Recognizing Tweet Relevance with Profile-specific and Profile-independent Supervised Models

Kathy Lee, Ashequl Qadir, Yuan Ling, Joey Liu, Sadid A. Hasan, Vivek Datla, Aaditya Prakash, Oladimeji Farri

Artificial Intelligence Laboratory, Philips Research North America, Cambridge, MA, USA {firstname.lastname, kathy.lee_1, dimeji.farri}@philips.com

Abstract

In the 2017 TREC (Text Retrieval Conference) Real-Time Summarization (RTS) track, we explored supervised methods for identifying relevant tweets based on a user's interest profile. We primarily focused on two approaches: profile-specific and profile-independent. For profile-specific, we trained a model for each interest profile with features spe-In case of profilecific to the target profile. independent, a single model was trained with features that were general across all profiles. For training the supervised models, we used labeled data from the previous year's challenge. We additionally introduced a novel method for automatically labeling tweets with relevance scores. The method treated keywords from titles as an essential information and penalized the relevance score for a tweet when the keywords were absent; while treating keywords from description as supporting information, and rewarding the relevance score when these keywords were present. In scenario A (real-time push notification), our best run yielded 9.95% EG-p and 11.11% nDCG-p improvements over the median in batch evaluation. In scenario B (daily digest), our best run achieved 25.43% nDCGp improvement over the median.

1 Introduction

For TREC Real-Time Summarization (RTS) track,¹ the challenge of training supervised models arises due to 1) a large collection of interest profiles, 2)

diversity of topics, and 3) limitations for generating labeled data for new interest profiles. For a given set of interest profiles, human experts can manually label some training data and train one supervised model per profile. This solution is difficult to scale if new interest profiles are introduced since a trained model would only be tied to a specific interest profile that the framework has been exposed to. An alternative approach can be to create a single profile-independent supervised model to determine profile relevance of a tweet, which would then be applicable for any interest profile. In this case, feature modeling needs to be carefully designed so that the extracted features are not tied to any specific profile.

For the TREC-2017 RTS challenge, we explored the use of both profile-specific and profileindependent supervised models for recognizing tweet relevance. We used the labeled data from the past year's challenge (Lin et al., 2016) as the training data to build a single profile-independent supervised regression model. In this method, we created training instances by pairing up tweets with interest profiles and the trained regression model predicts a real-valued relevance score. As features, different overlap statistics between a tweet and a profile were used (e.g., number of overlapping words, phrases, parts-of-speech tokens, etc.). We additionally explored an attention-based deep learning model that learns semantic characteristics of the words in a profile that may be important for recognizing relevance. For making push decisions we used a threshold relevance score.

We also introduced a novel method for automatically labeling new data to train a dedicated supervised model for each profile (profile-specific). In this method, we collected tweets using keywords ex-

http://trecrts.github.io/

tracted from the profiles, and assigned a relevance score based on a method that penalized for any missing title keywords, but gave a reward for additional keywords from profile descriptions.

We submitted a total of six automatic runs, three for real-time push notification (scenario A) and three for daily digest (scenario B). For scenario A, our three runs corresponded to 1) our best method in scenario A from the 2016 challenge, 2) a profileindependent supervised regression model trained on labeled data from the past challenges which exploits features based on different overlap statistics between a tweet and a profile, and 3) a combination of (1) and (2). For scenario B, our three runs corresponded to 1) our best method in scenario B from the RTS 2016 challenge, 2) a profiledependent supervised regression model trained on data that were automatically labeled with our novel method of reward/penalty-based relevance scoring, and 3) a profile-independent attention-based Convolutional Neural Network model that automatically determines which profile words may need more focus when searching for relevant tweets, trained on the data from the past challenges. In scenario A, our best run yielded 9.95% EG-p (+0.0104) and 11.11% nDCG-p (+.0185) improvements over the median in batch evaluation. In scenario B, our best run achieved 25.43% nDCGp (+0.0558) improvement over the median.

