Filtering and Ranking for Social Media Monitoring

Arlind Kopliku* — **Paul Thomas**** — **Stephen Wan***** — **Cécile Paris*****

* *IRIT, Paul Sabatier University, Toulouse, France*
arlindkopliku@yahoo.com
** *CSIRO, Canberra, Australia*
paul.thomas@csiro.au
*** *CSIRO, Sydney, Australia*
{stephen.wan,cecile.paris}@csiro.au

ABSTRACT. *Social media monitoring is fast becoming a staple activity for public relations and communications staff, who have a growing mandate to track mentions of organisational entities, projects, or products in social media. However, this task is not trivial because: 1) the mentions may be found across a variety of social media and 2) the keywords used for detecting mentions are often ambiguous. We address here classification and ranking of search results collected from a range of social media. Experiments with a labelled corpus demonstrate system effectiveness in filtering out non-relevant results and gains in efficiency by improving result ranking, in both regards outperforming commonly-used alternatives.*

RÉSUMÉ. *La veille sur les média sociaux est aujourd'hui une activité commune parmi le personnel de relations publiques et de communication, qui doit surveiller autour d'entités, projets et produits. Cette tâche n'est pas facile car 1) l'information se trouve sur plusieurs média différents et 2) les mots clés nécessaires sont souvent ambigus. Ce papier adresse la classification et le trie de résultats provenants de média sociaux. Les évaluations avec un corpus expérimental prouve l'éfficassité de la classification et aussi du gain en performance avec le trie. Les deux approches dépassent les techniques conventionnelle.*

KEYWORDS: *social media, classification, aggregated search*

MOTS-CLÉS : *media sociaux, classification, recherche agrégée*

1. Introduction

Social media such as microblogs, online forums, and blogs are of interest to a growing number of organisations as they grow their online engagement with the public, track trends, and strive to understand people's opinions of their services, policies, and products (Lakkaraju *et al.,* 2011, Paris *et al.,* 2011). In the government sector in particular, one of the aims of social media monitoring is to improve government services and better serve the community: and this is the context in which this work is situated. In this scenario, social media monitors examine how information about services is disseminated; ensure that information provided by the community is correct; check that that citizens applying for services are appropriately catered for; identify potential problems, such as long queues or service failures; and find opportunities for further informing citizens about available services. This requires communication staff to browse through a large amount of social media content related to specific government departments, services, or policy, and engage with online communities when appropriate and beneficial for clarifying information about services (Paris *et al.,* 2011).

A great deal of social media is available via search engines and public APIs, but the amount of data presents a problem as much as an opportunity: with the many millions of online groups and media reports of Facebook and Twitter exceeding hundreds of millions of users, [1] it is easy to become swamped in data. Thus, computational support for social media monitors—such as online filters, summarisers, and visualisations—can potentially save time, allowing already time-poor staff to priortise their actions more effectively. Commercial products presently available for social media search include Google Alerts, Social Mention, Twitter, WhosTalkin, Hootsuite, and FriendFeed.These tools generally filter results simply by keyword, without any means for the user to provide further context: monitors must manually sift through each mention. Additionally, some engines are specific to a particular medium or site; some others provide cross-media search, but group results according to source and do not attempt to integrate them. As well as search tools, specialist monitoring tools such as Radian 6, Alterian, and BuzzNumbers can help analyse and archive content. However, these tools do not aggregate arbitrary user-defined data sources.

This provides the application context for this work: how does one combine existing search technologies with text analysis tools to facilitate a social media monitoring task, particularly one that focuses on improving services? Here we examine the first step, further refining the aggregated results from the publically available search tools already in use by social media monitors. That is, we do not attempt to reimplement search mechanisms which generally require significant computing infrastructure. Rather we focus on how results are presented in a social media monitoring interface, using methods for ranking and filtering for aggregated search.

---

1. For an example of a report on registered Twitter users, see e.g. `http://mashable.com/2010/09/03/twitter-registered-users-2`; for Facebook users, `http://blog.facebook.com/blog.php?post=106860717130`

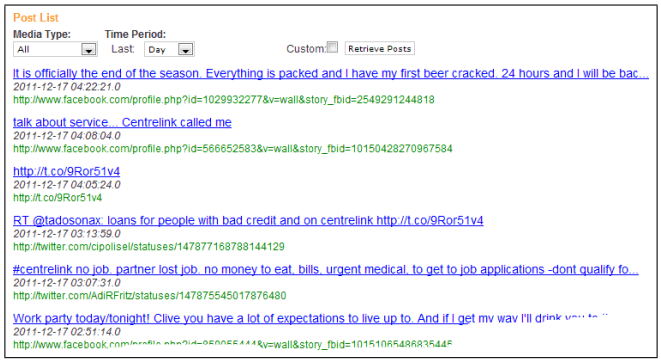

Figure 1. *"Post list" interface for social media monitors, presenting posts from several sources. We want to show a representative set of posts which are relevant to the monitored topic, without irrelevant posts.*

