J Intell Inf Syst (2016) 47:247–265
DOI 10.1007/s10844-016-0410-y

RedTweet: recommendation engine for reddit

Hoang Nguyen[1] · Rachel Richards[1] · Chien-Chung Chan[1] · Kathy J. Liszka[1]

Received: 13 October 2015 / Revised: 6 April 2016 / Accepted: 11 April 2016/
Published online: 10 May 2016

Abstract Twitter and Reddit are two of the most popular social media sites used today. In this paper, we study the use of machine learning and WordNet-based classifiers to generate an interest profile from a user's tweets and use this to recommend loosely related Reddit threads which the reader is most likely to be interested in. We introduce a genre classification algorithm using a similarity measure derived from WordNet lexical database for English to label genres for nouns in tweets. The proposed algorithm generates a user's interest profile from their tweets based on a referencing taxonomy of genres derived from the genre-tagged Brown Corpus augmented with a technology genre. The top K genres of a user's interest profile can be used for recommending subreddit articles in those genres. Experiments using real life test cases collected from Twitter have been done to compare the performance on genre classification by using the WordNet classifier and machine learning classifiers such as SVM, Random Forests, and an ensemble of Bayesian classifiers. Empirically, we have obtained similar results from the two different approaches with a sufficient number of tweets. It seems that machine learning algorithms as well as the WordNet ontology are viable tools for developing recommendation engine based on genre classification. One advantage of the WordNet approach is simplicity and no learning is required. However, the WordNet classifier tends to have poor precision on users with very few tweets.

Keywords Social networking · Genre classification · Ensemble classifiers · Twitter · Reddit · WordNet

✉ Chien-Chung Chan
chan@uakron.edu

Kathy J. Liszka
liszka@uakron.edu

[1] Department of Computer Science, University of Akron, Akron, OH, 44325-4003, USA

1 Introduction

Twitter[1] is a social media networking service that allows users to read and send 140 character messages called Tweets. In general, users who visit Twitter want to keep up with their friends, find updates about their favorite celebrities, or get close to real-time information about some viral current events. They also use their profile and tweets indirectly to express interests : hobbies, news, romance, and movies. The network now has over 270 million active users which makes it a virtual gold mine for data collection (Taylor 2014). We believe that each profile can provide insight subjects users want to read. We wish to use this data to provide personalized recommendations for viral stories posted on Reddit.[2]

Reddit is a popular website that constantly updates user submitted content such as news, images, videos, blogs, and books. Since Reddit has a huge number of topics, a given reader is not likely to be interested in everything available. For example, for the year 2015, there were 255,671 authors with more than 70 millions submissions and over 668 million comments.[3] It is often the case that a reader might scroll through myriad threads on Reddit before finding a title that catches their eye.

The objective of this study is to form an interest profile from a user's tweets, to recommend loosely related Reddit threads which the reader is most likely to be interested in. Instead of dealing with the problem at the topic classification level, we treat it as a genre classification problem as first proposed in our previous work (Nguyen et al. 2015). Given a tweet, we want to deduce what genre(s) it might fall under if those words in the tweet were used in formal texts. From there, we keep track of how many tweets fall under which genre, and generate a list of Reddit threads which similarly fall under those genre and are proportional to the interests of the user.

Our basic idea is to use a taxonomy of genres as a common vocabulary for matching interest profiles of Twitter users to the Top 50 subreddits of Reddit threads. We use the genre-tagged Brown Corpus (Francis and Kucera 1979) augmented with a technology genre as a referencing taxonomy of genres, and each genre is represented by a bag of words extracted from documents in the referencing corpus. A bag of words is a multiset where each element is a word paired with a frequency count.

The referencing taxonomy of genres is used to automate the process of transforming a user's tweets into an interest profile represented as a bag of genres where each genre is paired with a frequency count. It is also used to train classifiers for genre classification.

The output space of our problem is the collection of articles in subreddits. The identification of representative articles for each subreddit is a challenging problem that requires a separate work. For simplicity, we have manually assigned the Top 50 subreddits to the genres of the referencing taxonomy non-mutually exclusively. Once we have converted both the input and output spaces into bags of genres from the referencing taxonomy, the recommendation of Reddit stories can be made based on the top K genres in a user's interest profile.

The main focus of this work is on the task of genre classification from user's tweets. Previously, we have used an ensemble of three classifiers: 1) a classic Naive Bayesian classifier, 2) a Naive Bayesian classifier trained only on the part-of-speech of sentences, and 3)

[1] https://twitter.com

[2] http://www.reddit.com/

[3] The numbers were extracted from the raw data downloaded from Reddit with the help of Dr. Arvind Srinivasan of ZL Technologies in San Jose, CA.

a Naive Bayesian classifier which will only make a decision if the probability $P(x) \geq 0.9$. In this paper, we have introduced a new algorithm to the genre classification from tweets by using the WordNet lexial database (Fellbaum 1998). In the proposed WordNet approach, no learning is required, instead we use the similarity distance computed from the WordNet class hierarchies to do genre classification. For the machine learning approach, in addition to the ensemble classifier, we have applied another two well-known machine learning classifiers, the SVM and Random Forests. The classifiers are evaluated by using real life cases collected from Twitter. Emprically, we have obtained similar results from both approaches for users with a large number of tweets, such as a couple thousands of tweets. For a small number of tweets, the WordNet approach did not perform as well as the machine learning approach.

The rest of the paper is organized as follows. A brief review of prior work related to genre classification is given in Section 2. In Section 3, we present data collection used in our experiments. Section 4 gives the detail of data preprocessing of tweets and training sets. Machine learning classifiers are presented in Section 5. The WordNet approach is given in Section 6. Section 7 discusses Reddit and some experimental results using real life test cases, followed by conclusion and references.

