Crowdsourcing a News Query Classification Dataset

Richard M. C. McCreadie
Department of Computing Science
University of Glasgow
Glasgow, G12 8QQ
richardm@dcs.gla.ac.uk

Craig Macdonald
Department of Computing Science
University of Glasgow
Glasgow, G12 8QQ
craigm@dcs.gla.ac.uk

Iadh Ounis
Department of Computing Science
University of Glasgow
Glasgow, G12 8QQ
ounis@dcs.gla.ac.uk

ABSTRACT

Web search engines are well known for aggregating news vertical content into their result rankings in response to queries classified as news-related. However, no dataset currently exists upon which approaches to news query classification can be evaluated and compared. This paper studies the generation and validation of a news query classification dataset comprised of labels crowdsourced from Amazon's Mechanical Turk and details insights gained. Notably, our study focuses around two challenges when crowdsourcing news query classification labels: 1) how to overcome our workers' lack of information about the news stories from the time of each query and 2) how to ensure the resulting labels are of high enough quality to make the dataset useful. We empirically show that a worker's lack of information about news stories can be addressed through the integration of news-related content into the labelling interface and that this improves the quality of the resulting labels. Overall, we find that crowdsourcing is suitable for building a news query classification dataset.

General Terms: Performance, Experimentation

Keywords: News Query Classification, Crowdsourcing, Vertical Search

1. INTRODUCTION

General-purpose Web search engines are well known for integrating focused news vertical content into their search rankings when the user query is judged as holding some news-related intent [8, 12]. In particular, every user query submitted is classified as either having a news-related intent or not. If so, the Web search engine will aggregate appropriate news content into the search ranking returned. However, while it is likely that commercial search engines have large internal datasets for evaluating their news query classification performance, there currently exists no publicly available dataset upon which news query classification approaches can be evaluated and compared.

On the other hand, *crowdsourcing* [13] has been championed as a viable method for creating datasets both quickly and cheaply, whilst still maintaining a reasonable degree of quality [1]. We hypothesise that crowdsourcing is a suitable means to generate a news query classification dataset, which will be useful when investigating news query classification using evidence from various sources, e.g. the full query-log. In this paper, we detail the generation and validation of a such a dataset comprised of real user queries from a search engine query log and associated news classification labels crowdsourced using Amazon's Mechanical Turk.

In our study, we follow an iterative design methodology, as recommended in [2], during the creation our news query classification dataset. We propose multiple interfaces for crowdsourced query labelling and evaluate these interfaces empirically in terms of the quality of the resulting labels on a small representative sample of user queries from a Web search engine query log. Later, we use the best performing of these interfaces to generate our final news query classification dataset comprised of a larger query sample from the same log. We report the quality of our resulting news query classification dataset in terms of inter-worker labelling agreement and accuracy with regard to labels created separately by the authors. Moreover, we further investigate its quality in the form of an additional agreement study, in which crowdsourcing is leveraged for quality assurance.

Notably, one of the most interesting aspects of news query classification labelling is the temporal nature of news-related queries [16]. In particular, a query should only be labelled as news-related if there was a relevant noteworthy story in the news around the time each query was made. However, the query log we employ dates back to 2006 [7], hence there is no guarantee that our workers will remember what the major news stories were from the time of each query. In this work, we empirically investigate the effect that this has on labelling quality. Moreover, we propose the integration of news headlines, article summaries and Web search results into the interface seen by the workers to address this problem.

The main contributions of this paper are four-fold: firstly, we examine the suitability of crowdsourcing for the creation of a news query classification dataset; secondly we both propose and evaluate methods to overcome the temporal nature of news queries described above; thirdly, we investigate a novel application of crowdsourcing as a quality assurance tool; and lastly we propose some best practices based on experience gained from creating this dataset.

The remainder of the paper is structured as follows. In the next section, we further motivate the need for a news query classification dataset as well as describe related work in news-aggregation. Section 3 provides a brief background into crowdsourcing in general, as well as recent studies that have used crowdsourcing. In Section 4, we define the methodology used to create our news query classification dataset, discuss query sampling and describe the interfaces evaluated later. Section 5 covers our experimental setup. In Section 6, we empirically determine the best of our proposed interfaces, report on the quality of the news query classification dataset produced and provide some best practices when crowdsourcing. Finally, in Section 7, we provide concluding remarks.

2. NEWS AGGREGATION

News aggregation is an important problem in Information Retrieval (IR), with as much as 10% of queries possibly being news-related [5]. Moreover, classifying queries as news-related or not is a challenging task. In particular, the news-relatedness of a query is not solely dependent upon the terms it contains, but also the main news stories of the time. Hence, two identical queries made at dif-

CSE'2010, July 23rd, 2010, Geneva, Switzerland
ACM XXX-X-XXXXX-XXX-X/XX/XX.

ferent times may not always receive the same label. For example, the query 'ash' normally would not be seen as being news-related, since the dominant interpretation is the rock band with the same name. However, in April 2010 the query suddenly became so, as an ash cloud grounded aircraft in Europe and the United States. Furthermore, there are also notable challenges when classifying queries as news-related or not soon after a story breaks. Indeed, as news-related queries are now being submitted to Web search engines mere minutes after a newsworthy event occurs [19], new approaches need to be developed which can correctly classify such 'breaking news' queries [16]. Hence, the need is clear for a dataset as a basis upon which to investigate these classification challenges.

In addition, news aggregation has recently become a somewhat hot topic in IR. In particular, Arguello *et al*. [3] investigated the construction of offline classifiers for vertical content in the presence of user feedback, while Diaz *et al*. [8] investigated the online aggregation of news content using explicit user feedback. However, these studies use private datasets to evaluate performance.

