

Detection of Real-Time Intentions from Micro-blogs

Nilanjan Banerjee¹✉, Dipanjan Chakraborty¹, Anupam Joshi¹,
Sumit Mittal¹, Angshu Rai¹, and B. Ravindran²

¹ IBM Research - India, New Delhi, India
{nilanjba, cdipanjan, anupam.joshi, sumittal}@in.ibm.com,
angshurai1@gmail.com

² Indian Institute of Technology, Madras, India
ravi@cse.iitm.ac.in

Abstract. Micro-blog forums, such as Twitter, constitute a powerful medium today that people use to express their thoughts and intentions on a daily, and in many cases, hourly, basis. Extracting ‘Real-Time Intention’ (RTI) of a user from such short text updates is a huge opportunity towards web personalization and social networking around dynamic user context. In this paper, we propose novel ensemble approaches for learning and classifying RTI expressions from micro-blogs, based on a wide spectrum of linguistic and statistical features of RTI expressions (*viz.* high dimensionality, sparseness of data, limited context, grammatical in-correctness, etc.). We demonstrate our approach achieves significant improvement in accuracy, compared to word-level features used in many social media classification tasks. Further, we conduct experiments to study the run-time performance of such classifiers for integration with a variety of applications. Finally, a prototype implementation using an Android-based user device demonstrates how user context (intention) derived from social media sites can be consumed by novel social networking applications.

Keywords: Social networks · Micro-blogs · Intention mining

1 Introduction

Social networking sites like Facebook, Twitter, Plazes and Jaiku, provide easy means for users to share their thoughts and form a portal for media-rich social exchanges. Interestingly, much of what is exchanged is ‘context-indicative’- ranging from moods and opinions, to plans of a movie outing or a weekend hike. Prior work [2] report significant content (around 20%) of messages on Twitter contain expressions of user interest and intentions. With the growing popularity of several social exchange portals, that too on the now ubiquitous mobile phones, it is inevitable that there would be a demand for applications having capabilities to exploit ‘real-time’ user context, like “find me buddies who are available and

interested in seeing the Harry Potter movie tonight” or “Who all are interested in playing badminton today eve?”. As such, applications like FourSquare¹ and research studies [4] centered around *ephemeral social networking* is a recent area of focus in industry and academia.

While users are rampantly providing cues about their real-time intentions via mobile social media channels, state-of-the-art in social networking systems, unfortunately, is not automated to capture these free-text “context cues”. Users still need to manually organize such spontaneous events. A multitude of context-sensitive applications can benefit via automatic extraction of user’s intent from inputs available ubiquitously at the user’s finger tips and channelling them to relevant applications. Research along this vein has focused mainly on large-scale event extraction such as earth quakes, mass movements, and the like [15]. However, efficient and personalized extraction of real-time intention is challenging. This is the focus of our paper.

Conceptually, we define a Real-Time Intention (RTI) as “*text expression signifying an intent to perform an activity in near future*”. In particular, we focus on the two-class classification problem - is the update an RTI or a Non-Intention (NI)? Towards this, our goal is to investigate classification models that perform well with micro-blog feeds. The following challenges need to be addressed for such a task:

- *Limited Context Information*: Post size is restricted to a few characters (e.g. 140 in Twitter) – this gives a very short context window for traditional knowledge extraction algorithms [8] to be effective. Moreover, often the context is fragmented, making it difficult to mine the underlying intention, as in the tweet “Gorgeous evening. Out of work. Off to football. Life is sweet.”
- *Richness of Exchange*: Postings are of several kinds: (1) daily activities (2) conversations, discussions (e.g. using hash-tags in Twitter) (3) URL mentions (4) Random thoughts (e.g. moods, feedbacks). For example, “Watching X-Men movies is fun” expresses an opinion, whereas “Can’t wait to see the latest X-Men!!!” suggests an activity intention. Efficiently segregating data pertaining to RTIs from other postings therefore becomes challenging.
- *High Dimensionality*: The use of language in a micro-blog is often informal and sometimes grammatically incorrect or ambiguous, contains mis-spellings, spoken acronyms and morphological variants (e.g. “me wanna play ice hokey nowwwwww!!!”). These factors along with inherent vastness of English vocabulary and proper nouns make the data highly dimensional in nature.

