Generalized Task Markets for Human and Machine Computation

Dafna Shahaf*
dshahaf@cs.cmu.edu
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

Eric Horvitz
horvitz@microsoft.com
Microsoft Research
One Microsoft Way, Redmond, WA 98052

Abstract

We discuss challenges and opportunities for developing *generalized task markets* where human and machine intelligence are enlisted to solve problems, based on a consideration of the competencies, availabilities, and pricing of different problem-solving resources. The approach couples *human computation* with machine learning and planning, and is aimed at optimizing the flow of subtasks to people and to computational problem solvers. We illustrate key ideas in the context of Lingua Mechanica, a project focused on harnessing human and machine translation skills to perform translation among languages. We present infrastructure and methods for enlisting and guiding human and machine computation for language translation, including details about the hardness of generating plans for assigning tasks to solvers. Finally, we discuss studies performed with machine and human solvers, focusing on components of a Lingua Mechanica prototype.

1 Introduction

Connectivity, programmatic access, and the density of human resources on the Web has enabled *human computation* systems (von Ahn 2008), centering on the online recruitment of people to perform tasks. Human computation includes methods for acquiring human effort as an implicit output of peoples' engagements with online games. The enlistment of people to solve tasks and subtasks is also central in *task markets*, such as Amazon's Mechanical Turk (mTurk.com), which provides an online platform for specifying, recruiting, and reimbursing people for their efforts.

We focus in this paper on opportunities for broadening methods used in human computation to *generalized task markets* (GTM) which admit both human and machine problem solvers, and that employ learning and planning to coordinate and optimize the flow of tasks and subtasks to people and computational solvers. The decomposition and assignment of problems within generalized task markets can rely on distributed processes or on centrally coordinated monitoring and optimization. Research on generalized task markets is related to prior efforts on *complementary computing* on coupling human and machine solvers in an ideal manner via learning and planning (Horvitz and Paek 2007; Horvitz 2007; Kapoor et al. 2008). Work in this area has focused on methods that explicitly consider the competencies and availabilities of both computational and human resources to solve problems at hand.

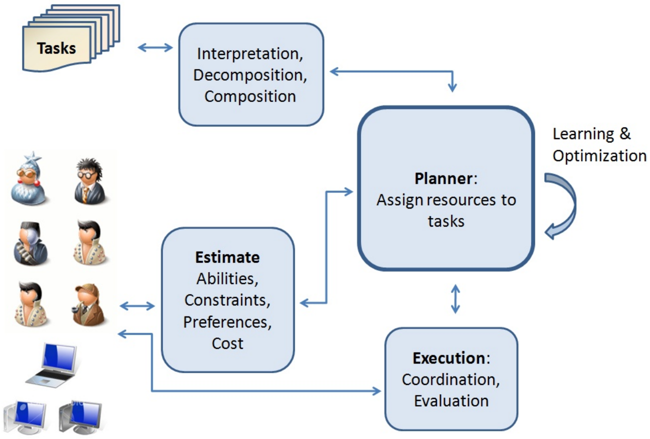
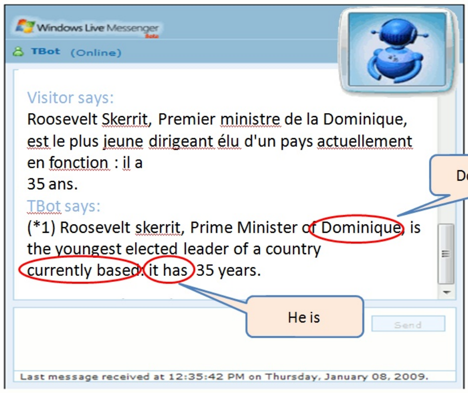

Tasks
Interpretation, Decomposition, Composition
Learning & Optimization
Planner: Assign resources to tasks
Estimate Abilities, Constraints, Preferences, Cost
Execution: Coordination, Evaluation

Figure 1: Schematic view of components of a generalized task market platform.

We foresee the rise of human-computer problem-solving ecosystems that make available pools of candidate perceptual, inferential, and motor competencies that can be harnessed and coordinated by generalized task market systems to create custom-tailored solutions to problems. On the way to such a potential future, several challenges must be addressed. In particular, learning, reasoning, and optimization will play a critical role in making such generalized task markets a reality.

GTM challenges include the development of components that interpret problem instances, decompose these posed problems into subproblems, federate these subtasks to human and computational solvers via computing ideal federation and sequencing plans, and finally composing answers via a combination of the results into overall solutions. Effective distribution and recombination of subproblems requires the system to identify, recruit, and reimburse human and machine solvers. As such, task market platforms may often be enhanced via the use of machinery for learning about competencies, availabilities, and the pricing of differ-

*Research performed during an internship at Microsoft Research.



Visitor says:
Roosevelt Skerrit, Premier ministre de la Dominique, est le plus jeune dirigeant élu d'un pays actuellement en fonction : il a
35 ans.

TBot says:
(*1) Roosevelt skerrit, Prime Minister of Dominique, is the youngest elected leader of a country currently based: it has 35 years.

Last message received at 12:35:42 PM on Thursday, January 08, 2009.