Figure 1 shows a monitoring tool we have developed for government applications (Paris *et al.,* 2011). This view presents social media messages ("posts") from several different sources including web forums, blogs, Facebook, and Twitter. In our user requirements and task analysis (Paris *et al.,* 2011), we identified that a list-based presentation provides a mechanism for communications staff to ensure that the monitoring activity is systematic. Social media monitors typically review this list to understand public opinion and identify followup actions. Consequently, we designed a list-based interface enhanced with meta-data information and the results of text analysis methods such as thread summarisation that would enable better use of staff time.

Ideally, this interface would show a representative set of posts relevant to any topic, without irrelevant posts. However despite several good public APIs, the volume of data makes retrieval for social media monitoring a non-trivial task. Queries can be broad, such as "child care" or "youth allowance" for governments, which makes many of the retrieved results not relevant or too general. In general, relevant content concerns a certain context including a domain (government, art, sport, ...) and geographical location (country, region, ...) and possibly an individual public service, product, or policy. Moreover, monitoring is mostly done with respect to recent posts, so the freshness of information is important.

In this paper, we propose adapting approaches from aggregated search to the social media monitoring: retrieving information of different types, each from its own source, and aggregating for monitors with the aim of increasing both precision and coverage. This work is different to existing aggregated search applications since (i) we emphasise the importance of context and time; (ii) we do not assume a primary source of data; and (iii) we rank results individually, not in blocks. We structure the remainder of this paper as follows. We provide our approach in Section 2; this contrasts with related work, detailed in Section 3. Our experimental setup and results are discussed in Sections 4–5. We outline future work in Section 6 and conclude in Section 7.

2. Our approach: Uniform ranking

The simplest way to retrieve posts for monitoring is to filter by keyword, on post content or possibly on tags or other metadata; and to sort by a generally-available feature such as the time of publication. This approach is taken by popular monitoring tools such as Social Mention and we considered this in our experiments. By contrast, we adapt techniques from aggregated search. We query sources (such as Twitter and Social Mention) through their public APIs [2], which we cannot influence. Our processing instead involves two parts of the aggregated search pipeline: query dispatch and result aggregation.

Query dispatch: Queries for monitoring can be broad, and the terms used can be ambiguous; this is especially true for government programmes which have similar names in different jurisdictions. (The phrases "child support", "age pension", or "unemployment", for example, are found across the English-speaking world.) Since our search engines are black-box tools which cannot easily be personalised, we instead modify queries at query dispatch time. Rather than issue one broad query to each source, we propose issuing multiple queries which incorporate context-specific terms. Concretely, let $d_1, d_2, \ldots, d_m$ be some discriminative terms for a context C. A discriminative term for a context C is a term which is more likely to appear in context C than in background text. We expand the query q with discriminative terms for C to increase the chances of having relevant and contextual results. We have two options. We can expand the query with many discriminative terms, obtaining the query $\langle q\ d_1\ d_2 \ldots d_m \rangle$; but this approach is not possible because most existing APIs do not deal well with long queries. As an alternative, we propose issuing multiple queries: $\langle q \rangle$, $\langle q\ d_1 \rangle$, $\langle q\ d_2 \rangle$, $\ldots \langle q\ d_m \rangle$. We use this second approach in the experiments below, both to gather data for human annotation and to gather posts to rank.

Result aggregation: Aggregation approaches to date have been developed for web search (Arguello *et al.,* 2011, Ponnuswami *et al.,* 2011). The assumption has been that the web (i.e. traditional web search) is the primary source, while results from other verticals are useful for some queries and for increasing diversity. In our application, there is no primary source—we should treat all sources equally. Following Liu, Yan and Chen (2009), we rank items singly instead of in blocks. This is for several reasons: first, we will have little or no multimedia content (such as videos, images, or maps). Our interface represents all our results as conventional two- or three-line text summaries, so blocks are not needed for layout. Further, a monitors' time is limited and expensive. Where possible, we should show the most relevant results from across all sources: low-quality results from one source should not crowd out higher-quality results from another, as may happen with fixed-size blocks. We propose a straightforward approach. We rank individual items according to their probability of relevance, as estimated by machine-learned models. (We stress here that we are not interested in perfect relevance estimates, but in order-preserving scores which will

2. Twitter API: `https://dev.twitter.com/docs`; Social Mention API: `http://www.socialmention.com/api/`

rank the most relevant results first.) Our post-retrieval features, which can be calculated for all sources, capture characteristics of text as well as geography and time; they are described in Section 4. We also include a feature representing the type of source—for example "microblog" or "forum"—since it has been shown that the impact of each feature depends on the type of content (Arguello *et al.,* 2011). We use the output of binary classifiers, trained on a small amount of labelled data, to estimate these probabilities of relevance.

