任务描述：使用亚马逊商品评论，进行切分、标准化（消除标点符号、大小写转换、去停用词等）、词干提取、词形还原以及高频词统计的训练。
输入：使用开源数据索引，下载Gift Cards五星评论（2972条）
https://nijianmo.github.io/amazon/index.html (https://nijianmo.github.io/amazon/index.html)
输出：每一步都需要输出，如
Tokenization: ["don't", 'hesitate', 'to', 'ask', 'questions', '.', 'he', 'works', 'happily', '.']
Normalization: ['dont', 'hesitate', 'ask', 'questions', 'works', 'happily']
Stemming: ['dont', 'hesit', 'ask', 'quest', 'work', 'happy']
Lemmatization: ['dont', 'hesitate', 'ask', 'question', 'work', 'happily']
Freq: 统计词频，按顺序从高到低列出前十个单词

Step 1 Tokenization:

In [2]:

```
import re
import nltk.tokenize as tk

with open (r'Gift_Cards_5.json','r') as file:
    data=file.read()
data=re.findall(r'"reviewText": "(.*?)"',data) #提取reviewText的内容，其中？的作用是实现非贪婪匹配

Tokenization = []   #存储全部的单词
for e in data:
    Tokenization += tk.word_tokenize(e)   #逐个拆分再添加
print(Tokenization)
```

```
['Another', 'great', 'gift', '.', 'Gift', 'card', 'for', 'my', 'daughter', 'Nic
e', 'present', 'My', 'niece', 'loved', 'this', 'birthday', 'greeting/gift', 'car
d', '.', 'fine', 'as', 'a', 'gift', '.', 'I', 'would', 'have', 'preferred', 'som
e', 'more', 'choices', '.', 'great', 'Very', 'cute', 'design', 'and', 'enjoyed',
'by', 'recipient', '.', 'I', 'used', 'the', 'text', 'option', 'to', 'send', 'thes
e', 'last', 'minute', 'gift', 'cards', 'to', 'my', 'Granddaughters', '(', 'via',
'their', 'mom', "'s", 'phone', ')', '.', 'Works', 'really', 'well', ',', 'you',
'get', 'a', 'confirmation', 'email', 'that', 'it', 'has', 'been', 'received',
',', 'and', 'a', 'confirmation', 'that', 'the', 'cards', 'have', 'been', 'redeeme
d', '.', 'Granddaughter', "'s", 'very', 'happy', 'with', 'the', 'card', 'design',
'.', 'I', 'love', 'the', 'options', 'you', 'have', 'when', 'you', 'send', 'Amazo
n', 'gift', 'cards', '!', 'This', 'was', 'for', 'a', 'gift', 'and', 'it', 'was',
'well', 'received', '.', 'Great', 'way', 'to', 'send', 'a', 'last', 'minute', 'bi
rthday', 'gift', 'to', 'someone', 'through', 'email', '.', 'Cant', 'go', 'wrong',
'with', 'Amazon', 'gift', 'cards', '!', 'guess', 'it', 'was', 'appreciated', '?',
'I', 'bought', 'several', 'of', 'these', 'cards', 'for', "'stocking", 'gifts',
"'", 'for', 'the', 'church', 'staff', '.', 'Glad', 'that', 'Amazon', 'actually',
'had', 'a', 'Real', 'Christmas', 'themed', 'card', '.', 'was', 'gift', 'great',
'they', 'are', 'gift', 'cards', 'Xmas', 'gift', '.', 'This', 'was', 'a', 'birthda
```

Step 2 Normalization:

In [3]:

```
from nltk.corpus import stopwords

pattern=re.compile("[^a-zA-Z0-9\n ]")      #数字字符的正则匹配
Normalization = []
for e in Tokenization:
    e = re.sub(pattern,"",e).lower()      #将所有非数字字符的符号转化为空，大小写转换
    e = tk.word_tokenize(e)               #文本标记化/分词
    e = [w for w in e if(w not in stopwords.words('english'))]  #去停用词
    Normalization += e                    #t添加到输出
print(Normalization)
```

```
['another', 'great', 'gift', 'gift', 'card', 'daughter', 'nice', 'present', 'niec
e', 'loved', 'birthday', 'greetinggift', 'card', 'fine', 'gift', 'would', 'prefer
red', 'choices', 'great', 'cute', 'design', 'enjoyed', 'recipient', 'used', 'tex
t', 'option', 'send', 'last', 'minute', 'gift', 'cards', 'granddaughters', 'via',
'mom', 'phone', 'works', 'really', 'well', 'get', 'confirmation', 'email', 'recei
ved', 'confirmation', 'cards', 'redeemed', 'granddaughter', 'happy', 'card', 'des
ign', 'love', 'options', 'send', 'amazon', 'gift', 'cards', 'gift', 'well', 'rece
ived', 'great', 'way', 'send', 'last', 'minute', 'birthday', 'gift', 'someone',
'email', 'cant', 'go', 'wrong', 'amazon', 'gift', 'cards', 'guess', 'appreciate
d', 'bought', 'several', 'cards', 'stocking', 'gifts', 'church', 'staff', 'glad',
'amazon', 'actually', 'real', 'christmas', 'themed', 'card', 'gift', 'great', 'gi
ft', 'cards', 'xmas', 'gift', 'birthday', 'gift', 'everyone', 'loved', 'getting',
'christmas', 'thanks', 'xmas', 'stocking', 'stuffer', 'loved', 'easy', 'gift', 'k
new', 'recipient', 'would', 'love', 'nice', 'sent', 'one', 'relatives', 'town',
'thanked', 'came', 'time', 'exactly', 'described', 'nice', 'card', 'gift', 'lot
s', 'thank', 'nieces', 'cute', 'love', 'gift', 'cards', 'fast', 'delivery', 'ok',
'looking', 'card', 'great', 'would', 'liked', 'come', 'tin', 'one', 'ones', 'orde
red', 'card', 'bent', 'half', 'could', 'nt', 'done', 'post', 'office', 'box', 'go
od', 'experiance', 'nice', 'look', 'always', 'great', 'gift', 'one', 'time', 'con
```

