Using Crowdsourcing to Compare Document Recommendation Strategies for Conversations

Maryam Habibi
Idiap Research Institute and EPFL
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
maryam.habibi@idiap.ch

Andrei Popescu-Belis
Idiap Research Institute
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
andrei.popescu-belis@idiap.ch

ABSTRACT

This paper explores a crowdsourcing approach to the evaluation of a document recommender system intended for use in meetings. The system uses words from the conversation to perform just-in-time document retrieval. We compare several versions of the system, including the use of keywords, retrieval using semantic similarity, and the possibility for user initiative. The system's results are submitted for comparative evaluations to workers recruited via a crowdsourcing platform, Amazon's Mechanical Turk. We introduce a new method, Pearson Correlation Coefficient-Information Entropy (PCC-H), to abstract over the quality of the workers' judgments and produce system-level scores. We measure the workers' reliability by the inter-rater agreement of each of them against the others, and use entropy to weight the difficulty of each comparison task. The proposed evaluation method is shown to be reliable, and the results show that adding user initiative improves the relevance of recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation, Retrieval models*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*

General Terms

Evaluation, Uncertainty, Reliability, Metric

Keywords

Document recommender system, user initiative, crowdsourcing, Amazon Mechanical Turk, comparative evaluation

1. INTRODUCTION

A document recommender system for conversations provides suggestions for potentially relevant documents within a conversation, such as a business meeting. Used as a virtual secretary, the system constantly retrieves documents that are related to the words of the conversation, using automatic speech recognition, but users could also be allowed to make explicit queries. Such a system builds upon previous approaches known as implicit queries, just-in-time retrieval, or zero query terms, which were recently confirmed as a promising research avenue [1].

Evaluating the relevance of recommendations produced by such a system is a challenging task. Evaluation in use requires the full deployment of the system and the setup of numerous evaluation sessions with realistic meetings. That is why alternative solutions based on simulations are important to find. In this paper, we propose to run the document recommender system over a corpus of conversations and to use crowdsourcing to compare the relevance of results in various configurations of the system.

A crowdsourcing platform, here Amazon's Mechanical Turk, is helpful for several reasons. First, we can evaluate a large amount of data in a fast and inexpensive manner. Second, workers are sampled from the general public, which might represent a more realistic user model than the system developers, and have no contact with each other. However, in order to use workers' judgments for relevance evaluation, we have to circumvent the difficulties of measuring the quality of their evaluations, and factor out the biases of individual contributions.

We will define an evaluation protocol using crowdsourcing, which estimates the quality of the workers' judgments by predicting task difficulty and workers' reliability, even if no ground truth to validate the judgments is available. This approach, named Pearson Correlation Coefficient-Information Entropy (PCC-H), is inspired by previous studies of inter-rater agreement as well as by information theory.

This paper is organized as follows. Section 2 describes the document recommender system and the different versions which will be compared. Section 3 reviews previous research on measuring the quality of workers' judgments for relevance evaluation and labeling tasks using crowdsourcing. Section 4 presents our design of the evaluation micro-tasks – "Human Intelligence Tasks" for the Amazon's Mechanical Turk. In Section 5, the proposed PCC-H method for measuring the quality of judgments is explained. Section 6 presents the results of our evaluation experiments, which on the one hand validate the proposed method, and on the other hand indicate the comparative relevance of the different versions of the recommender system.

2. OUTLINE OF THE DOCUMENT RECOMMENDER SYSTEM

The document recommender system under study is the Automatic Content Linking Device (ACLD [15, 16]), which uses real-time automatic speech recognition [8] to extract words from a conversation in a group meeting. The ACLD filters and aggregates the words to prepare queries at regular time intervals. The queries can be addressed to a local database of meeting-related documents, including also transcripts of past meetings if available, but also to a web search engine. The results are then displayed in an unobtrusive manner to the meeting participants, which can consult them if they find them relevant and purposeful.

Since it is difficult to assess the utility of recommended documents from an absolute perspective, we aim instead at comparing variants of the ACLD, in order to assess the improvement (or lack thereof) due to various designs. Here, we will compare four different approaches to the recommendation problem – which is in all cases a cold-start problem, as we don't assume knowledge about participants. Rather, in a pure content-based manner, the ACLD simply aims to find the closest documents to a given stretch of conversation.

