Exercise 1

a)

$$c^* := argmax_c \log(p(c)) + \sum_w N(c,w) * \log p(w \mid c)$$

The Text-Classification process is divided into 3 / 4 steps (4 if an external vocabulary is used):

1. read vocabulary (if provided): in this step all words from the vocabulary file are inserted into the internal dictionary of words. This also means that no other words can be add to the dictionary anymore! Additionally it is possible to use the first n words of a vocabulary file only.

2. read training data file: here the training data file is read. Thereby all categories that appear are generated and each category is assigned a CountStructure, which is needed for storing the word counts for each word in every category. During this step for all texts in the training data set the word counts are stored for all words that appear in the texts of one category. This also means that some words that are part of the dictionary will have a count of 0 (sparse training data!). If no additional vocabulary was provided all words that appear will be add to the dictionary!

3. read test data file: in this step the program checks the test data set for words that have not appeared in the training data set. These words are also added to the global dictionary file which is used in every CountStructure assigned to any category. If a word is added to the dictionary, all CountStructures will also have this word in "their"dictionary (dictionary is a shared pointer). Including the words from the test data is necessary (and only done) if no dictionary is given, because after adding the test data words to the dictionary the counts of the words in all categories are smoothed: the number of occurrences of each word in the dictionary for every category is increased by 1, such that every event (word w in category c) has a quantity of at least 1. The relative frequency $p(w|c)$ is computed by: $\frac{N(c,w)}{\sum_{w'} N(c,w')}$. This guarantees that after smoothing the number of occurrences the probabilities still sum up to Unity. The relative frequency for each category c is defined as following: $p(c) := \frac{N(c)}{\sum_{c'} N(c')}$, which is the number of texts of a category divided by the total number of texts in the training data.

4. categorization of texts in test data set: at least the texts from the test file are assigned to the best fitting category.

Data structures:

- Dictionary: word $\leftrightarrow$ index
- CountStructure: index $\leftrightarrow$ quantities and sum of all quantities of all words in this CountStructure

b)

Error rates depending on vocabulary size: (test data: **3974 texts**)

dictionary size:	10^2	$5*10^2$	10^3	$5*10^5$	10^4	$5*10^4$	93508	102752
absolute error:	1911	1359	1308	1953	2585	3432	3490	3664
error rate:	0.48088	0.34197	0.32914	0.49144	0.65048	0.86361	0.87821	0.92199
log(0)-error:	2	59	144	1202	2048	3249	3340	3561

Problem:

$$confidence(c) := \underbrace{N(c)*\log(p(c))}_{\text{does not change if vocabulary size grows}} + \sum_w N(c,w)*log(\underbrace{p(w\mid c)}_{\text{if not seen in training} \Rightarrow =0})$$

But if: $p(w \mid c) = 0 \Leftrightarrow \log(p(W \mid c)) = -\infty$

If we have a greater vocabulary the number of unseen events increases if we do not increase the amount of training data (**sparse training data**)!

Solution: Erasing 0-occurences

$$p(w \mid c) := \frac{N(c,w)}{\sum\limits_{w'} N(c,w')}$$

$$p'(w \mid c) := \frac{N'(c,w)}{\sum\limits_{w'} N'(c,w')}$$

with: $N'(c,w) := N(c,w) + 1$

This new probability values $p'(w \mid c)$ still are a valid distribution!

c)

Error rates depending on vocabulary size: (test data: **3974 texts**)

dictionary size:	10^2	$5*10^2$	10^3	$5*10^5$	10^4	$5*10^4$	93508	102752
absolute error:	1904	1256	1082	785	731	608	610	610
error rate:	0.47911	0.31605	0.27227	0.19753	0.18395	0.15300	0.15350	0.15350
log(0)-error:	0	0	0	0	0	0	0	0