2 Prior work

Most genre classification works are related to text documents. In general, the concept of genre is regarded as a collection of documents with similar type or a group of texts sharing common communicative purpose, content, function or form (Kessler et al. 1997; Finn and Kushmerick 2006; Stamatatos et al. 2000; Karlgren and Cutting 1994). The topic of a text is what the text is about, and it is in theory regarded as orthogonal to genre (Stein and Meyer zu Eissen 2006). Empirically, it is shown that topic and genre are partially related (Finn and Kushmerick 2006). Many works have obtained good results by using bag of words as features for genre classification (Finn and Kushmerick 2006; Freund et al. 2006; Lewis 1992; Stamatatos et al. 2000). Syntactic features such as frequency of Part-Of-Speech (POS) has been used in (Finn and Kushmerick 2006; Feldman et al. 2009). Some research uses statistics derived from stylistic information such as frequency count of the most used punctuation marks (Stamatatos et al. 2000), and statistics derived from text such as character n-gram, sentence length, etc. (Freund et al. 2006; Karlgren and Cutting 1994). The benefits of using bag of words are simplicity and good performance when the topical distribution of documents are not varying (Finn and Kushmerick 2006; Lewis 1992). Some work applied genre classification as filters to improve the search of more relevant documents (Freund et al. 2006). Most of the previous works have applied machine learning algorithms approach to genre classification of text documents, and some works were on web pages (Meyer zu Eissen and Stein 2004; Qi and Davison 2009). Twitter data have been used in classification of Twitter users (Pennacchiotti and Popescu 2011) and analysis of genre for interaction (Westman and Freund 2010), which are of different emphasis from ours. In summary, bag of words and machine learning algorithms are popular representation and tools for genre classification.

3 Data collection

In this research, we collected data from four different resources: the Brown Corpus, technology articles from the Internet, Twitter, and Reddit. The Brown Corpus is a text collection

that was compiled in 1961 for linguistic research. It consists of 1,014,312 English words from 500 text samples classified into 15 genres: news, editorial, reviews, religion, hobbies, lore, government, learned, fiction, belles lettres, science fiction, mystery, adventure, romance, and humor. Since it is a well-annotated corpus, it is ideal for classifying tweets into genre interests.

The Brown corpus is dated, and therefore does not include many references to modern technology, which is a very popular topic. We use 40 random news articles, from 2014 and 2015, that are specifically about technology. This gives us a total of 26,201 unique words in our data bank formed from the Brown Corpus and our own technology corpus.

Table 1 shows the distribution of the sentences classified into each genre before and after oversampling. In order to create a more unbiased classifier, the oversampling is done uniformly instead of randomly, so for each sentence, a copy is added based on how many times there we wish to oversample. The total number of sentences in the genres of religion, reviews and technology are doubled and science fiction and humor are tripled. We choose uniform oversampling as to not accidently introduce biases towards certain words over others which fall under the same decision.

It is of note that we did oversample more to completely balance the data. In this scenario, we doubled the examples from news, editorial, religion, hobbies, lore, government, fiction, mystery, adventure, and romance. We quadrupled reviews and sextupled science fiction and humor. Belles lettres and humor remained the same. There were noticeable improvements in the classifier, however, we suspected it to be a result of overfitting and chose not to use this model.

Data from Twitter is collected via Python Twitter Tools by Mike Verdone (2015). Given a specific username, i.e. @username, we request as many tweets as possible from that user. We are limited to collecting the last 3,500 tweets per user as per restrictions from Twitter. This data is used to create an interest profile for each user.

Table 1 Distribution of sentences within the genres from the Brown Corpus and technology corpus used in our experiments

Genre	Original (59136)	OverSampled (68401)
Adventure	4637	4637
Belles lettres	7209	7209
Editorial	2997	4637
Fiction	4249	4249
Government	3032	3032
Hobbies	4193	4193
Humor	1053	3159
Learned	7734	7734
Lore	4881	4881
Mystery	3886	3886
News	4623	4623
Religion	1716	3432
Reviews	1751	3502
Romance	4431	4431
Science fiction	948	2844
Tech	1796	3592

The second column is the distribution of the original data set with total of 59,136 sentences, and the third column is the distribution after oversampling with total of 68401 sentences

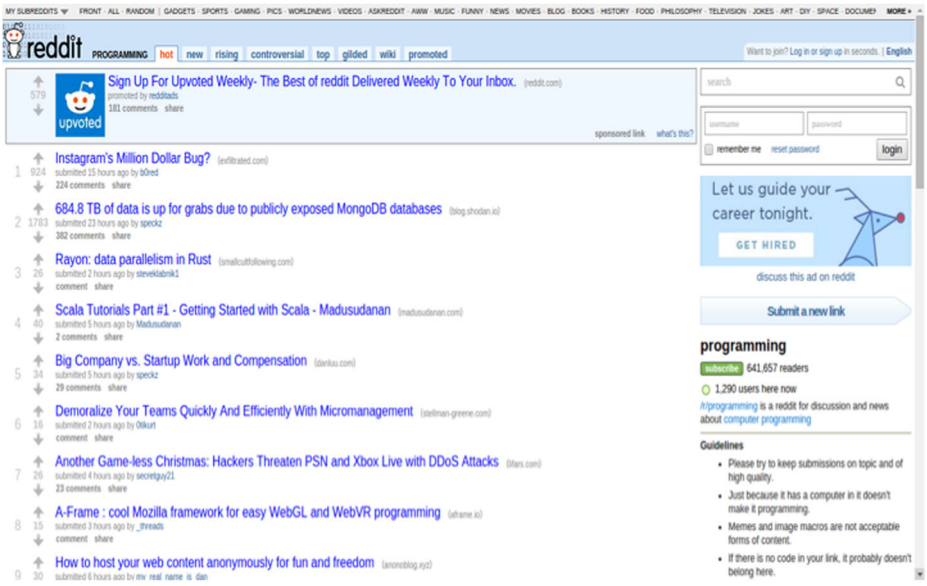

Fig. 1 Example of a subreddit from the programming page with some top trending articles

Reddit also has an API wrapper called PRAW, the Python Reddit API Wrapper (Boe 2015). We pull articles from the top 50 subreddits, which are sub communities defined by areas of interests. We have hand-tagged these subreddits with the 16 genres in our corpus so that we can recommend articles to the Twitter user. The hand tagging is done based on our own anecdotal understanding of the subreddits content. Figure 1 is an example subreddit page that contains articles that could be recommended. The top news stories in a given subreddit are determined by its readers whom upvote the thread if they find it interesting.