Figure 2: Sample case of machine translation. Errors are marked with circles and sample partial fixes are noted in text balloons.

ent problem-solving resources.

We shall illustrate key aspects of the challenge of developing a GTM platform for the task of language translation. We have constructed components and an overall prototype as part of the Lingua Mechanica project on methods for using a generalized task market for translation. The prototype formulates plans for acquiring translations via machine translation and from people with translation skills, and performs translation among multiple languages. The core approach is that people with different competencies are recruited to refine rough translations generated by a machine translation (MT) system or by other human translators. The system continues to optimize the flow of task assignments based on the expected utility of engaging human or machine assistance, and continues to collect data to refine its understanding of competencies and availabilities.

After an overview of key components of a GTM platform and service, we discuss the hardness of the GTM planning problem for language translation. We provide polynomial approximations to several classes of task federation models. Then, we present the use of games to provide incentives for people to refine and confirm translations. We describe results from studies of the behavior of people working with components of the Lingua Mechanica prototype. We discuss the distribution of language competencies among a population of contributors in a Lingua Mechanica study engaged to provide self-assessments of translation competency and to fix the output of machine translation. We explore the operation of the task federation planner of the prototype on planning problems defined by this user base. Then, we report on the experience with the use of a gaming component for acquiring implicit signals about the accuracy of translations via a game, and discuss the construction of a classifier for learning how to assign accuracy to translations being refined in the Lingua Mechanica pipeline.

2 GTM Planning Problem

As displayed in the schematic in Figure 1, a GTM platform includes components for accepting, interpreting, and decomposing a task into subproblems and for generating task federation plans. A GTM planner accesses information about the availabilities, abilities, and costs (and other preference information) of agents that can provide problem-solving effort in return for some reimbursement that provides incentives solve task subproblems. Execution is monitored and data is collected about multiple aspects of problem solving. A case library of interactions is used to build predictive models about the abilities, availabilities, and preferences of agents. These predictive models are harnessed in the federation planner.

The task federation problem lies at the heart of GTM operation. For this planning challenge, we consider a set of n agents, $\mathbb{A} = \{A_1, ..., A_n\}$. Each agent, A_i is associated with a set of abilities, $\{\mathit{ability}_j\}$. An ability is a pair $\langle b_k, v_k \rangle$ for b_k ability identifier and $v_k \in \mathbb{R}^+$ represents the skill level of the agent for that ability. Agents can represent any source of problem solving drawing upon human or computational resources. Abilities can span a spectrum of competency from such computational tasks as "Factor number N" to "Write a review for movie M." We shall focus on the planning problem for harnessing people and computation to perform translation among multiple languages.

Translation is an illustrative domain for human-machine collaboration within a generalized task market. Translating long documents is often a cumbersome task, requiring skilled bilingual people. Although machine translation (MT) has made significant progress in recent years, human refinements are still essential for natural-sounding results (see (Callison-Burch, Bannard, and Schroeder 2004)). Figure 2 displays an example of an online French-to-English machine translation service. The translation provided is far from perfect, but provides understandable output to English speakers. Moreover, providing fixes for MT output is easier, requiring less skill and can be performed by monolingual people.

In addition to agents, the planning problem is defined as a set of m independent high-level tasks $H = \{h_1, ..., h_m\}$ and a set T of low-level tasks. Each high-level task h_i can be solved via one or more solution (*scenarios*). Scenarios are represented as directed acyclic graphs (DAGs), where vertices are low-level tasks, and edges specify a partial precedence order: there is an edge from t_i to t_j if task t_j cannot be performed unless task t_i is already satisfied (e.g., because the output of t_i is needed as an input to t_j). A low-level task t_j is associated with a set demand_j of abilities.

Figure 3 displays a representation of a sample *high-level task* for translation. In this case, we wish to translate a paragraph from English to French. The figure shows three different scenarios for accomplishing this task: (a) acquire a bilingual person who can translate the paragraph directly, (b) acquire two bilingual people, one of whom can translate from English to an intermediate language (in this case, Italian), and the other from the intermediate language into French, or (c) use machine translation procedures to translate the paragraph from English to French and then engage a French-speaking person to correct the paragraph. Scenario b might be assumed a priori to require more competent language skills from both of the human translators, as the pipeline with two translations is likely to create more errors than single-stage translations. However, more generally, the errors incurred with each scenario will depend on the competencies of the human and computational agents.

More formally, we say that a task market *coalition* $C \subset$

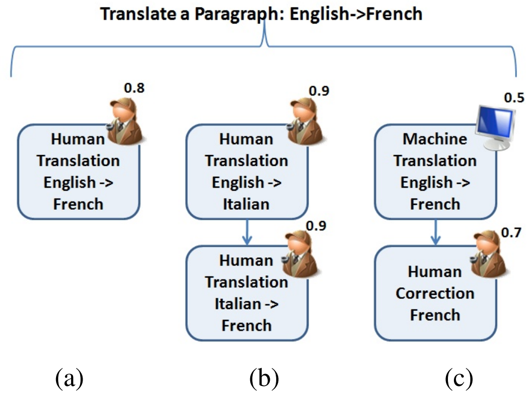

Translate a Paragraph: English->French
0.8
Human Translation English -> French
0.9
Human Translation English -> Italian
0.9
Human Translation Italian -> French
0.5
Machine Translation English -> French
0.7
Human Correction French
(a)
(b)
(c)

Figure 3: Three translation scenarios: (a) direct human translation, (b) human translation in two steps via an intermediate language, and (c) machine translation followed by repair by a person. Numbers refer to target quality.

