indicators as follows:

- user performance on certain task types (e.g., classification vs. co-reference);
- user performance on specific tasks such as sentiment evaluation or relation detection;
- user performance on specific domains: e.g., climate change, medical, general knowledge, finance, sports, etc.
- overall user performance, e.g., overall trust levels;
- average speed per task;
- availability - when was the user last seen online? what is the probability of the user coming online before the task deadline, based on his activity history?

uComp API	CrowdFlower API
Digital Profile	
Social ID URL of social profile, email	CF specific worker ID, external ID email, last IP address, recruitment channel
Personal Data	
name, gender, birthday	
Context Information	
location, locale	geographical location (country, region, city)
Skills	
spoken languages	
Community	
list of friends	
Game/session related information	
first/last login (date) visits right, wrong, unclear answers skipped relations, completion times	trust level = precision gold unit answers submission rate

Table 21: User and context information provided by the uComp framework and CF.

6.2 Utilisation of User Data

Basic information like the gender or birthday is used to gather statistical information. However, more specific data is used during playing the game and influences the experience. The list of the friends of the user is used to show relevant information (e.g. guesses of friends, or a comparison how good the user performs compared to his or her friends).

The spoken languages of the user are especially relevant for the language quiz, because through a promotion with Crowdflower it is possible to preselect the users who speak a certain language and show them relevant questions.

The game can also access the performance statistics of the users and if they come through a Crowdflower promotion (and will be paid for solving questions), they can be assessed with control questions. If a user fails to many control questions and therefor has a low performance score, the game will ban him from answering more questions and will not include his answers - also the users will not receive a payment. If they access the game through Crowdflower the user will be informed about this policy.

6.3 Candidate Ontology Models

An initial review of existing ontology models, revealed the following models as providing a potential basis for building the uComp user and context models that would cover the information described in the previous section:

Human Computation Ontology [4] Although not primarily focused on user profiling aspects, this ontology conceptualizes the connection between contributors and crowdsourcing tasks. It also relies on the Provenance Ontology (PROVO)[5] to describe the provenance of each contribution. See [9] for a description of this ontology.

SmartProducts user model [6] focuses on supporting ambient intelligent applications and therefore could contain reusable ontological models for specifying context information.

FOAF - Fried-of-a-friend [7] is a good candidate to be used for specifying personal information as well as links (relations) between players.

SIOC - Semantically-Interlinked Online Communities [8] provide ontological models for describing forums and social networking sites, including posts. It could be valuable at a later stage, if uComp decides to also record/archive relevant user content from social networking sites.

[4] http://swa.cefriel.it/ontologies/hc.html
[5] http://www.w3.org/TR/prov-o/
[6] http://projects.kmi.open.ac.uk/smartproducts/ontologies/v2.6/
[7] http://xmlns.com/foaf/spec/
[8] http://rdfs.org/sioc/spec/

7 Prioritisation Algorithms

We hereby provide a set of prioritisation algorithms that primarily focus on finalizing the given jobs by the given deadline. We start with an overview of related work before presenting these algorithms.

7.1 Overview of Related Work

There is a rich literature on approaches for assigning and scheduling crowdsourcing tasks under some constraints. For example, Minder et al [25] propose CrowdManager, a framework for optimizing the price and task allocation based on time, quality and budget constraints specified by the requester. Singer et al [38] focus on a pricing mechanism that 1) maximizes the number of tasks achievable within a budget and 2) minimizes payments for the tasks.

If the simplifying assumption is made that the worker's skills, speed and expected quality are known, then the assignment problem becomes a classical assignment problem solvable through linear programming, as has been done by Minder et al, who used an integer programing solution to the problem [25]. In a realistic crowdsourcing setting, however, there are several major challenges, from which we mention two:

1. **Workers arrive in an online fashion** in any realistic crowdsourcing setting and this makes assignment and planning of tasks difficult. To solve this problem some approaches opt for strategies to manage crowd-latency. For example, [25] relies on a **retainer model** in which case a set of workers are paid a low price to be available for whenever the tasks are posted. Such a retainer model approach has been originally introduced by [8]. There is no retainer model in uComp, therefore, task assignment will be performed among players that are currently online.

2. **There is little/no knowledge about workers** - crowdsourcing markets maintain little information about their workers, although prioritisation approaches require information about the speed, quality and acceptable costs of workers. To overcome this lack of information, various approaches implicitly ask workers to bid for tasks (e.g., specify their acceptable costs) as well as to perform some sample tasks in order to estimate their potential speed and expected quality [25, 38]. Other authors, more realistically, create worker models which they try to estimate from historic data gathered about the users [4] [18]. This approach seamed to be more feasible for uComp.

7.2 Mechanisms for Task Prioritisation

We define t_0 as the starting time of J_b, t_1 as the current time ($t_1 > t_0$) and t_{dl} as the deadline for the job.

At time point t_1:

- the time left to deadline is $t_{dl} - t_1$

- the number of missing judgments is the difference between the number of expected judgements for this job ($N * JPT$) and the judgments gathered so far (J_{J,t_1}), namely: $N * JPT - J_{J,t_1}$
- the completion speed Cs_{t_1} is $\frac{J_{J,t_1}}{t_1 - t_0}$

We can then compute an estimated time to completion ($T_{tc@t_1}$) as:

$$T_{tc@t_1} = \frac{N * JPT - J_{J,t_1}}{Cs_{t_1}} \tag{1}$$

A job's J_i priority at t_1, is the ratio of the estimated time needed for completion and the actual time available.

$$P_{J_i,t_1} = \frac{T_{tc@t_1}}{t_{dl} - t_1} \tag{2}$$

Ideally this ratio should always be < 1. Jobs should be ranked based on this priority. The priority of all jobs will be recomputed at given intervals of time. The frequency of this recomputation will be established experimentally.

Within a particular job J_i, individual tasks have their own priority level depending on:

1. the number of judgments they gathered so far. There are two options here: a) to prioritise those tasks that already have some of the required judgments in an effort to elicit all needed judgements or b) to prioritise those tasks that have no judgements yet in order to gather some judgments for them. We adopt the first approach and make task priority directly proportional to the completion rate of the task T_k at time t_1, which is $CR_{T_k@t_1} = \frac{JPT_{T_k@t_1}}{JPT}$, i.e., the ratio of judgments gathered for T_k at time t_1 and the expected judgements pert task.
2. the agreement level of the judgments gathered so far (for those tasks where inter-worker agreement can be computed). Tasks where the disagreement is high should be given higher priority than those for which an agreement seams to emerge already. Therefore, in the case of a classification style task, the task priority is indirectly proportional with the maximum inter-worker agreement achieved for one of the c categories of the classification task, that is $\max_{l=1,c}(IAA_{cat_l})$.

Therefore, for each task T_k of a job J_i at time t_1, the individual task priority can be computed as:

$$P_{T_k,J_i,t_1} = P_{J_i,t_1} + \frac{CR_{T_k@t_1}}{1 + \max_{l=1,c}(IAA_{cat_l})} \tag{3}$$

If there are no judgments available for T_k then its priority is identical to the job's priority.

