原著論文

Original Paper

マイクロタスク型クラウドソーシングプラットフォーム環境における精度向上手法の導入と評価

Development and Evaluation of Quality Control Methods in a Microtask Crowdsourcing Platform

芦川 将之
masayuki ashikawa
(株) 東芝 研究開発センター
Corporate Research and Development Center, Toshiba Corporation
masayuki.ashikawa@toshiba.co.jp

川村 隆浩
takahiro kawamura
（同　上）
takahiron.kawamura@toshiba.co.jp

大須賀 昭彦
akihiko ohsuga
電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, The University of Electro-Communications
ohsuga@uec.ac.jp

keywords: crowdsourcing, quality control, worker control

Summary

Open Crowdsourcing platforms like Amazon Mechanical Turk provide an attractive solution for process of high volume tasks with low costs. However problems of quality control is still of major interest. In this paper, we design a private crowdsourcing system, where we can devise methods for the quality control . For the quality control, we introduce four worker selection methods, each of which we call preprocessing filtering, real-time filtering, post processing filtering, and guess processing filtering. These methods include a novel approach, which utilizes a collaborative filtering technique in addition to a basic approach of initial training or gold standard data. For an use case, we have built a very large dictionary, which is necessary for Large Vocabulary Continuous Speech Recognition and Text-to-Speech. We show how the system yields high quality results for some difficult tasks of word extraction, part-of-speech tagging, and pronunciation prediction to build a large dictionary.

1. は じ め に

クラウドソーシングは，2006 年に Wired 誌の Jeff Howe によって提唱された．Crowd(群衆) ＋ Sourcing(調達) の造語であり,「企業，組織が，自社もしくはアウトソースの人材により実施していた業務を，よりオープンかつ不特定多数の Crowd(群衆) から人材を集め実施すること」と定義されている．企業などが目的 (需要) を提示し，それを不特定多数の情報発信者が参加して解決 (供給) することで大量の作業をこれまでよりも効率よく処理することが目的である．これまでは不特定多数の人間に対して目的を提供，結果の収集を行うことが難しかったが，インターネットの技術革新に伴い可能となった．

我々はこのクラウドソーシングの技術を様々な研究データの解析に用いている．研究データの作成は精度的な問題から自動化出来ないケースが多く，研究者，もしくは専門の技術を持った外部の業者といった人手による作業が必要になる．しかし，昨今の研究に用いられるデータはビッグデータと称される巨大なデータであることが多い．従来の人手による作業では巨大データを扱うにはコスト，速度の面から難しくなってきている．そこで，我々はクラウドソーシングを用いている．

既存のクラウドソーシングサービスとして Amazon Mechanical Turk[AMT] や Yahoo!クラウドソーシング [Yahoo!] などの様々なサービスが存在する．しかしこれらの外部サービスを研究データの作成に利用するには精度の面で問題があった．我々は作業 (タスク) の処理結果を研究データとして用いるため作業結果の品質を高く維持しなくてはならないが，そのためには外部のサービスが提供している機能の範囲では十分ではなく，さらに外部のサービスに新規の機能を追加することも難しい．我々はこれらの問題を解決するために，独自の精度向上手法の適用が可能なクラウドソーシングシステムをプライベートな環境下において構築し，様々な精度向上手法を組み込むことで問題の解決を試みている．

本研究では，フィルタリング手法を組み合わせることでコスト面を考慮しつつ，ワーカーを効率的にコントロールする精度向上手法を提案した．また，プライベート環境

下において精度向上手法を適用した独自のクラウドソーシングプラットフォームを用いて，実際に実務に適用することで精度向上が可能であることを確認した．これによりマイクロタスク型クラウドソーシングのサーバ側における精度向上の導入効果を実証出来た．

本稿では我々が対象としているマイクロタスク型のクラウドソーシングおよび関連研究に関して説明し (2 章)，外部のポイント業者が*1保持している会員を作業者 (ワーカー) 候補としたプライベートな環境下におけるクラウドソーシングシステム (PCSS) の構築方法と精度向上の手法について報告する (3 章)．さらに構築した PCSS を自然言語処理の研究に必要な語彙の収集にて実際に運用した例と精度向上手法の効果を報告する (4 章)．

2. マイクロタスク型クラウドソーシング

クラウドソーシングの定義は非常に緩やかなものであり，特定の目標に対して不特定多数の人間が関わって作業をしていればクラウドソーシングとして扱われている．本章では我々が対象としているマイクロタスク型のクラウドソーシングに関して説明する．

マイクロタスク型のクラウドソーシングとは，企業や組織が用意した大量の難易度の低いタスクを，数多くの不特定のワーカーが処理する形式のクラウドソーシングである． 企業が提示したタスクに対してワーカー候補が応募を行い，タスクを提示した企業や組織が募集に応じたワーカー候補から条件にマッチするワーカーを選び契約を行う形式のマッチング型クラウドソーシングと比較してタスクの難易度は低いことが多いため，一つのタスクにかかる時間は数秒から数分と非常に短いが，支払われる単価も低く設定されており大量にタスクを処理することが前提となっている．Amazon Mechanical Turk, Yahoo!クラウドソーシング，クラウドワークス [クラウドワークス]，ランサーズ [ランサーズ] などがこの形式のクラウドソーシングを行っている．

2·1 関 連 研 究

マイクロタスク型のクラウドソーシングの性能を測る指標は数多くあるが，本稿では「コスト」「精度」「速度」をクラウドソーシングの性能を測る指標として考える．これらは相互に負の相関関係を持つことが多い．例えば，コストを下げるために報酬を下げるとワーカーのモチベーションに負の影響がでてタスクに対する処理速度が低下する．また，精度向上のために一つの問題を複数のワーカーに出題する場合において，コストを下げるためには一問あたりのワーカー数を減らさなければならず，結果として精度も低下するなどである．

*1 自社の会員に対して他社のアンケート入力作業やサービスなどを紹介し，作業結果やサービス利用の対価として一定の条件で計算されたポイントを与えるサービス．ポイントは商品や現金と交換することが可能．

マイクロタスク型のクラウドソーシングはその特性上「安価で大量の処理が可能」という点に注目されることが多く，精度は優先度を低く設定されがちである．また，マイクロタスク型は一つ一つの作業の難易度が低いことも多く，精度を軽視させる要因の一つとなっている．しかし，今後のクラウドソーシングの普及に伴い，タスクの内容が多様化することが予想され，精度に関しても高レベルの要求がなされる可能性がある．

これまでもマイクロタスク型のクラウドソーシングの精度を向上させる方法に関して様々な研究がなされている．我々はこれらの研究を以下の 3 つのカテゴリに分類した．

(1) タスクに対する精度向上手法
(2) ワーカーに対する精度向上手法
(3) 作業出題者 (リクエスタ) に対する精度向上手法

(1) に関する研究はタスクのデザインに関する研究である．問題の表示方法や入力インターフェイスのデザインだけではなくタスクの進め方，出題方法などタスクに関する改善全般が該当する．タスクのデザインを改善することで精度向上につなげる研究 [Kittur 08]，タスクを複数に分割してワーカーの能力に応じて割り当てる研究 [松原 13]，タスクを複数のワーカーに出題し，結果を融合させることで精度を向上させる研究 [Dawid 79, Welinder 10, Whitehill 09, Mao 12]，ワーカーにタスク処理と同時に処理結果の精度への確信度を申告させる研究 [櫻井 12, 小山 13] などが行われている．既存のサービスにおいても正解が予めわかっている問題をタスクに混ぜ，その結果を用いてワーカーの能力をはかり選別する手法 [Yahoo!] などが行われている．