2 System Description for Scenario A

2.1 Real-Time Push Notification for Run 1

For our run 1 in scenario A, we used our best method from the 2016 TREC RTS challenge (scenario A) (Lee et al., 2016). This method uses a set of assorted textual features extracted from the interest profiles and determines relevance of a tweet based on a weighted relevance score. The seven categories of textual features that we used are:

- **Title words**: we extract all unigrams (individual words) from the profile title after excluding stopwords and punctuations.
- Title phrases: we extract all noun phrases and verb phrases that only appear in the title of an interest profile.

- **Noun phrases:** we identify all noun phrases from the title, description and narrative fields of the interest profiles.
- Phrases within quotations: we extract phrases from title, description, and narrative that appear within quotation marks. Intuitively, phrases within quotation carry special importance, and tweets that mention these phrases exactly, could be highly relevant to the profiles.
- Named Entity Phrases: we extract phrases that contain a named entity. For extracting named entities, we use the NLTK toolkit.
- Location Named Entity Phrases: We extract all named entity phrases that mention locations.
- TF-IDF phrases from narrative: We calculate TF-IDF scores for words in profile narratives, considering each narrative of an interest profile as a document. We take the top 10 words with the highest TF-IDF scores (excluding stopwords), and extract noun phrases and verb phrases that contain one of these high scoring TF-IDF words.

The textual feature categories are further expanded to include paraphrases of the extracted phrases so that phrases that are synonymous can also contribute towards measuring relevance. We use the PPDB Paraphrase Database (Ganitkevitch et al., 2013) (L-size) for the paraphrase-based feature expansion. We do not expand the named entity phrases and phrases within quotations. For the other textual features, we create four new categories with only the paraphrase terms. After feature expansion, we have a total of 11 categories of textual features.

To identify relevant messages from the Twitter feed, messages are first filtered based on language, minimal number of title words, presence of named entity or quoted phrase, etc. Finally, relevance with respect to feature categories and profiles are determined using the equations:

$$relevance(x, C_i) = \sum_{c \in C_i} \frac{l_c^2}{n_c \times max_n(C_i)}$$
 (1)

$$profile_relevance(x) = \sum_{i=1}^{C} w_i \times relevance(x, C_i)$$
(2)

Here, x is an input tweet, C_i is the set of textual features for the i^{th} textual feature category, l_c is the maximum number of rightmost words from phrase c that appears in the tweet consecutively and in the same order, n_c is the total number of words in c, and $max_n(C_i)$ is the maximum phrase length (in terms of words) among all of the phrases in C_i . Weights w_i are feature category weights estimated from the TREC RTS 2015 data, maximizing the Expected Gain (EG). The tweet that is assigned a high relevance score is then checked for novelty using a semantic similarity model (Hasan et al., 2015) and pushed to user if above a threshold of 0.75. More details on the method can be found in Lee et al. (2016)

2.2 Real-Time Push Notification for Run 2

For our run 2 in scenario A, we design a profile-independent supervised regression model. One of the important considerations in scenario A is *when* a tweet can be pushed to a user. So instead of a categorical relevant vs. not relevant decision, we use a regression model that allows us to directly predict a real-valued relevance score. As our regression algorithm, we use L2-regularized L2-loss support vector regression from the LIBLINEAR library (Fan et al., 2008). To create training instances, we use tweets from the 2016 TREC RTS challenge where each tweet is assigned a relevance score using batch evaluation. As our training instances, we pair a tweet with its respective interest profile and use the relevance score as the data label.

For this profile-independent regression model, the textual features described in Section 2.1 could not be directly used because they are lexical features closely tied to the respective interest profiles. For example, in profile RTS2, a user is interested in information related to Zica virus in Ecuador. One of our assorted feature types would identify Ecuador as a key named entity feature (among others) for finding tweets relevant to RTS2. But such lexical features are ideally useful for training a model specifically for RTS2, which limits the ability to train a new model for a profile if labeled training data is not

readily available.