The four compared versions are the following ones. Two "standard" versions as in [15] differ by the filtering procedure for the conversation words. One of them (noted AW) uses all the words (except stop words) spoken by users during a specific period (typically, 15 s) to retrieve related documents. The other one (noted KW) filters the words, keeping only keywords from a pre-defined list related to the topic of the meeting.

Two other methods depart from the initial system. One of them implements semantic search (noted SS [16]), which uses a graph-based semantic relatedness measure to perform retrieval. The most recent version allows user initiative (noted UI), that is, it can answer explicit queries addressed by users to the system, with results replacing spontaneous recommendations for one time period. These are processed by the same ASR component, with participants using a specific name for the system ("John") to solve the addressing problem.

In the evaluation experiments presented here, we only use human transcriptions of meetings, to focus on the evaluation of the retrieval strategy itself. We use one meeting (ES2008b) from the AMI Meeting Corpus [6] in which the design of a new remote control for a TV set is discussed. The explicit users' requests for the UI version are simulated by modifying the transcript at 24 different locations where we believe that users are likely to ask explicit queries – a more principled approach for this simulation is currently under study. We restrict the search to the Wikipedia website, mainly because the semantic search system is adapted to this data, using a local copy of it (WEX) that is semantically indexed. Wikipedia is one of the most popular general reference works on the Internet, and recommendations over it are clearly of high potential interest. But alternatively, all our systems (except the semantic one) could also be run with non-restricted web searches via Google, or limited to other web domains or websites.

The 24 fragments of the meeting containing the explicit queries are submitted for comparison. That is, we want to know which of the results displayed by the various versions at the moment following the explicit query are considered most relevant by external judges. As the method allows only binary comparisons, as we will now describe, we will compare UI with the AW and KW versions, and then SS with KW.

3. RELATED WORK

Relevance evaluation is a difficult task because it is subjective and expensive to be performed. Two well-known methods for relevance evaluation are the use of a click-data corpus, or the use of human experts [18]. However, in our case, producing click data or hiring professional workers for relevance evaluation would both be overly expensive. Moreover, it is not clear that evaluation results provided by a narrow range of experts would be generalizable to a broader range of end users. In contrast, crowdsourcing, or peer collaborative annotation, is relatively easy to prototype and to test experimentally, and provides a cheap and fast approach to explicit evaluation. However, it is necessary to consider some problems which are associated to this approach, mainly the reliability of the workers' judgments (including spammers) and the intrinsic knowledge of the workers [3].

Recently, many studies have considered the effect of the task design on relevance evaluation, and proposed design solutions to decrease time and cost of evaluation and to increase the accuracy of results. In [9], several human factors are considered: query design, terminology and pay, with their impact on cost, time and accuracy of annotations. To collect proper results, the effect of user interface guidelines, inter-rater agreement metrics and justification analysis were examined [2], showing e.g. that asking workers to write a short explanation in exchange of a bonus is an efficient method for detecting spammers. In addition, in [11], different batches of tasks were designed to measure the effect of pay, required effort and worker qualifications on the accuracy of resulting labels. Another paper [13] has studied how the distribution of correct answers in the training data affects worker responses, and suggested to use a uniform distribution to avoid biases from unethical workers.

The Technique for Evaluating Relevance by Crowdsourcing (TERC, see [4]) emphasizes the importance of qualification control, e.g. by creating qualification tests that must be passed before performing the actual task. However, another study [2] showed that workers may still perform tasks randomly even after passing qualification tests. Therefore, it is important to perform partial validation of each worker's tasks, and weight the judgments of several workers to produce aggregate scores [4].