3. Related work

We draw on both social media analysis and aggregated search.

Social media analysis and retrieval: Prominent issues for social media monitoring include on content summarisation; effective user interfaces; and trend analysis (Paris *et al.,* 2011). For example, recent research in information retrieval and computational linguistics has included work on opinion mining and social media recommendation (Lakkaraju *et al.,* 2011), event detection (Petrović *et al.,* 2010), conversation modelling (Ritter *et al.,* 2010), emergency situation awareness (Vieweg *et al.,* 2010), discussion similarity (Bernstein *et al.,* 2010), and analysing networks of influence (Kwak *et al.,* 2010). While such research is relevant to the task of social media monitoring in general, each of these relies on accurate retrieval of on-topic posts: in this paper we focus on work that can help identify relevant content. Our approach is based on machine-learned models of relevance. Machine learning has been used for a number of social media tasks recently: for example, topic modelling approaches such as Latent Dirichlet Allocation (LDA) (Blei *et al.,* 2003) have been applied to social media content to group related content together (for example, see Hong and Davison (2010) or Ramage, Dumais and Liebling (2010)) and to discussion forums (for example, see Rossi and Neville (2010)), and bookmarking and tagging sites like del.icio.us (Ramage *et al.,* 2009).

Supervised learners, similar to those we use here, have been used in other text categorisation problems such as detecting email spam (Sahami *et al.,* 1998) and predicting user-assigned tags (Ramage *et al.,* 2009). These techniques are typically used post-hoc, on information already received or retrieved; however, to our knowledge this has not been applied in a social media search tool. That is, relevance has been defined entirely by the search tools and generally there is no extra automated filtering. It is this gap that this paper seeks to address. Our approach also draws on information retrieval methods such as pseudo-relevance feedback (Salton *et al.,* 1990). We adapt these methods to define relevance to the social media monitoring context, and to present only content relevant to a particular organisation and location.

Aggregated search: Aggregated search is the task of retrieving and assembling information of potentially heterogeneous type (Murdock *et al.,* 2008, Paris *et al.,* 2010), and has the potential to be useful for social media monitoring. For this to be so, we identified two ways in which existing approaches could be extended. The first exten-

sion is to include clues about the context of the monitoring task. Secondly, aggregated search methods do not typically present search results so that single results interleaved regardless of media type. However, our task analysis with social media monitors suggests that users want such a presentation. In this remainder of this section, we present an overview of aggregated search to situate our work.

Cross-vertical aggregated search is one of the most successful approaches, and is implemented in major search engines (Arguello *et al.,* 2011, Lalmas, 2011, Kopliku, 2011). This relies on multiple "vertical" search engines (image search, video search, news search, ...) as well as traditional web search. They are combined in one aggregated system that allows information seekers to access results from different sources through one interface. For web search, it has been shown that relevant results become more diverse and are thus often complementary with each other (Kopliku *et al.,* 2011, Arguello *et al.,* 2009, Sushmita *et al.,* 2009). Research issues in aggregated search include query routing (Arguello *et al.,* 2009, Kopliku, 2011) and result aggregation (Arguello *et al.,* 2011, Sushmita *et al.,* 2009, Ponnuswami *et al.,* 2011). The first problem involves mostly vertical selection: deciding which sources should be used for a given query. The second problem, addressed by this paper, concerns the way returned results should be put together.

Result aggregation has seen two main approaches: *blended* and *unblended* aggregation (Sushmita *et al.,* 2009). The unblended approach keeps results of different types in separate panels. In the blended approach, results of different types are blended in the same ranked list. The latter has taken the lead both in research and industry (Lalmas, 2011), and is the focus of this work. Within the blended approach, we distinguish two techniques. Ponnuswami et al. rank search results in blocks by vertical (e.g. three images are ranked as a block, before or after a block of three web search results) (Ponnuswami *et al.,* 2011). They train their ranker on pairwise preferences between a block of web search results and a block of vertical search results. Similarly, Arguello et al.(Arguello *et al.,* 2011) rely on pairwise preferences. In their experiments a "learning to rank" technique outperformed a technique derived from classification and a voting approach. Liu et al. define a probabilistic model that enables ranking search results from different sources (Liu *et al.,* 2009). In contrast with other approaches, they rank single items (search results) instead of blocks of search results. In contrast to Liu et al. however, we focus exclusively on social media content.

4. Experiments

To evaluate our classifiers, rankers, and features, we use a simulated setting based on a real social media monitoring task. Manual relevance assessments on pre-selected queries serve as training and testing data.