4 Preprocessing

4.1 Tools used

The following tools are used for our data preprocessing:

- NLTK (Natural Language Toolkit) is used to import the tagged Brown Corpus (Bird 2015).
- Pattern (from CLiPs) is a web mining module for Python which has tools for data mining, natural language processing, machine learning, network analysis, and visualization (DeSmedt and Daelemans 2015). Pattern and Scikit-learn[4] packages are used for training set transformation and machine learning tasks.
- Python string replacement is used for cleaning tweets.
- WordNet is a module that contains a database of English words linked by their semantic relationship (Fellbaum 1998). This provides us an ontology of word relationships which

[4] http://www.scikit-learn.org

allows us to determine similarity between two given words. For these experiments, we use the WordNet module implemented by pattern.en (DeSmedt and Daelemans 2015), a part-of-speech tagger. This means we are limited to only using nouns.

4.2 Twitter data

We preprocess and transform tweets from Twitter in a five step process.

1. Extract only the text of the tweet (140 max characters) from the JSON response.
2. Remove all words which start with an @. These correspond to a tweet being directed at a specific user. Which user it is directed towards is irrelevant data.
3. Remove all URLs using the regular expression:

```
\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*
```

4. Remove all non-alphabetic characters. Numbers and punctuation tend to be irrelevant. It also removes the # symbol in front of hashtags leaving only the term. This also has the convenient side effect of removing all emoticons.
5. Remove stopwords and stem using Porters algorithm.

For example, the tweet

```
WFAA's @johnmccaa says Terrell Owens is getting a $50 severance
from Indoor Football League team. #haha
```

will become

```
 WFAA Terrel Owen sever Indoor Footbal League team haha.
```

4.3 Training sets

The Brown Corpus with text files for 15 genres is imported via NLTK (Natural Language Toolkit). The 40 random technology articles we collected for the corpus are stored in a text file. They are read in and processed using a Python script. All the sentences are read into memory and processed into three training data sets. Each training set undergoes a different set of operations and transformations.

Training Set 1 is derived from the sentences in our genre-tagged corpus with oversampling ratios as shown in Table 1. For each sentence, stopwords are removed using the pattern.en stopwords dictionary (DeSmedt and Daelemans 2015). After all stop words have been removed, each remaining word is stemmed using Porters algorithm in pattern.vector (DeSmedt and Daelemans 2015). Testing of the classifiers concluded that Porters algorithm provided better results than simple lemmatization. The remaining sentence is then labeled with the appropriate genre.

Training Set 2 is derived from the key part-of-speech for each sentence in our corpus with similar oversampling ratios. For each sentence, a tagger tags each word in the sentence with a part-of-speech tag. From the tagged sentence, object phrase, subject phrase, and verb are extracted using the pattern.en module.

Table 2 shows the result of how processing the sentence "Gene Roddenberry created the Star Trek TV Series over 50 years ago," would be divided. Each object phrase, subject phrase, and verb are labeled with the appropriate decision label (which is derived from the sentence they belong to) and added to the training set as separate examples (so three total).

Table 2 The results of parsing the sentence Gene Roddenberry created the Star Trek TV Series over 50 years ago, into object phrase, subject phrase, and verb

Example	Part-Of-Speech	Decision label
Gene roddenberry	Subject phrase	Science fiction
Created	Verb	Science fiction
Star trek TV series	Object phrase	Science fiction

(Before stop words and stemming is applied)

Each example has stopwords removed. Each word in the example is stemmed using Porters algorithm. The original part-of-speech tag is discarded. For example, a word like "invent" might appear in verb form or noun form (inventor) and we do not want to say that it is a science fiction word only if it is a verb. Also, due to the stemming, the original part-of-speech and any tenses become irrelevant anyway.

Training Set 3 is used to create the WordNet classifier. For each genre, we scan over all sentences contained within that genre sample. We form a set of nouns that appear five or more times within those samples. After this, all stop words are removed from the set. Once finished, we have 16 sets of nouns which are not necessarily mutually exclusive. In fact, we expect that there are nouns between these sets which overlap.

The overlapping words are retained. Due to the uncertain nature of genre decision making, a noun may be associated with multiple genres. If we were to remove all intersecting words, we run the risk of removing words which are key nouns that represent a genre.

4.4 Feature set creation

In order to build a classifier, we need to define sentence features. This is accomplished by using the TF-IDF value of a word. The term frequency-inverse document frequency is a popular measure used to evaluate the importance a word is to a document in a corpus (Manning et al. 2008; Salton et al. 1975). The importance increases proportionally to the number of times a word appears in a document but then offset by the frequency of the word in the corpus. For example, though the words "the" and "is" may occur frequently in one text, they are not as important in this text since they occur frequently in other texts. The formula for TF-IDF is

$$w_{x,y} = tf_{x,y} \times \log(N/df_x)$$

where $tf_{x,y}$ is the frequency of word x in document y, df_x is the number of documents containing word x, and N is the total number of documents. A document for each genre is created and represented as a vector indexed by a dictionary of terms provided by the corpora and the value of each entry is its TF-IDF. These vectors are used to train the machine learning classifiers applied in our experiments.