$\mathbb{A}$ is defined as a group of agents which cooperate in order to achieve a common task. We assume that a coalition can work on a single (low-level) task at a time. We consider situations where agents may or may not be members of more than one coalition. The utility achieved by performing the task t_j by coalition C is $U(C, t_j)$. The utility depends on tasks demands and the coalition's composite abilities. The utility of a high-level task h_i depends on the utilities of the low-level tasks that comprise the high-level task.

We now consider approximations to the task federation planning problem. We first assume modularity of effort, which asserts that the combined abilities of a coalition of agents is the sum of the abilities of the participating agents. If agents participate in more than one coalition, they can distribute their abilities among the coalitions. In the context of a binary utility model, where a task is either completed for full utility or not completed generating no value, modularity of effort means that t_j can be satisfied by a coalition of agents C only if their combined abilities are higher than the task's demands. Otherwise, $U(C, t_j) = 0$. A high-level task h_i is satisfied if all of the low-level tasks t_j of at least one scenario are satisfied. Later, we shall consider other types of utility functions.

Definition 2.1 (Coalition Problem)**.** *Given $\langle \mathbb{A}, H, T \rangle$, the coalition problem is to assign tasks $t \in T$ to coalitions of agents $C \subseteq \mathbb{A}$ such that the total utility is maximized and the precedence order is respected.*

In general, agents and tasks are both associated with sets of constraints (e.g., dollar value, availability, time criticality). In this formulation, the problem is to maximize the utility under all of the constraints.

3 Hardness and Approximations for the GTM Planning Problem

We shall now review the hardness of the GTM federation planning problem and provide polynomial approximations for models of different expressiveness.

Simple Tasks

Let us start with a simple problem: A high-level task h_i is composed of a single scenario, which is composed of a single low-level task t_i.

Hardness The problem is $\mathcal{NP}$-hard. The decision problem is in $\mathcal{NP}$ (given a solution, it can be verified in polytime). We show a reduction from weighted exact set cover.

Definition 3.1. *Given a set X and a set $\mathcal{S}$ of subsets of X with associated rewards $R(S)$ for each $S \in \mathcal{S}$, an exact set cover $\mathcal{S}^*$ is a subset of $\mathcal{S}$ such that every element in X is contained in exactly one set in $\mathcal{S}^*$. The goal is to find the exact cover with the maximal reward.*

Weighted exact set-cover is NP-hard. Given a weighted exact set-cover instance, we construct a coalition instance: Let $\mathbb{A} = X$. Each $S_j \in \mathcal{S}$ defines a high-level task h_j. h_j is composed of a single scenario and a single low-level task t_j, requiring $\langle b_j, |S_j| \rangle$ ($|S_j|$ units of ability b_j). Each $A_i \in S_j$ has 1 unit of ability b_j ($\langle b_j, 1 \rangle$). Therefore, the only coalition able to perform t_j (and thus, h_j) is S_j. Define the evaluation function such that $\sum U(S_j, t_j) = R(S_j)$.

A solution to the coalition problem corresponds exactly to a solution to weighted exact set cover. Thus, the problem is $\mathcal{NP}$-hard.

Tractable Approximation via Limiting Coalition Size The number of possible coalitions considered in GTM plan generation is exponential in n. Thus, a natural way to reduce the search space is to restrict the maximal size of a coalition to k, thus reducing the number of coalitions to $O(n^k)$. Such a restriction is reasonable for language translation as most tasks do not require more than a few participants.

We consider a graph, in which the agents are vertices and tasks are a collection of weighted hyperedges. A hyperedge corresponds to assigning these agents to the task; its weight is the expected reward. This is also a useful framework for the case of extra constrains (e.g., that status of the availability of an agent that is assigned to a coalition): in this case, hyperedges correspond to the possible coalitions.

Every task is identified with a color, and all of its edges are of this color. We want to find a maximum value matching in the graph such that only one edge of each color is used.

Claim 3.2. *A greedy strategy for selecting agents is a constant-factor approximation to the coalition problem (k is constant).*

The proof is based on a standard greedy analysis. Any coalition chosen for the solution contains at most k agents, and we compare their assignment to the optimal solution's assignment.

Tractable Approximations via Special Utility Functions We now relax the assumption of limited coalition size. The number of coalitions we need to consider is exponential in n again. We now seek tractability by introducing constraints on the utility functions to restrict our search space. We can assume *submodularity* (diminishing returns, $u_i(S \cap S') + u_i(S \cup S') \leq u_i(S) + u_i(S')$), subadditivity ($u_i(S \cup S') \leq u_i(S) + u_i(S')$), or superadditivity.

As an example, suppose that the utility functions are monotone and submodular.