(2) に関する研究は作業を行なうワーカーに関する研究である．ワーカーに信頼度の高いワーカーを紹介させる研究 [西 13]，作業結果を学習データとしてスパムワーカーを排除する研究 [Halpin 12]，ワーカーのタスクに非依存な行動からワーカーの能力を予測する研究 [Kilian 12]，ワーカーのランキングを行うことで低品質ワーカー，スパムワーカーを排除する研究 [Raykar 11] ，データに対するラベリングを行なうタスクにおいて高品質ワーカーと低品質ワーカーを判定する閾値を算出することで，低品質なワーカー排除し最適なデータを得るためのを研究 [Donmez et al., 2009] などが行われている．既存のサービスにおいても，ワーカーに事前テストを受けさせてリクエスタが必要に応じてワーカーを選別する手法 [AMT] などが行われている．

(3) に関する研究はタスクを提供するリクエスタに関する研究である．不適切なタスクはワーカーのモチベーションを下げ，結果としてワーカーの品質低下につながる．この不正リクエスタを排除することで全体の精度を保つ研究 [馬場 13] などが行われている．

また，(1) と (2) の組み合わせである，事前に事前テストを受けさせてワーカーを選別し，さらに出題方法の調

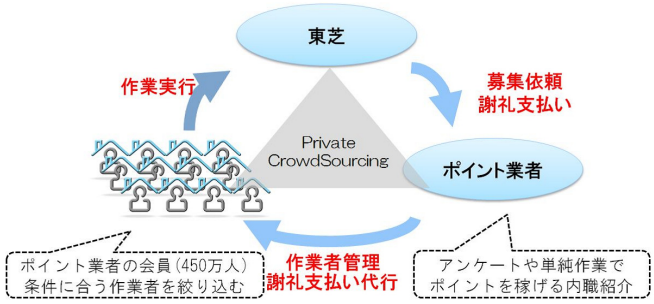

図 1 ポイント業者を経由したクラウドソーシングシステムの構築

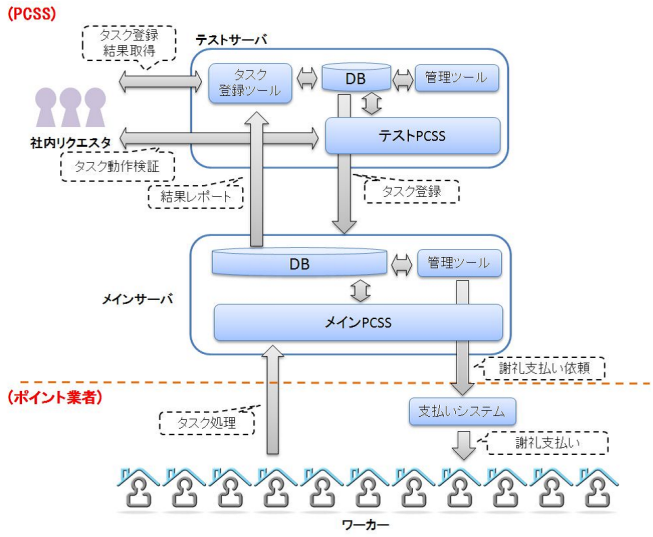

図 2 PCSS のシステム構成

整でワーカーを選別する研究 [Kazai 11] なども行われている．しかしこの研究では特定のタスクを対象としたリクエスタ視点で行われている研究であり，複数の種類のタスクが発生した場合は対応が難しいという問題がある．

PCSS では主に (2) のワーカーに対する精度向上手法を中心に行っている．(1) に関してはシステム外の精度向上手法に関する事項であるため，タスク内容に依存することが多くシステム側で対応しにくいという問題がある．実際に PCSS を運用するにあたってはリクエスタのタスクの内容に応じて対策を行っているが，PCSS における機能とは異なるため本稿では触れない．また，(3) に関してはプライベートなクラウドソーシングという特性上リクエスタが明確であるため，不正なリクエスタは存在せず対策は不要である．

3. PCSS の構築

研究データの構築には大量のタスクを高速に処理しなければならず，そのために，我々は前章で述べたマイクロタスク型のクラウドソーシングを用いている．しかし，既存のマイクロタスク型のクラウドソーシングサービスを研究データ構築に利用するには精度の点に問題がある．

我々はタスクの処理結果を研究データとして用いるため作業結果の品質を高く維持しなくてはならないという点があり，そのためには外部のサービスが提供している精度向上のための機能の範囲では十分ではないことが多い．また外部のサービスに精度向上のための新規機能を追加することも難しいという問題がある．

そこで，これらの問題を解決するために，プライベートな環境下において様々な精度向上手法を適用したマイクロタスク型のクラウドソーシングシステムを構築した．我々はこのクラウドソーシングシステムをプライベートクラウドソーシングシステム（PCSS）と呼称している．本章では PCSS の構築方法に関して述べる．

3·1 システムの構築

本節ではプライベート環境下におけるクラウドソーシングの構築方法に関して述べる．クラウドソーシングは不特定多数の人によって動作するシステムであり，システムを構築しただけでは動作しない．システムに対してタスクを提供するリクエスタと，タスクを処理するワーカーが必要となる．システムは両者の仲介を行い，様々な面でサポートを行うことで全体的な効率の向上を図っている．

プライベートなクラウドソーシングを構築するにあたって一番の問題はワーカーの募集である．Amazon Mechanical Turk のように既に周知のサービスであればワーカーの募集は容易だが，無名の状態から必要な人数を集めるには多大なコストがかかる．一方，Amazon Mechanical Turk のように誰でも作業ができる環境ではワーカーの質を管理するコストが大きく，タスク結果の質が低下してしまうという問題もある．PCSS では，ワーカーの募集をネットワークリサーチを行なっているポイント業者へと委託した．ポイント業者は既にリサーチ対象となるユーザを数百万規模で管理しており，これらのユーザを PCSS のワーカー候補とした．それらのワーカー候補に対して「作業可能な時間」「熱意」「希望時給」「学歴」「基本的な IT スキル」などのアンケートを実施し，各項目の能力が高いワーカー候補に対してプライベートクラウドソーシングへの案内を送付した．対象となったワーカー候補者の合計は 8 万人であり，これは PCSS におけるタスクの処理量が増えるに応じて募集を数回にわたって行った結果である．我々はこの絞り込みを「事前フィルタリング」と呼称している．これにより我々はポイント業者のユーザをワーカーとして作業を提供し，Web 経由で作業可能とし，さらにポイント業者を経由してワーカーに報酬を支払うという図 1 の構成を構築している．

システムは Perl で構築された CGI と，MySQL を用いたデータベースのサーバから構成されており図 2 のような構成となっている．リクエスタは Web インターフェイス経由でタスクをデータベースに登録し，ワーカーは

データベースに登録されたタスクに対して Web インターフェイス経由でタスク処理を行い，結果をデータベースに登録する．リクエスタはタスク処理が完了次第，結果をデータベースから取得する．

以上のように外部のサービスを利用することなく独自の環境下においてタスクを出題しワーカーに作業をしてもらうプライベートなクラウドソーシング環境を構築することが出来た．本システムは 2011 年 11 月から運用を継続しており，表 1 に示す運用実績を持っている．

表 1 PCSS の運用実績

運用開始	2011 年 11 月
ワーカー総数	1568 人
毎月実績のあるアクティブなワーカー	150 人
問題数	570 万件

3·2 ワーカー管理による精度向上手法

PCSS の環境を構築しただけでは得られるタスクの処理結果の精度は低いため，研究データとして使用するには十分ではない．本節では PCSS におけるタスク処理結果の精度向上手法に関して述べる．