Our goal is to produce a standard dataset such that news query classification approaches both old and new can be easily evaluated and compared. We examine the suitability of crowdsourcing to build such a dataset from a query log provided by a real Web search engine. In the following section, we provide a brief background into crowdsourcing and motivate its application for creating a news query classification dataset.

3. CROWDSOURCING

In this work, we propose to build a news query classification dataset using crowdsourcing [13]. Crowdsourcing is an attractive option for researchers and industry alike as a method for dataset generation. In particular, simple repetitive *jobs*, e.g. query labelling, can be completed at a relatively small cost, and often very quickly [1]. However, crowdsourcing has also been the subject of much controversy as to its effectiveness, in particular with regard to the lower quality of work produced [4], the lack of motivation for workers due to below-market wages [6] and susceptibility to malicious workers [10]. In general, the advantages of generating a news query classification dataset using crowdsourcing are easily quantified. Indeed, the total cost of the experiments reported in this paper is less than $200, with even the longest single job taking less than 3 days to complete. Still, the quality of the resulting labels may be questionable, due either to insufficient worker understanding of the job given to them, or a lack of important information needed to complete the job satisfactorily.

In this work, we use CrowdFlower, which is an on-demand labour website providing job creation, monitoring and analytical services on top of crowdsourcing marketplaces, most notably Amazon's Mechanical Turk (MTurk). MTurk is a service that can provide real human judgements for a variety of simple repetitive jobs. In particular, the crowdsourcer defines a *human intelligence task* (HIT), where a *worker* views an interface containing instructions on how to complete the HIT, typically along with some content to be processed, and then uses the same interface to provide feedback to the system. Workers are normally paid a small sum of money or micro payment for each HIT they complete.

Notably, there have been multiple studies into crowdsourcing with MTurk to date, which provide useful information for those wishing to use crowdsourcing for IR-related tasks. In particular, Snow *et al*. [21] and Callison-Burch [6] investigated the accuracy of labels generated using MTurk within a natural language processing context. They concluded that 'expert' levels of labelling quality can be achieved by having multiple workers complete each crowdsourcing job and taking a majority label. Indeed, for the experiments reported in this paper, we require that three individual workers label each query, taking the majority result.

Date	Time	Query	Query ID
2006-05-01	00:00:08	What is May Day?	37afe7af832649d2
2006-05-08	14:43:42	protest in Puerto rico	71ddb381f574410e

Table 1: Two example queries from the MSN Web search query log for May 2006.

Kittur *et al*. [14] also examined labelling quality when moderating Wikipedia pages. Their results highlight the need to validate the output produced by each worker, pointing out that some workers produce random or malicious labels. Following this recommendation, for both query samples used in this paper we create ground truth labels to vet our workers' output in an online manner. In this way, we can detect and then eject poorly performing workers from our jobs early on in the evaluation, saving us money and hopefully improving the quality of the final labels produced. Indeed, we later empirically show through experimentation the effect that validation has on the quality of our resulting labels.

Lastly, Alonso *et al*. [1] tackled the related task of building a TREC-style ad-hoc test collection using crowdsourcing. Our work differs from this, in that we propose to have workers label queries as news-related or not, instead of labelling a document's relevance to a query. Moreover, our news query classification task also has a novel temporal component which needs to be addressed, i.e. that a query's news-relatedness is dependent not only on the terms it contains, but also on the news stories that were important at the time the query was issued. Furthermore, unlike in [1], we also perform an empirical analysis of the produced labels' quality.

Taking this prior work into account, in the next section, we detail the methodology and data used to generate our news query classification dataset. Most notably, the creation of our two query sets, including validation queries, and the five job interfaces that are used during evaluation.

4. METHODOLOGY

The task that we address in this paper is the creation of a general, high quality dataset to evaluate approaches to news query classification. In particular, the dataset in question is comprised of a set of queries, each to be labelled as holding a news-related intent or not. The performance of any news query classification approach can then be evaluated on how well it performs against the dataset. In this work, we use real queries sampled from American users of the MSN Search engine (now Microsoft Bing) query log from 2006 [7], while we propose to generate the news query classification labels through crowdsourcing. Two example queries from the MSN query log are shown in Table 1. In particular, for each user query q from the set of all queries to be included in the dataset, we want our crowdsourced workers to classify q as holding a news-related intent or not for the time t, i.e. the time the query was made.

However, there are some important challenges that we need to address during the crowdsourcing of our dataset to ensure that the resulting labels are of a high enough quality for the dataset to be useful. Firstly, we identify the possibility of workers choosing labels in a random or malicious manner, which we propose to address in two ways, namely through having multiple workers label each query in addition to worker validation. Secondly, we also note that our workers' lack of information about the main news stories from the time of the query may hinder labelling quality. We propose to mitigate this by integrating news content from the time of the query into the job interface. Indeed, we empirically examine the integration of both news article content, i.e. headlines and news summaries, and Web search engine result rankings into the job interface, to determine which best overcomes the workers' lack of information and so provides the highest quality labels.

In order to investigate these challenges, and indeed our proposed solutions to them, we follow an iterative design methodology [2].

In particular, we begin by creating a small set of queries of approximately $\frac{1}{10}$th of the final desired dataset size for testing purposes. This query set is referred to as the *testset*. This testset is advantageous as it allows us to much more cheaply and quickly investigate the challenges described earlier. Indeed, it is important to note that prototyping, while a valuable tool when crowdsourcing, can dramatically increase the total cost of the task one wishes to address. Using the testset, we empirically evaluate the effect of validation using a baseline interface and subsequently test our proposed alternative interfaces integrating news content in an iterative manner.

Having determined the most effective job design, we then create a full size query sample, denoted the *fullset*, and crowdsource labels for it, hence creating our final dataset. We lastly evaluate the quality of this dataset in terms of inter-worker labelling agreement, accuracy with regard to labels created manually by the primary author of this paper and also in the form of a meta-agreement study performed using crowdsourcing. In the remainder of this section, we describe our query sampling and worker validation approaches, in addition to the interfaces used in later experiments.