Application setting: The Australian Commonwealth government delivers "human services"—such as income support, employment assistance, pensions, and child

(a)	Study expense changes for students receiving Austudy, **ABSTUDY** and Youth allowance http://fb.me/Eyfnl4Qc
(b)	Luckily in NY state, **Medicare** will not reimburse for treatment performed by ATCs/massage therapists. My job in my state is secure!
(c)	Ok so I'm officially going to put *(user)* on **child support**

Figure 2. *Relevant and non-relevant posts. (a) is relevant: it is discussion of a government programme. (b) is not relevant: it is a result of ambiguity, and in this instance is a programme in a different jurisdiction. (c) is not relevant: it mentions a programme, but does not discuss it or offer an opinion. A simple filter will include (b) and (c).*

support—through an agency called Centrelink. [3] Centrelink is a large organisation, dealing with around one third of the Australian population in 2008–9; they are accordingly high-profile and often discussed in traditional and social media. Centrelink employs social media monitors to gauge public opinion about their services; what they like, what they dislike, and what problems people encounter when applying for services. In our simulated application, the goal is to assist monitoring social media for conversations and comments around Centrelink or any of its programmes. Monitors are assumed to use an simple list interface like that in Figure 1, and their time is limited. Ranking posts toward the top is therefore useful if they discuss, for example, pensions in Australia or conditions at Centrelink offices; on the other hand posts should be ranked lower if they discuss pensions in other countries, or mention Centrelink offices only as landmarks. Figure 2 shows examples.

Data sets: We trained and tested our model with several types of post, collected from two search engines and responding to ten different topics.

Sources: We used four types of post: news articles, which are from online news sources; microblogs, including Facebook status updates and Twitter; forums, which includes online bulletin boards; and question answering, which includes community question-answering sites such as Yahoo! Answers. Posts were retrieved by Yahoo! BOSS, [4] with a custom list of targets, and Social Mention.

Queries: Queries were chosen by Centrelink social media monitors to represent topics of ongoing interest. All were the names of government agencies or programmes: *abstudy*, *austudy*, *baby bonus*, *centrelink*, *centre link* (note whitespace), *child support*, *medicare*, *newstart*, *paid parental leave*, and *youth allowance*. In each case, as outlined in Section 2, we issued three forms of the query: unmodified, with *australia* appended, and with *centrelink* appended (except where this was redundant). For each query we collected about 30 results of each of the four types, for a total of about 120 posts. Some sources returned fewer than 30 results for some queries.

3. `http://www.centrelink.gov.au`
4. `http://developer.yahoo.com/search/boss/`

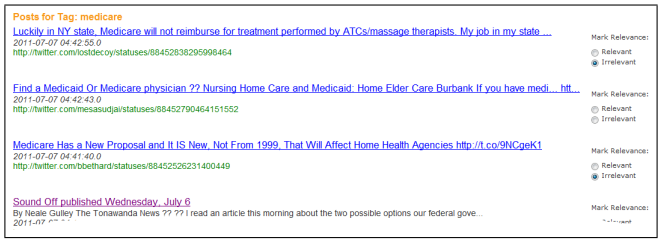

Figure 3. *Interface for annotators' relevance judgements.*

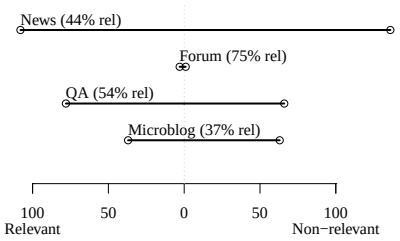

Figure 4. *Relevance labels in each type of post. Overall, 46% of posts were labelled relevant.*

Human assessments: Each post was assessed for relevance, using the tool illustrated in Figure 3. Five judges we used; two were authors of this paper and three were colleagues with knowledge of Centrelink and the social media monitoring task. Three topics were assessed by a single judge, six by two judges working independently, and one topic was assessed by three independent judges. Inter-judge agreement was very good (Krippendorff's $\alpha = 0.93$).

A document was labelled relevant if it was considered relevant by at least one assessor. Around 46% of posts were labelled relevant, varying from 37% to 75% for each source (Figure 4).

Features: Each type of post, and each source, suggests different features. For instance, results obtained through the Social Mention API provide a timestamp as well as geographical coordinates when available. Results coming from Yahoo! BOSS will also come with a timestamp, but they will not provide geographical coodinates. Blogs and news stories may include comments; tweets may include hashtags. We try to compute features uniformly across all sources. We used four groups of post-feature which are applicable across types. They are organised in four groups: *match features*, *geographical features*, *context features* and *time features*. The list of features is not meant to be exhaustive nor optimal, although the number of features is comparable to existing work which uses post-retrieval features for aggregated search (Arguello *et al.,* 2011). Query match features are common in the literature: they compute matching

scores in a uniform way across different types of results. Context and geographical features adapt the aggregated search system for a specific application. The time feature ensures recency. All our features are normalised to the interval $[0, 1]$.