PCSS における精度向上手法は主にワーカーに対する管理を中心に行っている．クラウドソーシングは「不特定多数の外部の人間」に作業を委託する仕組みであるため，ワーカーの品質は様々である．特定のカテゴリのタスクにおける正解率が高い高品質ワーカーや正解率が低い低品質ワーカー，リクエスタの出題意図に沿った回答ができるスキル保持ワーカーや意図に反した回答をする負スキル保持ワーカー，全体の正解率が低いまたはスクリプトなどで処理を行う，システムから排除対象となるスパムワーカーなどのワーカーが存在する．既存のクラウドソーシングサービスでは数多くのリクエスタから数多くのタスクを受け入れているため，ワーカーが行うタスクは多種多様となり，結果としてタスク単位におけるワーカーの行動情報が少なくなり，ワーカーのコントロールが難しくなっている．PCSS ではプライベートという特徴上タスクのカテゴリが限られているため，タスクカテゴリに対するワーカーの行動情報は相対的に多くなっており，そのワーカーの行動情報を活かすことでワーカーの特性に応じた適切なタスクを与え，低品質ワーカーおよびスパムワーカーを排除することを可能としている．

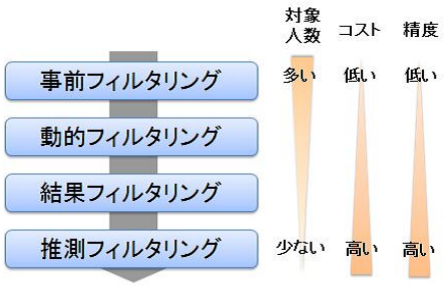

図 3 PCSS におけるワーカーに対する精度向上手法

以下にワーカーに対する PCSS の精度向上手法を (1) 事前フィルタリング，(2) 動的フィルタリング，(3) 結果フィルタリング，(4) 推測フィルタリング，の 4 つのカテゴリに分類した．それぞれのフィルタリングではコストと精度が異なり，コストが高いフィルタリングは低品質ワーカーを排除する精度が高い．我々はコストの問題から，対象のワーカーの数に応じてフィルタリングを適用している (図 3)．それぞれのフィルタリングに関して詳細を述べる．

§1 事前フィルタリング

ポイント業者からワーカーを募集する際に行うフィルタリングである．ポイント業者は数百万人の会員を有しており，これらのすべての会員をワーカーとして扱うのはコスト的に現実的ではなく処理能力的にも過剰である．また，これらの会員には ICT の素養が低い，Web における継続的な作業を望んでいない，などの PCSS に不適である会員も多く存在しており，このような明らかに高品質なワーカーになりえないワーカー候補を排除するために事前のアンケートを用いてフィルタリングを実施している．アンケート内容は「作業可能な時間」「熱意」「希望時給」「学歴」「基本的な IT スキル」などの基本的な設問に加えて，ワーカー募集の目的に応じた設問を追加して実施している．例として文法に関する技術を有するワーカーを募集したい場合は文法に関する設問を追記するなどの対応ができる．

§2 動的フィルタリング

ワーカーがタスク処理をしている際に行うフィルタリングである．(1) 事前フィルタリングにて最低限の品質を確保できたワーカーであるが，すべての低品質なワーカーを排除できたわけではない．また，人間は時間の経過に応じて能力が上下するため，初期の品質判定が継続するとは限らない．そのため，タスク処理を進めていく課程で動的にワーカーのフィルタリングを行うために正解率と経験値という 2 点の項目を設けている．

正解率は「正解数/総作業数」で算出し，一定値以下のワーカーはスパムワーカーとみなし，以降の PCSS におけるタスク処理を禁止する．また同様に,「正解数 - 不正解数」で算出される経験値を設定し，一定の経験値を持つワーカーに対して高報酬，高難易度のタスクを提供している．難易度の基準はリクエスタによって異なるが，多数決による正解判定を行なうタスクで結果が分散してしまう，タスクの完了まで時間がかかるなどのケースでは難易度が高いと判定される場合が多い．これらの数値は図 4 のように作業中に画面に常に表示している．正解率が一定値以下になることでタスク処理ができなくなるこ

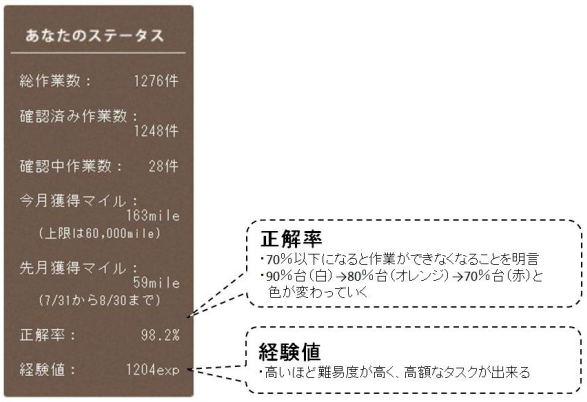

図 4 ワーカーに表示されるステータス画面

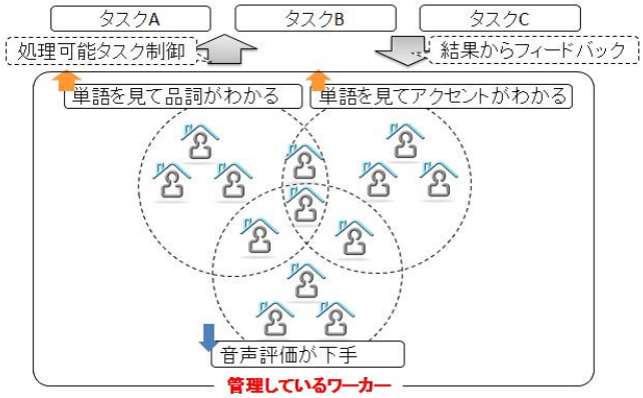

図 6 タスク処理結果を用いたワーカーの特徴付け

ワーカーID	読み付け			読み仮名判定			品詞判定		
	正解率	正解位数	不正解数	正解率	正解位数	不正解数	正解率	正解位数	不正解数
101	92.6	14368	1147	94.6	4295	244	88.6	474	61
102	96.2	53463	2066	97.1	23028	695	83.6	504	99
103	97.1	1455	43	98.3	5385	94	100.0	10	0
104	94.5	10247	597	95.8	28824	1250	83.9	3406	654
105	91.9	452	40	0.0	0	0	0.0	0	0
106	98.0	16010	329	99.3	11558	82	93.2	68	5
107	95.2	64775	3240	94.7	42631	2398	88.7	11915	1517
108	97.0	44965	1375	98.2	39815	741	89.4	1831	216
109	97.4	69290	1863	97.9	25541	543	84.4	1094	202
110	96.4	65581	2462	97.0	29294	903	90.5	7435	780
111	94.6	2164	124	95.7	44	2	100.0	13	0
112	90.4	64183	6792	95.0	52971	2814	82.4	17895	3833
113	95.9	77179	3313	96.1	94042	3841	89.8	13512	1532
114	95.5	121979	5746	95.9	85658	3629	47.1	3985	4483
115	90.0	98895	11040	97.0	114462	3533	92.7	11622	912

図 5 タスク別ワーカー結果精度 (一部)

とはワーカーに明言してあり，ワーカーはこの数値表示によって精度に対する注意を喚起される．これらの数値は作業者の行動によって変化するという点でゲームメカニクスにおける得点制度と同等に考えることができる．得点制度はゲームメカニクスとしては一般的であり，利用者のモチベーションを向上させるための手法として用いられている [Ahn 2008]．一方，これらの数値を算出するためには正解率が必要であり，ワーカーによって入力された結果の合否判定を行わなければならない．合否判定に用いる手段としては多数決を用いる手法が提案されている [Snow 08]．我々も主に多数決にて正解を決定しており，アンケートなど正解がない場合にはタスク説明に正解が無い旨を明記し，正解率は変動させない．