To create a profile-independent model, we instead determine how much these features overlap between an interest profile and a tweet in their respective feature categories, and derive statistics with respect to the categories instead of the features themselves so that they can be generalized across all profiles. For the above example, instead of relying on specific features such as "Ecuador", our new general features would now try to assess the importance of having an overlap between a tweet and a profile for different feature categories (e.g. named entity) to determine the tweet-topic relevance. To achieve this, for each feature type described in Section 2.1, we create five binary overlap features to use in our regression model. These features are:

- Full overlap: If all of the phrases from a feature type are present in a target tweet, we set this feature to 1 (or 0 otherwise).
- Much overlap: If more than half of the phrases from the feature type are present in a target tweet but not all, we set this feature to 1 (or 0 otherwise). We require that a minimum of three phrases are in the respective feature type.
- **Moderate overlap**: If half of the phrases from a feature type are present in a target tweet, we set this feature to 1 (or 0 otherwise).
- **Some overlap**: If less than half of the phrases from the feature type are present in a target tweet but more than one, we set this feature to 1 (or 0 otherwise). We require that a minimum of three phrases are in the respective feature type.
- Bare-minimal overlap: If one of the phrases from a feature type are present in a target tweet, we set this feature to 1 (or 0 otherwise). We require that a minimum of three phrases are in the respective feature type.

In addition to the feature types described in Section 2.1, we additionally use person, organization, nouns, verbs, adjectives and unigrams, bigrams, and trigrams when deriving the overlap features. From the profile, we only use title. We also experimented with descriptions and narratives but found the title part of the profile to be more effective with this modeling.

Once the support vector regression model is trained on the paired tweets and profiles, as we monitor the twitter stream, for a candidate tweet, we pair the tweet with each interest profile and apply the trained model. If the predicted relevance score for a profile-tweet pair is above a threshold value 0.5, we push the tweet to the respective user.

2.3 Real-Time Push Notification for Run 3

For run 3, we combined outputs of our run 1 and run 2 models.

3 System Description for Scenario B

3.1 Daily Digest for Run 1

For run 1 in scenario B, we use our last year's best run for scenario B (Run1 from 2016), the details of which can be found in Lee et al. (2016).

3.2 Daily Digest for Run 2

For the run 2 in scenario B, we build profile-dependent supervised models trained for each profile separately. The major obstacle for building such models is the availability of training data with respect to all interest profiles. Past challenge data are insufficient for this purpose since they do not have relevance judgment for the new profiles. To this end, we devise a method to automatically generate labeled training data allowing us to train supervised models for a given profile.

The main assumption of our automatic data labeling method is that the information present in the title of an interest profile is the most important part of the profile, while the information from the description provide additional supporting information. Under this assumption, we design a reward-penalty driven method for assigning a relevance score to a tweet for a given profile. First, we create an initial set of candidate relevant tweets (streamed from Twitter during the week before the challenge) for a given profile where each tweet contains at least one title word (limited to nouns, verbs and adjectives) from the profile. We assign a zero score to each tweet in the candidate tweets set as the initial relevance score.

We then make an assumption that if all words from this target profile's title are present in a tweet, then the tweet is a relevant tweet for that profile. For these tweets, we do not update the relevance score. But for the tweets that do not have all of the words from the profile's title, we want to penalize the assigned relevance score to reflect that the tweet may be missing some key information for establishing relevance. As the penalty, we remove 1 point from the relevance score for each missing title word. For example, if a profile title is "Zica virus in Ecuador", and the only common words in the tweet are "Zica" and "virus", but "Ecuador" is missing, then the score is updated to -1 from 0 to reflect the missing title word "Ecuador" (we only judge overlap or missing words with respect to nouns, verbs and adjectives).