Query match features: Most APIs will not disclose scores for the posts they return; at any rate, relevance scores are not generally comparable between different search engines. Instead, we process results after retrieval to compute query match scores (features). We compute match features similarly to past work (Arguello *et al.,* 2011, Joachims, 2002, Liu, 2009). We use four features which relate to the similarity of the document text with the query; we also compute the same four features with respect to the document title, for a total of eight query match features for each post. The text-similarity features are: (1) the cosine similarity between the query and the document/title representation, (2) the maximum number of query terms appearing consecutively in the document/title representation, (3) the percentage of query terms appearing in the document/title representation (4) the percentage of the document/title representation that matches query terms. Text has stopwords removed and terms stemmed with the Porter stemmer.

Geographical features: Monitoring applications are often connected to a specific geographical context. This is certainly the case with our example—there are, for example, several "child support" schemes around the world but only the Australian programme is of interest here. In the presence of geographically ambiguous queries, it is important to retrieve content related to the right area. For this reason we define three geographical features:

– Some APIs return geographical coordinates related to the results. If these coordinates fall within a relevant area—Australian territory, in our case—the *coordinates feature* is set to 1, otherwise it is 0.

– Many terms are strongly related to a geographical content rather than others. For instance, words such as "Sydney", "Melbourne", or "Canberra" are likely to appear in documents related to Australia. We manually defined a set of words which are very likely to be discriminative for Australia. The *geographical words feature* is set to 1 if and only if one or more of these terms appears in the post.

– The *URL domain feature* is set to 1 if and only if the URL of the result falls in the top-level domain of a relevant country; here, it is set for the doman "`.au`".

Context-based features: There are different ways to compute a contextual score of a term (Robertson, 1991). We compute the contextual score $cw(t, C)$ of a term t for a context C using web-based statistics that we can obtain from search engines. This approach is advantageous in that we do not have to index a huge collection of web documents and the statistics will be continuously up to date. Concretely, we try to compute the co-relation between a given term t and the context using the most discriminative terms for the context.

Let z be a highly discriminative term for the context C. In the Centrelink monitoring task, z is equal to "Centrelink". We now compute the contextual weight $cw(t, C)$

using pointwise mutual information between the terms t and z, relying on search engine hits counts (Popescu *et al.,* 2005, Turney, 2001). We have:

$$cw(t, C) = \mathrm{PMI}(z, t) = \frac{\mathrm{Hits}(z \wedge t)}{\mathrm{Hits}(z)\,\mathrm{Hits}(t)}$$

where $\mathrm{Hits}(x)$ is the number of hits reported for a web search for $\langle x \rangle$. The contextual score $cw(d, C)$ of a document d is computed as the mean per-term score:

$$cw(d, C) = \frac{\sum_{t \in d} cw(t, C)}{|d|} \qquad [1]$$

We also compute a contextual score for the title, supposing that titles have a higher impact on relevance. By analogy with Equation 1, we have

$$cw(\mathrm{title}, C) = \frac{\sum_{t \in \mathrm{title}} cw(t, C)}{|\mathrm{title}|}\ .$$

Time feature: The freshness of information can be very important in monitoring applications, including ours. We define a time score which is computed for results where the timestamp can be identified. By convention, we score 0 for all results older than a year, or results where time is not available. Results which are newer than an hour are given a score of 1. Results in between get a score which decreases logarithmically with the age of the post (Figure 5):

$$tw(t) = \begin{cases} 1 & \text{if } t < 1 \text{ hour} \\ \dfrac{\log(1 \text{ year}) - \log(t)}{\log(1 \text{ year}) - \log(1 \text{ hour})} & \text{if 1 hour} \leq t \leq 1 \text{ year} \\ 0 & \text{if } t > 1 \text{ year or no } t. \end{cases}$$

Summary: We generate a total of 14 features per document: eight features for the query match, three geographical features, two context-based features and one time feature.

4.1. *Classifiers and baselines*

We use two machine-learned models for classification: support vector machines and decision trees. We compare these with two simple filters and a naïve Bayes approach.

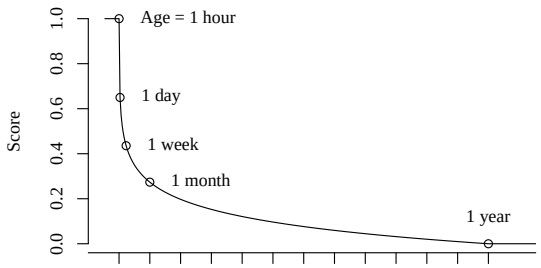

Figure 5. *Time feature, $tw(t)$, as a function of post age.*

1) Support vector machines are commonly used and typically effective for text classification problems (Joachims, 1998). We use the LIBSVM library (Chang *et al.,* 2011), encapsulated in the WEKA toolkit (Hall *et al.,* 2009), with a polynomial kernel. Further tuning is of course possible. All features from Section 4 are used as input.