現時点ではインターフェイス上の制限から，ワーカーが確認することが出来るのはすべてのタスクの全体平均正解率のみである．しかし，動的フィルタリングをこの全体平均正解率のみで行うとフィルタリング効果が低いことがわかっている．我々は図 5 のようにカテゴリごとに正解率をワーカーに明示せず別途管理している．現在このカテゴリは 13 種類存在し，リクエスタからタスクの公開申請を受けた時にシステム管理者が手動で分類を行っている．本稿ではこの 13 カテゴリのうち 4 章で事例として紹介した自然言語処理に関する 4 カテゴリ「単語判定カテゴリ」「読み付けカテゴリ」「品詞カテゴリ」「アクセントカテゴリ」に関して述べる．図 5 を見ると正解率が低いカテゴリの作業総数が非常に少なく，正解率が高いカテゴリの作業総数が多いケースが散見される．これにより正解率が下がるような難易度の高いタスクをワーカーが避ける傾向があることがわかる．しかし全体正解率のみで判断を行った場合,「賃金が高く難易度も高いタスク A」と「賃金が低く難易度も低いタスク B」があった場合，ワーカーはタスク A を処理し，全体平均正解率が下がるとタスク B を行なって全体平均正解率を回復させるという行動をとることがあった．これは該当するカテゴリの正解率が低いにもかかわらず大量に作業を処理しており，かつ全体正解率は高いというワーカーの存在から判明した．これらのワーカーを低品質ワーカーと呼称し表 6 に数と割合 (低品質ワーカー数/ワーカー数) を示す．このような低品質ワーカーの行動に対応するため特定カテゴリにおける作業総数が 50 以上になったワーカーをアクティブワーカーとし，特定のカテゴリにおけるアクティブワーカーの精度が一定値以下になった場合は，そのカテゴリに属するタスクを隠し，処理をさせないようにすることでワーカーの行動コントロールを行っている．

§3 結果フィルタリング

(2) 動的フィルタリングは正解を判定することが出来る作業に対してのみ有効であり，アンケートや文章作成のような明確な正解がなく，多数決も実施しにくいタスクにおいては適用できない，またワーカーが低品質ワーカーであると判明し，出題停止に至るまでに多くの低品質なデータが算出されてしまうという欠点がある．我々はこの問題に対し，図 6 のように，ワーカーのタスク処理結果からワーカーの特徴を判別する結果フィルタリングを用いて対応している．

明確な正解がないタスクでも，リクエスタの意図に沿った内容か否かという判定は存在しており，この判定をリクエスタにタスク毎に行わせるには大きなコストがかかる．このようなタスクに関して，リクエスタは他のリクエスタの類似したタスクの結果や，小規模のテスト用タスクを実施した結果などから，正解率の高いワーカーや出題意図に沿った回答をしているワーカーを選別し，以

降のタスクは条件に該当するワーカーのみに出題することで結果精度を向上させることができる．この選別基準はシステム側で明確に定めておらず，リクエスタによって異なる．これらのワーカーの情報で優秀なワーカーを判別するための情報を「スキル」，低品質なワーカーを判別するための情報を「負スキル」と呼称している．「負スキル」はカテゴリごとに作成可能であり，「負スキル」保持ワーカーは該当するカテゴリのタスク以外は作業できるため，全ての作業が不可能となっているスパムワーカーとは異なる．例えば「品詞」のカテゴリのタスクの正解率が高いワーカーには「品詞」のスキルを付与し，「品詞」のタスクは「品詞」スキルを持つワーカーにのみ出題することで精度向上を行っている．これらのスキルはリクエスタ間で共有して使用することが出来るため，新規のリクエスタも初回からスキルを保持するワーカーにタスクを処理させることが可能である．

§4 推測フィルタリング

(2) 動的フィルタリングや (3) 結果フィルタリングは何らかのタスクの処理結果をワーカーの行動コントロールに流用したものであり，ワーカーがスパムワーカー，低品質ワーカーであった場合はワーカーの行動コントロールが出来る段階に達した時点で低品質な処理結果を残してしまっている事が多い．これらのデータは再処理が必要であり，大量のワーカーによって短時間で大量のタスク処理が行われるマイクロタスク型のクラウドソーシングでは時間，賃金ともに再処理のコストが大きくなってしまう．そこで，我々は更に低品質なタスク処理結果を削減するために，ワーカーの特性から行動を推測し，事前にタスクに不適切なワーカーをフィルタリングすることで精度向上を試みている．ワーカーに対するタスクの割り当てに関する研究として様々な研究がなされている．タスクの内容やワーカーのタスクに対する完遂率をベースにタスクの推薦を行なう研究 [Ambati et al., 2011] では低品質ワーカーに対する対応が取られておらずタスク推薦の効果があらわれるまでに多くの低品質データが発生してしまう問題がある．我々は推測フィルタリングに至るまでの複数のフィルタリングで低品質ワーカーを可能な限り少なくすることで，低品質データの発生を最低限におさえている．またワーカーの行動履歴，ワーカーのタスクに対する嗜好からワーカーにタスクの推薦を行なう研究 [Yuen et al., 2012] でも対象となるワーカーが膨大になった場合のコストが大きいという問題がある．我々は前述の様に推測フィルタリングに至るまでの複数のフィルタリングで対象となるワーカーの数を削減し，必要なコストを最低限に抑えている．また，タスクの難易度レベル，ワーカーのスキルのレベルを推測した結果からワーカーにタスクの推薦を行う研究 [Ho et al., 2013] でも対象となるタスクのカテゴリが限られているという問題がある．我々は複数のカテゴリを管理し，タスクをカテゴリに分類することで複数のタスクカテゴリを対象とすることを可能としている．

ワーカーID

ワーカーID	101	102	103	104	105
101	1	0.43	−0.4	0.13	0.59
102	0.43	1	−0.07	0.76	0.58
103	−0.4	−0.07	1	−0.38	0.79
104	0.13	0.76	−0.38	1	0.51
105	0.59	0.58	0.79	0.51	1
106	0.31	0.92	0.24	0.62	0.51
107	−0.27	0.77	−0.54	0.86	−0.1
108	0.36	0.93	−0.26	0.68	0.11
109	0.73	0.86	−0.36	0.97	0.18
110	0.69	0.93	−0.23	0.82	0.59
111	−0.61	0.1	0.61	−0.39	0.24
112	0.18	0.56	0.21	0.38	0.23
113	0.1	0.46	0.16	0.04	−0.04
114	0.79	0.82	−0.44	0.97	0.49
115	0.11	0.07	0.67	0.29	0.78

図 7 ワーカー間類似度 (一部)

そのために我々はワーカーの類似性を利用した協調フィルタリングを用いて，ワーカーが未作業のカテゴリのタスクの結果精度の推測を行い，精度が低いと推測されるカテゴリのタスクは最初から処理させないという方法を用いている．協調フィルタリングとは多くのユーザの嗜好情報を蓄積し，あるユーザと嗜好の類似した他のユーザの情報を用いて自動的に推論を行う方法である．我々はユーザの嗜好情報の代わりにワーカーを特徴づける情報として，タスクのカテゴリ毎の結果精度を用いている．ワーカーをカテゴリ毎の結果精度のパターンで比較し，類似したワーカーの情報を用いて，未作業のカテゴリのタスクの結果精度の推測を行う．

具体的な例として表 2 のようなケースでは，次のようなパターンが考えられる．

(1) ワーカー a とワーカー b の類似度が高いのでワーカー a のタスク B に対する結果精度からワーカー b が未作業のタスク B の結果精度を推測する．ワーカー a のタスク B に対する結果精度が高いのでワーカー b のタスク B の結果精度も高く推測されるため，ワーカー b にはタスク B を積極的に勧める．

(2) また，ワーカー a とワーカー c の類似度も高いのでワーカー a のタスク E に対する結果精度からワーカー c が未作業のタスク E の結果精度を推測する．ワーカー a のタスク E に対する結果精度が低いのでワーカー c のタスク E の結果精度も低く推測されるため，ワーカー c にはタスク E を作業させない．

