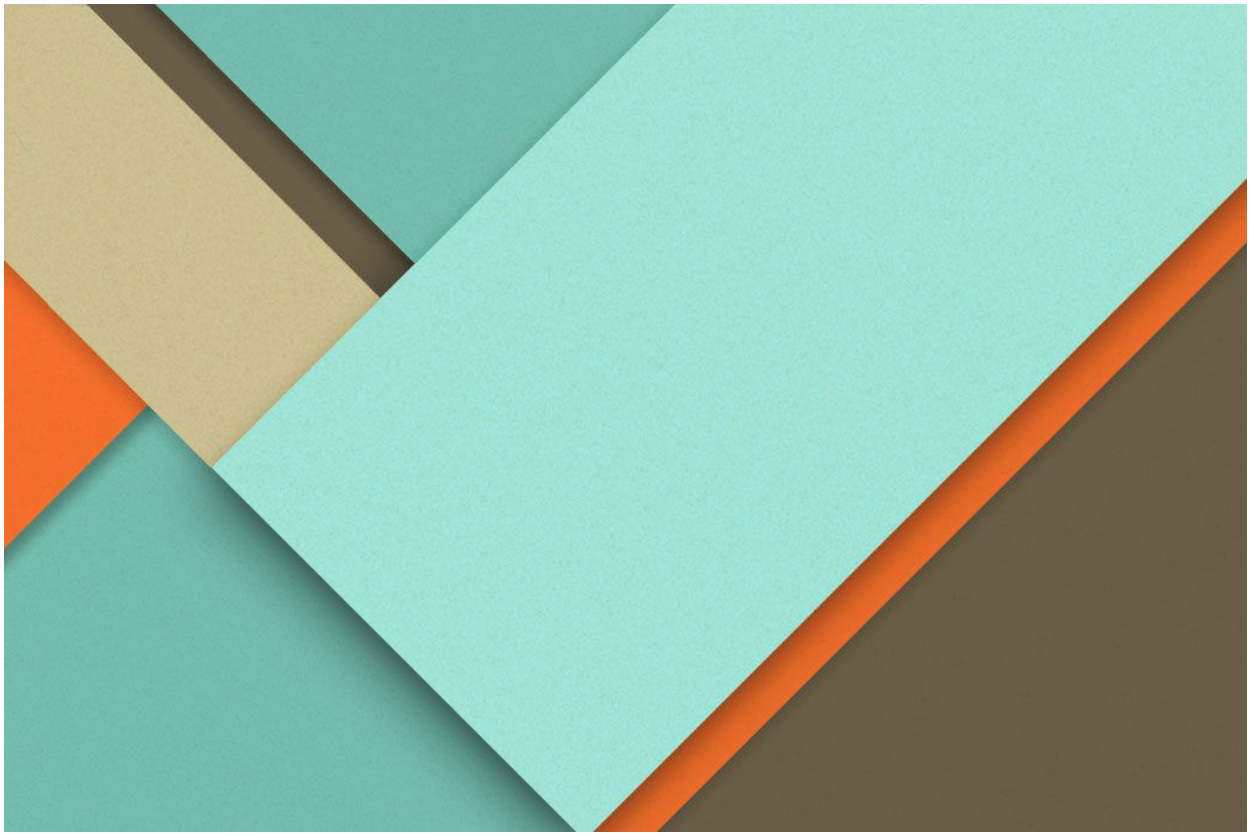

Inference of User Interests from Multiple Social Networks

Steven Kester Yuwono (A0080415N)

Choo Jia Le (A0116673A)

1. Overview

This assignment aims to design an effective multi-label multi-source classifier and then infer interests of users based on a given dataset collected from multiple social networks. In this study, three social media platforms are explored namely Twitter, Facebook, and LinkedIn. There are several challenges, such as pre-processing of raw data, extracting and aggregating suitable informative features from multiple sources, and designing effective classifier.

