Data Mining Capstone Task 3

September 27, 2015

1 Abstract

In this task, I applied **SegPhrase** to perform the dish list expansion for cuisine *Chinese*, and managed to score 12.

Besides **SegPhrase**, I also experimented and compared the results from topic mining (**ToPMine**), and mutual information calculations (**Word2Vec**). According to an empirical judgement and the course autograder, **SegPhrase** produced the best quality. After reading the paper of **SegPhrase**, I am trying to explan / understand why **SegPhrase** out-performs other approaches.

2 Implementations

2.1 Dish Expansion Using Topic Mining

I tried the "Comparative Text Mining" topic modeling algorithm implemented by the java tools.

However, the result is in the form of unigram term probablities. While dish names are mostly multi-gram terms. So the result is actually useless for this task.

I also tried the **TopMine** algorithm, with minimum-support being 10, max-size of phrase being 8, the number of topics being 5, and all other parameters being default as shipped from here.

Although from the file *topPhrases.txt*, I can get a list of multi-gram terms representing the phrases of top occurrences, by examining the content, one can easily come to a conclusion that the quality is not so good. Because although there are some dish terms, many of them are common phrases that are irrelevant to dishes.

Table 1 shows the top-10 phrases. Obviously "food was good" is not a dish phrase.

If we try to understand why topic mining is not producing good-quality results here, we can notice that dish-name-expansion focuses on *dishes*. Although review texts may be talking about dishes (i.e. the topic terms may include dishes), there are quite many noises. For example, the judgements of a restaurant or a dish can introduce terms like "pretty good", which are not what we want.

So empirically, we can realise that we not only need to mine popluar phrases, we also have a specific target.

2.2 Dish Expansion Using Word Association

I here tried **Word2Vec** in python, with at most 2-gram terms as training input for word2vec, with all default parameters.

After having got the trained model, I calculate the most similar phrases of all positive-labeled terms (i.e. the dishes).

dim sum
Chinese food
fried rice
Chinese restaurant
food was good
egg rolls
pretty good
orange chicken
lunch specials
Panda Express

Table 1: Top phrases of **ToPMine**

string beans
crispy noodles
salt pepper
broccoli
favorite dish
snow peas
beef broccoli
moo goo
moo shu
sesame chicken
egg fu

Table 2: Top phrases of **Word2Vec** using unigram and bigram input

Table 2 shows the top-10 phrases.

Empirically, the result is better than the one of topic mining approach, that most of the top phrases are real dish names. But there are still some noises. For instance, "favorite dish" is one of the top-10 phrases.

Since dishes with their ranking review terms are common to co-occur, it may not be surprising to see "favorite dish" to show up here, even though it is just a general term.

And another weakness of this approach is that it is time-consuming to train **Word2Vec** with higher-gram inputs. And dish names are sometimes very long. So in consideration of time and quality, I did not finalise my experiments here.

2.3 Dish Expansion Using SegPhrase

2.3.1 Implementation approach

When using **SegPhrase**,

- I manually collected 220 Chinese dish names from Wikipedia links: Chinese soups, Chinese desserts, and Chinese dishes;
- I applied auto-labeling feature in SetPhrase, using these 220 extra dish names combined with positive-labeled dish names in Task 3.1;

pork buns
noodle soup
hot pot
dim sum
bbq pork
shaved ice
pork belly
peking duck
chow mein
beef noodle soup

Table 3: Top quality phrases of **SegPhrase**

- I enabled wordnet noun word cleanup feature;
- I used default parameters for everything else. (support threshold is 10, max iteration is 5.)

The submitted list is a combination of:

- positive manual-labeled dish names;
- positive auto-labeled dish names;
- all manually collected dish names from Wiki;
- top-quality phrases mined by **SegPhrase**.

I scored 12 based on this approach.

Table 3 shows the top-10 high quality phrases (i.e. dish names) mined by **SegPhrase**. As is shown, all of them are indeed Chinese dish names.

2.3.2 Analysis

To analyse why **SegPhrase** can achieve a good result, we need to understand what **SegPhrase** actually does.

In a top-level view, as I understand it, SegPhrase firstly does frequent phrase mining, and do feature extration upon them. Then, it does classification and phrase segmentation, and repeat. This yields the final results. In real implementation, wordnet tool is also applied to filter out non-noun phrases.

The features involve many considerations of generating high-quality phrases, which not only include the mutual information calculations, text retrieval techniques (TF-IDF), but also other subtle factors such as if terms are quoted, or if terms start or end with a stopword.

By doing classification, **SegPhrase** focuses on the dish name mining, therefore in the final result, it is likely to see dish-irrelevant phrases are assigned with a lower quality value. This resolves the problem of topic mining approaches.

Also, the features that are used for classification have included many considerations, including but not limited to topic mining considerations (TF-IDF, frequent terms mining), word association

(mutual information calculations, which in this case can mine dish-relevant phrases). So it may not be surprising to see **SegPhrase** outperforms other algorithms with separate considerations.

Also it is quite convenient to add more features, such as POS tagging from natural language processing, which might boost the final quality even more.

Besides, in **SegPhrase** implementation, it provides an auto-labeling feature using external knowledge information (in this case, Wiki dish names). It actually uses KMeans clustering algorithm and manages to introduce a prior knowledge to the algorithm when doing phrase mining. This is a useful when manual labeling data is insufficient.

2.3.3 Opinion of the Results

In my opinion, the expanded dish names by **SegPhrase** is useful and makes sense, judging by empirical judgements and autograder grading results. One can set a threshold of quality and use the top phrases safely.

If one demand even better results, she can apply more features to the **SegPhrase** framework.

The high-quality expanded dish names can be used to understand which dishes are popular, in terms of region, time, social networks, and so on.

When these connections are constructed and analyzed, it may be benefitial to the restaurants, since they can better understand what kind of roles they are acting, how they should manage food ingredients suppliments over time, and what kind of customers they are most likely to serve.