Claim 3.3. *A greedy algorithm yields a $\frac{1}{2}$-approximation. In special cases where the submodular function is of a special type (discussed later), a $(1 - \frac{1}{e})$- approximation has been achieved; this is optimal for these problems.*

Those results follow from efficient approximation algorithms (Nemhauser, Wolsey, and Fisher 1978; Vondrak 2008) that exist for the Submodular Welfare Problem. The coalition problem can be easily formalized as a Welfare Maximization Problem. In this problem, m items are to be distributed among n players with utility functions $u_i : 2^{[m]} \rightarrow R^+$. Assuming that player i receives a set of items S_i, we wish to maximize the total utility $\sum_i u_i(S_i)$. In our case, an agent corresponds to an item and tasks correspond to players. For task t_i, utility function $u_i(C)$ is $U(C, t_i)$.

Tractable solutions can also be formulated for subadditive utility functions. The problem has been transformed into linear programming problem (Feige 2006), with the rounding of fractional solutions, achieving an approximation ratio of $\frac{1}{2}$. For the superadditive case, an approximation ratio of $\frac{\sqrt{\log m}}{m}$ can be achieved (Mirrokni, Schapira, and Vondrak 2008).

Tasks Consisting of a Single Scenario

We now move to the more complex case where a high-level task h_i takes the form of a single scenario. The scenario DAG corresponds to a partial ordering on low-level tasks t_j.

The hardness of the problem follows from the previous section. We build on the algorithms described by Shehory and Kraus, to provide a constant-factor approximation to the case of limited-size coalitions. The algorithm can be viewed as a centralized version of the methods described in (Shehory and Kraus 1995).

For each task t, we denote its set of predecessors, including t, by $P(t)$. The choice of t for coalition formation will depend on the costs and benefits of the formation of coalitions that perform all of the tasks in $P(t)$.

At each iteration, we choose in a greedy manner to perform the task t associated with the maximal $u(P(t))$ together with all of its predecessors, and form the required coalitions. We remove t, all of its predecessors, and the assigned agents from the list, and iterate. We note that only a small portion of the calculations needs to be updated.

Tasks with Multiple Scenarios

We now consider the most expressive case where there are multiple scenarios for solving tasks. As before, we assume that the size of coalitions is limited. We also limit the maximal number of low-level tasks executed within each scenario.

The algorithm we propose is a combination of two algorithms we have already discussed. We again construct a hypergraph, where a high-level task is a set of hyperedges, corresponding to coalitions that can carry out one of the scenarios. When a coalition is chosen, one can also optimize over the scenarios it can carry. The analysis of this case is similar.

Case of Transitive Tasks An interesting case of multiple scenarios for GTM solvers is where the tasks are *transitive*. Transitive tasks allow for a compact representation of many possible scenarios by encoding them as graph paths. Such transitivity is especially valuable in GTM for language translation. The task of translating from French to English is the same as translating from French to any intermediate language, and then to English. The path might even involve a chain of translations among several intermediate languages.

A natural way to solve the case of transitive tasks is with *multi-commodity flow*. We have a weighted multigraph $G = (V, E, w)$. Each $s \in V$ represents a language. Each problem-solving resource, whether a human or machine reasoner, corresponds to a set of directed edges. An edge weight represents the user's degree of competence at translating between the two languages represented by the nodes that it connects. As an example, in Figure 4 we characterize with numeric scores the competencies of different translators. We have one participant who speaks Hindi and Telugu well. Another participant is a Hindi speaker who knows some English and Kannada (dashed edges correspond to low weights on competency). Finally, we have several speakers of European languages.

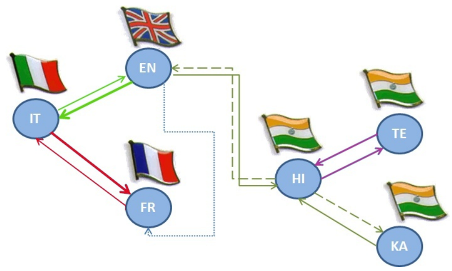


Figure 4: Translation abilities graph. Each resource is a collection of edges (in the same color). Dashed/thin edges correspond to low skills, thick edges correspond to high skills.

In addition, we have k tasks (commodities), where each one has source language $s_i \in V$, sink (target language) $t_i \in V$, and a starting flow $d_i \in R$ (in the basic scenario, $d_i = 1$). The goal of a GTM federation planner is to maximize the total throughput.

Note that unlike our earlier binary utility notion (a task is either completed for full utility, or generates no value), the linear program allows for a softer notion of task quality.
As an example, the GTM planning problem displayed in Figure 4 shows two candidate paths for English-to-French translation for the solvers available: direct translation or translating from English to French through a translation by one person from English into the intermediate Italian language and then by another person from Italian to French. The low weight on the direct edge indicates a low competency for the direct translation, but the other two edges are strong. However, as errors accrue over chains, the use of intermediate translations would tend to reduce the quality. How should we compare the output qualities?

In GTM for translation, edge weights represent the probability that the translation at each transition will reach a specified threshold of quality. Thus, the quality of a path is the *product* of its edge probabilities. We need to modify the traditional multi-commodity flow solution to handle this case. Each directed edge e has f_i^{in} , f_i^{out}, and a capacity c_e. The key concept is that when flow goes through an edge, it suffers some loss in quality. We seek some threshold quality at the sink. (Note that the product model is simplistic. In the future, we will need to either learn or model the way quality depends on competencies along the sequence.)