2. Implementation Details

2.1. Training Data

The training data used by the program contains texts in various formats. Twitter data is in JSON format, LinkedIn in HTML page format, and Facebook data is in unprocessed plain text.

Please refer to the training data summary below:

	# Users	# Interest Groups	# Sources
Training	420	20	3
Testing	150	20	3

2.2. Information Extraction / Features List

Twitter data is parsed and the following features are extracted from each tweet:

No	Features	Descriptions
1	Text	The content of the tweet
2	Retweets	The number of retweet for this particular tweet
3	Timezone	The timezone of the user's location when this tweet is posted
4	Friends Count	The number of users this account is following (AKA their "followings")

5	Followers Count	The number of followers this account currently has.
6	Favourites Count	The number of tweets this user has favorited in the account's lifetime.
7	User ID	ID of the person posting this tweet

Table 1. Feature List extracted from a tweet

After exploring all twitter's features mentioned above. We found out that only the text is meaningful and useful. Hence we will only use the tweet text in our system.

Facebook data is in plain text, with many unnecessary words.

Please refer to the following example:

> Timothy Nichols changed his profile picture. Share 9 people like this. Remove Lisa Nichols Well how did ya like being fired? Hahaha 1 July 25 at 2:21am Remove Timothy Nichols Haha. Well said July 25 at 2:32am Remove Trudy Patterson Boy I wish I would have done that a long time ago. 1 July 25 at 9:28pm"

Regular Expressions were implemented to detect unnecessary words/phrases which add noise to the data like the date, the number of likes, notification of changing profile picture, see more comments, etc.

After cleaning, it will look like this :

> Lisa Nichols Well how did ya like being fired? Hahaha Timothy Nichols Haha. Well said Trudy Patterson Boy I wish I would have done that a long time ago.

LinkedIn data is in HTML format. We used HTML parser to detect each section of LinkedIn profile. Please find the text section list below:

No	Section Title	Descriptions
1	Full Name	Full name of the user
2	Location	Geographical location
3	Current Position	The position currently held
4	Summary	The summary paragraph/text
5	Skills	List of skills acquired
6	Current Experience	Current job roles and description

7	Past Experiences	A list of past job roles and descriptions
8	Honors and Awards	List of honors and awards attained
9	Organizations	List of organizations which the user belongs to
10	Interests	List of interests

Table 2. Section List extracted from a LinkedIn profile

After some experiment to evaluate the importance of each LinkedIn section above, we have concluded that only summary, skills, and interests are useful. Including the rest of the section would make the performance poorer. Hence we only use text from the aforementioned 3 sections in our classifier.

2.3. Pre-Processing of Text Data

Before the text from multiple sources are used for classification, pre-processing are done to clean the data and get the best feature representation possible.

The following steps are performed:

1. Tokenization
2. Convert all words to lowercase
3. Removed url links with regex
4. Removed stopwords

2.4. Learning the representation of text

Using the whole text as a feature vector will result in a high dimension vector (thousands). Since too many features are not desirable (known as curse of dimensionality), we have explored to use LDA (Latent Dirichlet Allocation) to perform topic modelling. Topic modelling (LDA) is unsupervised, and hence able to run without any manual labelling/training.

We used MALLET[1] toolkit to perform topic modelling (LDA), and obtain the topic composition of documents which then used as the features for the learning classifier.

2.5. Multi-source fusion

We will use both early fusion and late fusion approach to combine features from multiple sources and evaluate their performance.

Early fusion is done by combining feature vectors from multiple sources before running the classifier/machine-learning algorithm.

[1] http://mallet.cs.umass.edu/topics.php

Late fusion is done using Ensemble learning. We assigned some weight to the confidence score with respect to each class in each source. The confidence scores are then added together to form the final score.