Assignment to appropriate platform. If the work mode is set to *quiz*, then the above algorithm can be used to prioritise the tasks. High priority tasks can be then assigned to workers that have a higher speed.

8 Applications

8.1 Climate Challenge

The Climate Challenge is an online competition in the tradition of games with a purpose that combines practical steps to reduce carbon footprint with predictive tasks to estimate future climate-related conditions [35]. The application is designed to increase environmental literacy and motivate users to adopt more sustainable lifestyles. Its feedback channels include a leaderboard and a visual tool to compare answers of individual players with (i) the average assessments of their direct social network contacts as well as the entire pool of participants, (ii) a selected group of experts, and (iii) real-world observations [34].

The Climate Challenge supports the following task types:

- Ten multiple choice questions per month, which are related to the topic of climate science. Since the answers to these questions are known, the players will get immediate feedback on the correctness of their answer. The players will get points for the right answer, and a point penalty for wrong answers.

- Opinion poll obtain survey-like results from players - for example, asking them whether they think governments should regulate the release of greenhouse gases, and to what extent depending on assumed cost factors. Opinion polls are bonus challenges, there is no fixed amount of monthly tasks. They offer a more open alternative to multiple choice questions - more flexible to use (e.g., in cases where there is no correct answer), and should appeal to a wider audience when tied to current events.

- Fifty sentiment detection tasks per month, where the players state whether they perceive certain words or phrases as positive, neutral or negative. Correct answers for this task type are not immediate available, as the assumed correct value will be calculated as an average once a certain task has been evaluated by ten different players. Therefore, points are distributed ex post based on how closely a player's assessment matches the average rating by other players.

- A monthly prediction task, which is promoted in a cooperation with Climate Program Office of the National Oceanic and Atmospheric Administration (NOAA), where the players are asked to predict future environmental conditions - e.g., What percentage of land area in the Northern Hemisphere will have a "White Christmas"? Three experts will also provide their answer to the monthly question. After the real-world answer is known (typically through actual measurements), the players are being awarded points depending on how closely their answer matches the observed value. A diagram lets them compare their own assessment with the average predictions of climate scientists, the predictions of all players, and the observed value.

- Five pledges per month ask participants for feedback on recommendations for reducing energy consumption and making sustainable lifestyle choice. They can declare if they have already adopted the recommendation, rate the environmental impact of this action, and share accepted pledges via their social media channels.

Each task type has a different mechanism to distribute points. For the prediction and sentiment detection tasks, the players will receive the points once the correct answers are available (i.e., at a future point in time). For multiple choice question, players will receive points immediately depending on whether the answer was correct or not. For pledges and opinion polls, players will receive a certain amount of points just for participating, because the task type does not have a right or wrong answer.

Since the main goal of the Climate Challenge is not only to collect answers, but also to raise awareness, the promotion and community aspect of the game has to be treated different than in the Language Quiz [34], which will be presented in the following section. As Sabou et al. [31] noted there are three different categories of crowdsourcing paradigms, and it is hard to mix the monetary reward system of Crowdflower with a gameplay-driven and altruistic reward system of the Climate Challenge. Instead, social media channels such as Facebook and Twitter were used to promote the game and create a stable community. This approach is also supported by Huberman [20], observing that ideas shared through friends' activities spread across social networks and can benefit the reach of a crowdsourcing application by creating widespread attention.

Another special aspect of the Climate Challenge is the recurring nature of the tasks. Inevitably, there is is a period where a prediction task is "open" for answers, followed by a waiting period until the real-world answer becomes available. Then the next prediction question can be asked. To account for this delay, and as an incentive measure, monthly game rounds were introduced where only a certain number of tasks per task type are being made available. This approach ensures that players will always find new tasks at the beginning of each month, and have a incentive to return to the game to find out about the correct answer to the previous question.

The created community was used as a platform to promote the new round each month, supplemented by e-mail notifications and Facebook announcements (to the approximately 2,500 followers of the Facebook community page) that new results were available. This approach was used as a regular reminder to re-activate players back every month, and to keep them engaged.

Since its launch in March 2015, the Climate Challenge attracted 3,236 unique users as of December 2015. From those, 644 created a user-account in the game leading to a high conversion rate of 20%. Out of those 644 users, 554 became active players. In total the game collected 680 prediction answers, 4,316 multiple choice answers, 15,979 sentiment answers, 2,003 pledge answers and 34 answers to the recently introduced opinion poll.

8.2 Language Quiz

The approach of the Language Quiz differs significantly from the Climate Challenge, even though both games are built upon the same framework. Instead of creating environmental awareness and promoting sustainable lifestyle choices, the Language Quiz aims to acquire multilingual language resources for research purposes.

A flexible and open architecture to support multiple languages, different types of task, and the inclusion of third-party tasks were among the key factors that have guided the development efforts. Conceptual insights gathered from analyzing the results of the Sentiment Quiz, a game

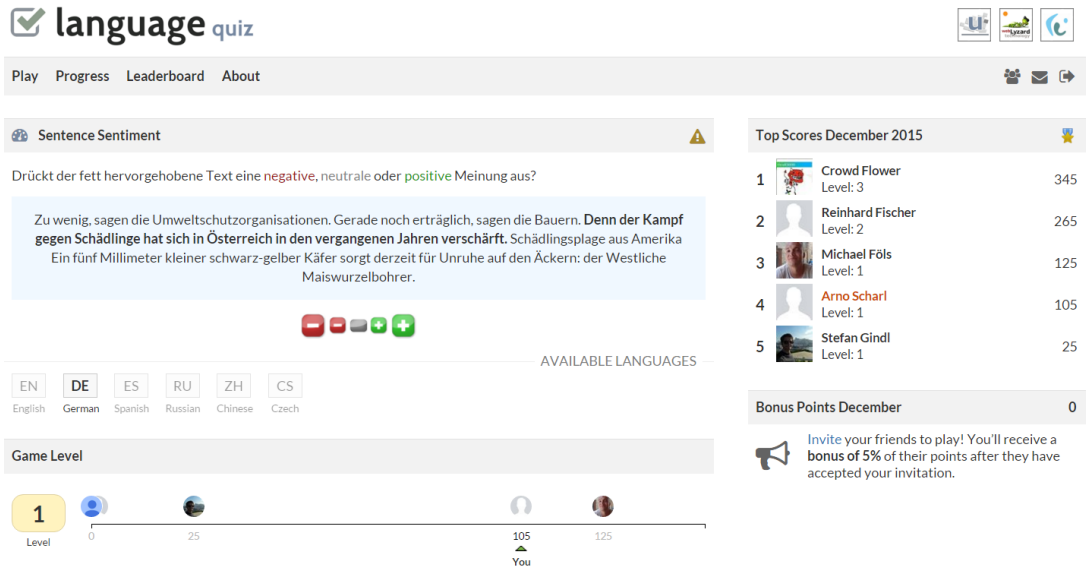


Figure 8: Main interface elements of the uComp Language Quiz (quiz.ucomp.eu)

with a purpose to assess sentiment terms [33], also influenced the design of the Language Quiz - although the two applications are based on a completely different technology stack.

