Milestone 1: Feature extraction

Zhanwang Chen

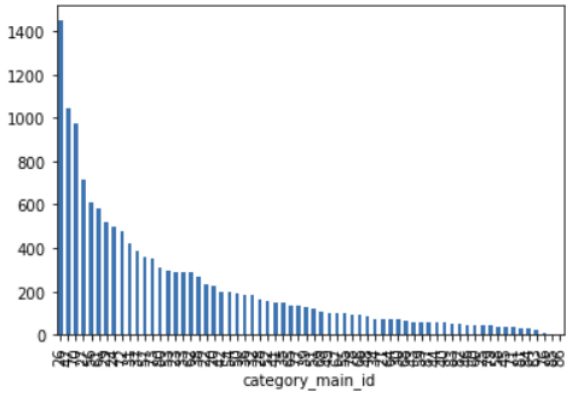
1.Feature extraction methodology

- **1.1 Overview of the data**

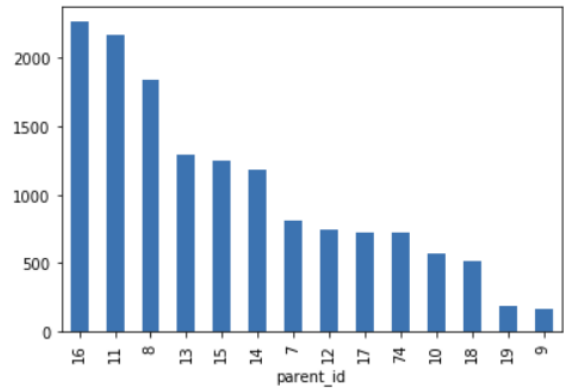
16052 rows of questions in question_train.csv, 1625 rows of garbage due to missing values or broken csv structure. 723 rows of duplicate questions.
The "category_id", "parent_id" in category.csv are used for constructing class hierarchy.

66 different category_main_id, 14 different parent_id, namely 66 subcategories, 14 parent category.

The distribution of number of question in different category are rather uneven, this is one of challenge for classification task. For example, the training test set split, some category may have very rare training sample or test sample and lead to poor results.



66 subcategories



14 parent categories

1.2 Pre-Processing Phase

The pre-processing steps include: tokenization, stop-word elimination, punctuations elimination, spell error correction, stemming and German characters transforming, were performed on the sms data.

1.3 Cleanup and correct spell-errors

This SMS contains spell-errors, incorrect punctuations, abbreviations, smiley face, special characters which are non-text etc. All numbers were remove during cleanup.
Enchant are used for spell errors, but this step takes super long time.

1.4 Stemming

SnowballStemmer were used for the German token stemming. Stemming reduces a word to its root or base form and thus reduces the number of word features to be processed.
This step significantly reduces the size of the data to be processed, saves time and memory space, and thus shorten the training time.
But this also has the risk that lose important semantic information which contribute to classification.

1.5 TF-IDF

sklearn TfidfVectorizer was used for get TF-IDF features.
TF-IDF is short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
It is based on the heuristic that a term is a good discriminator if it occurs frequently in a document but does not occur in many distinct documents of the corpus

1.6Word embedding

It is a limitation that the methods above learn such representations ignore the morphology of words, by assigning a distinct vector to each word.

1.7 Word2vec

Word2vec aims to make use of semantic relationships between words.

Thus in word embedding, stemming step can be omitted.

Its skipgram model, where each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations.

using word2vec can get the vectors for a list of tokens from SMS.
Machine learning algorithms require the text input to be represented as a fixed-length vector, like the common fixed-length vector representation: sklearn CountVectorizer , TfidfVectorizer
Given a sentence, how to get the vector of the sentence from the vector of the tokens in the

sentence?
Two methods were found:

1. Average of Word2Vec vectors : take the average of all the word vectors in a sentence. This average vector represent sentence vector.
2. Average of Word2Vec vectors with TF-IDF : take the word vectors and multiply it with their TF-IDF scores., and take the average represent sentence vector.

Average is to handle variable length sentences

Additional, using Doc2Vec[1] to train on the dataset, then use the sentence vectors.
It learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents, has potential to overcome the weaknesses of bag-of-words models like losing the ordering of the words and ignoring semantics.

1.8 FastText

In word2vec, there will be a out of vocabulary problem, you can not get word vector that did not appear in training data.

But FastText provide any vector representations for words not in the dictionary.
words are represented by the sum of its substrings, as long as the unknown word is made of known substrings, model will calculate a representation of it. Thus compare to bag-of-word model, spell errors has less effect on fasttext model.