In this paper, we detect (and classify) RTIs in micro-blogs by taking into account micro-blog characteristics described above and provide a platform for efficient classification. Our feature extractors are built around the central intuition that relationships ‘of several types’ between word classes would yield better discriminatory power. This also reduces the dimensionality of the feature space. Ensemble classifiers are well suited for combining features that observe different aspects of the data. We propose three ensemble techniques to combine information from the feature extractors - late fusion, early fusion, and multiple kernel

¹ <http://foursquare.com>

learning (MKL) [14], based on SVM classification models. Our experiments show significant improvement in classifying RTIs and NIs over commonly used classification techniques that use word level features.

We demonstrate a prototype implementation that is connected to Twitter and is capable of funneling interesting tweets to a user’s (Android) device. We believe that such selective context-cast of updates to interested users impacts a large class of current social networking applications. The R-U-In? application [4] that we developed earlier is a representative example of such emerging geo-social networking systems that can benefit from effective utilization of user context.

2 Related Work

Mining of user intention from natural language based dialogs have been a classical problem, researched extensively by computational linguists. A number of works [1] exist on the recognition of user intentions of different nature from dialogs between people and computers in co-operative task oriented environments. A natural progression of these works was to apply the techniques to mine user generated textual data obtained from either specialized application databases [13] or from the World Wide Web [6]. However, most of the work focused on the problem of document classification [11]. Work also exists towards extraction of user intentions and insights from informative one-to-one dialogs generated in a call center [5]. In comparison to dialogs and documents, micro-blog domain presents a unique platform where many of the user updates imply a ‘*state of mind*’ in a few words. Furthermore, the conversations on these platforms take place in a fragmented and ad-hoc fashion.

Social networking websites, blogs and discussion forums are fundamentally different in nature from micro-blogs as they usually contain a large set of semantically interlinked statements, describing a view point. The above cited works exploit this property and reveal relatively *static* or *long-term* trends in user interests, content generation patterns, etc. Our focus, on the other hand, is on capturing real-time intentions in micro-blogs which are characterized by limited contextual information and spontaneous, incoherent expression of *short-term* to RTIs.

In [2], we show that a fractional (20%), but quantitatively significant bulk of micro-blog content contains keywords indicative of user intentions, and reported results on keywords and n-grams commonly observed in Twitter for expressing real-time context. Inspired by these findings, we proposed [3] methods to extract linguistic and statistical features for RTI expressions, and demonstrated a heuristic-based fusion model for classifying RTIs. In this paper, we significantly enhance our initial work and focus on more effective techniques of building ensemble classifiers, and demonstrate an end-to-end system integrating such a classifier.

Hashtags are well-known to index and channel social media content to applications. However, they require user’s attention and are suited for stable topics of discussions rather than spontaneous indications of real-time intentions.

Along these lines, recent works like [15] exploit free-text updates as *Social Sensor data* to rapidly detect real-time events such as earth quakes, tornadoes, etc. Basic SVM models used in prior work is sufficient here as events are of a broader nature (e.g. earthquake) and the hints are significantly stronger with a large learning corpus. Basic SVM models using word-level features led to about 60% classification accuracy for our case, where events are more ephemeral in nature, and text expressions are much more varied, and learning corpus is relatively much lesser. Hence we focus on building better ensemble approaches.

3 Real-Time Intention (RTI) Classifier

We first provide the definitions of various concepts related to *Real-Time Intention* (RTI), followed by the description of our overall system for RTI detection and classification.

Terminology: Content-Indicative Word (CI word):- “*keywords that carry the central subject in an RTI.*” These are typically proper or common nouns (e.g. movie, football), but not necessarily restricted to these parts of speech (POS). Each CI word belongs to a category (class). For e.g. “football” belongs to “sports”.

Usage-Indicative Word (UI word):- “*Keywords that characterize the activity associated with a particular CI word.*” These can be either *Temporal* keywords : T-Words (e.g. evening, morning) or *Action* keywords: A-words (e.g. watch, go, see). We define T-words as words that describe the concept of *time* in a given statement. A-words are words that qualify the *action* associated with the CI word, normally verbs. Our definition is geared towards not imposing strong restrictions on the POS associated with these categories. Just like the CI words, each UI word also belongs to a category.

RTI:- “*A text expression containing one or more CI words providing class of intent; with one or more UI words that further qualify the intent, with no specific ordering.*”

Intuitively, our definition is generic and covers range of expressions to characterize RTIs in a single micro-blog, without grammatical constraints, while safely avoiding ambiguous expressions like “*excited about fishing*”. Note that we define all the terms using loose semantics since we do not want the definitions to be strictly grammatical or linguistically binding in nature.