Several other studies have focused on Amazon's Mechanical Turk crowdsourcing platform and have proposed techniques to measure the quality of workers' judgments when there is no ground truth to verify them directly [17, 19, 7, 10, 12]. For instance, in [5], the quality of judgments for a labeling task is measured using the inter-rater agreement and majority voting. Expectation maximization (EM) has sometimes been used to estimate true labels in the absence of ground truth, e.g. in [17] for an image labeling task. In order to improve EM-based estimation of the reliability of workers, the confidence of workers in each of their judgments has been used in [7] as an additional feature – the task being dominance level estimation for participants in a conversation. As the performance of the EM algorithm is not guaranteed, a new method [10] was introduced to estimate reliability based on low-rank matrix approximation.

All of the above-mentioned studies assume that tasks share the same level of difficulty. To model both task difficulty and user reliability, an EM-based method named GLAD was proposed by [19] for an image labeling task. However, this method is sensitive to the initialization value, hence a good estimation of labels requires a small amount of data with ground truth annotation [12].

4. SETUP OF THE EXPERIMENT

Amazon's Mechanical Turk (AMT) is a crowdsourcing platform which gives access to a vast pool of online workers paid by requesters to complete human intelligence tasks (HITs). Once designed and published, registered workers that fulfill the requesters' selection criteria are invited by AMT service to work on HITs in exchange for a small amount of money per HIT [3].

As it is difficult to find an absolute relevance score for each version of the ACLD recommender system, we only aim for comparative relevance evaluation between versions. For each pair of versions, a batch of HITs was designed with their results. Each HIT (see example in Fig. 1) contains a fragment of conversation transcript with the two lists of document recommendations to be compared. Only the first six recommendations are kept for each version. The lists from the two compared versions are placed in random positions (first or second) across HITs, to avoid biases from a constant position.

We experimented with two different HIT designs. The first one offers evaluators a binary choice: either the first list is considered more relevant than the second, or vice-versa. In other words, workers are obliged to express a preference for one of the two recommendation sets. This encourages decisions, but of course may be inappropriate when the two answers are of comparable quality, though this may be evened out when averaging over workers. The second design gives workers four choices (as in Figure 1): in addition to the previous two options, they can indicate either that both lists seem equally relevant, or equally irrelevant. In both designs, workers must select exactly one option.

To assign a value to each worker's judgment, a binary coding scheme will be used in the computations below, assigning a value of 1 to the selected option and 0 to all others. The relevance value RV of each recommendation list for a meeting fragment is computed by giving a weight to each worker judgment and averaging them. The *Percentage of Relevance Value*, noted PRV, shows the relevance value of each compared system, and is computed by assigning a weight to each part of the meeting and averaging the relevance values RV for all meeting fragments.

There are 24 meeting fragments, hence 24 HITs in each batch for comparing pairs of systems, for UI vs. AW and UI vs. KW. As user queries are not needed for comparing SS vs. KW, we designed 36 HITs, with 30-second fragments for each. There are 10 workers per HIT, so there are 240 total assignments for UI-vs-KW and for UI-vs-AW (with a 2-choice and 4-choice design for each), and 360 for SS-KW. As workers are paid 0.02 USD per HIT, the cost for the five separate experiments was 33 USD, with an apparent average hourly rate of 1.60 USD. The average time per assignment is almost 50 seconds. All five tasks took only 17 hours to be performed by workers via AMT. For qualification control we allow workers with greater than 95% approval rate or with more than 1000 approved HITs.

5. THE PCC-H METHOD

Majority voting is frequently used to aggregate multiple sources of comparative relevance evaluation. However, this assumes that all HITs share the same difficulty and all the workers are equally reliable. We will take here into account the task difficulty W_q and the workers' reliability r_w, as it was shown that they have a significant impact on the quality of the aggregated judgments. We thus introduce a new computation method called PCC-H, for *Pearson Correlation Coefficient-Information Entropy*.

5.1 Estimating Worker Reliability

The PCC-H method computes the W_q and r_w values in two steps. In a first step, PCC-H estimates the reliability of each worker r_w based on the Pearson correlation of each worker's judgment with the average of all the other workers judgments (see Eq. 1).

$$r_w = \frac{\sum_{a=1}^{A}\sum_{q=1}^{Q}(X_{qwa} - \bar{X_{wa}})(Y_{qa} - \bar{Y_a})}{(Q-1)S_{X_{wa}}S_{Y_a}} \quad (1)$$

In Equation 1, Q is number of meeting fragments, X_{wqa} is the value that worker w assigned to option a of fragment q, X_{wqa} has value 1 if that option a is selected by worker w, otherwise it is 0. $\bar{X}_{wa}$ and $S_{X_{wa}}$ are the expected value and standard deviation of variable X_{wqa} respectively. Y_{qa} is the average value which all other workers assign to the option a of fragment q. $\bar{Y_a}$ and S_{Y_a} are the expected value and standard deviation of variable Y_{qa}.