4.1 Query Log Sampling

Recall that we propose to use two sets of queries during our experimentation: a small query set, which we refer to as the testset, as well as a larger set of queries to be included in our final dataset, denoted the fullset. Importantly, the MSN Search engine query log [7] contains almost 15 million real user queries spread over the course of May 2006, which is many times the size of either of our desired query sets. Indeed, such a large query set could not be exhaustively labelled within a reasonable time-frame, even with crowdsourcing. Instead, we propose to sample the MSN query log to create our two query sets.

Notably, there are two desirable properties that we wish our sampled query-sets to hold. Firstly, we wish our samples to be *representative* [18]. This means that the sampling method chosen should maintain the statistics of the query log as a whole. Secondly, our samples need to be *unbiased* [17], i.e. every query within the query log should have an equal chance to be selected. Should the resulting sample lack either of these properties, our final dataset will be of limited use, as the dataset would not represent the querying behaviour of real users.

A well-known and straightforward sampling strategy is random sampling [22]. In random sampling, the query log is considered a 'bag' of queries. Queries are then iteratively selected at random from the query log without replacement. Notably, random sampling is unbiased. Indeed, each query has an equal chance of being selected. However, in practice, random sampling often produces an unrepresentative sample, as there is no guarantee that the selected queries will be spread over the entire log [17].

An alternative sampling strategy that has proved popular is known as systematic sampling [17]. Here, the query log is considered as a time-ordered stream, where queries are iteratively selected based upon a time interval. For example, given a time interval of three minutes, one query will be selected for every three minutes of log. A systematic sampling approach is advantageous, in that a fairly representative sample of the query log will be produced, with sampled queries being spaced evenly across the time-range of the log. On the other hand, systematic sampling is not unbiased, as the probability of a query being selected is independent from the density of queries within each time interval [18]. Hence, a query within a very dense time interval has a much lower probability of being selected that one from a sparser time interval.

In light of the drawbacks of these two prior approaches, the Poisson sampling strategy has been proposed as a means to create both an unbiased and representative sample [18]. In particular, Poisson sampling also treats the query log as a time-ordered stream.

Poisson Sampling - Pseudo-Code

```
1: Input
       query log: a temporally ordered stream of queries
       α: a parameter to control global the sampling rate
       querylogsize: the number of queries in query log
       μ, ϖ and ς: parameters to control the sampling rate over time
2: Output
       query-set: a set of sampled queries
3: Integer numToSkip = 1000
4: Integer pos = 0
5: Integer skipped = 0
6: for each query q in query log
7:     if skipped == numToSkip
8:         add q to query-set
9:         Integer newNumToSkip = 0
10:        double[] values
11:        while values[newNumToSkip] ≤ exp(numToSkip) loop
12:            newNumToSkip = newNumToSkip + 1
13             values[newNumToSkip]=rand(0,1)*values[newNumToSkip-1]
14:        end loop
15:        Double β = cos(((pos/querylogsize)*μ)+ϖ)+ς
16:        numToSkip = numToSkip * α * β
17:        skipped = 0
18:    end if
19:    skipped = skipped + 1
20:    pos = pos + 1
17: end loop
```

Figure 1: Pseudo-code interpretation of Poisson sampling.

Queries are then sampled probabilistically in an iterative manner based upon the density of queries within the current time interval. The idea is that, for a fixed time-interval, the number of queries sampled should reflect the query density, such that each query has an equal chance to be selected. For example, during a dense section of the query log, the probability of being sampled will be higher, such that the probability of sampling any query remains constant over time. Hence, we choose Poisson sampling to sample both query-sets used in this paper. The pseudo-code of our implementation of this sampling strategy is shown in Figure 1.

Notably, to control the overall rate at which we sample the log under this approach, and hence determine the final sample size, we introduced a parameter α on the distance between sampled queries, referred to as *numToSkip* in Figure 1. Furthermore, we note that queries near the start and end of the query log might be less useful when using the final dataset to evaluate approaches to news query classification. In particular, approaches that leverage evidence from other temporally close queries from the log [8] may be unfairly penalised due to a lack of available surrounding queries. As such, we favour our sampling towards the centre of the query log using a second parameter β on *numToSkip*. Specifically, the β value is dependent of the current position in the query log. *numToSkip* will be increased for queries near the start and end of the query log, hence the number of sampled queries will be less, while *numToSkip* is decreased for queries near the centre of the query log, thereby increasing the number of queries sampled. The exact distribution of β values is defined in terms of three parameters, μ, ϖ and ς, which were set experimentally to 3, 1.6 and 1.5 respectively, such that a β distribution with the above properties was observed.

Using this sampling method, we created the two aforementioned query-sets, i.e the testset and the fullset, from the MSN query log. In particular, through experimentation we selected an α value of 3 to create the fullset, resulting in a sample of just over 1000 of the 15 million original queries. However, we also noted that when creating the testset, the resulting sample was very sparse, i.e. only around 3 queries per day were sampled on average in our tests. Moreover, we later evaluate some interfaces that include news-related content for the day of a set of queries. To ensure that jobs have enough queries for a given day, we instead limit our Poisson sampling approach

	MSN query log	*fullset*	*testset*
Time Range	01/05 to 31/05	01/05 to 31/05	15/05
Number of Queries	14,921,286	1206	91
Mean Queries per Day	481,331	38.9	91
Mean Query Length	2.29	2.39	2.49

Table 2: Statistics for the MSN query log from 2006, as well as the sampled testset and fullset.

	fullset	*testset*
Time Range	01/05 to 31/05	15/05
Number of Queries	61	9
Mean Queries per Day	1.97	9
% of Target Query-set	5%	10%
News-Related Queries	47	5
Non-News-Related Queries	14	4

Table 3: Statistics for the validation sets created for the testset and fullset.

to only those queries from a single day. In particular, we use May 15th, which is the middle of the query log. In this way, we create a testset which is representative of a single day only, but where each job contains enough queries for it to be realistic. The assumption we make is that worker labelling performance will be similar for different days. With this restriction in place, we create our testset using an α value of 2, resulting in a query-set of approximately $\frac{1}{10}$th the size of the fullset. Statistics for both the query-sets created are shown in Table 2.