5 Machine learning approach

We have applied three different machine learning algorithms in our experiments: SVM (Support Vector Machine), Random Forests, and an ensemble of three Bayesian classifiers.

Table 3 Results of 10-fold cross-validation on a SVM classifier using Training Set 1

Classifier	Accuracy	Precision	Recall
SVM linear	91.9 %	58.3 %	60.0 %

5.1 SVM classifier

Support Vector Machine classifiers divide points of data via hyper-planes. It is one of the popular machine learning classifiers for text classification, because text has many features, document vectors are sparse, and text categorization problems should be linearly separable (Stein and Meyer zu Eissen 2006; Meyer zu Eissen and Stein 2004; Boser et al. 1992; Docs.opencv.org 2015). The SVM classifier we used was configured with a linear kernel, so separation of groups was done using a linear function. The SVM classifier is built using Training Set 1. The result of 10-fold cross validation for this classifier is shown in Table 3.

The SVM classifier has performance metrics similar to the ensemble of Bayesian classifiers used in our previous work (Nguyen et al. 2015). In all test cases of our experimental results, the SVM classifier suggested technology as a user's top genre of interest, even if the user only expressed a lower interest in the genre. It is possible that the words used in the technology samples are more distinct than other genres, leading to the classifier being very sensitive to those words. Since we cannot confirm why this happens, it was not included in the ensemble of Bayesian classifiers. However, we do not discount that it may still be viable after the 'technology' bias is fixed. Because of this possibility, we do compare its standalone results to the other classifiers.

5.2 Random forests classifier

Random Forests is a popular decision tree approach to machine learning for classification and regression (Brieman 2001). The basic idea is to create an ensemble of decision trees generated by randomly selected subset of features in order to minimize the impact of variance in training sets. We used the scikit-learn package to generate a random forest ensemble consisting of 100 decision trees using Training Set 1. The result of 10-fold cross validation for this classifier is shown in Table 4.

5.3 Bayesian ensemble classifier

Ensemble methods are learning algorithms that use multiple classifiers and then classify by using a weighted vote for their decisions. Originally, ensemble methods were Bayesian averages but now there are methods such as error-correcting output coding, bagging, and boosting (Dietterich 2000).

As stated earlier, a text may actually fall under multiple genres. Since one classifier can only provide one result, we use multiple classifiers to capture any latent extra genres a text might fall under. In the end, we sum up the decisions and output a count for each genre. We

Table 4 Results of 10-fold cross-validation on a Random Forests classifier using Training Set 1

Classifier	Accuracy	Precision	Recall
Random forests	55.24 %	55.0 %	55.0 %

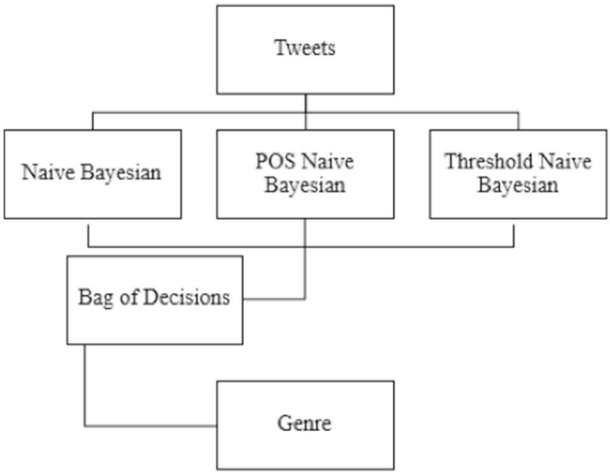

Fig. 2 The structure of Bayesian ensemble classifier in which a tweet is run through the three classifiers, results of each are summed in the bag of decisions

assume that even with false positive decisions being added to the bag, as long as the majority of decisions on a user's tweets are correct, the top genres in a user's profile will be correct.

We have used the ensemble of three Bayesian classifiers introduced in Nguyen et al. (2015), which consisting of a naive Bayesian classifier trained by using Training Set 1, a POS (Part-Of-Speech) naive Bayesian classifier trained by using Training Set 2, and a threshold biasing naive Bayesian classifier trained by using Training Set 1. Figure 2 is an overview of our ensemble classification system.

Tweets are collected from a user, preprocessed, transformed, and entered into each classifier. After the data is passed through all three classifiers, the decisions from all three are collected into a bag. Figure 3 is an example of what can happen in our ensemble approach. The example tweet will be preprocessed and transformed appropriately and entered into the three classifiers, Naive Bayesian, POS Naive Bayesian, and Threshold Naive Bayesian; decisions will be collected into a bag and tallied; and one or more genres will be predicted.

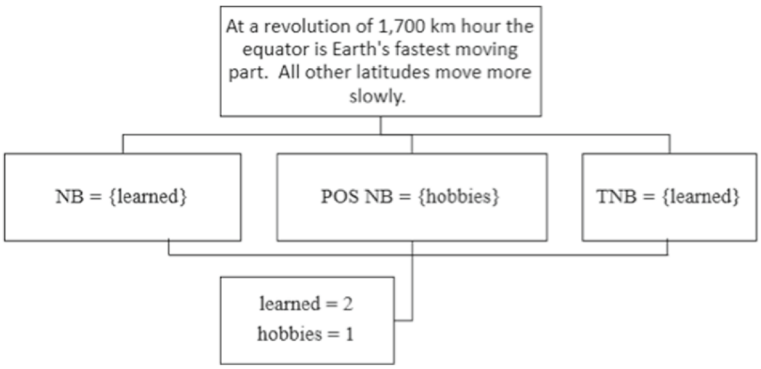

Fig. 3 Example of how genres are predicted for a tweet

In the example, the second classifier guesses wrong, but the biasing classifier causes the result to be in favor of the correct result.