Figure 1 schematically depicts our Real-Time Intention (RTI) Classification System that takes social updates of users, and classifies them into ‘Intentions’ or ‘Non-Intentions’. As shown in the figure, the classification process is a 3-step one. In the first step, dimensionality of data is reduced in a manner that preserves the relationships amongst words and noise is filtered. In the second step, we extract features using the reduced data dimension space. In the final step, we combine these features to classify an update as intention (RTI) or non-intention (NI). This classification can be used by a plethora of social networking applications.

Let us discuss the three core steps of the classification system in more detail.

3.1 Noise Filter and Dimensionality Reduction

To bootstrap, we first create a vocabulary of CI and UI words that are indicators of presence of RTIs, following an iterative seed-set expansion process [3] from a corpus of micro-blogs and external sources like WordNet.

Keeping in mind that the solution we propose needs to deal with micro-blog data where documents of interest are very sparsely distributed, we use a “*Noise Filter*” that tests every input micro-blog for presence of at least one CI word in the vocabulary. This filter is applied to a micro-blog before allowing it to be passed through the series of feature extractors and classifiers.

We then canonicalize the occurrences of CI and UI words by reducing them to their base category representation. For example “*football*” is converted to “*sport*”, “*sushi*” to “*food*”, etc. This reduces the vocabulary size and hence the dimensionality of the data in a simple and interpretable manner, while preserving the word positions, and hence relationships in the lower dimension. For e.g., the tweet “rushing through friday...is going bowling tonite” is interpreted as “<rushing>[A Word; category = *Verb_{move}*] through <friday>[T word; category = *TypeOfDay_{weekday}*]...is <going>[A word; category = *Verb_{generic}*] <bowling>[CI Word; category=*sports*] <tonite>[T word; category = *TimeOfDay_{evening}*]. These canonicalizations are used subsequently by several feature extractors as outlined in the next step.

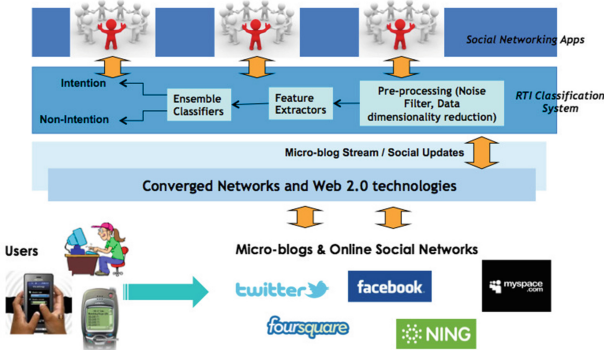


Fig. 1. Intention classification in micro-blogs

3.2 Feature Extraction

We learn a wide variety of information from each micro-blog by inspecting the linguistic relationships and statistical associations among words and phrases after representing the micro-blog in the reduced dimension of CI and UI word categories, for RTIs and NIs. Moreover, we inspect words and phrases *around* these canonicalized representations that show distinctive biases towards these classes. For these purposes, we use several ‘Feature Extractors’ that can inspect different aspects of information in the micro-blog, while handling issues of limited

context information, use of informal language and presence of noise. The output is a feature vector representing the micro-blog. We provide a summary of the features we use for the learning. The details of how these features are extracted can be found in [3].

Co-occurrence Feature Extractor: This extractor analyses the micro-blogs based on the following intuition – if more relevant words co-occur in a micro-blog, likelihood of the micro-blog expressing an intent increases. For this, we find all the gappy bi-grams between relevant CI and UI words and compute the co-occurrences. Gappy bi-grams as features condense the data further across several words while addressing the issue of informal language, and keeps the contextual information intact and interpretable. For each micro-blog, this feature extractor produces the number of co-occurrences between CI and A-words, CI and T-words, A-words and T-words.

POS Feature Extractor: This extractor exploits the fact that although micro-blogs often lack grammatical accuracy, at a sub-sentence level, a user is likely to arrange words in correct grammatical order. For example, consider a tweet: “*me want to watch movie tonight*” and “*me hungry got to eat something*”. Though both examples lack grammatical correctness, the words are more or less in correct grammatical order around the intention. We capture these sub-sentence grammatical constructs *around* the CI words, and produce a feature vector representing the nouns, past tense verbs, and adverbs around the CI words.