We shall first formalize this optimization problem as a linear program (LP), and then introduce rounding techniques to achieve a feasible solution. We can formalize the optimization problem as follows:

$$
\begin{aligned}
&\max \sum_{i} \sum_{e:\, \textit{edges to } t_i} f_i^{out}(e) \;\; \textit{s.t.} \\
&\qquad \sum_{i} f_i^{in}(e) \le c_e \\
&\qquad \sum_{e:\, \textit{edges from } s_i} f_i^{in}(e) = d_i \\
&\qquad f_i^{out}(e) = w(e) \cdot f_i^{in}(e) \quad \textit{(*Quality reduction*)} \\
&\qquad \sum_{w} f_i^{out}(w,u) = \sum_{v} f_i^{in}(u,v) \quad u \neq s_i, t_i \quad \textit{(*Flow*)}
\end{aligned}
$$

The GTM plan seeks to maximize the quality of commodity i reaching its sink. The flow on every edge (workload) is no more than the capacity of every edge. Alternatively, we can restrict total capacity per contributor.

Another interesting feature of transitive tasks is the feasibility of reusing intermediate results. Consider the sample goal of translating an English article into both Kannada and Hindi. The previous algorithm would treat them as two separate problems, translating directly from English into both. However, given the resources and competencies available, it may be more beneficial to translate English to Hindi, and then Hindi to Kannada. Alternatively, it may be better travel via an intermediate node for both.

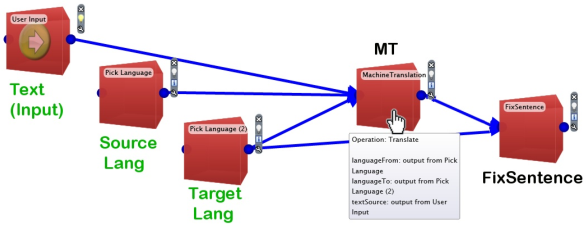
A similar flow formulation can be applied for re-using intermediate results. We define a variable for every $\langle$language,task$\rangle$ combination (the quality of translation for a specific task in that language). A task would have one source and possibly many sinks. The goal now is to minimize the number of intermediate translations, while enforcing minimal translation quality in all sinks. We note that this minimization is a constraint, not the objective in this case. The objective should be interpreted as a routing constraint. Longer paths cost more and are prone to the accrual of errors. While quality is acceptable, we prefer shorter paths.

Given a set of directions for tractable implementation of GTM federation, we will now move to describe studies of components of a Lingua Mechanica prototype.

4 Lingua Mechanica Prototype

We implemented a Lingua Mechanica prototype as a set of components for experimental studies of federation planner and of computational and human performance on the translation domain. In this section, we shall describe the prototype.

The Lingua Mechanica prototype includes components that interact with people, including a method for displaying sentences (including prior machine or human translations of sentences) and seeking direct translations, and tools for authoring and fielding of games for engaging people to provide human computation as part of gaming. We will later present more detail on several sample games and other interactive capabilities for capturing translation expertise.



Operation: Translate

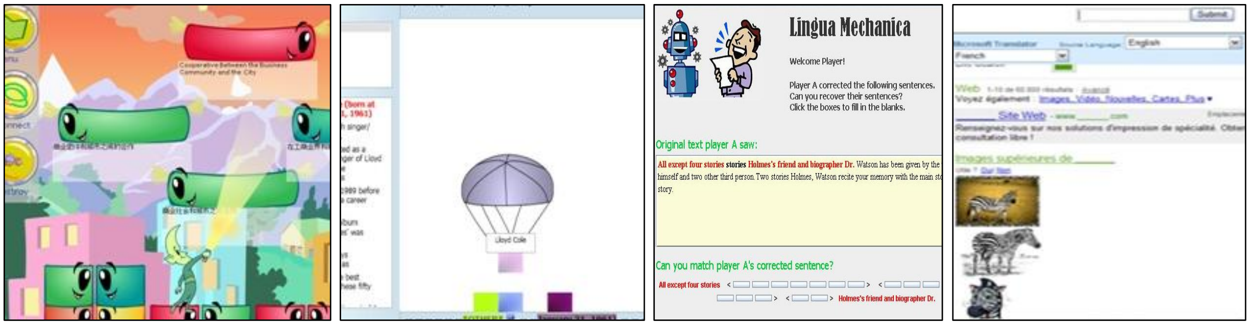
languageFrom: output from Pick Language
languageTo: output from Pick Language (2)
textSource: output from User Input

Figure 5: Screenshot of Lingua Mechanica interface for creating scenarios. The three blocks on the left correspond to inputs. The blocks on the right represent problem solving effort.