The value of r_w computed above is used as a weight for computing RV_{qa}, the relevance value of option a of each fragment q, according to Eq. 2 below:

$$RV_{qa} = \frac{\sum_{w=1}^{W} r_w X_{wqa}}{\sum_{w=1}^{W} r_w} \quad (2)$$

For HIT designs with two options, RV_{qa} shows the relevance value of each answer list a. However, for the four option HIT designs, RV_{ql} for each answer list l is formulated as Eq. 3 below:

$$RV_{ql} = RV_{ql} + \frac{RV_{qb}}{2} - \frac{RV_{qn}}{2} \quad (3)$$

In this equation, half of the relevance value of the case in which both lists are relevant RV_{qb} is added as a reward, and half of the relevance value of the case in which both lists are irrelevant RV_{qn} is subtracted as a penalty from the relevance value of each answer list RV_{ql}.

5.2 Estimating Task Difficulty

In a second step, PCC-H considers the task difficulty for each fragment of the meeting. The goal is to reduce the effect of some fragments of the meeting, in which there is an uncertainty in the workers judgments, e.g. because there are no relevant search results in Wikipedia for the current fragment. To lessen the effect of uncertainty in our judgments, the entropy of answers for each fragment of the meeting is computed and a function of it is used as a weight for each fragment. This weight is used for computing the percentage of relevance value PRV. Entropy, weight and PRV are defined in Eqs. 4–6, where A is the number of options, and H_q and W_q are the entropy and weight of fragment q.