2) C4.5 decision trees are less commonly used, but with a small number of features they have the advantage that the resulting model is easy to understand. Again we use a WEKA implementation of the algorithm, and all features are used.

3) As a baseline, we also consider a naïve Bayes classifier trained on the text of each post.[5] This approach is commonly used for text classification, for example by spam filters (Sahami *et al.,* 1998).

In this case match, geographical, context, and time features are not used.

4) Also as a baseline, we simply filter posts to keep only those with our query terms, and rank them according to their time of publication. (This is "simple filter" in Table 1.) This mimics the technique used by e.g. Social Mention or the Twitter search API; we expected this to have good recall but poor precision, especially since many query terms are ambiguous.

5) Finally, we filter posts to keep only those with the query term *and also* one or more of the geographical terms discussed in Section 4, again sorting by time ("geo filter"). This simulates a slightly more sophisticated use of traditional tools, emphasising precision at the expense of recall.

Figures reported for each machine-learned classifier (numbers 1 to 3) are over 10-fold cross-validation.

5. Results and discussion

Table 1 summarises our results. We consider system effectiveness as a filter (that is, a binary classifier) and as a ranker; in each case we compare to the three base-

5. In some cases, text was not available due to transient network or server problems. In these instances we used the title or snippet reported by the APIs.

	Classif. Accuracy	Ranking P@20	Ranking R@20
SVM, our features	0.82	0.75	0.07
C4.5 tree, our features	0.83	0.80	0.07
Naïve Bayes, terms	0.70	0.69	0.06
Simple filter, rank by time	0.46	0.20	0.02
Geo filter, rank by time	0.56	0.25	0.03

Table 1. *Summary of results. The C4.5 tree and support vector machine (SVM) use our features; the naïve Bayes classifier and filters use only the text of each post. Figures for C4.5, SVM, and naïve Bayes are means computed over 10 folds.*

lines described above. Since 46% of posts were labelled relevant, a majority classifier would achieve 54% accuracy but would mark all posts non-relevant.

5.1. *Classification*

The simple and geo filter are classifiers only—they simply categorise each post according to the presence of keywords. Classification is not central to our task, since we want to produce a ranked list, but it is useful for other tasks such as summarisation and trend analysis so we present overall classification accuracy here.

We gather posts by querying for topical terms, so the simple filter includes all posts; 46% of these are relevant to our context and the remaining 54% are from other jurisdictions (such as the Singaporean "baby bonus") or are non-relevant mentions (which use the key terms but are not discussing the programmes). Doing further filtering by geographic terms increases classification accuracy above that of a majority-class classifier, but accuracy is still only 56%. The naïve Bayes classifier significantly improves on the geo filter ($\chi^2 = 20.1$, df $= 1$, $p \ll 0.01$), as it can make use of varied weights instead of binary filtering and as it can make use of terms outside our short list. However, the naïve Bayes approach is in turn outperformed by both the SVM and C4.5 models ($\chi^2 = 18.8$ and 22.4 respectively, df $= 1$, $p \ll 0.01$) even though these models use only a handful of features and do not weight terms separately.

5.2. *Ranking*

Ranking, rather than classification, is central to our application. A ranking of posts can easily be induced from the naïve Bayes, SVM, and C4.5 models: we simply rank posts by $\Pr(\text{relevant}|\text{post})$, the estimated probability that a post is relevant. We compare this with output from the two filters, ordered by post date. We report both

precision and recall at twenty posts, which roughly corresponds to the number of posts a media monitor can see onscreen at once.

Ranking filtered results by time is crude but is currently provided by tools such as Social Mention and Twitter. It performs poorly in our experiments with a precision at twenty posts of only 0.20—so only four of the most recent twenty posts are relevant. Recall is correspondingly low at 0.02. Geo filtering, as expected, raises the precision slightly to 0.25 (five posts out of twenty). Recall at twenty is higher here too: since both filters return a large number of results, precision and recall at early ranks are directly proportional. Given longer lists, this would not be the case. The geo filter will only ever retrieve 19% of the relevant posts in our pool (that is, 81% of relevant posts do not have any of our geographic terms). Given poor results from a standard tool, it is tempting to disambiguate by using queries like *medicare australia* or *baby bonus singapore*; but it appears this is a poor strategy.