Finally, whenever a tweet has additional words from the description of an interest profile, we want to reward these cases. This is because, although the title generally specifies the main topic of an interest profile, a user may be interested in more specific information on the topic which can be reflected in the description. For example, for the profile title "heating pad recommendations", it becomes clear that the user is interested in knowing about which heating pads other consumers prefer and suggest. However, the description "What heating pads are recommended for treating low back pain?" further reveals that the user is mainly interested in heating pads in the context of back pain. Although an experience with a general heating pad may also apply in this case, when a tweet mentions a heating pad recommendation in the context of back pain, the tweet should be judged as more relevant.

To achieve this, when all of the title words are already present in a tweet, but the tweet also has additional words from the description of the interest profile, for each additional word from the description, we reward 1 point to the relevance score. For the above example profile, if a tweet mentions all of the title words "heating", "pad", "recommendation", and has the additional words from description such as "back" and "pain", the assigned relevance score is updated from 0 to +2 to reflect the presence of the supporting context "back" and "pain".

Once a set of tweets are automatically labeled with a relevance score using the method above, we train a L2-regularized L2-loss support vector regression from the LIBLINEAR library (Fan et al., 2008). As features for regression, we use bag-of-words.

To create a daily digest for a given interest pro-

file, we first use an initial filtering of all the streamed tweets to create a set of candidate tweets to rank. As the filtering method, we check for the following two conditions: 1) total number of unique title and description keyword overlaps are more than total number of unique title keywords from the profile, and 2) total number of unique title keywords overlaps are more than a half of total number of unique title keywords from the profile. After creating the candidate tweets set, the trained model is run on the new tweets to predict a relevance score. The tweets are sorted based on the predicted score and the top 100 ranked tweets above a threshold of -1.5 are selected to put in the batch/digest to send to the user.

3.3 Daily Digest for Run 3

For run 3, we employed an Attention-Based Convolutional Neural Network (ABCNN) model (Yin et al., 2015) which was designed to model sentence pairs by taking into account the interdependence between the two sentences, and has shown to achieve state-of-the-art performance in tasks such as paraphrase identification, answer selection and textual entailment. This model is profile-independent since it does not require profile-specific labeled data for training and can predict profile-tweet relevance score for previously unseen profiles. Evaluation data from past two years (RTS2016, Microblog2015) was used to train the model by setting a tweet and a user profile (topic title plus description) as a sentence pair. We used ABCNN-3 model with 2 convolution layers which computes attention weights on both the input representation and the output of convolution. For hyperparameters, we used 0.08 as the learning rate, 0.0004 for L2 regularization, batch size of 64, 20 epochs, and support vector machine (SVM) as classifier. If the final relevance score by SVM is above a threshold value, the model classifies the tweet-profile pair as relevant (i.e., the tweet is relevant to the user profile).

4 Evaluation

4.1 Post Hoc Batch Evaluation in Scenario A

For scenario A, tweets pushed are evaluated with the following evaluation metrics:²

• Expected gain (EG) (for an interest profile on a particular day) is defined as follows:

$$EG = \frac{1}{N} \sum G(t) \tag{3}$$

where N is the number of tweets submitted by a system and G(t) is the gain of each tweet.

 Normalized Cumulative Gain (nCG) (for an interest profile on a particular day) is defined as follows:

$$nCG = \frac{1}{Z} \sum G(t) \tag{4}$$

where Z is the maximum possible gain (given the ten tweets per day limit).

In determining gain G(t), not relevant, relevant, and highly-relevant tweets receive a gain of 0, 0.5, and 1.0, respectively. EG and nCG metrics have two variations each. For the EG-1 and nCG-1 variation of the metrics, on a silent day when there are no relevant tweets for a particular interest profile, a system receives a score of 1 (i.e., perfect score) if it does not push any tweet. That is, if the system pushes 0 tweets, it receives a score of 1. However, under the EG-p and nCG-p metrics, there is a penalty proportional to how "quiet" the system is. The score is one minus the fraction of the ten-tweet daily quota that is used. If it pushes 1 tweet, it gets a score of 0.9, if it pushes 2 tweets, it gets a score of 0.8, and so on, such that if a system uses up its quota of ten tweets for a silent day, it receives a score of zero. EG-p is the primary metric for judging systems in scenario A.