4.2 Validation of Worker Labelling

Earlier, we raised the possibility that workers might label our queries in a random or malicious manner [14]. Indeed, it is logical for a worker to try to maximise their profit while minimising the effort required if there is no penalty in doing so [10]. This is especially acute in our case, as the binary labelling jobs that we ask our workers to complete can be easily and quickly accomplished by selecting labels at random. Hence, to address this, we propose to perform online validation of our worker labels against a set of ground-truth query-label pairs. The idea is that should a worker try and 'game' our system by randomly labelling queries, their accuracy on the ground-truth queries will be low. As such, we can identify and then eject those workers from our jobs. This is advantageous not only because 'bad' worker labels can be ignored, thereby improving the quality of the resulting dataset, but workers ejected on such grounds are left un-paid.

To perform this type of validation, we create one validation set comprised of ground-truth queries and associated labels for each of the two query-sets previously described. As recommended by CrowdFlower, which supports this form of validation, we create validation sets of between 5% to 10% of the target query-set size[1]. Notably, when selecting validation queries, it is important to consider the background probability of queries belonging to each class. For example, in our case, at best only 10% of queries might be news-related [5]. Therefore, if we choose a representative distribution for our validation set, then just by labeling each query as non-news-related a worker would achieve 90% accuracy. Instead, we favoured our validation sets toward the news-related class, thereby forcing our workers to pick out the validation queries from the predominently non-news-related background queries. Statistics for our two validation sets are shown in Table 3.

4.3 Query Labelling Interfaces

Recall that the job that we want our workers to complete is the labelling of the queries in our aforementioned query sets as news-related or not. However, due to the temporal nature of news, the time at which each query was made must also be considered. In particular, a query should only be labelled as news-related if there was also a relevant noteworthy story in the news at the time the query was made. Indeed, even though two identical queries from different times may be judged, there is no guarantee that they should be assigned the same label. To this end, we need to design an interface comprised of two distinct components, namely: a set of instructions explaining the job the worker is to complete; and a labelling interface with which the user labels each query.

[1] `http://crowdflower.com/docs/gold`

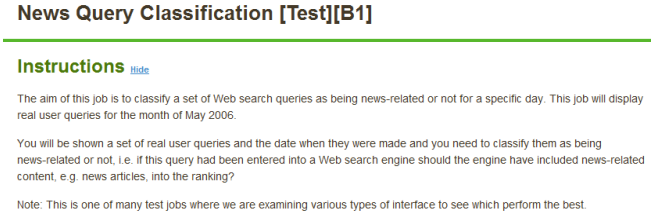

Figure 2: The basic instructions shown to our workers for each job.

Figure 3: The basic interface with which our workers label each query.

With this in mind, we designed our basic job interface as shown in Figures 2 and 3. In particular, the instructions we provide to each worker are presented by Figure 2. Notably, the first paragraph is a summary of the job and is displayed to workers searching for available jobs to complete. Hence, it is important that this succinctly describes what the worker will be asked to do. The second paragraph further clarifies the meaning of news-relatedness, as news-relatedness can be subjective in nature. For example, a worker might label the query 'ipad sales' as being news-related at the time this paper was written, as there was a news story indicating that the Apple iPad had sold over 2 million units. However, a worker uninterested in technology or holding anti-Apple views might label it otherwise.

Figure 3 shows the labelling interface with which our workers will interact during a job. Importantly, this labelling interface is replicated for each query. In particular, the workers are shown a query in addition to the date and time it was made, and asked to judge that query as being news-related or not. This interface combination is referred to as *Basic*.

However, earlier we identified our workers' lack of information about the news stories from the time of the queries as a factor which might hinder labelling quality. Indeed, we hypothesise that our *Basic* interface, as described above, will be insufficient to garner high-quality judgements, as our workers lack the information needed to accurately judge queries as news-related or not. Furthermore, the query log that we use dates from 2006, hence, our workers will likely remember little from that far back. Moreover, the nature of crowdsourcing, in particular the lack of worker motivation, makes it unlikely that workers will independently attempt to address this, e.g. by searching news archives.

To address this issue, we propose to incorporate news-related content from the time of the queries into our interface design. Our intuition is that by providing the workers with information about the news stories from around the time of the query, the workers will be able to make better informed judgements, hence increasing

```
Headline:
32 Dead in Iraq Attacks

Summary:
Two suicide car bombers tore into a checkpoint for Baghdad's airport, killing at least 14 people and
wounding 16. It was the first bomb attack in nearly a year aimed at the airport, and the worst in a
spree of violent assaults that left at least 32 people dead on Iraq's deadliest day in weeks.
```

Figure 4: Example headline and news summary extracted from a New York Times article.