To open the game engine for use by the partners of the project, an API was created which allows approved partners to send tasks to the game. Currently, Language Quiz supports multiple choice and sentiment assessment tasks, on both a term level and a a sentence level (the latter serving either stand-alone statements, or highlighted sentences in a context of a whole paragraph). When uploading the task data, the project owner can decide if the tasks should be sent (i) to the game, (ii) to CrowdFlower, or (iii) to a hybrid workflow that uses CrowdFlower to recruit game participants. The flexibility of this approach increases the usefulness of the framework as an evaluation tool, especially when the size of a community and its engagement level are difficult to assess ex ante, or if the task is rather repetitive in nature. This is often the case when acquiring language resources, as compared to games with a purpose such as the Climate Challenge, which offers higher intrinsic motivation due to a limited amount of hand-picked monthly tasks in a domain that is often characterized by altruistic motivation. Since the Language Quiz does not follow such a curated approach, the quality of the questions can vary.

The mechanics of the game were also adapted to support the hybrid scenario. If a user is sent to the game via Crowdflower, he would receive 20 questions. Upon successful completion of the game rounds, the user will receive a code which he can enter in his Crowdflower task to be paid. Since only the highest-rated Crowdflower users were being targeted, the quality of the results were correspondingly good. To filter out cheating players, each Crowdflower user has to answer three gold standard questions. In the case of a wrong answer on a test question, the user does not receive Crowdflower payments, and his answers will be deleted from the game. This ensures a consistent high quality of the obtained results.

The hybrid support of native gaming and paid Crowdflower evaluation means that the task

uploaders have the chance to leverage the benefits of both crowdsourcing categories depending on the requirements of the given task. The paid tasks also improve the user experience of the native game participants, since the correct answers and related game point calculations are available sooner. As Sabou et al. [32] have identified, games with a purpose and mechanized labour have different pros and cons in the categories speed, cost and quality of answers - using the uComp framework, task providers can make the decision for each task individually. Since its launch in October 2015, the Language Quiz attracted 814 unique users. 579 of these users created a user account, and 525 became active players. Out of the 525 active players, 503 were contacted via the Crowdflower campaign option. The game was also promoted at RANLP'2015 and through social media. 347 users became active players who submitted valid answers, 178 failed the test questions. In total 12,283 valid answers were submitted to the game, 7,459 of those answers where paid via Crowdflower and 4,824 where organic answers from the players.

Further promotion activities by USFD are planned for early 2016, pending ethical approval from the ethics committee. The approval is required since we will be recruiting human participants via the Univerisity's volunteer list and also via the University run GATE users community list.

9 Progress Monitoring

The progress monitoring serves as an incentive mechanism to engage the players. The uComp framework uses a dynamic level bar below the question box, to show the user at which level he currently is and where other players are on the same level (displayed as a horizontal bar that indicates how far other players are ahead or behind the user). This tool was designed to motivate the user to play more because he can always see how close other players are positioned around him and by playing the game the user can see his avatar bypass the other users to reward him visually for actively playing the game.

Another real-time indicator is the monthly high score table in the right sidebar, which shows the overall top three scores as well as the two participants who rank immediately before and after the player. This tool was designed to visualize the overall position of the player and to show how much distance is between the player and the top players. Like the level indicator, its dynamic updates are intended to encourage the players to keep playing, as they will always be visually rewarded by watching their avatar climb up the leaderboard.

Another incentive to motivate players are the dynamic progress bars shown in Figure 9, which indicate how many of the monthly tasks in each category have already been solved by the player - e.g., if a player has answered two of the five pledges, the corresponding bar on the progress monitor will be at 40 percent. This feature aims to trigger the user's desire for completion while they have not answered all the monthly questions. At the same time, the component serves as a navigational aid. Users can click on the desired task type to only show questions of this type as long as new ones from the monthly pool of questions are available.

The progress page also includes an archive with historic performance data in order to amplify the motivation to complete the monthly set of questions. With a similar intention, the game offers a leaderboard (see Figure 3) that lists the top three players of each month. This feature

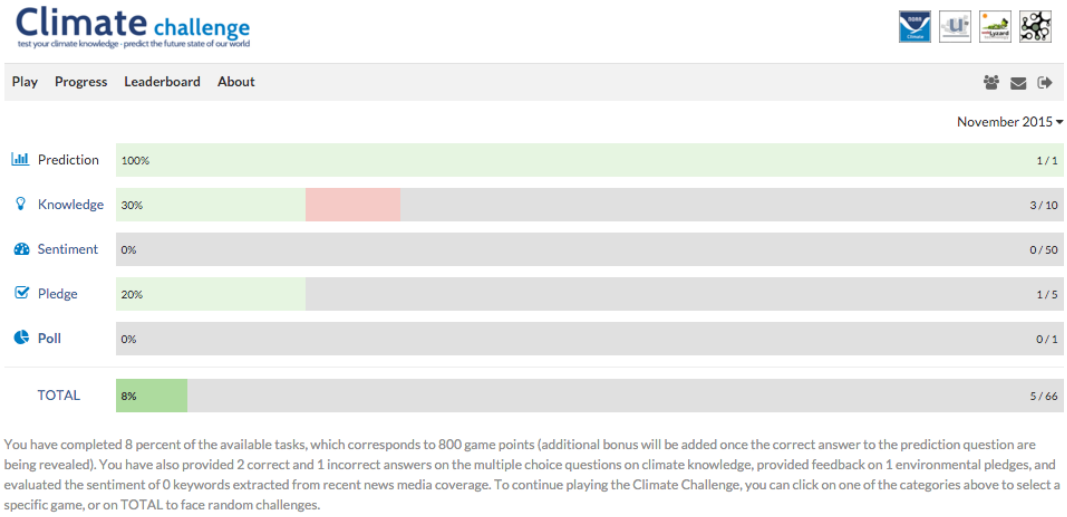


Figure 9: Progress page of the Climate Challenge

complements the progress page archive and the dynamic sidebar elements, designed to support the players' long-term motivation and to grant them "bragging rights" as they become part of the official history of the game.

In addition to the progress indicators for players, the Language Quiz as an open platform for trusted third parties to upload their own tasks provides a progress monitoring system for administrators as well (the Climate Challenge with its curated system of monthly tasks does not require such a feature). This system is embedded in the API. After creating a job for the game, the task owner will receive a Job ID. Afterwards he can send a status request with the ID to the game. This works both for Crowdflower and game tasks. The returned result indicates if the job has been completed; and, if not, how many answers still have to be collected. For game tasks, it is also possible to request the preliminary set of answers collected up until now, even if the job has not yet been completed.

9.1 Quality Control

Measures of quality control vary according to the type of task, and according to the application. Quality control measures in the Climate Challenge, for example, are only required for specific tasks - e.g. the multiple choice questions or the language resource acquisition task for assessing the sentiment of extracted climate change keywords.