実際に我々が推測フィルタリングを行なうにあたって，必要なワーカーの類似度を計算するためにピアソン相関係数を用いている．ピアソン相関係数は協調フィルタリングにて類似度を判定する際に用いられることの多い値である．全ワーカーの集合を W，その要素を u, v，全タスクカテゴリの集合 T，その要素を i, j とする．この時あるワーカー u のタスクカテゴリ i における結果精度を $r_{u,i}$，ワーカー u の結果精度の平均を $\bar{r_u}$ とした場合，ワーカー u とワーカー v の類似度 $S_{u,v}$ は式 1 のようになる．

表 2 ワーカーに対する協調フィルタリングのデータ例 (「-」部分は未作業)

	タスク A	タスク B	タスク C	タスク D	タスク E
ワーカー a	98%	95%	99%	-	50%
ワーカー b	99%	-	97%	-	60%
ワーカー c	98%	99%	90%	-	-

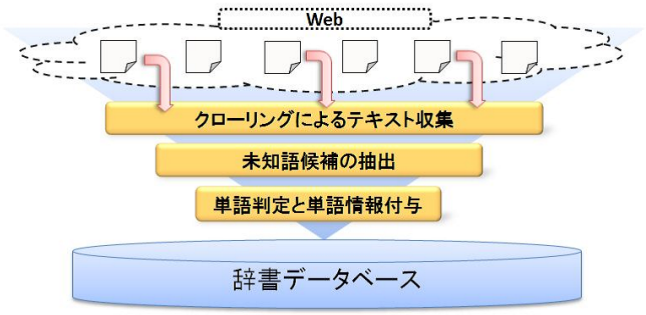

図 8 語彙抽出フロー

$$S_{u,v} = \frac{\sum_{i \in T}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{u \in W}(r_{u,i} - \bar{r_u})^2}\sqrt{\sum_{v \in W}(r_{v,i} - \bar{r_v})^2}} \quad (1)$$

式 1 を用いて各ワーカーの類似度を計算した結果は図 7 のようになった．この結果よりワーカー間の類似度は一定ではなく，類似しているワーカーと類似していないワーカーが存在することがわかる．得られたワーカー間の類似度を元に，ワーカー u がまだ作業していないタスク i における予測タスク結果精度 $P_{u,i}$ は式 2 のように計算することができる．

$$P_{u,i} = \bar{r_u} + \frac{\sum_{v \in W}(r_{v,i} - \bar{r_v})S_{u,v}}{\sum_{v \in W} \mid S_{u,v} \mid} \quad (2)$$

このようにして得られた予測タスク結果精度を元に，リクエスタによってタスクが出題されたタイミングでカテゴリの判定，カテゴリに応じた推測フィルタリングを実行する．その結果に基づきワーカーが得意と予想されるタスクをワーカーに優先的に提示し，不得意と予想されるタスクをワーカーに表示しないという方法で結果精度の向上を試みている．

推測フィルタリングにて協調フィルタリングを用いるに当たって，全員の正解率が高いタスクで発生した低品質ワーカーを推測できないという問題がある．この問題に対して我々は動的フィルタリング，結果フィルタリングで対応を行っている．

4. PCSS を用いた語彙収集の事例

PCSS を用いて知識処理研究に必要な語彙を収集した事例について述べる．収集のフローを図 8 に示す．まず始めに，Web クローラを用いた大規模テキストの収集を行い，続いて収集したテキストから未知語の候補を自動抽出する．そして最後に，PCSS を用いて未知語候補から単語として適当なものだけを絞り込み，知識処理研究の一環である音声認識や音声合成の辞書を構築するために必要な品詞や読み仮名，アクセントの単語情報を付与する．

4·1 クローリングによるテキスト収集

Web テキストには，固有名詞や新語などの未知語が頻繁に出現する．こうした未知語を獲得するコーパスとして，Web テキストを収集する．本稿では，OpenDirectory[*2]の URL をシードとして，Apache Nutch[*3]を用いて収集した．

獲得したテキストの情報を表 3 に示す．5.2 億ページから日本語文 125 億文を得ることが出来た．

4·2 未知語候補の抽出

4·1 節で収集したテキストから未知語の候補を抽出する．抽出処理は以下のステップで行った．

(1) テキストに対して点予測手法 [森 11] による単語分割を実施
(2) 単語分割結果から辞書未登録文字列を取得
(3) 単語分割結果を用いて単語 Ngram を作成
(4) 単語 Ngram を用いて辞書未登録文字列の中から未知語候補を選出

この一連の処理によって 125 億文のテキストから 23 万語の未知語候補を抽出することができた．

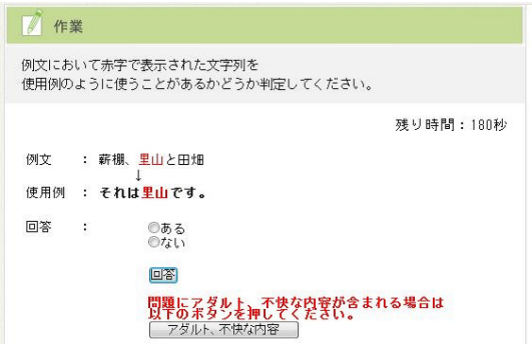

図 9 単語判定タスク

*2 http://www.dmoz.org/World/Japanese/
*3 http://nutch.apache.org/

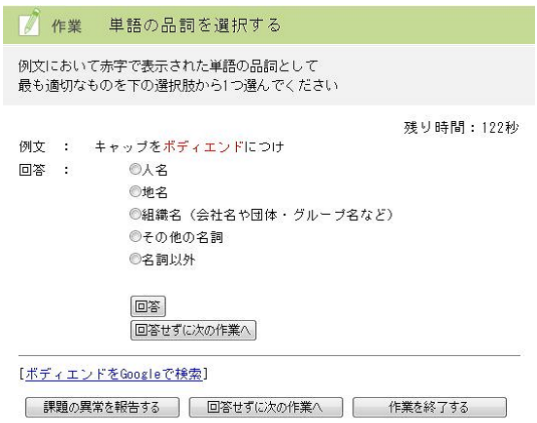

図 10 品詞付与タスク

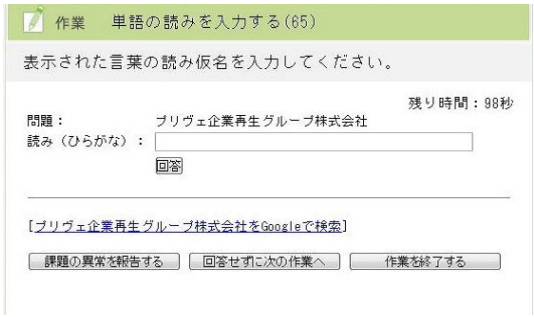

図 11 読み付与タスク

4·3 単語判定と単語情報付与

4·2 節の方法で作成された未知語候補には，単語として適当でないものが残っている可能性が高い．また，抽出した単語に対して音声処理に必要な情報を付与しなくてはならない．これらの情報収集を PCSS の以下の 4 タスクとして行った．

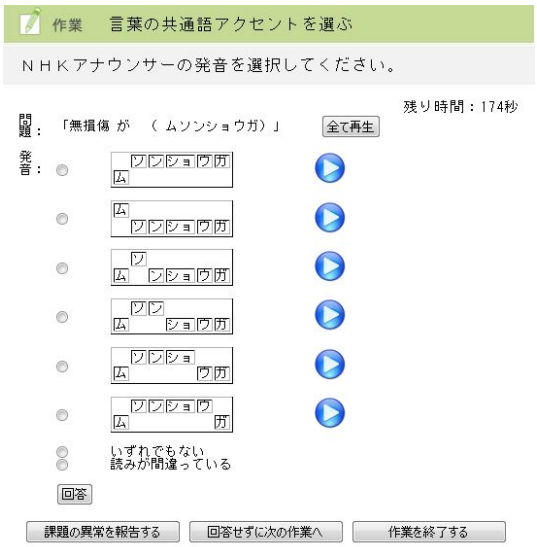

図 12 アクセント付与タスク

(1) 単語判定タスク

タスクデザインを図 9 に示す．このタスクではワーカーに対して 4·2 節の方法で作成された未知語候補を「それは (未知語候補) です」という問題文に加工して表示し,「問題文は日本語して自然か否か」という選択をさせた.「日本語として自然である」と回答された場合，その文章に含まれる未知語候補を未知語として扱う．