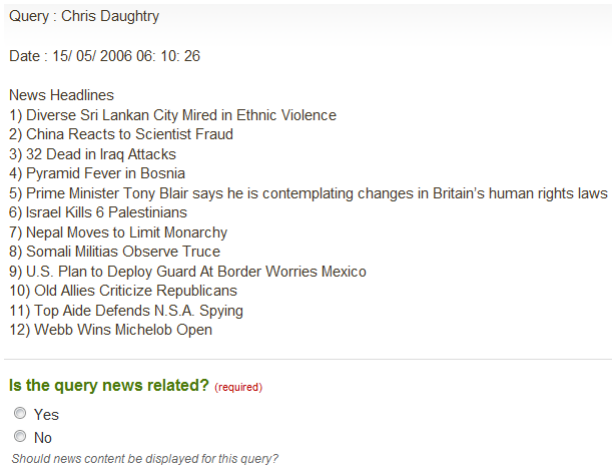

Figure 5: Examples of additional content added to the labelling component of the *Headline_Inline* interface.

labelling quality. In particular, we identify two sources of news-related content that our workers might find informative, i.e. news articles (headlines and/or news summaries) and Web search results.

Initially, to supplement our interface with news article content, we use a collection of news headlines and article summaries from the time of the query log, as provided by the New York Times. An example headline and associated news story summary is shown in Figure 4. It is likely that the more relevant news-related information our workers have available the better they can judge each query. For instance, providing the summary in addition to the headline might enable our workers to identify queries that do not contain terms in the headline but were related to the news story. Consider, for example, the news summary shown in Figure 4. A worker might be able to label the query 'baghdad airport' as news-related having read the news summary but would be unlikely do so from the headline alone.

However, we suspect that the amount of useful information we can provide our workers with is limited, as overloading the interface with surplus irrelevant content will cause our workers to either ignore it or reject the job in the first place. To test this, we experiment using three alternative interfaces, varying the level of supporting information provided.

In the first interface, we include 12 news headlines, like the one shown in Figure 4 within the job instructions. This provides the worker with some news story context from the time of the queries being labelled. The resulting interface is denoted *Headline*.

For our second interface, we test the effect that the level of detail of information provided to our worker has on labelling quality. In particular, in addition to the headlines used in the above interface, for each of those headlines, we also included its associated news story summary. This interface we denote *Headline+Summary*.

However, we are concerned that our workers might not read the headlines provided in the instructions. So, for our third interface, instead of including the headlines within the instructions, we moved them into the labelling interface for each query. An example of this interface is shown in Figure 5. Notably, the headlines will always be onscreen at the time the worker makes each judgement, hence making it more likely that our workers will refer to these headlines when judging. We denote this interface *Headline_Inline*.

As an alternative to providing news headlines, we examine the usefulness of incorporating Web search results. In particular, we

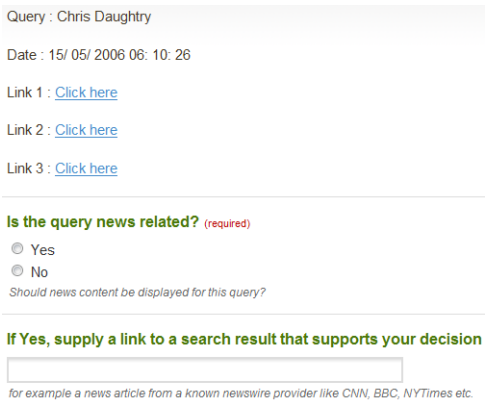

Figure 6: Examples of additional content added to the labelling component of the *Link-Supported* interface.

propose to provide automatically generated search links to the three top search engines, i.e. Bing, Google and Yahoo!, for each query and time. Each link initiates a search by the associated search engine. The query for each search constitutes the original user query and the date that the query was submitted, e.g. '4th May 2006 Moussaoui Verdict'. In this way, we hope that the Web search results will be similar to those that might have been returned for the original query at the time it was made. The modified labelling interface is shown in Figure 6. Note that we do not make the target of the search link immediately clear by obscuring it behind a 'click here' anchor. We do this on the suspicion that, should we show our workers the search engines, then only the worker's favoured engine would be clicked. This interface is referred to as *Link-Supported*.

Interestingly, the *Link-Supported* interface also provided us with the opportunity to gain additional feedback from our workers. In particular, as we are requesting our workers to inform their decision based on Web search results, we can also gain useful feedback from the search result that the worker based their decision on (if any). Hence, for this interface we also ask our worker to provide the URL of the appropriate Web result. We hypothesise that this can later be used to validate the judgements through examination of the supporting URLs. Indeed, we investigate this later in Section 6.5.

5. EXPERIMENTAL SETUP

In this section, we detail the experimental setup for our crowdsourcing experiments. In particular, Section 5.1 defines the specific research questions that we address in our experiments, while in Section 5.2, we describe the settings we used for our crowdsourcing jobs. Lastly, Section 5.3 describes the evaluation measures used.

5.1 Research Questions

In Section 6, we investigate the following research questions:

1. How well do users agree on news query classification labels using our basic interface? (Section 6.1)
2. What effect does online validation of the worker labels have on overall label quality? (Section 6.2)
3. Are any of the proposed alternative interfaces effective at countering our workers' lack of information with regard to the main news-stories of the time? (Section 6.3)

Interface	Time per query	Total cost
Basic	5s	$1.30
Headline	10s	$4.39
HeadlineInline	10s	$4.59
Summary	15s	$5.56
Link-Supported	22s	$8.78

Table 4: Mturk estimated worker time to spend per query and the cost incurred over the 100 query testset.

4. Is the resulting news query classification dataset of good quality? (Section 6.4)

5. Is crowdsourcing useful for post labelling quality assurance? (Section 6.5)

5.2 Crowdsourcing Settings

When submitting a job, there are multiple variables which determine how it is handled by MTurk. Firstly, the crowdsourcer must specify the amount paid to each worker on a per job basis. In our case, we paid workers based on the estimated amount of time it would take to judge each query at a rate of $2 per hour. The estimated time per query and the resulting cost paid on a per-interface basis for the testset queries are shown in Table 4. Secondly, the crowdsourcer has the option to limit the worker pool based on geographical location. Since the MSN query log is predominantly American, we limited our worker pool to the USA only. To control validation, we set the validation cutoff, i.e. the level at which a worker is ejected from the evaluation, to 70% (should they get more than 30% of the validation queries wrong they are ejected from the evaluation without remuneration). Lastly, we set the level of redundancy for judging our queries at three, whereby each query will be judged by three unique workers. Note that we do not use the four redundant judgements recommeded by Snow *et al*. [21], as we wish to take a majority vote to determine the final label.