Task types such as the pledges or the polls, where the answer is a matter of opinion and cannot be considered right or wrong, do not yield quality metrics. Similarly, answers to the prediction questions cannot be immediately evaluated because the real-world answers are not known when the question is asked. The multiple choice questions, by contrast, are instantly evaluated and the user will receive points for right answers and a point penalty for wrong answers. In all four cases it is not possible for players to cheat.

For the sentiment detection tasks in the Climate Challenge (as well as all tasks in the Lan-

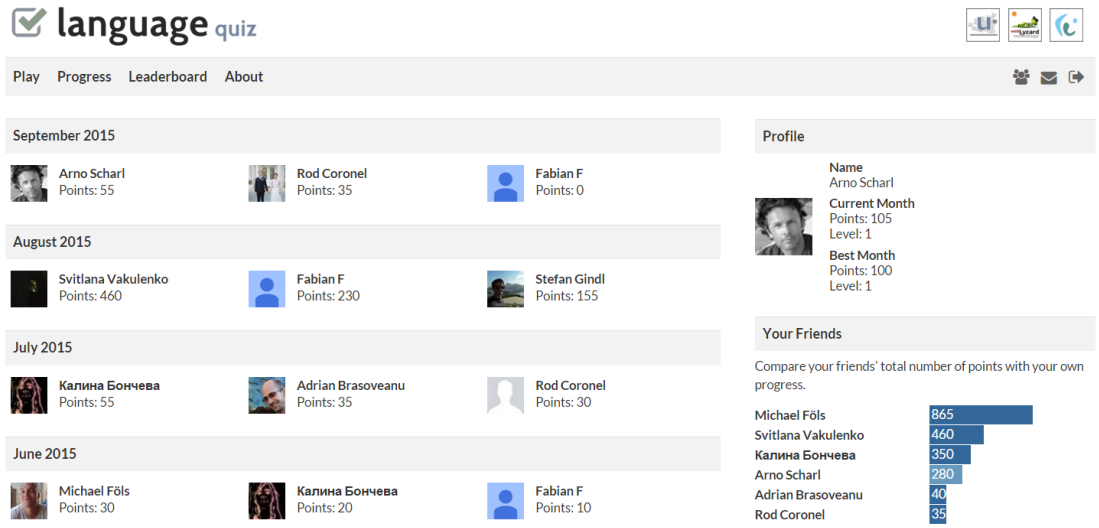


Figure 10: Leaderboad of the Language Quiz

guage Quiz), however, the computation of quality metrics is important. One possible approach would be the evaluation of users based on gold standard questions. Because of the recurring nature of the game and the limited testing ability of the other task types, a cross-validation approach has been chosen (which can be configured via the uComp API) - collecting a total of ten answers per question. The "correct" sentiment will be determined based on the mean of the assessments, and is then used to award game points. The standard deviation of the received answer serves as an indicator of ambiguity, and can be used to decide whether the acquired language resources are valid and should be considered for further processing.

Cheating prevention becomes essential in the case of the Language Quiz, which offers the option to send tasks to Crowdflower and/or to recruit game participants via Crowdflower. The API allows developers to include gold standard data in their set of uploaded tasks. This data is then forwarded to Crowdflower to include test questions and refuse payment based on the number of failed tasks.

Hybrid Crowdsourcing Model. If Crowdflower is only used as a promotional tool to recruit players for a certain task, there are two systems in place to prevent cheating:

- Users have to insert a code in the Crowdflower submission form, which they will receive after having answered 20 questions in the game. This code is a complex set of characters and special characters, which cannot be guessed.

- Crowdflower allows to specify a minimum time that it usually takes to complete a task (filtering out users who do not provide a genuine answer); e.g. if the minimum time is set to two minutes, but the answers is received within 30 seconds, it can be assumed that the user has not really tried to solve the problem.

Test Questions. If Crowdflower is used to the generate the results (i.e. the uComp API only serving as an interface between the task owner and the crowdsourcing marketplace), every

player has to correctly answer three test questions as part of the batch of 20 questions that need to be answered to receive payment. This flags users who try to cheat, or those who are lack the expertise for a specific task type (e.g. a lack of language skills). Answers of players who fail one test question will be disregarded, and they will not receive the payment. This system will be explained in advance to ensure that players pay attention to the quality of their answers.

Player Selection. The Crowdflower marketplace distinguishes three level of players, depending on the average quality of their answers. If tasks are made available to all levels, they will be solved more quickly at reduced accuracy. The quality requirements and additional complexity introduced by redirecting participants to the game interface suggests to only consider Level 3 players to solve the tasks, except in very time-critical situations. Those players will always be more careful when solving a task, as they risk their Level 3 status if they fail too many test questions.

9.2 Incentive Mechanisms

Crowdsourcing strategies can be classified into three categories [31]: mechanised labour offering financial rewards, games with a purpose with gameplay incentives, and approaches based on altruistic work. The Language Quiz uses the first two strategies, while the Climate Challenge relies on gameplay features and altruistic incentives.

Ranking System. The main incentive mechanisms in the Climate Challenge are traditional gameplay mechanisms, which are used to engage the users: (i) a real-time leaderboard that shows how they bypass other players and rank higher while playing the game; (ii) a leaderboard with historic data on the top three monthly players, and (iii) a leveling system to motivate users to reach higher levels and surpass their previous record.

Sharing of Game Content. Players can also invite their friends to participate in the game, in general or specific to certain tasks - players can use Facebook, Twitter, Google+ or E-Mail to notify others when they accept a pledge. A strong incentive to share are bonus points that players receive when their friends follow their invitation.

Prizes and Bragging Rights. Each month, the best players receive small prizes such as coffee mugs or t-shirts, which on purpose do not represent a significant monetary value. But even with this physical reward system in place, we consider the in-game mechanics and the achieved bragging rights as the most important incentives. The game relates to climate change as an important environmental topic, where players can demonstrate their knowledge vis-à-vis other players and friends. Previous work has shown that games that tackle a specific domain will benefit from the intrinsic motivation of supporting a good cause [29].

Notification System. Another mechanism to motivate players are push notifications via the Facebook API and/or E-Mail notifications, have proven to be a very effective tool to (re-)engage players at the start of a new monthly game round. Users who have signed up are notified once new answers to their open questions are available. Notifications are only sent out once or twice each month, to minimize the risk of loosing subscribers. Weekly updates only go to players who play regularly, the other players will just receive rare reminders if they want to rejoin the game or other major announcements such as the final game round of the year. Players

have the option to unsubscribe from the notifications and not receive any E-Mail or Facebook notifications at all.

Multilinguality. For the Language Quiz, a major benefit of using Crowdflower as a promotional tool is the ability to recruit native speakers in many different languages [31]. Since the support for different languages is a key feature of the Language Quiz, this feature is very important for the overall success of the game. Players also remain motivated because they know that the task would contain test questions and that their payment (and indirectly their Crowdflower level) depends on the accuracy of their answers.