(2) 品詞付与タスク

タスクデザインを図 10 に示す．このタスクでは名詞とそれ以外の品詞に分ける作業を行なっている．名詞に関しては「人名」「地名」「組織名」「その他の名詞」に再分類している．(1) で単語として適切であると判定された未知語に単語抽出元の前後の文章を付与して問題文に加工して表示し,「人名」「地名」「組織名」「その他の名詞」「名詞以外」を選択させた．

(3) 読み付与タスク

タスクデザインを図 11 に示す．このタスクでは (2) で名詞と判定された未知語を問題として表示し，その読みを入力させ，その結果を未知語に対する読みと判定した．

(4) アクセント付与タスク

タスクデザインを図 12 に示す．このタスクでは (3) で付けられた読みから推定されるアクセント候補から合成した音声を用い，どれが自然かを選択させた．その結果を未知語に対するアクセントと判定した．

各タスクは 3 人に出題され，2 人以上一致した回答を有効なデータとして扱う．ただし，(1) の単語判定タスクは高精度であることを求められるため，3 人が一致した回答のみを有効なデータとして扱った．また，ワーカーが設問が不適切であると判断した場合は「パス」を選択できるようにしている．通常のパスであれば回答権は他のワーカーに移動するが，6 回以上パスが行われた場合はその問題は不適切と判定されて排除される．PCSS ではリクエスタからの中断依頼が無い限り，出題した全ての問題に対して回答かパスの処理が行われるまで出題される．各カテゴリにおけるタスク処理結果から無作為に 10000 件の結果データを抽出し，一致率を調査した結果を表 4 に示す．

4·4 得られた語彙数

以上の処理を用いて Web から得られた語彙数を表 5 に示す．125 億文の Web テキストから 14 万語の語彙を獲得することが出来た．獲得できた未知語の例としては「Siri」「あっちゃん」「先っちょ」「スンゲー」「ドm」「花立山」「えらそう」「やべえええええ」などが挙げられる．

4·5 PCSS の精度向上手法による精度改善効果

4·3 節で述べた各タスクの処理結果は 3·2 節で紹介した精度向上手法を用いることで精度が大きく向上してい

表 3 獲得した Web テキスト

獲得ページ数	517,239,154
日本語ページ数	319,570,805
文数	12,504,868,218

表 4 各タスクの作業結果における一致率

	3 人一致	2 人一致	不一致	不適切
単語判定カテゴリ (2 択)	71.1%	28.9%	0.0%	0.0%
読み付けカテゴリ	75.6%	16.7%	2.4%	5.3%
品詞カテゴリ	84.3%	2.4%	13.2%	0.1%
アクセントカテゴリ	66.0%	28.5%	3.9%	1.6%

表 5 未知語獲得数

未知語候補抽出数	227,367
未知語獲得数	138,546

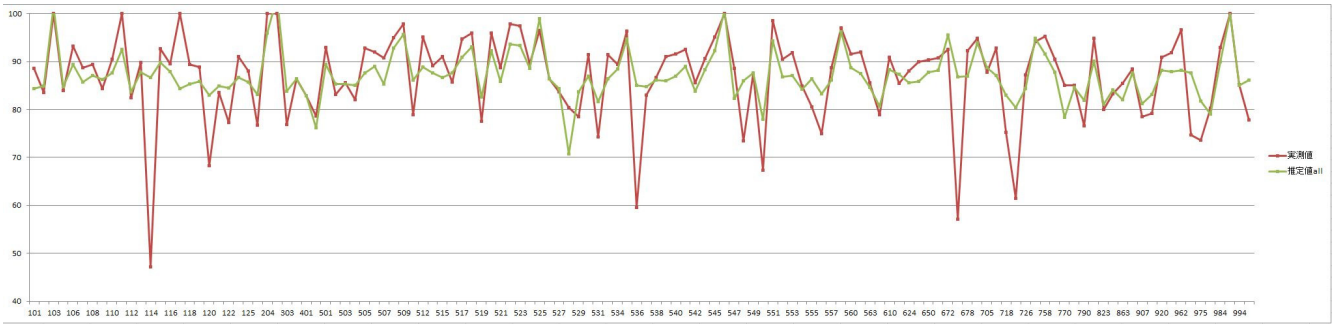

図 13 実測タスク結果精度 $M_{u,i}$ と予測タスク結果精度 $P_{u,i}$ の比較

る．本節では語彙収集における精度向上効果に関して述べる．

前述のように現在 PCSS では各タスクを 13 個のカテゴリに分類して管理を行っている．それらのうち本節では語彙収集の各タスクと対応している「単語判定」「読み付け」「品詞判定」「アクセント」の 4 つのカテゴリに関して述べる．各カテゴリにおける問題数，1 問でも回答したことがあるワーカー数，50 問以上回答したことがあるワーカーをアクティブワーカーと定義した時のアクティブワーカー数を表 6 に示す．

語彙収集における各フィルタリングによる精度改善効果は以下のようになった

§1 事前フィルタリング

ポイント業者の保持する会員のうち，8 万人に対して事前フィルタリングとなるアンケートを行った．アンケートの項目は「現在仕事を行っている場合は週にどれくらい仕事を行っているか」「副業にどれくらいの時間を使えるか」「最低欲しい時給はいくらか」「内職に興味はあるか」「学歴」「好きな科目」「PC 環境の有無」である．このアンケートの結果からワーカー候補として 2457 人に絞込み，クラウドソーシングへの勧誘を行った．最終的に勧誘に応じてクラウドソーシングの作業を一度でも行ったワーカーは 1630 人 (そのうち 62 人はスパムワーカーとして排除)，実際に毎月実績のあるアクティブなワーカーは 150 人前後となっている．

§2 動的フィルタリング

PCSS では動的フィルタリングにおいて排除すべき低品質ワーカーと判定する閾値を全体正解率 70%以下としている．これは運用開始時に閾値を 80%と仮設定した際に，不慣れなワーカーがすぐにスパムワーカー扱いされてしまいクレームが発生したためである．我々は結果精度を高く維持したい一方で，大量のタスクを高速に処理しなくてはならない．そのためにはワーカーをすぐに切り捨てるのは得策ではないと判断し，逐次閾値を下げていき，動的フィルタリング以外の精度向上手法を導入することで対応した．この全体正解率を用いた動的フィルタリングによって 1630 人のワーカーから 62 人のワーカーを低品質，スパムワーカーとして排除することができている．また各カテゴリにおける低品質ワーカーの数と割合 (低品質ワーカー数/ワーカー数) を表 6 に示す．これらの特定カテゴリにおける低品質ワーカーは該当するカテゴリに属するタスクが選択可能なタスク一覧から隠され作業ができなくなる．

§3 結果フィルタリング

結果フィルタリングを用いることによって得られたスキルを表 7 に示す．単語判定，読み付けに関しては動的フィルタリングで排除されたワーカー以外のワーカーには大きな能力の差異は見られなかったため，スキル保持

表 6 各カテゴリにおける値

	作業数	ワーカー数	アクティブなワーカー数	低品質ワーカーの数 (割合)
単語判定カテゴリ	1,652,271	454	353	50(11.0%)
読み付けカテゴリ	3,185,708	576	380	32(5.6%)
品詞カテゴリ	589,949	129	107	6(4.7%)
アクセントカテゴリ	1,270,618	358	276	33(9.2%)