• Gain Minus Pain (GMP), defined as follows:

$$GMP = \alpha \times G - (1 - \alpha) \times P \tag{5}$$

Here G (gain) is computed in the same manner as above; and P (pain) is the number of non-relevant tweets that are pushed, and controls the balance between the two. Evaluations are done at three α settings: 0.33, 0.5, and 0.66. The EG, nCG and

²http://trecrts.github.io/ TREC2017-RTS-guidelines.html

Table 1: Scenario A batch evaluation results. EG = Expected Gain (1 = with silent day reward, p = proportional silent day reward), nCG1 = Normalized Cumulative Gain (1 = with silent day reward, p = proportional reward on silent day), GMP = Gain Minus Pain (at $\alpha = 0.33$, 0.5 and 0.66).

Evaluation Metrics	EGp	EG1	nCGp	nCG1	GMP.33	GMP.5	GMP.66
Median	0.2194	0.1951	0.2095	0.1826	_	-0.1707	_
PRNA Systems							
PRNA Run 1	0.2090	0.1951	0.2052	0.1913	-0.1330	-0.0780	-0.0262
PRNA Run 2	0.2066	0.1914	0.2058	0.1906	-0.2630	-0.1707	-0.0839
PRNA Run 3	0.2298	0.2016	0.2280	0.1998	-0.3278	-0.2052	-0.0899

GMP metrics are used in a post hoc batch evaluation where relevant tweets are semantically clustered into groups containing tweets that share substantively similar information and judged as not-relevant, relevant, or highly relevant by the pools.

Table 1 presents the results of our three runs in scenario A and compares them with the median results among the participating teams. For both variations of the EG and nCG metrics, our run 3, which combines our last year's best run with this year's profile-independent supervised support vector regression model, performed the best among the three runs. Run 3 also outperformed the median results on these metrics. On the official EG-p metric, our run 3 improved the results by +0.0104 (9.95% of the median). Individually, run 1 and 2 could only perform better than the median results for the nCG-1 metric. However, on the GMP.5 metric, run 1 still achieved the best result among our three runs, which is +0.0927 higher than the median result (54.31% of the median).

4.2 Live User-in-the-loop Assessments for Scenario A

In a live user-in-the-loop assessments, tweets submitted by the participating systems are immediately routed to the mobile phone of a human assessor to be judged as relevant, not-relevant or redundant. Precision is calculated as the ratio of the relevant tweets to the judged tweets. In strict precision, relevant but redundant tweets are excluded. In lenient precision, relevant but redundant tweets are still included.

Table 2 presents the evaluation by mobile assessors for scenario A. Run 1 had the best results among the three runs. Our run 1 and run 3 both performed better than the median results where run 1 achieved

+0.0737 and +0.0609 better strict and lenient precision respectively, compared to the median results (21.66% and 14.59% of the median).

Table 2: Evaluation by mobile assessors in scenario A.

	Strict	Strict	Lenient	Lenient
	Precision	Utility	Precision	Utility
Median	0.3403	-805	0.4174	-456
PRNA Systems				
PRNA Run 1	0.4140	-262	0.4783	-66
PRNA Run 2	0.3346	-678	0.3912	-446
PRNA Run 3	0.3625	-852	0.4264	-456

Table 3: Top 5 scenario A runs evaluated by mobile assessors.

•		
	Team (Run)	Strict Precision
	WUWien (Run 1)	0.4337
	IRIT (Run 1)	0.4200
	PRNA (Run 1)	0.4140
	udel_fang (Run 1)	0.4096
	udel_fang (Run 2)	0.3980

For scenario A, a total of 41 runs were submitted from 15 participating teams. Table 3 shows the top 5 runs and strict precision based on evaluation by the mobile assessors. Our best model (PRNA Run 1) ranked third among the 41 runs.