10 Workflow Optimisation

Active learning is about a semi-supervised machine learning task and the improvement of its performance by asking answers to a perfect oracle for particular data points, for a survey of previous works see [36], [26] and [5]. In one of its variants, proactive learning proposed in [13], the perfect oracle is replaced by several imperfect ones with various characteristics, and the problem is recast into a decision-making under uncertainty to match the most appropriate oracle with the data point that will yield the largest performance improvement. Such approach results in an architecture comparable to HC computing, except for the share given to humans intervention which is sought to be minimal in active learning, while it is systematic in HC. [24] and [6] studied how to make active learning tolerant to random noise classification with a statistical approaches, while [41] addressed clustering and [6] combined smoothing techniques and clustering. Another recent work has considered latent variable models in the context of active learning [1] to combine information across multiple tasks via a shared intermediate layer (abstract feature space). For our purpose, active learning can contribute to workflow optimization along two dimensions:

1. given an information data point to collect, choose the most appropriate oracle [13], this point has been addressed in Section 6, essentially at HC task level with the contribution of the User Models. If we consider the level of a single data point, data clustering approaches like [41] or [6] can be deployed to improve global task performance.

2. given a dataset to submit to HC, rank the data items to be processed in decreasing order depending on their potential contribution to the global task. Here active learning can contribute a lot to HC, since task independent solutions like measures based on disagreement or information entropy have been investigated for a long time in active learning [3] [26].

Among the recent work were active learning has been applied to NLP we find, [37] who adapted uncertainty sampling for the learning with rationales framework in a text classification task. When the model is uncertain about an unlabeled document, the words/phrases contained in the document are searched among the rationales of the labeled documents. Application of active learning for speech understanding in noisy conditions was investigated in [17]. Four sequence labeling tasks : phrase chunking, part-of-speech tagging, named-entity recognition, and bio-entity recognition were the use cases which served to compare active learning strategies

in [22]. Active learning with clustering was applied to the task of word sampling in [14]. In the work of [11] cross-entropy measures are adapted to active learning for post-editing based incrementally retrained machine translation. Active learning is used with parallel corpora in [28] for helping to identify relations with a co-training strategy.

The principle of deploying together active learning and crowdsourcing is presented in [2] with a translation task as use-case. In their paper the authors address the three possible contribution of active learning for crowdsourcing, namely selecting the workers (oracles), selecting the input data and output data (information provided by the workers) at the sentence level. In [23] active learning and crowdsourcing are applied to opinion mining from microblogs. The authors propose a new non-linear distribution spreading algorithm (based on Delta IDF feature weighting) across the different classes to be learnt in order to identify discriminative features for mislabeled data. This last experience is an example of a recursive HC architecture where erroneous data points are fed-back to be re-annotated, similarly to the boosting approach in machine learning [16], which can be considered to be a particular case of active learning applied to mislabeled data points. Note also that since HC computation by its natures follows a map-reduce architectures, first splitting a complex annotation task among many workers (map), then collecting back the information produced with a strategy akin to ensemble learning, all the performance optimization developed for map-reduce approaches could be easily imported into an HC architecture.

In conclusion, active learning is a straighforward improvement of the HC computation workflow. In is simplest expression, it provides improved selection strategies for prioritizing the processing of the data points according to different strategies, the most frequently used being to first handle the most probables cases (data point with the highest predictability) or the most uncertain ones (data points with the highest disagreement).

10.1 Experiments on Sentiment Analysis Tasks

In collaboration with the organizers of the DEFT series of evaluation campaigns on text mining, uComp has provided the data and contributed to the different tasks definitions of the DEFT 2015 event, which provided a testbed for assessing how crowdsourcing could contribute to the evaluation paradigm. The event is reported on in the uComp deliverable: "D 5.3 Evaluation Report". On this occasion, a reference corpus of Twitter messages in French about climate change has been annotated by human annotators following the classical methodology of the evaluation paradigm [9]. The annotations describe at a fine grain level the opinion, sentiment or emotions expressed in the microblogs messages (see Figure 10.1 on page 36 for an example). In addition to the annotations at the word group level and relations between word groups, a global category for the opinion, sentiment or emotion of the whole message is chosen from a set of 20 categories. For the evaluation campaign four tasks were proposed:

1. determine the general polarity of tweets (positive, negative, neutral),
2. (a) identify the generic classes (opinion, sentiment, emotion, information) of the whole message,

[9] The anntation task have been subcontracted to ELDA.

(b) identify the specific classes (among 19 classes) of the whole message,

3. identify at word level the source, target, and opinion, sentiment, emotion focus, the possible sentiment intensity modifier and negation markers and the links between the identified word groups.

A total of 12 teams participated to the evaluation campaign, but none tried the taks 3 (fine grained annotation) which was judged to be too difficult. The corpus that resulted from the DEFT2015 campaign contains : manually annotated data by experts (gold standard), the output of the system from the 12 participating teams and the result of combining automatically the data annotated by the participating systems along different combination schemes. In the previous section one of the first clues mentionned to base one's preference for selecting target data points in order to improve annotation performance is the observed disagreement present in existing annotations. In the figure 12 on page 37, we show the disagreement distribution observed on the test data set for the three shared tasks of DEFT2015.

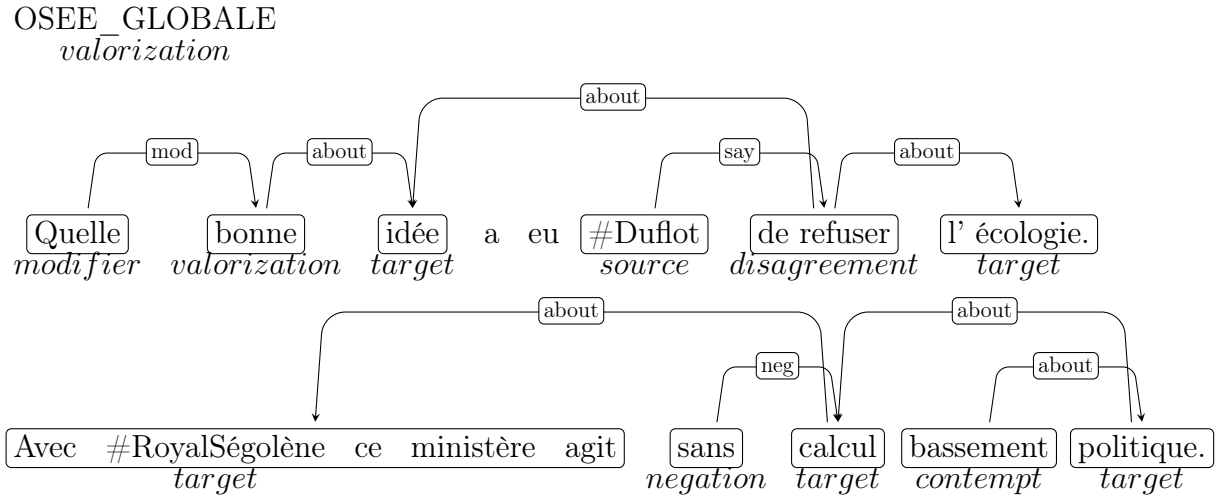
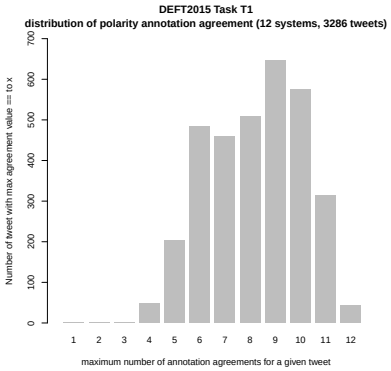
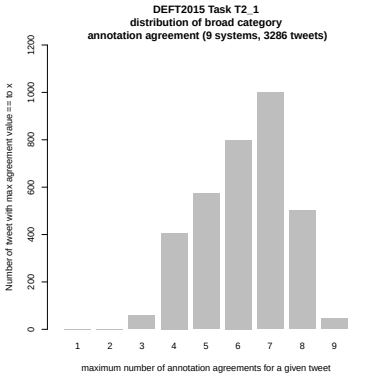