5.3 Measures

In this work, we measure the quality of our crowdsourced labels in two distinct ways. Firstly, we measure our worker agreement, i.e. how often our workers assigned the same label to each individual query. Our intuition is that should our workers often agree about which queries are news-related, then our confidence in the labels produced increases. In particular, we employ two well-known measures for evaluating agreement in user evaluations: Free-Marginal Multirater Kappa [20], denoted κ_{free}; and Fleiss Multirater Kappa [11], denoted κ_{fleiss}. Notably, the two Kappa agreement measures differ in the way each calculates the probability of agreement occurring by chance. Free-Marginal Multirater Kappa assumes that the chance of selecting a class is equal to one over the number of classes, i.e. 50%, while Fleiss Multirater Kappa takes into account the relative size of the classes, i.e. in our case that queries are more likely to belong to the non-news-related class. Indeed, recall that at best only 10% of queries might be news-related [5].

The second manner in which we evaluate the quality of our crowdsourced labels is against labels manually generated by the primary author of this paper. Our intuition is that the labels we generate will have a higher probability of being correct due to a longer time spent, in addition to news article access from the time of the queries. In short, we use the labels generated as an 'expert' ground-truth. From this, we report standard classification measures, precision and recall. To provide a combined measure, we also report overall classification accuracy.

It is worth noting that labelling was based upon knowledge of the mainstream news stories of the time. However, there exist queries which refer to news events not reported in mainstream news, e.g. a local football match, these are sometimes known as to as 'tail events' [19]. However, we believe that such tail events may be more difficult for assessors, leading to disagreement between our ground truth and the workers.

6. EXPERIMENTS AND RESULTS

In this section, we study the suitability of crowdsourcing as a means to generate labels for our news query classification dataset. In particular, Section 6.1 investigates the initial level of agreement observed between workers when labelling queries as news-related or not. Section 6.2 examines the effect of online validation of worker labelling quality. In Section 6.3, we investigate the alternate interfaces previously described in terms of labelling quality. In Section 6.4, we evaluate the quality in terms of agreement and accuracy on our final news query classification dataset, whilst we perform an additional meta agreement study into our dataset's quality in Section 6.5.

6.1 Worker Agreement

Following our methodology described earlier in Section 4, we begin by examining the workers' labelling agreement on the smaller testset. In particular, we wish to establish that crowdsourcing is indeed suitable for labelling queries as news-related or not, before committing to a labelling job over the fullset comprised of over 1000 queries. To this end, we evaluate worker agreement when labelling the queries of the testset using the *Basic* interface described in Section 4.3. A high level of agreement indicates that the resulting labels are of good quality, hence the labelling method is suitable. Importantly, there is no de facto standard for defining acceptable or significant levels of agreement. However, Landis and Koch [15] state that Kappa values over 0.61 indicate substantial levels of agreement, while values over 0.81 represent almost perfect agreement.

Table 5 reports two Kappa agreement measures, κ_{free} and κ_{fleiss} for labelling the testset queries with our *Basic* interface. As can be observed from the table, labelling using our basic interface provides a low level of agreement - $\kappa_{fleiss} = 0.2647$ - when worker pooling is restricted to the USA only[2]. This level of agreement is markedly lower than that which we would judge acceptable, hence in the following sections we examine methods to improve labelling quality in terms of agreement.

6.2 Importance of Validation

Previous work into crowdsourcing with MTurk has highlighted the importance of result validation [14], i.e the checking of worker input against a ground truth to prevent random or malicious results. Hence, we examine the importance of validation for our news query labelling job. In particular, the 9 validation queries are interspersed with the 91 queries of the testset. Workers are examined in terms of the percentage of the validation queries that they correctly label.

The first two rows of Table 5 report labelling agreement for the testset queries using our basic interface. We observe that both agreement and accuracy markedly improved over our earlier baseline run which did not validate worker input. Not only does this confirm the need to validate worker input, but it highlights the scale of the issue. In particular, 32% of queries judged during this job were rejected based on our validation. Such a high level of rejected judgements might be attributed to a lack of information provided to our workers, as no additional news content is provided at this stage. However, a large proportion of these judgements were also made

[2]Note that we also examined pooling workers from all countries, however, performance was markedly lower, i.e. $\kappa_{fleiss} = 0.0395$. This likely results from the predominantly American centric nature of the MSN query log, in addition to possible malicious worker spamming from other countries.

Interface	Query Set	Validation	Precision	Recall	Accuracy	κ_{free}	κ_{fleiss}
Basic	testset	✖	0.5	0.5714	0.9263	0.5833	0.2647
Basic	testset	✔	**1.0**	**0.5714**	**0.9681**	0.7373	0.3525
Headline	testset	✔	0.5	0.5714	0.9263	0.8	0.5148
Headline_Inline	testset	✔	**1.0**	0.2857	0.9474	0.7866	0.3018
Headline+Summary	testset	✔	**1.0**	**0.5714**	**0.9684**	0.83	0.5327
Link-Supported	testset	✔	**1.0**	**0.5714**	**0.9684**	**0.8367**	**0.5341**
Link-Supported	**fullset**	✔	**0.6761**	**1.0**	**0.9748**	**0.8358**	**0.7677**

Table 5: Quality measures for news query classification on the approximately 100 query testset with varying interfaces, in addition to the over 1000 query fullset using the *Link-Supported* interface.

within the first minute of the job, at a rate far exceeding a human's labelling ability. This indicates that there are auto-completing bots attempting jobs on MTurk, which need to be identified and filtered.