Rule-Based Feature Extractor: This extractor learns a set of conjunctive rules that capture words and phrases, containing CI/UI words, commonly used to express RTIs. This produces a set of *intention favorable* rules (*RTI-Rules*) and *non-intention favorable* rules (*NI-Rules*); the rules themselves may not necessarily be grammatically correct. We use a conjunctive rule learner algorithm [9] to learn word based rules for both classes and use presence of these rules as features. This extractor produces a vector representing the number of matched RTI-rules and NI-rules.

Dependency Based Feature Extractor: This extractor identifies *relationships* shared by words in RTI and NI sets based on their grammatical roles, beyond what can be captured by simpler co-occurrence and POS extractors. The intuition is that words play *different grammatical roles* when used to express RTIs as when used for NIs. We capture these roles played by CI, A and T-words and how they relate to words around them. Each role is represented as a *relation* between the source word and a target word. We consider frequently occurring relations as candidates in the feature vector. Given a micro-blog, the output here is a feature vector indicating the number of frequently occurring relations that are present in the micro-blog. For e.g., in the tweet “*out of a head baking meeting... now some soccer!!!*” the tuple “*sport,now*” can be identified as following a dependency relationship using this technique.

Δ -TFIDF Feature Extractor: This technique captures words whose usage is heavily biased towards either one of the sets. Δ -TFIDF driven SVM models have

been shown to improve performance in document classification tasks [12]. We first compute the TF-IDF values for different words separately for the RTI and NI sets for each category. Then the difference of two sets of TF-IDF values is assigned to each word as the Δ -TFIDF value. For non-discriminating words, Δ -TFIDF scores are nearer to 0. For each micro-blog, the output is: vector $V=[\Delta_1, \Delta_2.. \Delta_n]$, where n = no. of distinct words in the micro-blog, and Δ_i = Δ -TFIDF value for word w_i . By their very nature, TF-IDF based methods are grammar agnostic, hence Δ -TFIDF scoring efficiently handles informal usage of language while identifying discriminating utterances in micro-blogs. Also, since Δ -TFIDF are computed using canonicalized data, this approach is robust against emerging CI and UI words (and phrases) as they get accommodated in the vocabulary through regular updates to the different categories.

3.3 Multi-View Ensemble Classifiers

Each feature extractor looks at a different representation of the data. An ensemble classifier is suited for our classification task since they can combine insights from extractors that inspect different views of each micro-blog. Ensemble approaches have been found to be useful with respect to obtaining better classification performance, scale and parallelizability than single classifier approaches [7, 10].

We present three support-vector machine (SVM) based ensemble approaches with the objective of exploring various design choices between computational overhead and accuracy. We choose SVM-based fusion techniques as this provides a theoretically sound approach to determining decision boundaries from complex multi-dimensional datasets, and has been found useful for two-class pattern recognition problems in text classification [15].

Late Fusion Approach: In this approach, we treat the output of each feature extractor as a vector. We learn *individual* SVM models for each of these feature vectors from training data. For each micro-blog, SVM_i outputs ρ_i = predicted probability of the micro-blog containing RTI by SVM_i . Thereafter, overall relevance value for a micro-blog is computed as the weighted sum $S = \sum_{i=1}^5 \rho_i * w_i$, where w_i is normalized weight of the model SVM_i based on model accuracy. A micro-blog m contains an RTI if $S \geq \tau$ for m , NI otherwise.

Intuitively, this approach exploits the discriminatory power of individual feature extractors for classification, while compromising on the potential correlations present in the mixed feature space combining all feature extractors.

Early Fusion Approach: In this approach, all the feature vectors from different feature extractors are combined together to form a single, master feature vector. These feature vectors from the training data are then used to build a single SVM classification model which is used to perform classification. The prediction probabilities are used to assign class values after comparison with τ .

Intuitively, this approach exploits the true multi-dimensional feature representation, combining all features, right from the start. However, on the other hand, learning an effective model combining all features becomes more difficult

as the optimization problem being solved by SVM classifier involves more features simultaneously, incurring additional computational overhead as compared to Late Fusion.