The weight is learned by using **Genetic Algorithm** (very similar to "Hill Climbing" Optimization) to get the best accuracy.

$$Score(l) = \sum_{i=0}^{k} \frac{P(l)_i \times w_i}{k}$$

2.4. Learning the model

We explored **SVM** as our machine-learning algorithm using LIBSVM library. We used binary approach to solve the multi-label classification problem. One classifier instance is built for each class (interest group) with binary output. To obtain the confidence score for all the classes, binary classifier for each class is run and confidence score is recorded for that particular class and then rank the confidence scores across the classes.

After implementing the classifier, we realized that there is an unbalanced dataset problem. E.g. if there are 20% of negative training dataset and 80% positive training dataset, the classifier's positive-class prediction value will be very close to 80%.

Therefore **minority-oversampling technique** is adopted. This is to ensure that the classifiers prediction value is fair for all classes. We will use more of the minority training dataset to achieve equal number positive and negative training instances.

3. Testing and Evaluation

3.1. Methodology

To develop and tune our classifiers, we use the training set to train the classifier, and test the classifier on using the test set given.

We will be using P@K, R@K, and S@K evaluation measure where K=1 to 10.

P@K represents the proportion of the top K recommended interests that are correct.

$$P@K = \frac{|C \cap R|}{|C|}$$

R@K represents the proportion correct number of predicted interests (in the top K) out of the actual number of interests for that particular user.

$$R@K = \frac{|C \cap R|}{|R|}$$

S@K represents the mean probability that a correct interest is captured within the top K recommended interests.

$$S@K = 1 \; if \; (|C \cap R| \geq 1) \; : \; 0 \; otherwise$$

3.2. Results

We try to optimize the number topics for topic modelling (LDA). We use P@K, R@K, S@K where K is 5 to evaluate the performance. K is chosen to be 5 because we believe higher performance achieved at lower K is more important than higher K.

Number of Topics	5	10	20	35	50	100	200	300	1000
Facebook P@5	0.356	0.367	0.339	0.360	0.343	0.355	0.332	0.348	0.333
Facebook R@5	0.257	0.265	0.245	0.260	0.248	0.257	02.40	0.252	0.241
Facebook S@5	0.947	0.907	0.887	0.887	0.867	0.907	0.893	0.887	0.927
LinkedIn P@5	0.251	0.244	0.392	0.492	0.489	0.501	0.476	0.497	0.489
LinkedIn R@5	0.181	0.176	0.284	0.356	0.354	0.363	0.344	0.360	0.354
LinkedIn S@5	0.693	0.713	0.940	0.953	0.933	0.933	0.940	0.967	0.960
Twitter P@5	0.392	0.449	0.440	0.423	0.409	0.413	0.415	0.403	0.420
Twitter R@5	0.284	0.325	0.318	0.306	0.296	0.299	0.300	0.291	0.304
Twitter S@5	0.913	0.927	0.940	0.940	0.913	0.927	0.913	0.893	0.860

Table 3. Classifier's performance with respect to number of topics (LDA)

From the result (Table 3) above, it is observed that the optimum number of topics is 100. Larger number of topics yield poorer performance and converges to a certain value. Therefore we will be using 100 as the number of topics in further evaluation below.

Using Genetic Algorithm, we found that most of the weights favour LinkedIn and Twitter. Please find some of the best weights below:

Facebook Weight	LinkedIn Weight	Twitter Weight	Late Fusion P@5	Late Fusion R@5	Late Fusion S@5
24.753	695.971	141.733	0.476	0.344	0.947
25.392	964.318	285.489	0.473	0.342	0.947
37.385	784.276	224.330	0.472	0.341	0.947
3.470	340.835	129.659	0.472	0.341	0.947

Table 4. Late Fusion Best Weights

From the result above, we can see that the first set of weights yield the best result. Therefore we will use the best weights for late fusion.

The overall performance of the classifier using combination of sources are shown below. FB, LI, Tw, EF, LF denotes Facebook, LinkedIn, Twitter, Early-Fusion, and Late-Fusion respectively. Early-Fusion and Late-Fusion utilize all three sources (FB, LinkedIn, and Twitter).