Table 4: Time Latency in Scenario A.

latency (seconds)	mean	median	
PRNA Systems			
PRNA Run 1	50612.7	69.0	
PRNA Run 2	29994.0	78.0	
PRNA Run 3	39366.4	74.0	

Table 4 shows the time latency of the runs. The average latency is very high but median latency is

low. The main reason for the high average latency is the semantic similarity checking method we used to remove duplicate contents, which increases overall latency as the runs progress and the queue of the streamed tweets increases, resulting in very high latency in the later days of the challenge.

4.3 Post Hoc Batch Evaluation in Scenario B

For scenario B runs, **Normalized Discounted Cumulative Gain nDCG@10** is used as the evaluation metrics with two variants as the scenario A. nDCG@10-1 rewards a system for not pushing tweets on a silent day when there are no relevant tweets, and nDCG@10-p proportionally rewards depending on how quite a system is on a silent day.

Table 5: Scenario B batch evaluation results. nDCG = Normalized Discounted Cumulative Gain (1 = with silent day reward, p = proportional reward on silent day).

are, p proportional remains on silent ea,				
Evaluation Metrics	nDCGp	nDCG1		
Median	0.2194	0.1865		
PRNA Systems				
PRNA Run 1	0.2071	0.1914		
PRNA Run 2	0.2752	0.2400		
PRNA Run 3	0.2143	0.1686		

Table 5 compares the results of our runs with median results. Among our three runs, run 2, the profile-dependent support vector regression model trained with automatically labeled data, achieved the best results. This run yielded +0.0558 nDCG-p and +0.0535 nDCG-1 results over the median (25.43% and 28.69% of the median).

4.4 Evaluation of Scenario B Runs as Scenario A Runs

Table 6: Top 5 scenario B runs evaluated as scenario A runs by the mobile assessors.

une miceme acceptation				
Team (Run)	Strict Precision			
PRNA (Run 1)	0.4811			
IRIT (Run 1)	0.4666			
PKUICST (Run 3)	0.4625			
IRIT (Run 3)	0.4560			
IRIT (Run 2)	0.4497			

For scenario B, a total of 40 runs were submitted from 15 participating teams. Table 6 shows the top 5

scenario B runs and strict precision when evaluated as scenario A runs by the mobile assessors. Our best run (PRNA Run 1) ranked first among the 40 runs.

Table 7: Top 5 scenario B runs evaluated as scenario A runs by NIST assessors.

Team (Run)	EG-p	
PKUICST (Run 1)	0.2959	
adv_lirmm (Run 1)	0.2676	
PRNA (Run 2)	0.2674	
adv_lirmm (Run 2)	0.2641	
adv_lirmm (Run 3)	0.2620	

Table 7 presents the top 5 scenario B runs and EG-p when evaluated as scenario A runs by NIST assessors. Our best run (PRNA Run 2) ranked third among the 40 runs.

5 Conclusion

In this paper, we explored the use of both profile-specific and profile-independent supervised models for recognizing tweet relevance. In scenario A, our run 3, which combines our last year's best run with a new supervised regression model, yielded +0.0104 EG-p and +.0185 nCG-p improvements over the median. In scenario B, our run 2, a profile-dependent support vector regression-based model trained with automatically labeled training data, obtained nDCG-p of 0.2752, achieving +0.0558 nDCG-p improvement over the median. In future work, we will explore the use of automatic data labeling methods to label training data in large quality for training deep learning-based models to retrieve relevant tweets.

References

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

S. A. Hasan, Y. Ling, J. Liu, and O. Farri. 2015. Exploiting Neural Embeddings for Social Media Data Analysis. In *Proceedings of the Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015.*

- Kathy Lee, Ashequl Qadir, Vivek V. Datla, Sadid A. Hasan, Joey Liu, Aaditya Prakash, and Oladimeji Farri. 2016. Assorted textual features and dynamic push strategies for real-time tweet notification. In Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016.
- Jimmy Lin, Adam Roegiest, Luchen Tan, Richard Mc-Creadie, Ellen Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 real-time summarization track. Technical report, DTIC Document.
- Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *CoRR*, abs/1512.05193.