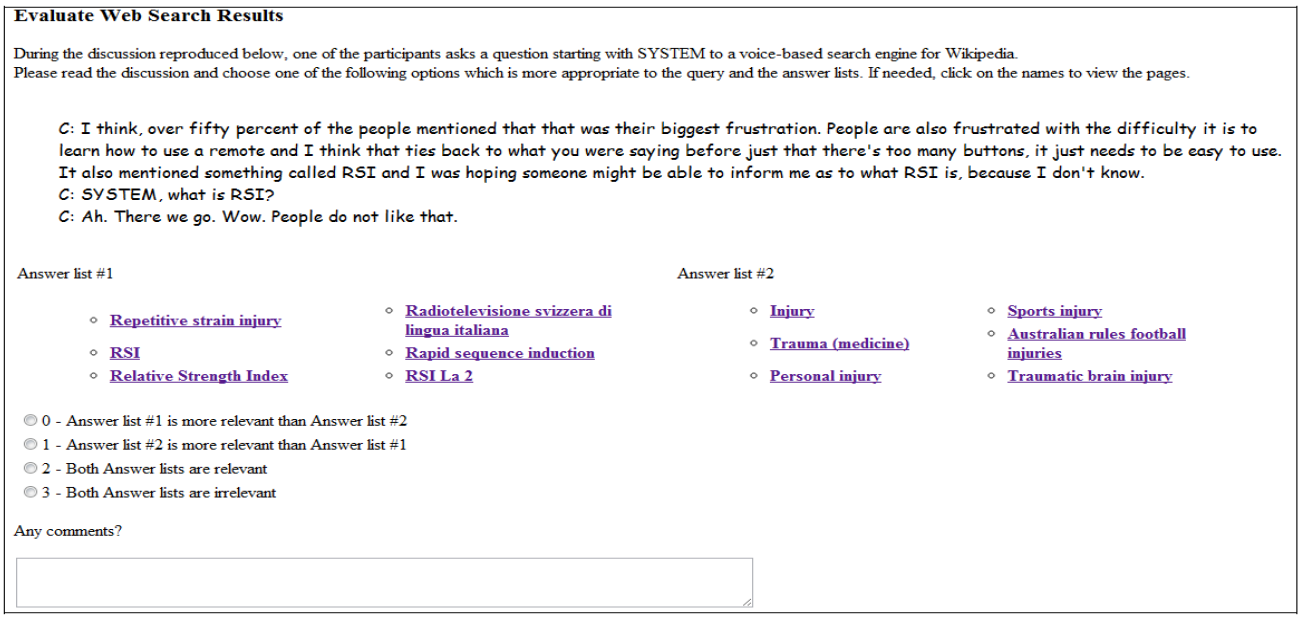

Figure 1: Snapshot of a 4-choice HIT: workers read the conversation transcript, examine the two answer lists (with recommended documents for the respective conversation fragment) and select one of the four comparative choices (#1 better than #2, #2 better than #1, both equally good, both equally poor). A short comment can be added.

$$H_q = -\sum_{a=1}^{A} RV_{qa} log(RV_{qa}) \quad (4)$$

$$W_q = 1 - H_q \quad (5)$$

$$PRV_a = \frac{\sum_{q=1}^{Q} W_q RV_{qa}}{\sum_{q=1}^{Q} W_q} \quad (6)$$

6. RESULTS OF THE EXPERIMENTS

Two sets of experiments were performed. First, we attempt to validate the PCC-H method. Then, we apply the PCC-H method to compute PRV for each answer list to conclude which version of the system outperforms the others.

In order to make an initial validation of the workers judgments, we compare the judgments of individual workers with those of an expert. For each worker, the number of fragments for which the answer is the same as the expert's answer is counted, and the total is divided by the number of fragments to compute accuracy. Then we compare this value with r_w, which is estimated as the reliability measurement for each worker's judgment. The percentage of agreement between each worker vs. the expert e_w and the r_w for each worker for one of the batches is shown in Table 1, with an overall agreement between these two values for each worker. In other words, workers who have more similarity with our expert also have more inter-rater agreement with other workers. Since in the general case there is no ground truth (expert) to verify workers judgments, we rely on the inter-rater agreement for the other experiments.

Firstly, equal weights for all the user evaluations and fragments are assigned to compute PRVs for two answer lists of our experiments, which are shown in Table 2.

Table 1: Percentage of agreement between a single worker and the expert, and a single worker and the other workers, for the KW system and 4-choice HITs

Worker #	e_w	r_w
1	0.66	0.81
2	0.54	0.65
3	0.54	0.64
4	0.50	0.71
5	0.50	0.60
6	0.50	0.35
7	0.41	0.24
8	0.39	0.33
9	0.36	0.34
10	0.31	0.12

In this approach, it is assumed that all the workers are reliable and all the fragments share the same difficulty. To handle workers' reliability, we consider workers with lower r_w as outliers. One approach is to remove all the outliers. For instance, the four workers with lowest r_w are considered outliers and are deleted, and the same weight is given to the remaining six workers. The result of comparative evaluation based on removing outliers is shown in Table 3.

In the computation above, an arbitrary border was defined between outliers and other workers as a decision boundary for removing outliers. However, instead of deleting workers with lower r_w, which might still have potentially useful insights on relevance, it is rational to give a weight to all workers' judgments based on a confidence value. The PRV for each answer list of four experiments based on assigning weight r_w to each worker's evaluation, and equal weights to all meeting fragments are shown in Table 4.

Table 2: PRVs for AW-vs-UI and KW-vs-UI pairs

All workers and fragments with equal weights		2-choice HITs	4-choice HITs
AW-vs-UI	PRV_{AW}	30%	26%
	PRV_{UI}	70%	74%
KW-vs-UI	PRV_{KW}	45%	35%
	PRV_{UI}	55%	65%

Table 3: PRVs for AW-vs-UI and KW-vs-UI pairs

Six workers and fragments with equal weights		2-choice HITs	4-choice HITs
AW-vs-UI	PRV_{AW}	24%	13%
	PRV_{UI}	76%	86%
KW-vs-UI	PRV_{KW}	46%	33%
	PRV_{UI}	54%	67%