表 7 結果フィルタリングによって「スキル保持」「負スキル保持」と判定されたワーカ数

スキル名	対象タスクカテゴリ	ワーカー数
単語を見て品詞がわかる	品詞カテゴリ	31
発音の正誤判定ができる	アクセントカテゴリ	207
複数候補から正しい発音を選択できる	アクセントカテゴリ	142
単語を見て発音を記述できる	アクセントカテゴリ	53
音声を評価するにあたって問題がある	アクセントカテゴリ	242

表 8 実測タスク精度と予測タスク精度の比較

	推測値誤差	推測高精度ワーカー数	推測高精度ワーカー正解数
単語判定カテゴリ	4.44	183	163
読み付けカテゴリ	3.69	219	194
品詞カテゴリ	4.45	23	23
アクセントカテゴリ	4.27	138	121

ワーカーの絞り込みは実施していない．動的フィルタリングと同様に，これらのスキルを元にタスク処理の可否を決定している．

品詞カテゴリタスクの作業結果を解析して作業結果が高品質であったワーカーには「単語を見て品詞がわかる」スキルが与えられる．リクエスタは難易度が高く精度を優先する品詞タスクを出題するときは「単語を見て品詞がわかる」スキル保持ワーカーのみに作業を出題して精度向上を行なう．

アクセントカテゴリに関しては複数の難易度の段階があり，リクエスタはその段階ごとにタスク化を行っている．さらにリクエスタは各難易度ごとにスキルを作成し，小規模タスクを行い，結果を人手でチェックして一定以上の正解率を持つワーカーに対してスキルを付与する．表 7 のアクセントカテゴリにおけるスキルは難易度が高い順で「単語を見て発音を記述できる」「複数候補から正しい発音を選択できる」「発音の正誤判定ができる」となっており，リクエスタは難易度が高いスキル（例：単語を見て発音を記述できる）を持つワーカーには難易度が低いスキル（例：複数候補から正しい発音を選択できる，発音の正誤判定ができる）を同時に付与する．つまり難易度が低いアクセント作業には高スキルワーカーから低スキル保持ワーカーまで全てに作業を行わせて処理速度を向上させ，難易度が高いアクセント作業には高スキル保持ワーカーのみに作業を行わせて処理速度を犠牲に精度を向上させる．また「音声を評価するにあたって問題がある」スキルは負スキルであり，一番難易度の低いタスクにおいて結果品質が低いワーカーに付与されるスキルである．このスキルが付与されたワーカーにはアクセントカテゴリに属するタスクの処理をさせないことで精度向上を行なう．

§4 推測フィルタリング

実際に推測フィルタリングを行うにあたって，式 2 で得られた予測タスク結果精度 $P_{u,i}$ の精度を確かめるために，今までの PCSS の運用データを用いて実験を行った．対象となったのは 2013 年 11 月時点で正解判定がある何らかのタスクを実施した経験のあるワーカー 792 人である．各ワーカーの結果精度をカテゴリ毎に集計し (図 5)，その集計結果を元にピアソン相関係数を用いてワーカーの類似度を計算した．図 5 で既に実際の解答履歴から算出されているタスク i におけるワーカー u の実測タスク結果精度 $M_{u,i}$ と，他のワーカーとの類似度から推測した予測タスク結果精度 $P_{u,i}$ を比較検証した．「品詞カテゴリ」を例に用いた場合，得られた実測タスク結果精度 $M_{u,i}$ と予測タスク結果精度 $P_{u,i}$ の比較は図 13 のような結果となる．各カテゴリにおける実測タスク結果精度と予測タスク結果精度の値の差の平均，予測タスク結果精度が 90%以上のワーカーを推測高精度ワーカーと呼称し，その人数，推測高精度ワーカーの実測タスク結果精度を調査し，実際に結果精度が 90%以上であるワーカーの数を推測高精度ワーカー正解数と呼称し，その人数を表 8 にしめす．

効果を確認するために，各カテゴリに対して精度向上適用前と適用後それぞれのタスクの処理結果から無作為に各カテゴリごとに 1000 件のデータを抽出し，人手によって合否を確認することで精度を計測した．対象とな

表 9 各カテゴリにおける精度向上効果

	精度向上適用前正解率	精度向上適用後正解率
単語判定カテゴリ	65.9%	89.6%
読み付けカテゴリ	56.3%	94.0%
品詞カテゴリ	71.0%	90.4%
アクセントカテゴリ	54.1%	98.7%

表 10 各カテゴリにおけるワーカー数

	精度向上手法適用前			
	高精度ワーカー数		高精度ワーカー以外のワーカー数	
	アクティブ	非アクティブ	アクティブ	非アクティブ
単語判定カテゴリ	15	0	8	0
読み付けカテゴリ	22	0	1	1
品詞カテゴリ	12	0	16	0
アクセントカテゴリ	6	0	1	0
	精度向上手法適用後			
	高精度ワーカー数		高精度ワーカー以外のワーカー数	
	アクティブ	非アクティブ	アクティブ	非アクティブ
単語判定カテゴリ	33	1	17	9
読み付けカテゴリ	51	4	8	5
品詞カテゴリ	12	0	0	0
アクセントカテゴリ	8	0	0	0

るデータは実務上の測定であるため同一の問題ではないが，同一条件で行った Web クローリングによって取得した 125 億文の Web テキストデータに対して，同一の辞書で形態素解析を行い，得られた未知語候補 22 万語を単語判定，読み付け，品詞付け，アクセント付けの各カテゴリにおけるタスクで処理した結果のデータである．結果を 表 9 に示す．このように複数の精度向上手法により，実際に研究データに利用可能なデータの取得効率が向上していることがわかる．また各カテゴリにおける精度向上適用前，精度向上適用後の各 5000 件のデータに対して「作業を行った高精度アクティブワーカー数，高精度非アクティブワーカー数」「作業を行った高精度ワーカー以外のアクティブワーカー数，高精度ワーカー以外の非アクティブワーカー数」を表 10 に示した．用いたデータは表 9 で用いたデータと同一条件で抽出した．単語判定カテゴリや読みつけカテゴリに関しては難易度が低く，リクエスタから処理速度が優先とされているため結果フィルタリング，推測フィルタリングは用いていない．そのため精度向上手法適用前，精度向上手法適用後ともに高精度ワーカーと通常ワーカーが混在して作業を行っている．その後高精度ワーカー以外のワーカーのうち低品質ワーカー (結果精度 70 %以下) は動的フィルタリングで排除されるため，表 9 で示したように事前フィルタリング，動的フィルタリングで十分な精度向上効果を得ることができている．品詞カテゴリやアクセントカテゴリに関しては難易度が低く，リクエスタから精度が優先とされているため結果フィルタリング，推測フィルタリングを用いて精度改善を試みた．高精度ワーカー以外のワーカーは結果フィルタリング，推測フィルタリングで事前に排除されるため，精度向上手法適用後は作業をすることはない．また高精度ワーカーのみが処理を行なうため，ワーカー数が制限され，処理速度が低下するが，リクエスタから精度が優先とされているため，速度に関しては問題視されていない．

4·6 精度改善手法における考察

以上のように，我々は研究データの作成に PCSS を用いるにあたって，結果データの品質を重視している．初期の PCSS では事前フィルタリングと動的フィルタリングを用いて運用していたが，得られたタスク処理結果は研究データとして利用できるデータとして満足行くものではなかった．PCSS を運用していく過程でタスクのカテゴリ管理，結果データの解析などを導入することで，ワーカーには画一的なスパムワーカーだけではなく，特定のカテゴリが得意なワーカー，不得意なワーカーが存在することがわかってきた．これらのワーカーは自らの得意不得意を意識せず，報酬や興味に応じて作業を行なうため，結果として低品質な結果データの算出につながっている．しかしこれらのワーカーはスパムワーカーと異なり適当な回答や適当な入力を行なうという悪意のある行動は少なく，適切なコントロールを行なうことで得意分野を活かすことができると判断した．その結果，結果フィ