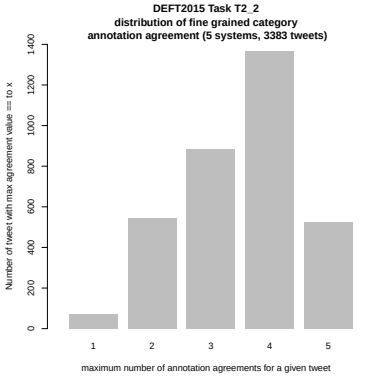
Figure 11: The message is: "*What a good idea (Mrs) #Duflot has had to refuse (the Ministry of) Ecology. With (Mrs) Ségolène Royale this Ministry acts without crassp political scheme.*". The tweet global sentiment expressed (OSEE_GLOBALE) is *valorization*. The relation *neg* and *mod* link the words indicative respectively of a negation ("sans", i.e. *without*) and of an intensity modifier ("Quelle", i.e. *What a...!*) to the sentiment expression they modify, here respectively "calcul" (*scheme*) and "bonne" (*good*). The links *about* connect the sentiment expressions to the objects they qualify, note here that the word "calcul" (*scheme*) a priori not bearing any opinion/sentiment/emotion value receives it through a chain of *about* links from "bassement" (*crassp*). Finally, the relation *say* connects the group of words referring to the sentiment holder to the sentiment expression.




```
4873...593 = + - = + = = = = - + = 7
4873...568 = = . = = - - + = + = = 7
4873...896 = = + = = = + = = = = + 9
4873...920 + + . + = - - + + + + = 7
4873...216 - - - + - - - - - - = = 9
...
```




```
4873...593 INF INF OP INF OP INF INF INF INF 7
4873...568 INF INF . INF INF OP EMO INF INF 6
4873...896 INF INF OP INF INF OP INF INF OP 6
4873...920 OP OP . OP INF OP OP OP INF 6
4873...216 OP OP OP INF EMO OP OP INF OP 6
....
```




```
4873...593 VALO COLERE VALO VALO INFO  3
4873...568 VALO VALO PEUR PEUR INFO  2
4873...896 VALO VALO VALO VALO INFO  4
4873...920 DESACC VALO VALO DESACC DESACC  3
4873...216 VALO VALO DEVALO DEVALO INFO  2
...
```

Figure 12: In the first column the distribution of the maximum number of concording annotation per tweet is shown for DEFT2015 task-1, task-2.1 and task-2.2, from top to bottom, with respectively 12, 9 and 5 systems and 3286 tweets, 3,383 tweets, 3,383 tweets. The right column displays a excerpt of the corresponding combined annotations.

From the three previous graphs of Figure 12 on page 37, we see that the agreement classes which have the highest counts (647 for task1, 1,000 for task2.1 and 1,366 for task2.2) are in the higher part of the range of agreement values (value 9 out of 12 for task1, value 7 out of 9 for task2.1 and value 4 out of 5 for task2.2), cf Table 38 on page 38.

count task1	value task1	count task2.1	value task2.1	count task2.2	value task2.2
0	1	0	1	70	1
0	2	2	2	542	2
0	3	58	3	883	3
49	4	406	4	**1,366**	**4**
204	5	573	5	522	5
483	6	798	6	-	-
460	7	**1,000**	**7**	-	-
509	8	501	8	-	-
647	**9**	45	9	-	-
576	10	-	-	-	-
314	11	-	-	-	-
44	12	-	-	-	-

Table 38: Distribution of the agreement counts for DEFT2015 task1, task2.1 and task2.2.

Sorting the tweets for annotation according to the maximal agreement among the systems annotations is clearly the best strategy here, since we will be maximizing both the confidence in the annotation and the number of tweets processed.

But since the data has been collected in the scope of an evaluation campaign, we have not only the annotation provided by the participating systems, but an estimation of the performance of each systems measured on the gold standard data. Thus we can refine a little more the previous choice under such conditions, by weighting the vote of each system for an annotation value, according to the precision performance of the system, as measured in the evaluation campaign, see Figure 13 on page 39. It is this strategy that has been used for sorting a corpus extracted form the data annotated during the DEFT2015 evaluation campaing, before sending it for crowdsourcing annotation. The corpus is made of 4258 tweets for task1, 4282 tweets for task2.1 and 6496 twets for task2.2 and its annotation is currently in progress.

10.2 Experiments on Named Entity Annotation Tasks

In named entity annotation tasks (i.e. marking names of people, organisations, locations), crowdworkers are sometimes are faced with passages of text which bear no entities. These blank examples are especially common outside of the newswire genre, in e.g. social media text [19].

```
4873...896 <+, 0.835696 > <=, 5.37248 >
4873...648 <+, 0.700729 > <-, 0.699273 > <=, 4.80818 >
4873...216 <-, 4.70666 >  <=, 1.50152 >
4873...593 <+, 2.00403 >  <-, 0.558989 > <=, 3.64516 >
4873...568 <+, 1.29227 >  <-, 1.39778 >  <=, 3.51812 >

4873...648 <INFORMATION, 2.44851 >  <OPINION, 0.097393 >
4873...593 <INFORMATION, 2.21321 >  <OPINION, 0.332688 >
4873...920 <INFORMATION, 0.430081 > <OPINION, 2.11582 >
4873...896 <INFORMATION, 1.87631 >  <OPINION, 0.66959 >
4873...216 <EMOTION, 0.332688 >     <INFORMATION, 0.382598 > <OPINION, 1.83062 >

4873...896 <ENNUI, 0.0929964 >         <VALORISATION, 0.45777 >
4873...568 <PEUR, 0.176961 >           <VALORISATION, 0.373805 >
4873...593 <COLERE, 0.0994085 >        <DERANGEMENT, 0.0929964 > <VALORISATION, 0.358362 >
4873...920 <DESACCORD, 0.358362 >      <VALORISATION, 0.192405 >
4873...216 <DEVALORISATION, 0.176961 > <ENNUI, 0.0929964 >       <VALORISATION, 0.280809 >
```

Figure 13: Example of ordering obtained by weighting the vote of a system for an annotation value by its precision performance measure,for DEFT2015 task1, task2.1 and task2.2

While finding good examples to annotate next is a problem that has been tackled before, these systems often require a tight feedback loop and great control over which document is presented next. This is not easily achieved in a crowdsourcing scenarios, where large numbers of documents (e.g. tweets) are presented for annotation simultaneously, in order to leverage crowdsourcing's scalability advantages. The loosened feedback loop, and requirement to issue documents in large batches, differentiate the problem scenario from classical active learning.