The performance of each classifier used in the ensemble approach is shown in Table 5. Notice that the Threshold Naive Bayesian outperforms the other two classifiers in the area of precision and recall though accuracy has been sacrificed. This trade-off is typical for dealing with imbalanced data in machine learning algorithms. It is preferred, since the precision and recall rates are more relevant than the accuracy in our current application.

The threshold biasing classifier will only classify a new case if the probability $P(h|D) \geq 0.9$. Otherwise, it will not classify the case at all. In cases where users have very few tweets, this classifier would not be able to produce accurate results. We call it a "biasing" classifier because it biases the results of the ensemble towards the decisions with the highest chance of being correct. Each result produced from this classifier has a high probability of correctness whereas the other two classifiers will make a decision even under very low probabilities.

Since we use a bag approach, where all decisions are considered correct, the final interest profile is a list of name-value pairs of $\langle genre, count \rangle$ where genre is one of the 16 genres and count is the number of times it appeared in the bag of decisions. The following is a sample output of interest profile for all the tweets of one user:

```
{'lore': 764, 'belles_lettres': 1388, 'learned': 1333,
'hobbies': 792, 'news': 933, ....}
```

6 WordNet-based genre classification

6.1 Using WordNet for interest profile generation

Unlike the machine learning approach, which attempts to classify individual tweets using a model learned from previous example sentences, the WordNet method proposed in the following will focus specifically on nouns. As they teach in grade school, nouns refer to objects, places, and things. It is a natural assumption that nouns which occur frequently are representative of objects, places, and things which appear in a specific genre of text.

Let $n(*)$ be the set of all nouns in the English language supported by WordNet (Fellbaum 1998). Let $g(*)$ be the set of 16 genres (15 from the Brown Corpus plus technology). Our referencing corpus consists of genre-tagged sample documents from the Brown Corpus and technology articles downloaded from the Web.

Table 5 The statistics of using the three classifiers in the ensemble method

Classifier	Accuracy	Precision	Recall
Naive Bayesian	91.7 %	58.8 %	58.2 %
POS Naive Bayesian	92 %	57.2 %	55.8 %
Threshold Naive Bayesian	79.1 %	76.9 %	78.2 %

Note that the numbers for the threshold Naive Bayesian is only for cases where it did classify and ignores the cases when it predicts NONE

For a given user, we extract all nouns which appear more than five times within the user's tweets into a set of nouns. Note that the threshold for word frequency could be data and application dependent. For users with a small number of tweets, the word frequencies tend to be small. These tweets have been preprocessed beforehand so stopwords are not included. Let T be a set of pairs, $(x, freq(x))$, where x is a noun which appears in a given user's tweets, $freq(x)$ represents the frequency at which that noun appeared in the user's tweets, and $freq(x) \geq 5$. The set T denotes the interest profile of a user as a bag of nouns frequently appearing in the user's tweets.

For each genre g in $g(*)$, we extract a set U_g of all nouns from its corresponding sample documents in the referencing corpus with a frequency greater than five. Let C be a set of pairs, (g, U_g) where g is a genre in $g(*)$ and U_g is a subset of $n(*)$ such that each noun x in U_g, $freq(x) \geq 5$. The set C denote the referencing taxonomy of genres derived from the referencing corpus where each genre is represented by a bag of nouns. The sets of nouns associated with each genre are not necessarily mutually exclusive. This is expected. It is quite common for a word to appear in more than one genre due to the fuzzy nature of English literature and genre.

To determine which genres a user finds interesting, we assign a score to each genre g based on the similarity of nouns in T and U_g. The similarity of two nouns is measured by their distance in the WordNet class hierarchy. The result is denoted by the set P consisting of a list of genres $g_1, ..., g_{16}$, paired with a list of scores, $w_1, ..., w_{16}$. The set P represents the interest profile of a user as a bag of genres.

6.2 Algorithm for interest profile generation

In the following, we describe how to generate the interest profile from a user's tweets based on bag of words representation and WordNet similarity.

For each pair $(x, freq(x))$ in T, we construct a set called *labels* for storing all genres that x might be labeled. For each pair (g, U_g) in C, we assign the set U_g of nouns associated with genre g to the variable l. For each noun y in l, we calculate the WordNet similarity value between x and y by the distance of the two terms in the WordNet class hierarchies.

If that similarity value is greater than or equal to 0.9, we immediately assume that x can be labeled with g. We add g to the set of labels for x. Next we look up g in the set P and add the frequency of the noun x to the score for g. We break out of the innermost loop and move on to the next genre. It should be noted that when calculating the score, the original frequency of the noun y found in the Brown Corpus is not considered or normalized. It may be of interest to factor this value in for future improvements.

It is possible that a noun might fall under multiple genres. This is expected because, as stated earlier, the nouns associated with each genre are not mutually exclusive. It is also possible that the synonym sets for each noun are not mutually exclusive.

The reason we use a similarity threshold of 0.9 is to cut down on processing time. Though it may be possible to only assign one genre based on the maximal similarity in all genres, the computational time was too long to justify.

The algorithm terminates once all nouns in T have been processed. The set P is the final output, and the resulting scores are used to make recommendations. Higher scores indicate more interest in a topic than lower scores.