6.3 Supplementing Worker Information with News Content

Recall that we identified the worker's lack of knowledge about the major newsworthy stories of the time around which the queries were submitted as an important limitation on our workers. Moreover, we proposed four alternative interfaces, which provide workers with addition news-related information in an attempt to address this. In this section, we evaluate the quality of the labels produced using these alternative interfaces. Our intuition is that the use of this additional evidence will allow our workers to make informed decisions about the queries, thus improving the final label quality measured in terms of worker agreement.

Once again, Table 5 presents our workers' labelling performance in terms of agreement for the testset queries using the four interfaces previously described in Section 4.3. As can be observed from the table, providing our workers with either news headlines (*Headline*), news summaries (*Headline+Summary*) or a ranking of search results (*Link-Supported*) from the time of the query, we can markedly increase worker agreement. However, we also noted that for the *Headline_Inline* interface, placing news headlines prominently with each query instead of within the instructions decreased agreement between our workers. We suspect that this results from some workers only matching the query against the provided headlines, causing them to miss other newsworthy queries for which a headline was not provided. Indeed, the low recall against our manually judged query set confirms this.

Overall, we observe that our *Link-Supported* interface obtained the highest worker agreement, i.e. $\kappa_{free} = 0.8367$ and $\kappa_{fleiss} = 0.5341$. Furthermore, the high overall labelling accuracy of 0.9684 obtained for this interface, in addition to the strong level of agreement shown, attests the suitability of crowdsourcing for labelling queries as news-related or not. [21] noted that in crowdsourcing there is a trade-off between the number of workers assigned to each job and the resulting quality. We note that the *Link-Supported* job cost over 6 times that of *Basic*. However, although we could conceivably have 6 times as many workers perform labelling using *Basic* for the same cost as *Link-Supported*, we would not expect accuracy with *Basic* to markedly increase as the information available to each worker remains constant.

6.4 Evaluating our News Query Classification Dataset

Having observed that crowdsourcing appears to be suitable for generating news query classification labels upon the 100 testset queries, we now build the our full news query classification dataset comprised of the 1206 queries from the fullset in addition to the 61 queries in its associated validation set and evaluate the quality of the resulting labels. The last row in Table 5 shows label quality for the fullset using our *Link-Supported* interface with validation. We report precision, recall and accuracy over our manually judged labels, in addition to the agreement measures κ_{free} and κ_{fleiss}.

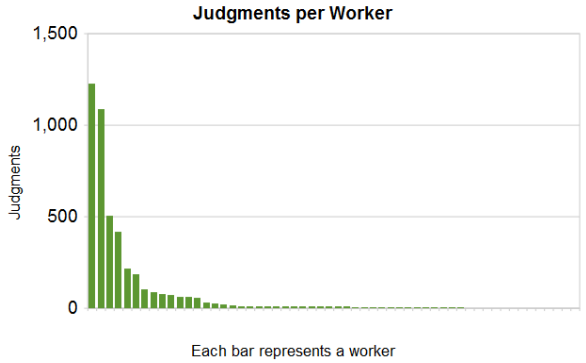

Figure 7: Number of judgements made by each worker when labelling the fullset queries using the Link-Supported interface.

As can be observed from Table 5, over the larger query set, the accuracy of our crowdsourced labels is high, i.e. over 90%. Indeed, all of the queries judged by the author as being news-related were also labelled as such by the workers (100% recall). In comparison to our results on the testset, this recall increase indicates that the testset queries were more dificult. Reported precision on the other-hand is markedly lower than on the testset. This is to be expected, as was noted earlier, there likely exist news-related queries that refer to tail events. For these queries, the workers may be more likely to disagree with our ground truth labels. Additionally, agreement between our workers is also high, indeed higher than shown on the smaller testset. This may have resulted from workers becoming more proficient at the job as they judge more queries and moreover pass a larger number of validation queries. Indeed, Figure 7 shows the number of judgements per worker. We observe that the majority (over 70%) of our judgements were completed by 3 workers, completing between 500 and 1200 queries each.

6.5 Crowdsourcing Additional Agreement

In the previous experiments, we used crowdsourcing as a means to label queries as news-related or not for a specific time. However, we also hypothesised that crowdsourcing could also be used as a quality assurance tool. In this section, we further evaluate the quality of our news query classification dataset by crowdsourcing additional agreement labels.

Intuitively, we could return the labels produced for our news query classification dataset to MTurk, asking a second group of workers to validate the quality of those labels under the same conditions as the original job. However, this is similar in effect to increasing the redundancy for the original job. Instead, we propose to leverage the URLs that each worker was asked to provide under the Link-Supported interface when labelling a query as news-related.

In particular, we ask our workers to validate each of the queries in the fullset which were judged news-related by our original workers, based upon the content of the linked Web page by that URL. Should the URL support the original worker's label, then this increases confidence that the label is correct. Hence, the labelling quality of the query subset judged news-related can be determined by the proportion of queries within that set that are supported by their linked Web page. In particular, we ask each worker to label

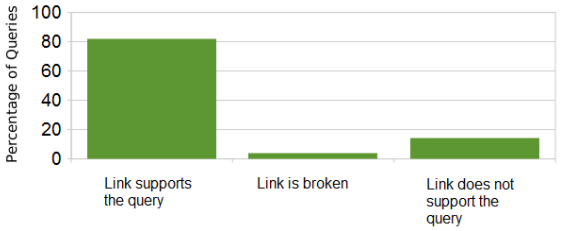

Figure 8: Percentage of URL's that were labelled as either broken, supporting the news-relatedness of the query or not supporting the news-relatedness of the query during the quality assurance job.

the URL as either being broken, supporting the news-relatedness of the query or not supporting the news-relatedness of the query. Notably, this can be seen as a form of meta agreement, as we measure how well the new workers agree with the original workers based on evidence provided by those original workers.