Fasttext supervised model can get a vector representation for sentence, experiment was made, model was train on labeled questions but accuracy is very low, could be due to not enough learning epoch.

2 Feature selection strategy

Mutual information and chi square test in scikit-learn are chosen for feature selection. SelectPercentile class are used to measure how many percent of features should keep, depend on the classifier. Appropriate dropping can improve the accuracy, because get rid of some noise. After the specific point, the accuracy begin to drop down. The parameters can be tuned by gridsearch.

2.1 Mutual information

Mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables measures the information that X and Y share: It measures how much knowing one of these variables reduces uncertainty about the other.
However, omitting features that don't have mutual information (MI) with this concept might cause you to throw the important features. There are cases in which a single feature is useless but given more features it becomes important.

Consider a concept which is the XOR of some features. Given all the features, the concept is

totally predictable. Given one of them, you have 0 MI.

```
['alt',
 'bedeutet',
 'bekomm',
 'best',
 'deutsch',
 'deutschland',
 'frau',
 'geburtstag',
 'gibt',
 'gross',
 'heisst',
 'jahr',
 'kommt',
 'lang',
 'mann',
 'mensch',
 'nam',
 'pass',
```

The words that has high Mutual information score

2.2 Chi square test

Chi Square Test is used in statistics to test the independence of two events.
Here, we want to test whether the occurrence of a specific token and the occurrence of a specific class are independent.

If the two events are dependent, we can use the occurrence of the feature to predict the occurrence of the class. We aim to select the features, of which the occurrence is highly dependent on the occurrence of the class.

The higher value of the chi2 score, the more likelihood the feature is correlated with the class, thus it should be selected for model training.

```
['abtreib',
 'adolf',
 'advanced',
 'angibt',
 'anwendungsgebiet',
 'autogrammadress',
 'beitragsbemessungsgrenz',
 'bettina',
 'bislang',
 'canada',
 'candy',
 'clk',
 'colada',
 'colbi',
 'dacht',
 'depression',
```

The words that has high chi2 score

2.3 over-fitting.

Overfitting is an issue here, especially some of the categories contain some little samples, the model might very easy to get wrong prediction for unseen data.
Cross-Validation was used to minimize over-fitting, although it is not enough.

2.4 Multi-class feature selection

Here most the model above used one-against-all method.

2.5 significant differences between the categories

in some subcategories, some are very difficult to distinguish.

2.6 Interpretable

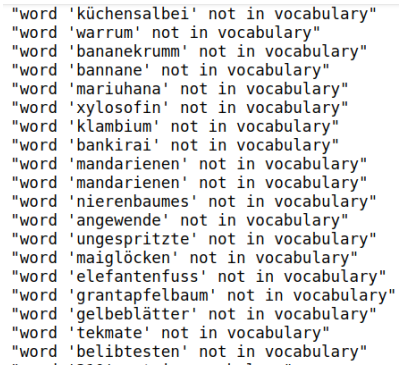
The features get from above models are interpretable.

3 Discussion

- **challenges of the data set?**

The distribution of number of question in different category are rather uneven, in parent category and subcategory.

- **What does (not) work and why (not)?**



Word2vec out of vocabulary problem

Word2vec can't get word vector that did not appear in training data.

- **What do you think is possible on the data set? Do you think this will be useful in business?**

It is possible that there are some questions that related to some business product like shoes, sports equipment users

We can do market analysis on the data, aligning the users into a distinct segment and can tailor the needs according to the users. find a segment of customers based on vulnerability and the business could offer them with special offers and enhance satisfaction. Recommendation system can recommend related products to the users or show related ads according to users' interest.

In terms of text classification, we can use this kind of method to classify a SMS or email as legitimate or mark it as spam.

With the rapid growth of online text information, text classification is one of the key techniques for organizing text data, extract the valuable information and hidden patterns from users' text information.

- **Are there things you would like to try but (until now) have not tried?**

1. Using Named-entity recognition (NER) locate and classify named entities in text names of persons, organizations, locations, brand name that can be used as important feature for classification.

2. Make use of WordNet to enrich the semantic meaning, capture the relations between the words, thus enhance machine learning algorithms.

References

[1] Distributed Representations of Sentences and Documents
http://cs.stanford.edu/~quoc/paragraph_vector.pdf
[2] Efficient Estimation of Word Representations in Vector Space
https://arxiv.org/abs/1301.3781
[3] Enriching Word Vectors with Subword Information
[4] Bag of Tricks for Efficient Text Classification
[5] FastText.zip: Compressing text classification models
[6] Multi-class feature selection for texture classification