Algorithm 1

```
Input: n(*) = set of all nouns supported by WordNet
g(*) = { g1,..., g16} , set of referencing genre labels
T = { ( x, freq(x)) | x in a user's tweets, freq(x) >= 5 and x
in n(*)}
C = { ( g, Ug) | for all y in Ug, freq(y) >= 5 and y in n(*)}

Output:
P = { ( g1: w1),..., ( g16: w16)}

begin
for each genre g in g(*) do
    w[g] = 0; //genre scores are initialized to zero
for (x, freq(x)) in T do
{ labels = empty set;
   for (g, l = Ug) in C do
     for y in l do
     { if (similarity(x, y) >= 0.9) {
          labels = labels + { g} ;
          w[g] += freq(x);
         }
        break;}
      }
}
end;
```

7 Results and real life tests

7.1 Reddit

Submissions to Reddit are organized into a large number of subreddits, and the number of articles or news in each subreddit is also numerous. Currently, RedTweet does not classify the contents of subreddits automatically. In Nguyen et al. (2015), we have manually assigned genres to subreddits. Table 6 shows the top 50 subreddits in terms of popularity. The top genres of interest are chosen as they are the most likely points of interest and the limit of five subreddits is due to the limited amount of requests allowed by the API. Table 7 shows those top 50 subreddits organized into the 16 genres. These subreddits have been categorized by hand. Subreddits may fall under multiple genres. Stories recommended to the user are pulled from these subreddits.

For this research, we define "interest profile" as the Top-K (for K = 5) genres which a user is most likely interested in. Each tweet within a user's history is assigned labels by a classifier. These labels are aggregated together into a bag of genre-count pairs. The top five genres are then used to generate subreddits recommendation. For each genre in the user's

Table 6 The top 50 subreddits available at redditlist.com in the subscribers column

Top 50 subreddits				
Todayilearned	personalfinance	politics	pics	askreddit
Worldnews	art	mildlyinteresting	photoshopbattles	videos
Movies	oldschoolcool	nottheonion	wtf	diy
Music	listentothis	history	earthporn	fitness
News	internetisbeautiful	gadgets	adviceanimals	lifeprotips
Bestof	getmotivated	dataisbeautiful	space	books
Explainlikeimfive	creepy	futurology	funny	philosophy
Television	nosleep	documentaries	gifs	gaming
Sports	food	jokes	aww	showerthoughts
Technology	science	tifu	askscience	
			iama	

These 50 are hand labeled with one or more genres

interest profile, five random subreddits are selected which fall under that genre. The top ten articles from each subreddit are pulled into a bucket for each genre. A random subset of that bucket is recommended to the user. This subset is weighted based on the proportion that the respective genre took up in their interest profile. For example, if the classifier assigned "science fiction" appears twice as much as "romance," the user would be recommended twice as many science fiction threads than romance threads.

7.2 Real life tests

7.2.1 Experiment 1

In our previous paper, we applied the Bayesian ensemble classifier to the Twitter profile of two famous figures, Hillary Clinton (@hillaryclinton) and Neil DeGrasse Tyson (@neiltyson) (Nguyen et al. 2015). We could only verify loosely that the results were viable based on what we knew about these two people from their accomplishments, careers, etc. The number of top-K recommendation is fixed at $K = 5$. In this work, we have extended the experiment to include the use of Random Forests and WordNet classifiers.

The results for two well-known people who have very different amounts of tweets will allow us to compare classifier results of small tweet samples versus larger data. Hillary Clinton is a well-known political figure who would seem to have a great interest in news. Neil DeGrasse Tyson is an astrophysicist, cosmologist, author, and science communicator and has become a popular TV science expert. It seems appropriate that he would show a high interest in the areas of belles-lettres and learned, which contains the area of science. The samples for results presented were taken on July 1, 2015 (a prior version of these results were taken on April 25, 2015 but are now being updated to match a more recent tweet sample). Hillary Clinton, @hillaryclinton, had 902 tweets at the time. Neil DeGrasse

Table 7 Subreddits organized into genres

News	Editorial	Reviews	Religion	Hobbies	Iore
Todayilearned	AskReddit	AskReddit	books	AskReddit	gaming
Worldnews	videos	videos	Documentaries	pics	Showerthoughts
Movies	bestof	gaming	philosophies	videos	Jokes
Music	television	movies		gaming	history
News	politics	Music		EarthPorn	nosleep
Bestof	mildlyinteresting	books		books	creepy
Explainlikeimfive	DIY	television		AdviceAnimals	
Television	Fitness	LifeProTips		television	
Sports	Showerthoughts	Fitness		sports	
Politics	history	food		DIY	
Mildlyinteresting	Futurology	gadgets		Fitness	
Nottheonion	Documentaries	Documentaries		food	
History	personalfinance	listentothis		photoshopbattles	
Gadgets		Art		InternetIsBeautiful	
Dataisbeautiful				history	
Futurology				dataisbeautiful	
Documentaries				listentothis	
Personalfinance				Art	
Art				OldSchoolCool	
Government	Mystery	Adventure	Romance	Humor	Tech
AskReddit	movies	pics	AskReddit	funny	AskReddit
Worldnews	explainlikeimfive	videos	movies	pics	gaming
Politics	books	gaming	aww	videos	technology
Philosophy	mildlyinteresting	movies	books	gifs	space
Worldnews	space	gifs	InternetIsBeautiful	WTF	photoshopbattles
News	nosleep	EarthPorn	GetMotivated	AdviceAnimals	InternetIsBeautiful
	creepy	books		mildlyinteresting	gadgets
		AdviceAnimals		Showerthoughts	dataisbeautiful
				Jokes	Futurology
				tifu	science
				photoshopbattles	
				GetMotivated	
				nottheonion	
Learned	Fiction	Belles lettres	Science fiction		
AskReddit	gaming	pics	gaming		
Todayilearned	movies	todayilearned	movies		
Science	books	books	technology		
IAmA	photoshopbattles	LifeProTips	books		
Technology	nosleep	Showerthoughts	space		
Askscience	Art	tifu	creepy		
Explainlikeimfive	creepy	GetMotivated	nosleep		
EarthPorn		philosophy			