In order to show that our method is stable on different HIT designs, we used two different HIT designs for each pair as mentioned in Section 4. We show that PRV converges to the same value for each pair with different HIT designs. As observed in Table 4, PRVs of AW-vs-UI pair are not quite similar for two different HIT designs, although the answer lists are the same. In fact, we observed that, in several cases, there was no strong agreement among workers to decide which answer list is more relevant to that meeting fragment, and we consider that these are "difficult" fragments. Since the source of uncertainty is undefined, we can reduce the effect of that fragment on the comparison by giving a weight to each fragment in proportion of the difficulty of assigning RV_{ql}. The PRV values thus obtained for all experiments are represented in Table 5. As shown there, the PRVs of AW-vs-UI pair are now very similar for 2-HIT and 4-HIT tasks. Moreover, the difference between the system versions is emphasized, which indicates that the sensitivity of the comparison method has increased.

Moreover, we compare the PCC-H method with the majority voting method and the GLAD method (Generative model of Labels, Abilities, and Difficulties [19]) for estimating comparative relevance value through considering task difficulty and worker reliability parameters. We run the GLAD algorithm with the same initial values for all four experiments. The PRVs which are computed by majority voting, GLAD and PCC-H are shown in Table 6.

As shown in Table 6, PRVs which are computed by the PCC-H method for both HIT designs are very close to those of GLAD for the 4-choice HIT design. Moreover, the PRV values obtained by the PCC-H method for the two different HIT designs are very similar, which is less the case for majority voting and GLAD. This means that PCC-H method is able to calculate the PRVs independent of the exact HIT design. Moreover, the PRV values calculated using PCC-H are more robust since the proposed method is not dependent on initialization values, as GLAD is. Therefore, using PCC-H for measuring the reliability of workers judgments is also an appropriate method for qualification control of workers from crowdsourcing platforms.

The proposed method is also applied for comparative evaluation of SS-vs-KW search results (semantic search vs. keyword-based search). The PRVs are calculated by three different methods as shown in Table 7. The first method is the majority voting method which considers all the workers and fragments with the same weight. The second method assigns weights computed by PCC-H method to measure PRVs, the third one is the GLAD method. Therefore the SS version outperforms the KW version according to all three scores.

Table 4: PRVs for AW-vs-UI and KW-vs-UI pairs

All workers with different weights and parts with equal weights		2 choices HIT design	4 choices HIT design
AW-vs-UI	PRV_{AW}	24%	18%
	PRV_{UI}	76%	82%
KW-vs-UI	PRV_{KW}	33%	34%
	PRV_{UI}	67%	66%

Table 5: PRVs for AW-vs-UI and KW-vs-UI pairs

All workers with different weights and fragments with different weights (PCC-H method)		2-choice HITs	4-choice HITs
AW-vs-UI	PRV_{AW}	19%	15%
	PRV_{UI}	81%	85%
KW-vs-UI	PRV_{KW}	23%	26%
	PRV_{UI}	77%	74%

7. CONCLUSION AND PERSPECTIVES

In all the evaluation steps, the UI system appeared to produce more relevant recommendations than AW or KW. Using KW instead of AW improved PRV by 10 percent. This means that using UI, i.e. when users ask explicit queries in conversation, improves over AW or KW versions, i.e. with spontaneous recommendations. Nevertheless, KW can be used as an assistant which suggests documents based on the context of the meeting along with the UI version, that is, spontaneous recommendations can be made when no user initiates a search. Moreover, the SS version works better than the KW version, which shows the advantage of semantic search.

As for the evaluation method, PCC-H outperformed the GLAD method proposed earlier for estimating task difficulty and reliability of workers in the absence of ground truth. Based on the evaluation results, the PCC-H method is acceptable for qualification control of AMT workers or judgments, because it provides a more stable PRV score across different HIT designs. Moreover, PCC-H does not require any initialization.