The machine-learned strategies again outperform the simple filters. On average 16 of the top 20 posts are relevant if ranked by a C4.5 tree; 15 of 20 are relevant if ranked by an SVM. Both are better than the naïve Bayes classifier, the C4.5 tree significantly so (Welch's $t = 2.6$, df $= 17.2$, $p < 0.01$). The performance of the the naïve Bayes classifier, which performs significantly bettern than chance, suggests that the task is amenable to an automated approach. However, the problem is not trivial with a 30% error margin, suggesting that it is worthwhile investigating alternative feature sets (in addition to terms) and classification algorithms. Indeed, this error margin is reduced by approximately 50%, to 75–80% precision, if we use SVM or C4.5 classifiers. The increased precision at early ranks does not come at the cost of lower recall further down the list. The SVM would, at the limit, return 81% of relevant results; the C4.5 tree would return 84%.

Since monitors tend to spend only 20–30 minutes in a session, early precision is paramount and the figures of 75–80% here are a substantial advance on the simple strategies currently used. In future work, we intend to study what effect this might have on the task prioritisation of the social media monitors.

Removing the media type, time, the geographic features, or the context features individually does not materially effect accuracy, precision, or recall of the C4.5 model, which suggests that several features here carry the same information. Removing all these features, however, and relying only on query match features does degrade performance: accuracy drops a little to 0.79, but P@20 and R@20 each drop a relative 9%. Using only cosine similarity, not our other match features, sees a drop to 64% accuracy, and a relative 35% drop in P@20 and R@20. Although ad-hoc, the features in Section 4 are clearly useful.

6. Future work

The results presented in this paper, though promising, were obtained in an experimental setting. In future work, we intend to integrate the proposed methods into our

social media monitoring prototype. This would allow us to perform extrinsic evaluation with users to complement the results we present here. Integration with a live tool would allow us to collect relevance judgements in a realistic setting, which could be used for further experimentation as well as to further tune our models. This would also allow reexamining our prior task analysis: the ability to quickly find relevant posts may change the way social media monitors prioritise their tasks. We also intend to refine our current features and investigate others, such as those available through topic modelling. Analyses such as LDA may provide a means to explicitly model the semantic similarity of different words, such as synonyms. Additionally, the data, by the very nature of social media, is streaming and we intend to examine update mechanisms to our language models to cater for this.

Finally, although we described a mechanism for ranking social media content so that single posts of different types are interleaved, it may be desirable to re-use this method to rank over blocks of content. In our paper, the block size is currently of a single post, however this may be parameterised—for example, to capture entire threads on forums or Twitter and rank them as a single object.

7. Conclusions

Social media monitoring is an increasingly common part of government and commercial practice. Applications which support monitoring staff—for example summarisers, trackers, or sentiment analysis—rely on accurate retrieval and ranking of topical posts. To our knowledge, the system described here is the first application of techniques from aggregated search to problems of social media monitoring. Simple retrieval and ranking techniques, as used for example by Twitter, are ineffective since keywords may be used in a number of contexts. Simple keyword filtering includes too many non-relevant posts, and while adding additional contextual terms—such as our geographical words—is an obvious refinement, it is too restrictive for our application.

Further, relevant results in our application may come from any number of media. We want to present these in a single interface, without favouring any one site; and we would prefer to show all relevant posts, without block-based ranking forcing posts from one media to crowd out those from another. Blended ranking, with post-level scoring and features in common to all media, offers a useful approach. Using either support vector machines or C4.5 decision trees, common learners and ad-hoc features work well; without sophisticated tuning or feature selection it is possible to achieve early precision of 0.80 and classification accuracy of 0.83. Compared with either filtering and simple ranking, as used at present, or with naïve Bayes classifiers, this is a substantial improvement for the government application and for the time spent by monitors. The features used here are useful for our monitoring application,

Integrating this ranking technique into the interface of Figure 1 has the potential to save staff time on common monitoring jobs. Such improvements of precision and

recall at early ranks will also improve downstream processing such as summarisation, trend analysis, and visualisations.

8. References

Arguello J., Diaz F., Callan J., « Learning to Aggregate Vertical Results into Web Search Results », *Proc. Conference on Information and Knowledge Management*, p. 201-210, 2011.

Arguello J., Diaz F., Callan J., Crespo J.-F., « Sources of evidence for vertical selection », *Proc. International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 315-322, 2009.

Bernstein M. S., Suh B., Hong L., Chen J., Kairam S., Chi E. H., « Eddi: interactive topic-based browsing of social status streams », *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, ACM, New York, NY, USA, p. 303-312, 2010.

Blei D. M., Ng A. Y., Jordan M. I., « Latent Dirichlet allocation », *Journal of Machine Learning Research*, vol. 3, p. 993-1022, March, 2003.