Table 7 (continued)

Learned	Fiction	Belles lettres	Science fiction
AdviceAnimals		OldSchoolCool	
LifeProTips		dataisbeautiful	
Mildlyinteresting		listentothis	
DIY		Art	
Space		IAmA	
InternetIsBeautiful		Documentaries	
History		worldnews	
Gadgets		history	
Dataisbeautiful			
Futurology			
Documentaries			
Personalfinance			
Philosophy			

Tyson, @neiltyson, had 3242 accessible tweets (limited by Twitter's API) at the time. Tables 8 and 9 show the output for Hillary Clinton and Neil DeGrasse Tyson, respectively, from the WordNet decision algorithm side-by-side with the results from the Bayesian ensemble and Random Forests classifiers. Percentages given in the table show what percentage of total decisions that genre took up. As one can tell, the WordNet decision algorithm produces similar top 5 genres for both test cases. In Table 8, we can see that 3 of the 5 are the same, though they appear in different orders, with News being the top genre. In Table 9 we can see that all 5 are the same, with Learned and Belles Lettres being the top 2 in both. The Random Forests outputs are around 60 % in agreement with the other two classifiers.

7.2.2 Experiment 2

In order to verify our results in a more concrete way, we have collected six samples of Twitter profiles from regular people. Due to time constraints, the sample is not large enough to have a meaningful statistical interpretation, which will be addressed in our future work. However, the experiments do provide some useful insights regarding the WordNet-based approach to genre classification.

Table 8 The top five genres predicted for Hillary Clinton and their respective percentages of total decisions made

Bayesian ensemble	WordNet output	Random forests
News: 22.2 %	News: 9.0 %	Belles Lettres: 22.02 %
Belles Lettres: 10.2 %	Editorial: 8.3 %	News: 13.93 %
Editorial: 10.9 %	Learned: 8.3 %	Romance 10.45 %
Reviews: 9.6 %	Lore: 8.1 %	Tech 10.22 %
Tech: 9.2 %	Belles Lettres: 7.9 %	Learned 7.87 %

Table 9 The top five genres predicted for Neil DeGrasse Tyson and their respective percentages of total decisions made

Bayesian ensemble	WordNet	Random forests
Belles Lettres: 16.5 %	Learned: 8.1 %	Belles Lettres: 15.58 %
Learned: 16.2 %	Belles Lettres: 7.9 %	Romance: 10.58 %
News: 10.9 %	Lore: 7.8 %	Tech: 10.55 %
Hobbies: 9.6 %	News: 7.3 %	News: 9.04 %
Lore: 9.2 %	Hobbies: 6.9 %	Learned: 8.76 %

Rather than focusing on the subreddit threads which our RedTweet ensemble outputs, we focus on the top-K genres specified by the users with varying values for K. Reddit threads and suggestions change constantly so it is impossible to determine effectiveness based on the threads recommended. Instead the genres the users claim they are interested is compared against the interest profile generated by the different classifiers evaluated in this experiment.

Real life Twitter users were asked to provide answers to a survey administered via Google Forms. These questions were:

1. What is your Twitter username? (i.e. @username)
2. Of the 16 genres, select all genres of which you are interested. This includes what you like to watch (TV/web), read (books/articles), write, etc.
3. Of those genres selected in question 2, select your top 5 interests.

The survey was taken by six individuals. Five of them were selected for analysis. The sixth was excluded because that user (@ambiixrox) only had one tweet ever made. These five individuals have a diverse number of tweets ranging between 100 and 3000 tweets. This diversity in tweet count gives us a reasonable approximation of how well a classifier might work across a spectrum of (prolific and non-prolific) Twitter users. The individuals vary in

Table 10 Respondent information. The respondent's username, number of tweets, and the number of responses they provided to Q2[1]

Case #	Username	# of tweets	# picked in Q2
1	Schrommboy33	3189	13
2	beorn_identity	148	8
3	dylanconqueso	484	8
4	sudoxnerdx[2]	131	3
5	jt_borders	2808	7
6	ambiixrox[3]	1	*

[1]Thanks to Mitch Schromm, Nickolas Beorn, Dylan Turner, Michael Griffith, Jared Borders, and Amber Cericola-Woods for taking part in the survey.

[2]User only indicated 3 genres of interest. Extra guesses were considered false positives.

[3]User had one tweet and was therefore excluded from results.

Table 11 Results of recall measures of Experiment 2 on the five respondents

Case #	Ensemble	SVM	Random forest	WordNet
1	11/13	11/13	11/13	10/13
2	5/8	5/8	5/8	3/8
3	4/8	5/8	4/8	4/8
4	1/3	1/3	1/3	0/3
5	5/7	5/7	4/7	4/7

background but all fit within the 18 - 25 age group. The summary of survey results can be found in Table 10.

The users' tweet histories were sampled on August 31st, 2015. The RedTweet method was run on each respective sample. The classifier in every case produces a ranking of the 16 genres, which will be matched with the K genres specified by a user. Each result indicates what the classifier guesses to be the user's top-K genres of interest. These results are then compared to the user's responses in Q2 and Q3 using two measurements: recall and precision.

Values of Q2 denote the number of genres that a user might be interested in, so it is used to compute the rate of recall. Values of Q3 corresponds to the actual interested genres of a user; therefore, it is used to compute the rate of precision. Strictly speaking, the measures used here are relative measures, in the sense that they are conditioned on the number of genres specified by a user. However, the classifiers are trained by a fixed number of 16 genres, and we assume that a classifier always returns an aggregated ranking of a list of 16 genres. Therefore, the Top-K genres specified by a user are used to match the top-K genres predicted by a classifier. In this experiment, the value of K is varying, not necessarily fixed.