The prototype also provides functionalities for defining tasks. Common tasks are input via task templates. As an example, the prototype provides a "Translate a page" option. The author of the task provides the system with several task parameters, including the content that needs to be translated and the source and destination languages. In addition, the author of the task can change the metadata associated with the task, including parameters describing the desired quality, deadlines, and how much (if any) they are willing to pay for human contributions. The price and deadline affects the result: the planner can route the request through a professional translator (high-quality, slow, expensive) or through MT (low-quality, quick, cheap). Translation challenges can also be defined as custom-tailored high-level tasks with associated parameters, and details about the flow of machine and human effort on tasks can be specified. We use a Popfly-like[1] interface for such authoring. Figure 5 displays an example graphical specification of tasks and execution. Task authors define and connect blocks together into a DAG, such that the output of one block can be used as inputs to others. The specifications correspond to *scenarios* described in Section 2. The sample specification in Figure 5 corresponds Scenario c in Figure 3, where the output of the "Machine Translation" block is the input of "Fix Sentence." We implemented the GTM planning algorithm for task federation that executes the flow procedure described in Section 3. Rather than having a single node representing a language, the planner considers two nodes for each language (before improvement, after improvement), extending the plans to include monolingual people who can improve the quality of a sentence that has been translated into their language by a person or machine. A monolingual person corresponds to a directed edge between those two nodes. We set the 'after improvement' node as our sink node for tasks. Finally, we add an edge of no cost (and no quality loss associated) between each such pair of nodes, so the sink is reachable even if no monolingual people are involved.

Several other changes were made with the planning algorithms. First, to speed up the assignment process, we defined equivalence classes of abilities. Instead of using the fine-grained $\langle b_i, v_i \rangle$ abilities, we mapped them to equivalence classes (thus mapping agents and tasks). For example, we defined three levels of English proficiency. We refer to this

[1] Popfly.com is a website allowing users to create mashups.



Lingua Mechanica

Welcome Player!

Player A corrected the following sentences.
Can you recover their sentences?
Click the boxes to fill in the blanks.

Original text player A saw:

All except four stories stories Holmes's friend and biographer Dr. Watson has been given by the himself and two other third person. Two stories Holmes, Watson recite your memory with the main st story.

Can you match player A's corrected sentence?

All except four stories < □ □ □ □ □ □ □ > < □ □ □ □ □ □ > < □ □ > Holmes's friend and biographer Dr.

Figure 6: Games for translation. From left to right: Moon Climb, WordTetris, Hangman, Dictionary Builder

primary method for task federation as the *Flow* procedure.

When the greedy algorithm picked a coalition to perform a task, equivalent tasks and coalitions are identified and we assign as many of those as possible. The approximation guarantee still holds, and computation is significantly faster.

We also integrated logging functions to capture traces of the detailed activity of people making contributions and machine components, as well as information about the availability of people. Such data can allow the system to learn from data so as to better optimize task federation. There are multiple places for learning about machine reasoners, as well as people and their competencies and availabilities in a GTM system. As an example, the online nature of the system makes learning about the availability of people an attractive functionality. A planner might not try to execute a translation to Chinese now if it knows some highly-skilled contributors will log in soon. We will describe one of the learning tasks in more detail in Section 6.

5 Acquiring Human Computation

We have explored several methods for engaging people to provide human computation for translation, including the use of games with a purpose and providing monetary rewards or lotteries for rewards.

Accessing Human Computation via Games

Incentives for recruiting human computation include providing monetary rewards, receiving points, achieving skill levels, increased rankings or reputation in a community, altruism, patriotism, and playing or competing in games. There has been growing interest in engaging people with participation in fun or challenging online *games with a purpose*. This idea was first explored by (von Ahn and Dabbish 2004), who used a matching game to seek labels about images from people. We designed and implemented several different games for enlisting human computation for translation, drawing assistance from MT experts. Screenshots from these games are displayed in Figure 6. The game of MOON CLIMB challenges bilingual players to match a sentence with its best translation. With DICTIONARY BUILDER, players translate specific words (e.g., words that are not available in a bilingual dictionary or words assigned low confidence by a translation system). The players are shown translated search results for this word (images and snippets), and try to guess the missing word. We shall provide more detail on two additional games, HANGMAN and WORDTETRIS. Additional information on these games is provided in (Shahaf and Horvitz 2009).

The goal of HANGMAN is to fix a machine-translated paragraph. Players switch between serving as a *fixer* and as a *guesser*. The fixer receives content translated by MT and improves it; the guesser sees the translated version *and* the fixer's sentence, with the changes displayed as empty spaces. The guesser tries to fill in the missing parts, one letter at a time, in a Wheel-of-Fortune manner. Both players receive points for each word correctly identified.

WORDTETRIS also centers on the fixing of machine translations. Two players who speak the target language see blocks attached to chunks of a sentence falling from the sky. Each block may have several alternatives corresponding to MT uncertainty (e.g., "il" is either "he" or "it," as displayed in Figure 2). The initial choice is random per user. Users choose the relative location of the block (effectively choosing word ordering), and its phrase, applying wildcards and blanks when needed. Their goal is to match the sentence with their partner. A complete match makes a whole row disappear, while partial matches result in partial disappearances. The game ends when the stack of blocks reaches the top of the playing field and no new blocks are able to enter.