Figure 8 reports the percentage of URLs that were placed in each class. We observe that the vast majority (82%) of URLs were judged as supporting the query as being news-related. This is very promising, as it shows that not only were our original workers effective at finding news-related queries, but they were making informed decisions based upon the Web search results provided. Moreover, these results further support the claim that our resulting news query classification dataset is of good quality.

Overall, we conclude that crowdsourcing is indeed suitable for labelling queries as news-related or not, as attested by the high levels of agreement between our workers, the high labelling accuracy upon manually judged labels and in terms of meta-agreement with a second set of workers. Moreover, from the experience that we gained from the creation of the resulting dataset, we suggest the following best practices when crowdsourcing.

1. **Be aware of geographical differences**: Worker performance varies from location to location. Consider from where your workers will be best able to complete your task.
2. **Online worker validation is paramount**: You need to evaluate worker performance to detect bots and poor quality workers early within the evaluation.
3. **Provide workers with as much information as possible**: Workers are not experts at most jobs. Overcome this by providing additional relevant information or external resources that workers can quickly and easily refer to.
4. **Workers can learn**: Workers are real people and can learn to become better at a job over time. This is true not only over a single large job, but equally over all the jobs submitted. Indeed, we observed that there was a notable worker overlap between our runs using the testset.
5. **Consider meta-agreement for additional validation**: Try to collect additional feedback from the workers. There may be too much data to be evaluated by hand, but crowdsourcing can be used for evaluation as well as data creation.

7. CONCLUSION

In this paper, we investigated building a dataset for the news query classification problem. In particular, we proposed the crowdsourcing of labels from Amazon's Mechanical Turk as a faster and cheaper method than relying on specialist annotators. Using queries sampled from a real search engine query log we experimented to determine the effect of worker validation, in addition to methods for mitigating the lack of information about the news stories from the time of the queries on the part of our workers.

We have shown that online validation of workers is paramount, and that by supplying workers with Web search rankings or related news article content for the query, we could dramatically increase labelling quality. Moreover, we have shown the suitability of crowdsourcing for the news query classification problem as well as its application in practice. Indeed, we created a new dataset of sufficient quality, both in terms of inter-worker agreement and also against a set of manually judged queries. Furthermore, we have also examined the resulting dataset using a novel application of crowdsourcing as a quality assurance tool, showing that crowdsourcing can be useful as a means to calculate addition agreement between users and also confirming the high level of quality shown by our dataset. Lastly, based upon the experience we have gained from this study, we have provided a set of best practices to help future researchers design robust crowdsourcing experiments.

8. REFERENCES

[1] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.

[2] O. Alonso. Crowdsourcing for relevance evaluation. *Tutorial at ECIR'10*, Milton Keynes, UK.

[3] J. Arguello, F. Diaz, J. Callan, and J. F. Crespo. Sources of evidence for vertical selection. In *Proceedings of SIGIR'09*, Boston, MA, USA.

[4] J. Atwood. Is Amazon's Mechanical Turk a failure?, 2007. `http://www.codinghorror.com/blog/2007/04/is-amazons-mechanical-turk-a-failure.html`, accessed on 02/06/2010.

[5] J. Bar-Ilan, Z. Zhu, and M. Levene. Topic-specific analysis of search queries. In *Proceedings of WSCD'09*, Barcelona, Spain.

[6] C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of EMNLP'09*, Singapore.

[7] N. Craswell, R. Jones, G. Dupret and E. Viegas. In *Proceedings of WSCD'09*, Barcelona, Spain.

[8] F. Diaz. Integration of news content into Web results. In *Proceedings of WSDM'09*, Barcelona, Spain.

[9] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3), 2007.

[10] J. Downs, M. Holbrook, S. Sheng, and L. Cranor. Are your participants gaming the system? Screening Mechanical Turk workers. In *Proceedings of CHI'10*, Atlanta, GA, USA.

[11] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[12] S. Hansell. Google keeps tweaking its search engine, 2008. `http://www.nytimes.com/2007/06/03/business/yourmoney/03google.html`, accessed on 08/06/2010.

[13] J. Howe. The rise of Crowdsourcing, 2006. `http://www.wired.com/wired/archive/14.06/crowds.html`, accessed on 02/06/2010.

[14] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceeding of CHI'08*, Florence, Italy.

[15] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[16] R. McCreadie, C. Macdonald, and I. Ounis. Insights on the horizons of news search. In *Proceedings of SSM'10*, New York, NY, USA.

[17] H. C. Ozmutlu, A. Spink, and S. Ozmutlu. Analysis of large data logs: an application of Poisson sampling on Excite Web queries. *Information Processing & Management*, 38(4):473–490, 2002.

[18] S. Ozmutlu, A. Spink, and H. C. Ozmutlu. A day in the life of web searching: an exploratory study. *Information Processing & Management*, 40(2):319–345, 2004.

[19] J. Pedersen. Keynote speech. In *Proceedings of SSM'10*, New York, NY, USA.

[20] J. J. Randolph. Free-marginal Multirater Kappa (multirater K free): an alternative to Fleiss' Fixed-marginal Multirater Kappa. In *Proceedings of JULIS'05*, Joensuu, Finland.

[21] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP'08*, Honolulu, HI, USA.

[22] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

[23] E. M. Voorhees. TREC: Continuing information retrieval's tradition of experimentation. *Commun. ACM*, 50(11):51–54, 2007.

[24] E. M. Voorhees and D. M. Tice. Building a question answering test collection. In *Proceedings of SIGIR'00*, Athens, Greece.