The comparative nature of PCC-H imposes some restrictions on the evaluations that can be carried out. For instance, if N versions must be compared, this calls in theory for $N * (N-1)/2$ comparisons, which is clearly impractical when N grows. This can be solved if *a priori* knowledge about the quality of the systems is available, to avoid redundant comparisons. Moreover, an approach to reduce the number of pairwise comparisons required from human raters proposed in [14] could be ported to our context. For

Table 6: PRVs computed by the majority voting, the GLAD, and the PCC-H methods

Methods pairs		Majority voting, GLAD, PCC-H	
		2-choice HITs	4-choice HITs
AW-vs-UI	PRV_{AW}	30%, 23%, 19%	26%, 13%, 15%
	PRV_{UI}	70%, 77%, 81%	74%, 87%, 85%
KW-vs-UI	PRV_{KW}	45%, 47%, 23%	35%, 23%, 26%
	PRV_{UI}	55%, 53%, 77%	65%, 77%, 74%

Table 7: PRVs for SS-vs-KW

Method pair		Majority voting, GLAD, PCC-H
		4-choice HITs
SS-vs-KW	PRV_{SS}	0.88%, 0.88%, 0.93%
	PRV_{KW}	0.12%, 0.12%, 0.07%

progress evaluation, a new version must be compared with the best performing previous version, looking for measurable improvement, in which case PCC-H fully answers the evaluation needs.

There are instances in which the search results of both versions are irrelevant. The goal of future work will be to reduce the number of such uncertain instances, to deal with ambiguous questions, and to improve the processing of user-directed queries by recognizing the context of the conversation. Another experiment should improve the design of simulated user queries, in order to make them more realistic.

8. ACKNOWLEDGMENTS

The authors are grateful to the Swiss National Science Foundation for its financial support under the IM2 NCCR on Interactive Multimodal Information Management (see www.im2.ch).

9. REFERENCES

[1] J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges and opportunities for information retrieval: Report from SWIRL 2012. *SIGIR Forum*, 46(1):2–32, 2012.

[2] O. Alonso and R. A. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 153–164, 2011.

[3] O. Alonso and M. Lease. Crowdsourcing 101: Putting the "wisdom of the crowd" to work for you. WSDM Tutorial, 2011.

[4] O. Alonso, D. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42:9–15, 2008.

[5] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254, 1996.

[6] J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation Journal*, 41(2):181–190, 2007.

[7] G. Chittaranjan, O. Aran, and D. Gatica-Perez. Exploiting observers' judgments for nonverbal group interaction analysis. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 2011.

[8] P. N. Garner, J. Dines, T. Hain, A. El Hannani, M. Karafiat, D. Korchagin, M. Lincoln, V. Wan, and L. Zhang. Real-time ASR from meetings. In *Proceedings of Interspeech*, pages 2119–2122, 2009.

[9] C. Grady and M. Lease. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL-HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 172–179, 2010.

[10] D. R. Karger, S. Oh, and D. Shah. Budget-optimal crowdsourcing using lowrank matrix approximations. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2011.

[11] G. Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 165–176, 2011.

[12] F. K. Khattak and A. Salleb-Aouissi. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011.

[13] J. Le, A. Edmonds, V. Hester, and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation : The effects of training question distribution. In *Proceedings of the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation*, pages 17–20, 2010.

[14] X. Llorà, K. Sastry, D.E. Goldberg, A. Gupta, and L. Lakshmi. Combating user fatigue in iGAs: Partial ordering, support vector machines, and synthetic fitness. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO '05)*, pages 1363–1370, 2005.

[15] A. Popescu-Belis, E. Boertjes, J. Kilgour, P. Poller, S. Castronovo, T. Wilson, A. Jaimes, and J. Carletta. The AMIDA automatic content linking device: Just-in-time document retrieval in meetings. In *Proceedings of Machine Learning for Multimodal Interaction (MLMI)*, pages 272–283, 2008.

[16] A. Popescu-Belis, M. Yazdani, A. Nanchen, and P. Garner. A speech-based just-in-time retrieval system using semantic search. In *Proceedings of the 49th Annual Meeting of the ACL*, pages 80–85, 2011.

[17] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labeling of venus images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1085–1092, 1994.

[18] P. Thomas and D. Hawking. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 94–101, 2006.

[19] J. Whitehill, P. Ruvolo, T.-F. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2035–2043. 2009.