d) ConfusionMatrix

```
   | C1    C2    C3    C4    C5    C6    C7    C8    C9    C10   C11   C12   C13   C14   C15   C16   C17   C18   C18   C20
-------------------------------------------------------------------------------------------------------------------------------
C1 | 0.805 0     0     0     0     0     0     0     0     0     0     0     0.005 0.005 0.015 0.010 0     0.01  0.005 0.24
C2 | 0     0.76  0.104 0.01  0.025 0.080 0.02  0.005 0     0     0     0.010 0.055 0.015 0.015 0.005 0     0.005 0.005 0.005
C3 | 0     0.025 0.617 0.03  0.015 0.040 0     0.005 0     0     0     0.005 0     0.005 0     0.005 0     0     0     0
C4 | 0     0.05  0.098 0.815 0.035 0.035 0.06  0     0     0     0     0     0.04  0     0     0.005 0     0     0     0
C5 | 0     0.015 0.027 0.065 0.884 0.020 0.015 0     0     0     0     0.005 0.005 0     0.005 0     0     0     0     0
C6 | 0     0.03  0.071 0.005 0     0.788 0.01  0     0     0.005 0     0.005 0.005 0     0     0     0     0     0     0
C7 | 0     0     0     0.03  0.010 0     0.815 0.015 0.015 0     0     0.005 0.01  0     0     0     0     0     0     0
C8 | 0     0     0     0     0.005 0     0.01  0.93  0.015 0.005 0     0     0.01  0.005 0     0     0     0     0     0
C9 | 0.005 0     0     0     0     0.010 0.01  0.01  0.955 0.005 0     0     0.015 0     0.005 0     0.005 0.005 0     0
C10| 0     0     0.016 0     0     0     0     0     0     0.965 0.005 0     0     0     0     0     0     0.005 0     0
C11| 0     0     0     0.005 0     0     0.005 0     0     0.01  0.97  0     0     0.01  0     0.005 0     0     0     0
C12| 0     0.055 0.016 0     0     0.005 0     0.01  0     0     0.005 0.939 0.055 0     0.005 0     0.025 0.005 0.005 0
C13| 0     0.03  0.005 0.035 0.005 0     0.04  0.005 0.01  0     0     0.005 0.775 0.015 0.01  0     0     0     0     0
C14| 0     0.005 0     0     0     0.005 0     0     0     0     0.005 0.005 0.015 0.895 0.01  0     0     0.005 0     0
C15| 0.005 0.015 0     0.005 0     0.010 0     0     0.005 0     0.005 0.005 0.005 0.015 0.905 0.005 0     0     0.01  0.005
C16| 0.02  0     0     0     0     0     0     0     0     0.005 0.005 0     0.005 0.005 0     0.939 0     0.01  0     0.065
C17| 0     0.005 0     0     0.005 0.005 0     0.005 0     0     0     0.010 0     0.005 0     0     0.91  0     0.135 0.045
C18| 0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0.925 0.06  0
C19| 0.005 0.005 0.021 0     0.015 0     0.01  0.015 0     0.005 0.005 0.005 0     0.01  0.025 0.005 0.020 0.03  0.715 0.035
C20| 0.16  0.005 0.021 0     0     0     0.005 0     0     0     0     0     0     0.015 0.005 0.020 0.040 0     0.065 0.605

C1 = alt.atheism
C2 = comp.graphics
C3 = comp.os.ms-windows.misc
C4 = comp.sys.ibm.pc.hardware
C5 = comp.sys.mac.hardware
C6 = comp.windows.x
C7 = misc.forsale
C8 = rec.autos
C9 = rec.motorcycles
C10= rec.sport.baseball
C11= rec.sport.hockey
C12= sci.crypt
C13= sci.electronics
C14= sci.med
C15= sci.space
C16= soc.religion.christian
C17= talk.politics.guns
C18= talk.politics.mideast
C19= talk.politics.misc
C20= talk.religion.misc
```

In the ideal case only the diagonal should be filled with "1"s all other fields in the matrix should have been "0"! The columns show the true categories and in the rows the hypothesized categories are shown. If a field is not "0" that is not part of the diagonal, there has been a false assignment of the category to the text.

We are also able to identify clusters of categories which have lots of words in common. This leads to the phenomenon that the categories of these clusters are often flipped with each other. E.g. alt.atheism and talk.religion.misc: 16% of the texts of alt.atheism are assigned to talk.religion.misc. The other way round is even bigger: 24%. Assignments to other classes are much lower.

An other cluster if formed by the categories which deal with computers: comp.* and science: sci.*. The clusters of categories which can be recognized are all in similar super-classes: the categories alt.atheism and talk.religion.misc seem to deal with religious or philisophical questions and thus might have a similar domain-specific vocabulary. This might lead to the incorrect assignment of the categories because the word counts of several texts differ from the word counts of the training data.

e)

The results on the spam-corpus show that in 63 of 743 texts the wrong category is assigned. The confusion matrix is:

```
                correct category:
                ham             spam
ham             0.973   0.181
spam    0.026   0.819
```

This shows that most of the errors appear in the assignment of spam texts to ham. The probability of an assignment of ham texts to spam is much less. The error rate on the spam-corpus is 8.84354%.