Multiple Kernel Learning (MKL) Approach: This is a theoretically stronger and intuitive approach of combining information from multiple sources by altering SVM internals [16]. We use a MKL formulation on the lines of SimpleMKL [14] for our classification task. We first build a master kernel using a linear combination of kernel values obtained from different feature extractors as follows: $\sum_{i=1}^L w_i k_i(\mathbf{x}_{in}, \mathbf{x}_{im})$, where $n, m = 1, \dots, D$, $\sum_{i=1}^L w_i = 1$, $0 \leq w_i \leq 1$, L is the number of different kernels (corresponding to different feature extractors), and \mathbf{x}_{in} is the feature vector obtained from feature extractor i corresponding to micro-blog n . The individual kernels used are linear kernels. So the SVM optimization problem now becomes the one of maximizing:

$$\tilde{F}(\mathbf{a}, \mathbf{w}) = \sum_{n=1}^D a_n - \sum_{n=1}^D \sum_{m=1}^D a_n a_m t_n t_m \sum_{i=1}^L w_i k_i(\mathbf{x}_{in}, \mathbf{x}_{im}) \quad (1)$$

Where $\mathbf{a} [a_1 \dots, a_D]$ are Lagrangian coefficients and $\mathbf{t} [t_1 \dots, t_D]$ the class labels. Once the optimal weights are obtained, a test micro-blog \mathbf{x} represented by multiple feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_L$ is classified by the class value obtained from the reformulated function $sign(y(\mathbf{x}))$, where $y(\mathbf{x}) = \sum_{n=1}^D a_n t_n \sum_{i=1}^L w_i^* k_i(\mathbf{x}_i, \mathbf{x}_{in})$. This may lead to a suboptimal solution due to stagnation at a local optima given the non-smooth nature of the solution space. To address this, we follow standard practise of choosing best values through multiple random initiations of \mathbf{w} .

4 Experiments and Evaluation

We implemented the RTI Classifier in JAVA, using APIs provided by libsvm, openNLP and Stanford's Dependency Parser² to build the different modules. We used the maximum entropy based POS tagger from OpenNLP³ toolbox as the POS tagger.

Training and Test Data: For the purpose of learning the classification models and evaluating its discriminatory power, we use the data set available with [2]. This data set contains over 20 million publicly available tweets and a set of manually labelled micro-blogs containing a mixture of RTIs and NIs, totalling to 13206 tweets. The RTIs fall in five categories: *Movie*, *Sports*, *Music*, *Food*, *Dance*. For each category, a tweet that means an intention to perform an activity strictly in the near future was marked as an RTI. All other tweets, including ones that indicate activities being performed at present (e.g. watching a movie) or past activities and other irrelevant ones were marked as NI.

We used a 10-fold cross validation process for performance evaluation. Experiments were performed on a 2.83 GHz, 64 bit quad-core Intel processor system

² <http://nlp.stanford.edu/software/lex-parser.shtml>

³ <http://incubator.apache.org/opennlp/>

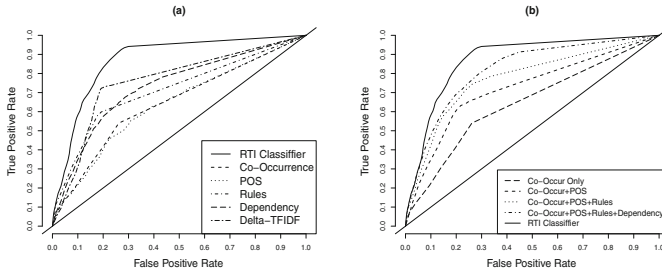


Fig. 2. ROC curves for RTI classifier using late fusion approach ($RTI_{class} = sports$) (a) SVMs built on individual feature extractors Vs RTI classifier, (b) Incremental contributions of individual feature extractors to RTI classifier performance. Results for other categories were similar.

with 4 GB RAM and 6 MB L2 cache. Different embodiments of RTI Classifier incorporate one of the three ensemble learning approaches discussed before. Along with each ensemble classifier that makes use of all feature extractors, we also study the corresponding classifiers that utilize only a single feature extractor, named as *Individual_{svm}*. This is to understand incremental contributions of different feature sets. We use the area under the Receiver Operating Characteristic (ROC)⁴ curve (referred to as AUC) to compare performance. This provides useful insights into expected performance of the classifiers over the entire operating region. The plots use “RTI Classifier” as the uniform label to represent respective ensemble approaches.