K =	1	2	3	4	5	6	7	8	9	10	Ave
FB-P@K	0.307	0.300	0.318	0.330	0.355	0.368	0.354	0.358	0.360	0.350	0.340
LI-P@K	0.440	0.483	0.513	0.500	0.501	0.493	0.485	0.467	0.456	0.431	0.477
Tw-P@K	0.527	0.477	0.438	0.413	0.413	0.416	0.408	0.399	0.391	0.403	0.428
EF-P@K	0.433	0.417	0.364	0.360	0.375	0.397	0.405	0.401	0.401	0.395	0.395
LF-P@K	0.453	0.490	0.487	0.473	0.476	0.458	0.456	0.438	0.426	0.406	0.456

Table 5. Overall P@K

K =	1	2	3	4	5	6	7	8	9	10	Ave
FB-R@K	0.044	0.087	0.138	0.191	0.257	0.319	0.359	0.415	0.469	0.506	0.278
LI-R@K	0.064	0.140	0.223	0.289	0.363	0.428	0.491	0.540	0.593	0.623	0.375
Tw-R@K	0.076	0.138	0.190	0.239	0.299	0.361	0.413	0.462	0.509	0.582	0.327
EF-R@K	0.063	0.121	0.158	0.208	0.271	0.344	0.410	0.464	0.522	0.572	0.313
LF-R@K	0.066	0.142	0.211	0.274	0.344	0.397	0.462	0.506	0.554	0.587	0.354

Table 6. Overall R@K

K =	1	2	3	4	5	6	7	8	9	10	Ave
FB-S@K	0.307	0.540	0.687	0.813	0.907	0.940	0.973	0.980	0.993	1.000	0.814
LI-S@K	0.440	0.733	0.887	0.927	0.933	0.967	0.973	0.987	0.993	0.993	0.883
Tw-S@K	0.527	0.667	0.787	0.827	0.927	0967	0.980	0.993	1.000	1.000	0.867
EF-S@K	0.433	0.640	0.740	0.820	0.900	0.947	0.947	0.960	0.987	0.987	0.836
LF-S@K	0.453	0.760	0.900	0.940	0.947	0.960	0.980	1.000	1.000	1.000	0.894

Table 7. Overall S@K

As observed from the table above, LinkedIn performs the best if compared from the Average Recall and Precision where K is 1 to 10. However, Late-fusion performs best in the S@K evaluation metric, according to the average S@K from 1 to 10.

We have explored topic modelling with bigram as well. However it did not significantly improve the performance (worse instead). Therefore bigram is not included in our final system. Please find the result below:

K=	5	10
Bigram FB-P@K	0.368	0.353
Bigram LI-P@K	0.485	0.452
Bigram TW-P@K	0.412	0.413
Bigram EF-P@K	0.325	0.382
Bigram LF-P@K	0.465	0.406
Unigram FB-P@K	0.355	0.350
Unigram LI-P@K	0.501	0.431
Unigram TW-P@K	0.413	0.403
Unigram EF-P@K	0.375	0.395
Unigram LF-P@K	0.476	0.406

Table 8. Bigram and Unigram Comparison

4. Conclusion

From the result shown above, it is observed that LinkedIn is the most useful source, followed by Twitter, and then Facebook. We have also found the 100 is the best number of topic for topic modelling in this case.

After trying to get the optimum weight for Late-Fusion by Genetic Algorithm, we observed that Late-Fusion's performance is limited by the performance of the best source, i.e. it is very difficult to outperform all the three sources significantly. Therefore Late-Fusion is not effective in this case. Early Fusion is also found not to be very effective as the performance is always worse than the best single-source classifier (LinkedIn).

It is also observed that noise (words that are not important) in the data causes a drop in the performance of the classifier, therefore we did data cleaning and chose the most useful text features from LinkedIn.

Bigram was explored and found to be not very useful in this task, therefore it is not adopted in our system.

From the results above, LinkedIn performs the best overall, followed by Late-Fusion and, Twitter.