We hypothesise that these blank examples are of limited value as training data for statistical entity annotation systems, and that it is preferable to annotate texts containing entities over texts without them. We evaluated this hypothesis directly, in the context of named entity recognition (NER) [12]. Effectively, this offers a new pre-annotation task: predicting whether an excerpt of text will contain an entity that the workflow should send to the crowdworkers.

The goal is to reduce the cost of annotation, or alternatively, to increase the performance of a system that uses a fixed amount of data. As this pre-annotation task tries to acquire information about entity annotations before they are actually created – specifically, whether or not they exist – we call the task "pre-empting" and integrate it in the crowdsourcing workflow.

Unlike many modern approaches to optimising annotated data, which focus on how to best leverage annotations (perhaps by making inferences over those annotations, or by using unlabelled data), we examine the step before this – selecting what to annotate in order to boost later system performance.

For details on the experiments please refer to our peer-reviewed publication [12]. In summary, we:

- demonstrate that entity-bearing text results in better NER systems;

- introduce an entity pre-empting technique;
- examine how pre-empting entities optimises corpus creation, in a crowdsourcing scenario.

We demonstrated that entity pre-empting makes corpus creation quicker and more cost-effective [12], i.e. choosing to annotate texts that are rich in target entity mentions is more efficient than annotating randomly selected text. Larger samples can be used for training social media pre-empting; though we only outline an approach using 1 000 examples, up to 15 000 have been annotated and made publicly available for some entity types. Though demonstrated with named entity annotation, it can apply to other annotation tasks, especially when for corpora used in information extraction, for e.g. relation extraction and event recognition.

10.3 Experiments on Contextualized Sentiment Analysis

In the contextualized sentiment analysis task, crowdworkers are shown sentences in two different settings: On the one hand, a sentence is shown in isolation, on the other hand the sentence is shown with two preceding and two succeeding sentences as context, the sentence of interest highlighted in bold font face. The crowd workers have to decide whether the sentence of interest is negative, neutral or positive on a five-step scale. Figure 10 shows the user interface for this task.

The motivation for these experiments were contradictory results of an approach to create a context-aware sentiment analysis algorithm [44]. The algorithm takes co-occurring terms as contextual information for assessing a sentiment term's polarity. This approach showed promising results on document-level sentiment on various types of reviews. Applying the same approach on sentence-base sentiment produced unsatisfactory results when the training data was only annotated on the document level.

Faced with the decision how to move to the finer sentence level, we were faced with two problems: First we needed to decide what amount of context would be appropriate. Second, we needed to create the corresponding training corpus.

A random sample comprising 40.000 documents from the Media Watch on Climate Change was used for these Experiments. Each sentence of these documents got assigned two sentiment values: One with a sentiment algorithm ignoring contextual information and one with a sentiment algorithm that takes contextual information into account and was trained on product reviews. An initial set of 500 sentences where the two assigned scores differed was chosen randomly as input for the uComp Language Quiz.

	without context	**with context**
total number of tasks	465	478
tasks with absolute majority	298	283
mean majority	6.2	6.0
Krippendorff's alpha	0.225	0.165

Table 41: Crowd sourcing results for the experiments on contextualized sentiment analysis.

In total 10000 judgements were gathered, 478 sentences with context and 465 sentences without got exactly 10 judgements. For investigating the results, we relaxed the classes to three: negative, neutral and positive. It turned out that the selected sentences were judged very different by the individual crowdworkers.

Table 10.3 shows the data of the results. The results are notable because they show that the the raters had stronger agreement when the sentences were displayed in isolation. Additionally, the majorities were stronger in this data set.

With these experiments we showed that:

- For humans, presenting the surrounding sentences as context does not help in agreeing on a sentence's sentiment value. The surrounding sentences introduce noise instead of disambiguation support.
- Only small parts of data acquired this way could be used as training data, if the majority and inter-rater agreement were strong.

These results were pivotal for our work: Given that adding context by taking surrounding sentences into account does not help coming to an agreement among humans, it is questionable if this approach to context-dependent sentiment analysis can lead to promising results.

11 Summary

This deliverable presented the theoretical background, practical implementation and usage of the uComp Human Computation (HC) framework. It covered the full life cycle of HC applications including system design, multi-channel task deployment, task management, user profiling, quality control methods, and workflow optimization.

The human computation framework of uComp is flexible and modular in terms of progress monitoring, quality control, and incentive mechanisms for different types of tasks. This flexibility is reflected in the development and launch of two applications with distinct characteristics, the Climate Challenge and the Language Quiz:

- For gameplay-driven games like the Climate Challenge, incentive mechanisms include leaderboards, real-time level indicators and progress graphs. Cross-player validation prevents cheating, while social sharing and other social display mechanics increase the applications' visibility.

- Task-centric games like the Language Quiz can benefit from promotional campaigns on marketplaces such as Crowdflower to recruit players and kickstart the game. This approach includes test questions and leverages internal Crowdflower mechanism - for example, selecting only reliable Level 3 users).

The human computation framework of uComp is well suited to handle both type of games, and includes the necessary tools to keep heterogeneous user groups engaged while ensuring a consistent high quality of answers.

The framework bridges the gap between crowdsourcing marketplaces and games with a purpose by allowing the outsourcing of tasks to CrowdFlower, or using CrowdFlower as an effective promotion tool to quickly acquire a critical mass of players for a game.

References

[1] Ayan Acharya. *Knowledge Transfer Using Latent Variable Models*. PhD thesis, Department of Electrical and Computer Engineering, The University of Texas at Austin, August 2015. URL `http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=127528`.

[2] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Active learning and crowd-sourcing for machine translation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7. URL `http://www.lrec-conf.org/proceedings/lrec2010/pdf/244_Paper.pdf`.

[3] Stéphane Ayache and Georges Quénot. Evaluation of active learning strategies for video indexing. In *Proceedings of the th IEEE International Workshop on Content-Based Multimedia Indexing (CBMI'07)*, Bordeaux, France, June 2007. URL `https://hal.inria.fr/hal-00953887`.

[4] David F. Bacon, David C. Parkes, Yiling Chen, Malvika Rao, Ian Kash, and Manu Sridharan. Predicting Your Own Effort. In *Proc. of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 695–702, 2012.

[5] Maria-Florina Balcan and Ruth Urner. *Encyclopedia of Algorithms*, chapter Active Learning - Modern Learning Theory. Springer, 2015. doi: DOI10.1007/978-3-642-27848-8_769-2. URL `http://repository.cmu.edu/cgi/viewcontent.cgi?article=1007&context=machine_learning`.

[6] Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. *Journal of Machine Learning Research Workshop and Conference Proceedings*, 37, July 2015. URL `http://jmlr.org/proceedings/papers/v37/berlind15.pdf`.

[7] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: A Word Processor with a Crowd Inside. In *Proc. of the 23rd ACM Symposium on User Interface Software and Technology*, 2010.