ルタリング，推測フィルタリングの導入に至り，現在は精度の高い結果データを得ることが可能となっている．このように PCSS では低品質なワーカーを排除するフィルタリングを中心に行っている．一方で低品質なワーカーが作業を継続することによってスキルが向上し，高品質ワーカーとなるケースも存在し，そのようなケースを有効に活用して精度を向上させる手法が必要となる．しかし本研究ではこのケースに関してまだ有効な手法を確立できておらず今後の課題である．

5. まとめと今後の課題

本研究ではマイクロタスク型のプライベートなクラウドソーシングシステムを構築し，システム内にて様々な精度向上手法を組み込むことで高精度なタスク処理結果をえた．ワーカーとなる権利を完全に不特定多数の人に開放するのではなく，事前フィルタリングでワーカー候補を絞込み，タスク処理過程による動的フィルタリング，結果フィルタリング，推測フィルタリングを繰り返すことで高精度なワーカーを維持し，研究データに利用可能な精度を持つタスク処理結果を得ることが出来た．

さらに，PCSS を用いた研究データの構築の実例として自然言語処理の研究に用いる未知語の収集を示した．Web からクローラを用いて日本語 125 億文の Web テキストを収集し，形態素解析で得られた未知語候補に対して PCSS を用いることで 14 万語の未知語を抽出することに成功した．

PCSS ではマイクロタスク型クラウドソーシングの特徴上，ワーカーやタスクの数が飛躍的に拡大することが予想されており，ワーカーの行動をコントロールするにあたって，それらの規模に対応した計算方法を想定しなければならない．また，今後クラウドソーシングを用いた就労形態が一般的になった際に，簡易にワーカーを排除することは効率的な面からも社会的な面からも問題がある．そのため，低品質ワーカーに対しては排除だけではなく低品質ワーカーを高品質ワーカーにするための手法を検討するなど新たな精度向上施策を検討していくことが今後の課題である．

謝　　辞

本稿は，下郡信宏氏，池田朋男氏，西山修氏，中田康太氏，有賀康顕氏，宮村祐一氏との議論を基礎に置いています．またこのような機会を与えて頂いた当社関係者に感謝します．

◇ 参　考　文　献 ◇

[Ahn 2008] Ahn, L., Dabbish, L., “Designing games with a purpose”, Communications of the ACM, pp. 58-67, (2008)
[AMT] Amazon Mechanical Turk, https://www.mturk.com/mturk/
[Ambati et al., 2011] Ambati, V. et al., “Towards task recommendation in micro-task marketss”, In proc. of HCOMP, (2011).
[馬場 13] 馬場 雪乃, 鹿島 久嗣, 木下 慶, 山口 豪志, 秋好 陽介, “機械学習による不適切なクラウドソーシングタスクの検出”, DEIM, (2013).
[クラウドワークス] クラウドワークス, http://crowdworks.jp/
[Dawid 79] Dawid, A.P., Skene,A.M.,“Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm”,Journal of the Royal Statistical Society, (1979).
[Donmez et al., 2009] Donmez, P. et al., “Efficiently learning the accuracy of labeling sources for selective sampling”, In proc. of KDD, 2009.
[Halpin 12] Halpin, H., Blanco, R.,“Machine-Learning for Spammer Detection in Crowd-Sourcing”, HCOMP, (2012)
[Ho et al., 2013] Ho, C. J., et al., “Adaptive Task Assignment for Crowdsourced Classification”, In proc. of ICML, (2013).
[Kazai 11] Kazai, G., Kamps,J., Koolen,M., Milic-Frayling, N., “Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking”, SIGIR, (2011).
[Kilian 12] Kilian, N., Krause, M., Runge, N., Smeddinck, J., “Predicting Crowd-Based Translation Quality with Language-Independent Feature Vectors”, HCOMP, (2012)
[Kittur 08] Kittur, A., Chi, E., Suh, B., “Crowdsourcing user studies with mechanical turk”, CHI, (2008).
[小山 13] 小山 聡, 馬場 雪乃, 櫻井 祐子, 鹿島 久嗣, “クラウドソーシングにおけるワーカーの確信度を用いた高精度なラベル統合”, 人工知能学会全国大会, (第 27 回), (2013).
[ランサーズ] ランサーズ, http://www.lancers.jp/
[Mao 12] Mao, A., Procaccia, A., Chen, Y., “Social Choice for Human Computation”, HCOMP, (2012).
[松原 13] 松原繁夫, 水島拓也, “クラウドソーシングにおける複数タスク割当て”, 人工知能学会全国大会, (第 27 回), (2013).
[森 11] 森 信介, 中田 陽介, Graham, N., 河原 達也, “点予測による形態素解析” , 自然言語処理, Vol.18, no. 4, pp. 367-381, (2011).
[西 13] 西 智樹, 小出 智士, 大野 宏司, 長屋 隆之, “ソーシャルネットワークを用いたクラウドソーシングの品質向上”, 人工知能学会全国大会,(第 27 回)(2013).
[Raykar 11] Raykar, V., Yu, S., “Ranking annotators for crowdsourced labeling tasks”, NIPS, (2011).
[櫻井 12] 櫻井 祐子, 沖本 天太, 岡雅 晃, 兵藤 明彦, 篠田 正人, 横尾真, “クラウドソーシングにおける品質コントロールの一考察”, JAWS, (2012).
[Snow 08] Snow, R., O’Connor, B., Jurafsky,D., Ng, A.Y., “Cheap and Fast But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks”, EMNLP, (2008).
[Welinder 10] Welinder, P., Branson, S., Belongie, S., Perona, P., “The Multidimensional Wisdom of Crowds”, NIPS, (2010).
[Whitehill 09] Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movelan, J., “Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise”, NIPS, (2009).
[Yahoo!] Yahoo!クラウドソーシング, http://crowdsourcing.yahoo.co.jp/
[Yuen et al., 2012] Yuen, M. C., et al., “TaskRec: probabilistic matrix factorization in task recommendation in crowdsourcing systems”, In proc.of ICONIP, (2012).

〔担当委員：鹿島 久嗣〕

2014 年 2 月 12 日　受理

著 者 紹 介

芦川 将之(正会員)

1999 年早稲田大学理工学部情報学科卒業. 2001 年同大学院理工学研究科修士課程修了. 現在, 株式会社東芝研究開発センターにて大規模データ処理の研究・開発に従事.

川村 隆浩(正会員)

1994 年早稲田大学大学院 理工学研究科 電気工学専攻 修士課程了. 同年 (株) 東芝入社. 現在, 同社 研究開発センター主任研究員. 2001–2002 年米国カーネギ・メロン大学 ロボット工学研究所 客員研究員 兼任. 2003 年より電気通信大学大学院 情報システム学研究科 客員准教授 兼任. 2007 年より大阪大学大学院 工学研究科 非常勤講師 兼任. 工学博士 (早稲田大学). 主としてセマンティック Web, エージェント技術の研究・開発に従事. 情報処理学会会員.

大須賀 昭彦(正会員)

1958 年生. 1981 年上智大学理工学部数学科卒. 同年 (株) 東芝入社. 同社 研究開発センター, ソフトウェア技術センター等に所属. 1985〜1989 年 (財) 新世代コンピュータ技術開発機構 (ICOT) 出向. 2007 年より, 電気通信大学 大学院情報システム学研究科 教授. 2012 年より, 国立情報学研究所 客員教授 兼任. 工学博士 (早稲田大学). 主としてソフトウェアのためのフォーマルメソッド, エージェント技術の研究に従事. 1986 年度情報処理学会論文賞受賞. IEEE Computer Society Japan Chapter Chair, 人工知能学会理事, 日本ソフトウェア科学会理事を歴任. 情報処理学会, 電子情報通信学会, 日本ソフトウェア科学会, IEEE Computer Society 各会員.