4.1 Classification Results

SVM Ensemble vs. *Individual_{svm}* Classifiers: Fig. 2 shows the comparative performance of Late Fusion Learning against SVMs built using individual

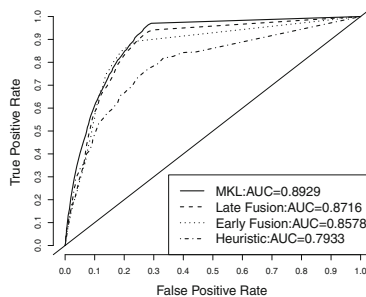


Fig. 3. Performance comparison of heuristic, late fusion, early fusion and MKL approach ($RTI_{class} = sports$). Results for other categories were similar.

⁴ http://en.wikipedia.org/wiki/Receiver_operating_characteristic

feature extractors. We observe that the fusion approach performs better than the $Individual_{svm}$ classifiers. This shows that different features jointly contribute additional information towards building better decision boundaries for our task. Among different $Individual_{svm}$ classifiers, POS and Co-occurrence features produce similar results; Dependency and Rule based features perform better than these two; while Δ -TFIDF features due to its capability of exploiting discriminatory words performs best. Finally, from Fig. 2b, we notice that single classifiers in fact contribute to various degrees to provide an overall improvement to the ensemble.

Cross-comparison of Ensembles: Fig. 3 shows performance comparison of all three ensembles and the best prior result on this data set using a heuristic-based linear classifier [3]. In the heuristic-based approach, each feature extractor adds a bias in the classification of a given micro-blog. For each extractor, a relevance score R_{c_i} is computed, which is the confidence given by the extractor to a given micro-blog to contain an RTI for class c_i . The combined relevance value S is a weighted linear sum of the individual R_{c_i} scores. A micro-blog is classified as an RTI of class c_i if $S >$ discrimination threshold τ ; NI otherwise.

We observe from the AUC that the ensemble approaches provide substantial improvement over the known heuristic. Going into deeper analysis, Late and Early Fusion approaches work similarly, specially at lower values of τ . However, Late Fusion performs better than Early Fusion at higher values of τ . This indicates that combining features across different dimensions early is having a slight deteriorating effect for our data-set. This has also been reported for other classification tasks. We observe for MKL approach, the performance is sustained till TPR (True Positive Rate) beyond 90%. It also largely provides the best TPR for the same FPR compared to others. Moreover, we found that time delay in MKL is comparable to the two fusion approaches. This makes it best suitable for applications requiring high precision and recall. We wish to point out that by nature of formulation of the MKL approach used here, it was not guaranteed that optimal weights would be obtained. However, our trainings with different random seeds resulted in identical performance plots.

We obtained accuracies of 52–65% from experiments conducted using bag of words as features using the standard SVM classifier (as well as J48 decision trees, naive bayes). Across all categories, we obtained \approx 20–25% improvement in AUC, on use of different ensemble approaches. This provides evidence that the feature extraction techniques are generic towards deriving useful aspects of RTI. Further, the ensemble techniques provide effective integration of these features, leading to an effective classification system.

4.2 Run-Time Performance

We measure classification delay of the complete system under two conditions: (1) Without Noise Filter (2) With Noise Filter, on a stream of 50,000 tweets, containing sparsely located RTIs, emulating real-life situations. Table 1 presents the results. Heuristic approach has minimum classification delay (sec/tweet) owing

Table 1. Run-time performance of different approaches

Approach	W/O Noise Filter (<i>sec/tweet</i>)	W/ Noise Filter (<i>sec/tweet</i>)	W/O Noise Filter (<i>tweets/month</i>)	W/ Noise Filter (<i>tweets/month</i>)
Heuristic	0.237	0.0189	10×10^6	137×10^6
Late Fusion	0.677	0.0216	3.8×10^6	120×10^6
Early Fusion	0.840	0.0227	3×10^6	114×10^6
MKL	0.736	0.0219	3.51×10^6	118×10^6

to its simplicity. We observe that Late Fusion approach performs slightly better than Early Fusion and MKL approach while MKL approach, in turn, performs slightly better than Early Fusion approach. We note that the ensemble systems with Noise Filter (last column) are capable of catering to the overall tweet generation rate of 100 million tweets per month⁵. Going forward, we intend to improve the throughput further via parallel computation of features, kernels and individual classifier scores.