In a validation study, we found that participants overall found WORDTETRIS to be the most engaging game out of the four tested. Most players became comfortable with WORDTETRIS within 1-3 rounds. People enjoyed the challenges provided by the game, which includes a speeding up with play. On the downside, translations decline in quality as the game speeds up.

Evaluating Contributors

It is vital to have a means of determining the quality of solutions provided by human computation. In preliminary studies, users' linguistic abilities are determined through a short questionnaire about the languages that they know and their self-assessment about their competence. Contributors can take optional qualification tests to raise the system's confidence about their abilities (and to be offered better challenges). Beyond explicit testing, with time and repeat contributions, the system can learn about competencies using instream testing based on known answers and on alignments of output with those received from others working on the same instance, and a natural language tool that provides scores on grammar correctness (Cherry and Quirk 2008).

6 Experiments

We now review results from studies of several aspects of the Lingua Mechanica prototype, including the performance of the task federation planner and the performance of people on two interrelated translation tasks.

Participant Base

For the studies, we sought a set of participants with a representative distribution over translation skills. We collected a snapshot of language competencies from 388 participants from 70 different countries willing to contribute to Lingua Mechanica, out of a larger group of 2,500 people we invited to help. The invitees were selected at random from the full Microsoft employee address book, spanning all Microsoft sites throughout the world. An incentive of a lottery ticket for a gift certificate was provided to invitees.

Before being engaged in translation challenges, the participants were asked to provide information on their country of origin, current location, and language competencies. The participants were asked to rank their skill level in reading and writing for the set of fifteen popular languages under consideration. Participants assessed for each language whether they were a beginner, intermediate, or advanced, based on the topics they can understand (very limited vocabulary/ conventional/ sophisticated) and the amount of editing required to fix sentences they construct.

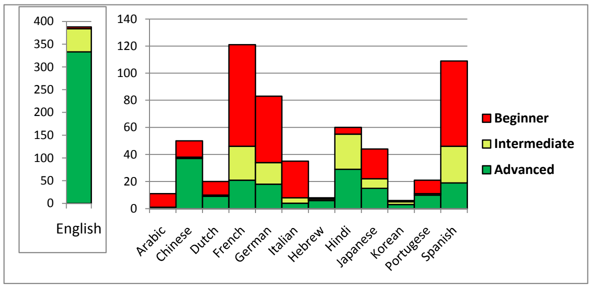


Figure 7: Main languages spoken by study participants, broken down by skill level.

Figure 7 shows the main languages spoken by the participants. To assess the accuracy of self-assessments of competency, participants were asked to answer a short list of questions for each language they speak. The questions were designed to test grammar, vocabulary, and familiarity with idioms. We found that in general, people who assess themselves as experts typically can be trusted. Interestingly, we found experts could sometimes even recover mistranslated cultural terms. For example, several experts managed to reverse-translate the mistranslated Sherlock Holmes book title, ”bright red novel study,” back to ”A Study in Scarlet.” However, people who report themselves as being intermediate in skill often over-estimate their abilities. Indeed, we had several people entering comments at the end of the competency survey such as, “My French is probably rustier than I remembered.”

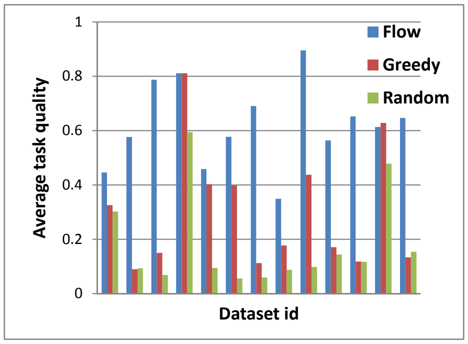


Figure 8: Comparative analysis of federation planners. For each dataset, we show the average task quality for Flow, Greedy and Random strategies.

Studies of GTM Planner

We first report on a comparative analyses of the performance of GTM planning algorithms. We used the inputs gathered from participants as a proxy and source of distributional information. We ran task federation algorithms on a set of benchmark cases, each containing representative tasks and agents. Benchmark cases are lists of tasks, each defined by source and target languages and the minimal quality desired. We compared the average task quality generated by a rounding of the primary Flow procedure, as described in Section 3, with the performance of a greedy and a random selection strategies. Average task quality is the average over the end quality over all tasks in the dataset. A task which was not completed is considered to have zero quality. The Greedy strategy iterates over the tasks, selecting paths of acceptable quality that use the fewest resources (humans and computers each cost a single unit; some edges are free, as discussed in Section 4). The Random strategy chooses uniformly at random among the same paths. The results (Figure 8) show that the performance of Flow nearly always dominates (by a significant factor) Greedy, which in turn nearly always dominates Random.

The datasets consisted of random translation tasks. The language pairs were determined from our survey distribution: some datasets focused on translation between languages with high correlation (e.g. Spanish/Portuguese), while others focused on the opposite case (e.g. German/Chinese).

Studies of Human Repair of Machine Translation

We also explored the abilities of the participants in our study to repair machine-translated sentences in the languages they indicated they had some degree of competence with. The base machine translations and the repaired sentences were checked for *grammatical correctness* and *understandability* by experts, checking whether the intended meaning of the sentence is clear, even if there are some mistakes. We expected that experts would be accurate raters. We confirmed that their relative ratings were frequently consistent with other experts.