Step 3 Stemming:

In [4]:

```
import nltk
Stemming = []
for token in Normalization:
    pt_stem= nltk.stem.porter.PorterStemmer().stem(token)
    Stemming.append(token)
print(Stemming)
```

```
['another', 'great', 'gift', 'gift', 'card', 'daughter', 'nice', 'present', 'niec
e', 'loved', 'birthday', 'greetinggift', 'card', 'fine', 'gift', 'would', 'prefer
red', 'choices', 'great', 'cute', 'design', 'enjoyed', 'recipient', 'used', 'tex
t', 'option', 'send', 'last', 'minute', 'gift', 'cards', 'granddaughters', 'via',
'mom', 'phone', 'works', 'really', 'well', 'get', 'confirmation', 'email', 'recei
ved', 'confirmation', 'cards', 'redeemed', 'granddaughter', 'happy', 'card', 'des
ign', 'love', 'options', 'send', 'amazon', 'gift', 'cards', 'gift', 'well', 'rece
ived', 'great', 'way', 'send', 'last', 'minute', 'birthday', 'gift', 'someone',
'email', 'cant', 'go', 'wrong', 'amazon', 'gift', 'cards', 'guess', 'appreciate
d', 'bought', 'several', 'cards', 'stocking', 'gifts', 'church', 'staff', 'glad',
'amazon', 'actually', 'real', 'christmas', 'themed', 'card', 'gift', 'great', 'gi
ft', 'cards', 'xmas', 'gift', 'birthday', 'gift', 'everyone', 'loved', 'getting',
'christmas', 'thanks', 'xmas', 'stocking', 'stuffer', 'loved', 'easy', 'gift', 'k
new', 'recipient', 'would', 'love', 'nice', 'sent', 'one', 'relatives', 'town',
'thanked', 'came', 'time', 'exactly', 'described', 'nice', 'card', 'gift', 'lot
s', 'thank', 'nieces', 'cute', 'love', 'gift', 'cards', 'fast', 'delivery', 'ok',
'looking', 'card', 'great', 'would', 'liked', 'come', 'tin', 'one', 'ones', 'orde
red', 'card', 'bent', 'half', 'could', 'nt', 'done', 'post', 'office', 'box', 'go
od', 'experiance', 'nice', 'look', 'always', 'great', 'gift', 'one', 'time', 'con
```

Step 4 Lemmatization:

In [5]:

```
import nltk.stem as ns
Lemmatization = []
lemmatizer = ns.WordNetLemmatizer()
for token in Stemming:
    Lemmatization.append(lemmatizer.lemmatize(token))
print(Lemmatization)
```

```
['another', 'great', 'gift', 'gift', 'card', 'daughter', 'nice', 'present', 'niec
e', 'loved', 'birthday', 'greetinggift', 'card', 'fine', 'gift', 'would', 'prefer
red', 'choice', 'great', 'cute', 'design', 'enjoyed', 'recipient', 'used', 'tex
t', 'option', 'send', 'last', 'minute', 'gift', 'card', 'granddaughter', 'via',
'mom', 'phone', 'work', 'really', 'well', 'get', 'confirmation', 'email', 'receiv
ed', 'confirmation', 'card', 'redeemed', 'granddaughter', 'happy', 'card', 'desig
n', 'love', 'option', 'send', 'amazon', 'gift', 'card', 'gift', 'well', 'receive
d', 'great', 'way', 'send', 'last', 'minute', 'birthday', 'gift', 'someone', 'ema
il', 'cant', 'go', 'wrong', 'amazon', 'gift', 'card', 'guess', 'appreciated', 'bo
ught', 'several', 'card', 'stocking', 'gift', 'church', 'staff', 'glad', 'amazo
n', 'actually', 'real', 'christmas', 'themed', 'card', 'gift', 'great', 'gift',
'card', 'xmas', 'gift', 'birthday', 'gift', 'everyone', 'loved', 'getting', 'chri
stmas', 'thanks', 'xmas', 'stocking', 'stuffer', 'loved', 'easy', 'gift', 'knew',
'recipient', 'would', 'love', 'nice', 'sent', 'one', 'relative', 'town', 'thanke
d', 'came', 'time', 'exactly', 'described', 'nice', 'card', 'gift', 'lot', 'than
k', 'niece', 'cute', 'love', 'gift', 'card', 'fast', 'delivery', 'ok', 'looking',
'card', 'great', 'would', 'liked', 'come', 'tin', 'one', 'one', 'ordered', 'car
d', 'bent', 'half', 'could', 'nt', 'done', 'post', 'office', 'box', 'good', 'expe
riance', 'nice', 'look', 'always', 'great', 'gift', 'one', 'time', 'convenient',
```

Step 5 Freq:

In [6]:

```
from collections import Counter
result = Counter(Lemmatization).most_common(10)
result
```

Out[6]:

```
[(’gift’, 1700),
 (’card’, 1127),
 (’great’, 712),
 (’love’, 413),
 (’good’, 283),
 (’amazon’, 277),
 (’christmas’, 186),
 (’nt’, 185),
 (’like’, 173),
 (’perfect’, 158)]
```