5 System Demonstration

In this section, we present how we leveraged the RTI Classifier and integrated Twitter with R-U-In? [4] - an activity-oriented social networking system for users to collaborate and participate in activities of mutual interest. R-U-In? essentially enables a user interested in a certain activity (e.g. movie, game of tennis, etc.) to

**Fig. 4.** Integration of Twitter with R-U-In? exploiting RTI Classification

⁵ <http://blog.twitter.com/2010/02/measuring-tweets.html>

look for people who have expressed similar interests and to schedule an activity based on their availability over preferred modes of communications. However, following conventional models, R-U-In? depends on *structured* user interests that are expressed explicitly with some tools or applications (e.g., SMS, gtalk, etc.). RTI Classifier brings the power of intelligent suggestions inferred from implicit expression of user interests (e.g., somebody tweeting about an activity-related intention) and enhances the overall R-U-In? experience.

We consider the following scenario to illustrate this claim. Let us assume that Joshua has already scheduled a movie-watching activity with some of his interested and available friends following the procedure described in [4]. Now, another of his friends, Sam, gets interested in going for a movie and uses his Android based mobile phone to express his intention. Figure 4a, shows a screenshot of Sam's Android phone, where he logs into the R-U-In? portal and uses the Twitter box to post the tweet "*wish i had someone to go for movie with today*". This tweet is automatically captured by RTI Classifier that declares this tweet as carrying real-time intention along with keywords describing the intention (e.g. category, and any temporal word describing time). This interest along with Sam's location (obtained from other offline channels such as phone's inherent location tracking mechanism or GPS) is semantically matched thereafter by R-U-In?'s match-making engine and a number of matching activities in Sam's neighborhood is shown in the Google map of R-U-In? portal. Figure 4a also shows Sam's current location and the matching activities. A click on a matching activity shows further details of the activity (e.g., time, location, owner of the activity) and an option to join it. This is shown in Fig. 4b. Sam selects an activity of interest and ends up joining his friends over the movie.

We argue that our RTI Classifier can be similarly integrated with several other social networking applications, thus progressing the state-of-art in social communication.

6 Conclusion

This paper carried out a thorough analysis of micro-blogs towards detection and classification of Real-Time Intentions (RTIs) of users. We presented feature extractors that obtain relational features in a reduced dimension of the complex micro-blog data along with the performance evaluation of ensemble learning approaches for combining these features. We believe our methodology, system demonstration and performance evaluation offer significant insights to applications aiming to exploit free-text intentions from social media.

References

1. Allen, J.: Recognizing intentions from natural language utterances. In: Brady, M., Berwick, R.C. (eds.) *Computational Models of Discourse*, pp. 107–166. MIT Press, Cambridge (1983)

2. Banerjee, N., et al.: User interests in social media sites: an exploration with micro-blogs. In: International Conference on Information and Knowledge Management (CIKM) (2009)
3. Banerjee, N., et al.: Towards analyzing micro-blogs for detection and classification of real-time intentions. In: ICWSM (2012)
4. Banerjee, N., et al.: R-U-In? - exploiting rich presence and converged communications for next-generation activity-oriented social networking. In: International Conference on Mobile Data Management Systems (MDM), Services and Middleware (2009)
5. Chalamalla, A.K., et al.: Identification of class specific discourse patterns. In: ACM Conference on Information and Knowledge Management (2008)
6. Chen, Z., et al.: User intention modeling inweb applications using data mining. World Wide Web: Internet and Web Inf. Syst. **5**, 181–191 (2002)
7. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
8. Han, J., et al.: Mining frequent patterns without candidate generation (2000)
9. Han, J., et al.: Data Mining Concepts and Techniques, Chap. 6, 2nd edn, pp. 318–327. Morgan Kaufmann, San Francisco (2006)
10. Hastie, T., et al.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009)
11. Liu, R.-L., et al.: Incremental context mining for adaptive document classification. In: KDD, pp. 599–604 (2002)
12. Martineau, J., et al.: Delta TFIDF: an improved feature space for sentiment analysis. In: AAAI International Conference on Weblogs and Social Media (2009)
13. Nasukawa, T., et al.: Text analysis and knowledge mining system. IBM Syst. J. **40**(4), 967–984 (2001)
14. Rakotomamonjy, A., et al.: SimpleMKL. J. Mach. Learn. Res. (JMLR) **9**, 2491–2521 (2008)
15. Sakaki, T., et al.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW (2010)
16. Varma, M., et al.: Morje generality in efficient multiple kernel learning. In: ICML (2009)