[8] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. In *Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 33–42. ACM, 2011.

[9] I. Cellino. Geospatial Dataset Curation through a Location-based Game. *Semantic Web Jornal, Accepted for publication, Available at http://www.semantic-web-journal.net/content/geospatial-dataset-curation-through-location-based-game-0*.

[10] J. Chamberlain, K. Fort, U. Kruschwitz, M. Lafourcade, and M. Poesio. Using Games to Create Language Resources: Successes and Limitations of the Approach. In I. Gurevych and K. Jungi, editors, *The People's Web Meets NLP. Collaboratively Constructed Language Resources*. Springer, 2013. To Appear.

[11] Aswarth Abhilash Dara, Josef van Genabith, Qun Liu, John Judge, and Antonio Toral. Active learning for post-editing based incrementally retrained mt. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 185–189, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/E14-4036`.

[12] Leon Derczynski and Kalina Bontcheva. Efficient named entity annotation through pre-empting. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 123–130, Hissar, Bulgaria, September 2015. INCOMA Ltd. Shoumen, BULGARIA. URL `http://www.aclweb.org/anthology/R15-1018`.

[13] Pinar Donmez and Jaime G. Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th Conference on Information and Knowledge Management (CIKM'08)*, Napa Valley, California, USA, October 2008. ACM. URL `http://www.cs.cmu.edu/~pinard/Papers/cikm0613-donmez.pdf`.

[14] Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. Formalizing word sampling for vocabulary prediction as graph-based active learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1374–1384, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1143`.

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

[16] Yoav Freund and Robert E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, September 1999. URL `http://www.site.uottawa.ca/~stan/csi5387/boost-tut-ppr.pdf`.

[17] Hossein Hadian and Hossein Sameti. Active learning in noisy conditions for spoken language understanding. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1081–1090, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/C14-1102`.

[18] John Joseph Horton and Lydia B. Chilton. The Labor Economics of Paid Crowdsourcing. In *Proc. of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 209–218. ACM, 2010.

[19] Yuheng Hu, Kartik Talamadupula, Subbarao Kambhampati, et al. Dude, srsly?: The surprisingly formal nature of Twitter's language. *Proceedings of ICWSM*, 2013.

[20] Bernardo A Huberman. Crowdsourcing and attention. *Computer*, 41(11):103–105, 2008.

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

[22] Diego Marcheggiani and Thierry Artières. An experimental comparison of active learning strategies for partially labeled sequences. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 898–906, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1097`.

[23] Justin Martineau, Lu Chen, Doreen Cheng, and Amit Sheth. Active learning with efficient feature weighting methods for improving data quality and classification accuracy. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1104–1112, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-1104`.

[24] Maria Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*, Stateline, Nevada, December 2013. URL `http://www.cs.cmu.edu/~ninamf/papers/statistical-active-learning-nips.pdf`.

[25] P. Minder, S. Seuken, A. Bernstein, and M. Zollinger. CrowdManager - Combinatorial Allocation and Pricing of Crowdsourcing Tasks with Time Constraints. In *Workshop on Social Computing and User Generated Content in conjunction with ACM Conference on Electronic Commerce (ACM-EC)*, 2012.

[26] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. Technical Report T2009:06, Swedish Institute of Computer Science.

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

[28] Longhua Qian, Haotian Hui, Ya'nan Hu, Guodong Zhou, and Qiaoming Zhu. Bilingual active learning for relation classification via pseudo parallel corpora. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-1055`.

[29] W. Rafelsberger and A. Scharl. Games with a Purpose for Social Networking Platforms. In *Proc. of the Conf. on Hypertext and Hypermedia*, pages 193–198, 2009.

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

[31] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl. Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. In *9th Language Resources and Evaluation Conference (LREC-2014). Reykjavik, Iceland: European Language Resources Association (ELRA)*, 2014.

[32] Marta Sabou, Kalina Bontcheva, Arno Scharl, and Michael Föls. Games with a Purpose or Mechanised Labour? A Comparative Study. In *Proc. of the 13th International Conference on Knowledge Management and Knowledge Technologies (iKNOW)*, 2013.

[33] A. Scharl, M. Sabou, S. Gindl, W. Rafelsberger, and A. Weichselbraun. Leveraging the Wisdom of the Crowds for the Acquisition of Multilingual Language Resources. In *Proc. of the LREC*, 2012.

[34] A. Scharl, M. Föls, and D. Herring. Climate Challenge - Raising Collective Awareness in the Tradition of Games with a Purpose. In *14th Brazilian Symposium on Human Factors in Computer Systems (IHC-2015)*, pages 506–509, 2015.

[35] Arno Scharl, Michael Föls, David Herring, Lara Piccolo, Miriam Fernandez, and Harith Alani. Application design and engagement strategy of a game with a purpose for climate change awareness. In *3rd International Conference on Internet Science (INSCI-2016)*, page Forthcoming.

[36] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009. `http://burrsettles.com/pub/settles.activelearning.pdf`.

[37] Manali Sharma, Di Zhuang, and Mustafa Bilgic. Active learning with rationales for text classification. In *North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, 2015. URL `http://www.cs.iit.edu/~ml/pdfs/sharma-naaclhlt15.pdf`.

[38] Yaron Singer and Manas Mittal. Pricing Mechanisms for Crowdsourcing Markets. In *Proc. of the 22nd International Conference on World Wide Web*, WWW '13, pages 1157–1166, 2013.

[39] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and Fast—but is it Good?: Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proc. of EMNLP*, pages 254–263, 2008.

[40] Stefan Thaler, Elena Simperl, and Stephan Wölger. An Experiment in Comparing Human-Computation Techniques. *IEEE Internet Computing*, 16(5):52–58, 2012.

[41] Ruth Urner, Sharon Wulff, and Shai Ben-David. Plal: Cluster-based active learning. *Journal of Machine Learning Research Workshop and Conference Proceedings*, 30, June 2013. URL `http://jmlr.org/proceedings/papers/v37/wei15.pdf`.

[42] L. von Ahn. Games With a Purpose. *Computer*, 39(6):92 –94, 2006. ISSN 0018-9162. doi: 10.1109/MC.2006.196.

[43] A. Wang, C.D.V. Hoang, and M. Y. Kan. Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation*, 47(1), 2013.

[44] A. Weichselbraun, S. Gindl, and A. Scharl. Extracting and grounding contextualized sentiment lexicons. *IEEE Intelligent Systems*, 28(2):39–46, March 2013.

[45] Gerhard Wohlgenannt, Albert Weichselbraun, Arno Scharl, and Marta Sabou. Dynamic Integration of Multiple Evidence Sources for Ontology Learning. *Journal of Information and Data Management*, 3(3):243–254, 2012.

[46] L. Wolf, M. Knuth, J. Osterhoff, and H. Sack. RISQ! Renowned Individuals Semantic Quiz - a Jeopardy like Quiz Game for Ranking Facts. In *Proc. of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 71–78. ACM, 2011. ISBN 978-1-4503-0621-8. doi: 10.1145/2063518.2063528. URL `http://doi.acm.org/10.1145/2063518.2063528`.