For Experiment 2, the performance of classifiers is evaluated in terms of Recall and Precision. The results are shown in Tables 11 and 12. Classifiers from the machine learning approach seem to have similar performance, and they are better than the WordNet classifier in both measures. The fourth respondent appears to be a difficult case for all the classifiers, and it has the smallest number of tweets and interested genres. In addition, the WordNet classifier has poor performance over cases with small number of tweets, especially, in precision. It seems that machine learning approach is not as sensitive to the size of tweets as the WordNet classifier.

Table 12 Results of precision measures of Experiment 2 on the five respondents

Case #	Ensemble	SVM	Random forest	WordNet
1	2/5	2/5	2/5	1/5
2	4/5	2/5	3/5	1/5
3	2/5	3/5	2/5	0/5
4	1/3	1/3	1/3	0/3
5	3/5	4/5	3/5	2/5

8 Conclusion

In this paper, we have applied natural language processing tools and machine learning algorithms to develop a recommendation system for subreddit articles based on interest profiles derived from users' tweets. The genre-tagged Brown corpus updated with technology related texts is used as a referencing taxonomy for representing interest profiles. Thus, the recommendation problem can be treated as a task of genre classification. We introduced a simple WordNet-based genre classifier based on a similarity measure derived from the WordNet ontology. We used two real life test cases to evaluate performance of the WordNet classifier in comparison to some of the well-known machine learning classifiers. Our preliminary experiments show that all evaluated classifiers have similar performance when the number of tweets are sufficiently large, around 1000 or more, and it also shows that the WordNet approach has poor precision when the number of tweets is small. Further experiments are required in order to establish a statistical confirmation.

References

Bird, S. (2015). 'Natural Language Toolkit NLTK 3.0 documentation', Nltk.org. [Online]. Available: http://www.nltk.org/. [Accessed: 27- Apr- 2015].

Boe, B. (2015). PRAW: The Python Reddit Api Wrapper PRAW 2.1.21 documentation, Praw.readthedocs.org. [Online]. Available: https://praw.readthedocs.org/en/v2.1.21/. [Accessed: 27- Apr- 2015].

Boser, B.E., Guyon, I.M., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers, In Haussler, D. (Ed.) *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory* (pp. 144–152). Pittsburgh, PA: ACM Press.

Brieman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

DeSmedt, T., & Daelemans, W. (2015). Pattern — CLiPS, Clips.ua.ac.be. [Online]. Available: http://www.clips.ua.ac.be/pattern. [Accessed: 27- Apr- 2015].

Dietterich, T. (2000). Ensemble Methods in machine learning. *Multiple Classifier Systems*, *1857*, 1–15.

Docs.opencv.org (2015). Introduction to Support Vector Machines OpenCV 2.4.11.0 documentation. [Online]. Available: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. [Accessed: 27- Apr- 2015].

Feldman, S., Marin, M.A., Ostendorf, M., & Gupta, M.R. (2009). Part-of-speech histograms for genre classification of text. In *2009. ICASSP 2009. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4781–4784): IEEE.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*: MIT Press.

Finn, A., & Kushmerick, N. (2006). Learning to classify documents according to genre. *Journal of the American Society for Information Science and Technology*, *57*(11), 1506–1518.

Francis, W., & Kucera, H. (1979). *Brown Corpus Manual*, 1st edn. Providen ce: Brown University.

Freund, L., Clarke, C.L.A., & Toms, E.G. (2006). Towards genre classification for IR in the workplace. *Proceedings of the 1st International Conference on Information Interaction in Context*, (p. 3036). New York, NY.

Karlgren, J., & Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. *Proceedings of the 15th Annual Meeting of the Association for Computational Linguistics*, (p. 10711075). Morristown, NJ.

Kessler, B., Nunberg, G., & Schtze, H. (1997). Automatic detection of text genre. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, (pp. 32–38). Morristown, NJ.

Lewis, D.D. (1992). Feature selection and feature extraction for text categorization. *Proceedings of the workshop on Speech and Natural Language*, 212–217.

Manning, C., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. New York: Cambridge University Press.

Meyer zu Eissen, S., & Stein, B. (2004). Genre classification of web pages. *KI 2004: Advances in Artificial Intelligence*, 256–269.

Nguyen, H., Richards, R., Chan, C.-C., & Liszka, K.J. (2015). *RedTweet: Recommendation Engine for Reddit*. Paris, France: MSNDS Workshop 2015. (to appear Proceedings of IEEE/ACM ASONAM 2015).

Pennacchiotti, M., & Popescu, A.na.-M.aria. (2011). A machine learning approach to twitter user classification. *ICWSM*, *11*, 281–288.

Qi, X., & Davison, B.D. (2009). Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*, *41*(2), 12.

Salton, G., Wong, A., & Yang, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Text genre detection using common word frequencies. *Proceedings of the 18th Conference on Computational Linguistics*, 808–814.

Stein, B., & Meyer zu Eissen, S. (2006). Distinguishing topic from genre. *Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06)*. Graz: Journal of Universal Computer Science.

Taylor, L. (2014). 10 Remarkable Twitter Statistics for 2015, Social Media Consultant — Social Media Agency — Social Marketing. [Online]. Available: http://lorirtaylor.com/twitter-statistics-2015/. [Accessed: 27- Apr- 2015].

Verdone, M. (2015). Python Twitter Tools (command-line client and IRC bot), Mike.verdone.ca. [Online]. Available: http://mike.verdone.ca/twitter/. [Accessed: 27- Apr- 2015].

Westman, S., & Freund, L. (2010). Information interaction in 140 characters or less: genres on twitter. *Proceedings of the third symposium on Information interaction in context*: ACM.