Chang C.-C., Lin C.-J., « LIBSVM: A library for support vector machines », *ACM Transactions on Intelligent Systems and Technology*, vol. 2, n° 3, p. 27:1-27:27, May, 2011.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H., « The WEKA Data Mining Software: An Update », *SIGKDD Explorations*, 2009.

Hong L., Davison B. D., « Empirical Study of Topic Modeling in Twitter », *Proc. Workshop on Social Media Analytics*, Washington, DC, July, 2010.

Joachims T., « Text Categorization with Support Vector Machines: Learning with Many Relevant Features », *in* , C. Nédellec, , C. Rouveirol (eds), *Proc. European Conference on Machine Learning*, p. 137-142, 1998.

Joachims T., « Optimizing search engines using clickthrough data », *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 133-142, 2002.

Kopliku A., Approaches to implement and evaluate aggregated search, Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre, 2011.

Kopliku A., Damak F., Pinel-Sauvagnat K., Boughanem M., « Interest and Evaluation of Aggregated Search (regular paper) », *IEEE/WIC/ACM International Conference on Web Intelligence, Lyon, 22/08/2011-27/08/2011*, ACM, http://www.acm.org/, p. 154-161, 2011.

Kwak H., Lee C., Park H., Moon S., « What is Twitter, a Social Network or a News Media? », *Proc. International Conference on World Wide Web*, p. 591-600, 2010.

Lakkaraju H., Ajmera J., « Attention prediction on social media brand pages », *Proc. ACM International Conference on Information and Knowledge Management*, p. 2157-2160, 2011.

Lalmas M., « Aggregated search », *in* , M. Melucci, , R. Baeza-Yates (eds), *Advanced Topics on Information Retrieval*, Springer, 2011.

Liu N., Yan J., Chen Z., « A probabilistic model based approach for blended search », *Proc. International Conference on World Wide Web*, p. 1075-1076, 2009.

Liu T.-Y., « Learning to Rank for Information Retrieval », *Foundations and Trends in Information Retrieval*, vol. 3, p. 225-331, March, 2009.

Murdock V., Lalmas M., « Workshop on aggregated search », *SIGIR Forum*, vol. 42, n° 2, p. 80-83, 2008.

Paris C., Wan S., « Listening to the community: social media monitoring tasks for improving government services », *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, ACM, New York, NY, USA, p. 2095-2100, 2011.

Paris C., Wan S., Thomas P., « Focused and aggregated search: a perspective from natural language generation », *Information Retrieval*, vol. 13, n° 5, p. 434-459, 2010.

Petrović S., Osborne M., Lavrenko V., « Streaming First Story Detection with application to Twitter », *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics*, p. 181-189, June, 2010.

Ponnuswami A. K., Pattabiraman K., Wu Q., Gilad-Bachrach R., Kanungo T., « On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals », *Proc. ACM International Conference on Web Search and Data Mining*, WSDM '11, p. 715-724, 2011.

Popescu A.-M., Etzioni O., « Extracting product features and opinions from reviews », *Proc. Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, p. 339-346, 2005.

Ramage D., Dumais S., Liebling D., « Characterizing Microblogs with Topic Models », *Proc. International AAAI Conference on Weblogs and Social Media*, p. 130-137, 2010.

Ramage D., Hall D., Nallapati R., Manning C. D., « Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora », *Proc. Conference on Empirical Methods in Natural Language Processing*, p. 248-256, August, 2009.

Ritter A., Cherry C., Dolan B., « Unsupervised Modeling of Twitter Conversations », *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics*, p. 172-180, June, 2010.

Robertson S. E., « On term selection for query expansion », *Journal of Documentation*, vol. 46, p. 359-364, January, 1991.

Rossi R., Neville J., « Modeling the Evolution of Discussion Topics and Communication to Improve Relational Classification », *Proc. Workshop on Social Media Analytics*, Washington, DC, July, 2010.

Sahami M., Dumais S., Heckerman D., Horvitz E., « A Bayesian approach to filtering junk e-mail », *Proc. AAAI Workshop on Learning for Text Categorization*, July, 1998.

Salton G., Buckley C., « Improving retrieval performance by relevance feedback », *Journal of the American Society for Information Science*, vol. 41, n° 4, p. 288-297, 1990.

Sushmita S., Joho H., Lalmas M., « A Task-Based Evaluation of an Aggregated Search Interface », *SPIRE '09: Proceedings of the 16th International Symposium on String Processing and Information Retrieval*, Springer-Verlag, Berlin, Heidelberg, p. 322-333, 2009.

Turney P. D., « Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL », *Proc. European Conference on Machine Learning*, p. 491-502, 2001.

Vieweg S., Hughes A. L., Starbird K., Palen L., « Microblogging during two natural hazards events: what twitter may contribute to situational awareness », *Proc. International Conference on Human Factors in Computing Systems*, p. 1079-1088, 2010.