Translation challenges were created by translating sentences drawn from Polish Wikipedia to multiple languages via the Bing Translator. We found that participants could improve machine translation to an acceptable level of understandability. Detailed traces of machine translations and repairs are provided in (Shahaf and Horvitz 2009).

We next explored if game playing could provide a valuable signal about the accuracy of the sentences that participants had attempted to repair in the first phase of the study. We invited a group of 6 new participants from our organization to play a version of the HANGMAN game (Section 5), for an incentive of a lunch coupon. In each trial of the game, a sentence that had been refined by participants in the earlier phase described above was displayed in Hangman style, with empty slots appearing in lieu of letters. Players were asked to try to fill in all of the missing letters in the sentence by issuing a sequence of letter guesses. Letters correctly guessed were immediately rendered in the appropriate places in the sentence. Players gained points for each completed sentence and worked as fast as they could under a total time allotted for the overall session.

We noted that poor translations would often take players longer to complete. As an example, some of the translators in the first phase of the refinement (especially beginners), fixed the machine translation in a way that is grammatically correct and understandable, yet wrong. For instance, one of the participants had translated that, "Sherlock Holmes wrote books about Arthur Conan Doyle." During the gaming study, players encountering such translations often looked confused, and paused to reflect about potential alternatives before guessing letters. Similarly, sentences using the wrong prepositions caused almost all translators to make the same mistakes (e.g. trying to type the correct preposition 'in', instead of the wrong 'at').

We collected logs of the guesses and timing and constructed a library of cases. Each case was labeled with a measure of the translation accuracy. Labels were provided by experts. Features included knowledge about the initial translator (skill level, both reported and assessed), knowledge about the sentence itself (number of words, grammar correctness score, number of conjunctions), and features drawn from logs of activity by players (the maximal pause time, minimal pause time, number of letters guessed correctly, and total time to guess). Those features were normalized across players (dividing by the length of the longest round for each player), to compensate for speed differences. We trained a classifier using logistic regression from the case library. We used leave-one-out cross-validation. The classifier allowed us to boost classification accuracy of the accuracy of translations by 12% (from a baseline marginal of $\sim 31\%$ to $\sim 43\%$). We believe that adding richer features could further improve the classification accuracy.

7 Summary and Directions

We described the opportunity for developing generalized task market platforms, with a focus on human-computer task markets that mesh human expertise and machine intelligence to solve problems. We discussed key components of such systems, centering on the challenge of generating plans for distributing subproblems to people and machines based on competencies, availabilities, and pricing of the different solvers. We examined the hardness of planning and introduced approximations. We described how we can harness a multi-commodity flow procedure for federating tasks. We illustrated ideas in the context of efforts on Lingua Mechanica, a human-computer task market prototype for language translation. We reviewed components of the prototype that provide methods for authoring translation tasks and for acquiring human computation via rewards and games with a purpose. Finally, we reviewed several evaluation studies performed to probe the performance of different aspects of the machine and human computation harnessed by a Lingua Mechanica prototype. Future research includes exploration of applications that involve the weaving together of broader classes of resources, including perceptual, inferential, and motor skills to solve new classes of problems. In the most general case, task markets should handle tasks requiring perception, reflection, and action in the world, generating solutions that weave together combinations of these skills. There are a number of challenges ahead with the creation of a rich ecosystem of machine and human resources and with operating generalized task markets in the open world. However, we foresee multiple opportunities on the horizon with the development and fielding of generalized task markets that can ideally tap human and machine intelligence.

Acknowledgments The authors thank Vikram Dendi, Paul Koch, Raman Sarin, Paul Newson and Tommy A. Brosman for their assistance on the Lingua Mechanica project.

References

Callison-Burch, C.; Bannard, C.; and Schroeder, J. 2004. Improved statistical translation through editing. In *EAMT-2004*.

Cherry, C., and Quirk, C. 2008. Discriminative, syntactic language modeling through latent svms. In *AMTA '08*.

Feige, U. 2006. On maximizing welfare when utility functions are subadditive. In *STOC '06*.

Horvitz, E., and Paek, T. 2007. Complementary computing: Policies for transferring callers from dialog systems to human receptionists. *User Modeling and User Adapted Interaction*.

Horvitz, E. 2007. Reflections on challenges and promises of mixed-initiative interaction. *AI Magazine* 28(2).

Kapoor, A.; Tan, D.; Shenoy, P.; and Horvitz, E. 2008. Complementary computing for visual tasks: Meshing computer vision with human visual processing.

Mirrokni, V.; Schapira, M.; and Vondrak, J. 2008. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *EC '08*. ACM.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming* 14(1).

Shahaf, D., and Horvitz, E. 2009. Investigation of human-computer task markets: Methods and prototype. Technical Report Microsoft Research Technical Report MSR-TR-2009-181, Microsoft Research.

Shehory, O., and Kraus, S. 1995. Task allocation via coalition formation among autonomous agents. In *IJCAI '95*.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *CHI*.

von Ahn, L. 2008. Human computation. In *ICDE*. IEEE.

Vondrak, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC '08*.