All the experiment data, file and program can be found in Github link:
https://github.com/junchaoIU/CISC7021/tree/master/Assignment1

NLP Assignment 1 Report

WU JUNCHAO
University of Macau
Mc25653@um.edu.mo

1 Environment Installation

1.1 VMware and Ubuntu Installation

Download the Ubuntu system[1] and VMware[2].

Load the Ubuntu system CD in VMware, configure basic information after startup, and install the virtual machine and Linux system.

1.2 Install GCC Compiler

Enter the console with the following command to install the gcc compiler:

sudo apt install gcc
sudo apt-get update

1.3 Install Tcl (Tool Command Language)

Download Tcl on the official website (http://www.tcl.tk/), extract it to the Home directory, and execute: *cd tcl8.6.12/unix/* (enter tcl8.6.12/unix/)

Execute the command under the tcl8.6.12/unix directory: *./configure* (configure Tcl)

Execute the command under the tcl8.6.12/unix directory: *Make* (compile the source package)

Execute the command under the tcl8.6.12/unix directory: *sudo make install* (installation)

1.4 Install SRILM

Download SRILM[3] on the official website, and press it to the Home directory (take the author's virtual machine as an example: /home/wujunchao/srilm/)

Modify the MakeFile file:

- SRILM = /home/wujunchao/srilm
- MACHINE_TYPE := i686-m64

Modify the srilm/common/makefile.i686-m64 file:

- GAWK = /usr/bin/gawk

Execute the command under the srilm directory: *make World* (compile all documents and modules)

Execute the command under the srilm directory: *make test* (compile and test, if there is no problem, the installation is successful)

After passing the above steps, the experimental environment will be loaded into /home/wwujunchao/srilm/bin/i686-m6, and subsequent experiments will be carried out under this path.

2 Priori Basic Data Cleaning

- Convert all English text to lowercase.
- Replace the English abbreviation. Mainly replace “'re”, “'ve”, “'m”, “'d”, “'ll”, “'t” with “ are”, “ have”, “ am”, “ would”, “ will”, “ not”.
- Perform digital cleaning and replace Arabic numerals with English. For example, 1 is replaced by one.
- Symbol cleaning, cleaning punctuation marks. Mainly include +, =, -, %, ?, !, (,), ^, / , ; etc.
- Make a symbol substitution. Mainly replace “&”, “\|”, “=”, “\$” with “ and ”, “ or ”, “ equal ”, “ dollar ”.

3 Experiment for Task 1

3.1 Data Set

Training data: from News Commentary, consisting of 300,000 lines of text.

Test data: from News Commentary, consisting of 90,000 lines of text.

3.2 Additional Data Preprocessing

Data preprocessing: Due to the need to build a word-based language model, remove the extra spaces between words and leave only one space. The cleaned training set is news_word.train, and the test set is news_word.test.

3.3 Experiment Procedure

A total of three sets of experiments are carried out. Taking 1 gram as an example, the experimental steps are as follows:

Generate the N-Gram To generate the n-gram count file from the corpus, execute the command:

./ngram-count -text news_word.train -order 1 -write news_word.train.count

the input file is news_word.train, generate 1-gram, and the output file is news_word_1-gram.train.count.

[1] https://ubuntu.com/download/desktop
[2] https://www.vmware.com/cn.html
[3] http://www.speech.sri.com/projects/srilm/download.html

Train Language Models To train the language model in the generated count file, execute the command:

```
./ngram-count -read news_word.train.count -order 1 -lm news_word_1-gram.train.lm
```

The input file is news_word_1-gram.train. count, 1-gram is generated, and the output file is news_word_1-gram.train.lm.

Calculate the Perplexity Use the generated language model to calculate the perplexity of the test set, and execute the command:

```
./ngram -ppl news_word.test -order 1 -lm news_word_1-gram.train.lm > task1_result_1-gram
```

The test set is news_word.test, and the output result is task1_result_1-gram.

3.4 Experiment Results and Conclusion

Features	Train	Test
sentences	299467	91381
words	6535812	1989384
OOVs	0	12596

Table 1: Statistical results of text features in the word-based dataset.

From the statistical results of the experiments in Table 1 and Table 2, it can be seen that when the word-based language model is transferred from the training set to the test set, its thesaurus is very large, and some OOVs will appear. Secondly, as the parameter N of the language model increases, the perplexity will continue to decrease and become flat, and the model performance will continue to improve and become saturated. In addition, the effect of the model on the training set is better than that on the test set. It can be seen that when the text gap between the training set and the test set is too large, the performance of the model will be degraded.

4 Experiment for Task 2

4.1 Data Set

As the same as task1.

4.2 Additional Data Preprocessing

Due to the need to build a Character-based language model, the extra spaces between characters are removed, and only one space is retained. The cleaned training set is news_ character.train, and the test set is news_character.test.

4.3 Additional Data Preprocessing

Due to the need to build a Character-based language model, the extra spaces between characters are removed, and only one space is retained. The cleaned training set is news_ character.train, and the test set is news_character.test.

4.4 Experiment Procedure

The experimental steps are the same as Task1, which will not be described redundantly here. A total of six groups of experiments are carried out.

4.5 Experiment Results and Conclusion

Features	Train	Test
sentences	299467	91381
words	33750588	10216880
OOVs	0	0

Table 3: Statistical results of text features in the character-based dataset.

From the statistical results of the experiments in Table 3 and Table4, it can be seen that when the character-based language model is transferred from the training set to the test set, although there are many words, the thesaurus is actually very small, and OOVs will not occur. Secondly, character-based language models perform better than word-based language models. The same as the task1 model is that with the increase of the parameter N of the language model, the perplexity will continue to decrease and become flat, and the model performance will continue to improve and become saturated.

5 Experiment for Task 3

5.1 Data Set

For Task3, we collected the English dataset from the **Europarl v7 Corpus** from WMT16 as the monolingual language model training data to augment our training data. (2.21 million lines)

Training data: from News Commentary and Europarlv7, consisting of 2.51 million lines.

Test data: from News Commentary, consisting of 90,000 lines of text.

Model	Logprob		ppl		ppl1	
	Train	Test	Train	Test	Train	Test
1-gram	-2.104566e+07	-6307528	1199.432	1121.535	1659.787	1551.659
2-gram	-1.48957e+07	-5062321	151.0908	280.3696	190.147	363.8163
3-gram	**-1.362508e+07**	**-4860715**	**98.47958**	**224.0012**	**121.5291**	**287.6705**

Table 2: Perplexity results of different language models on word-based datasets. Logprob = logP(T); ppl=10^{-{logP(T)}/{Sen+Word}}; ppl1=10^{-{logP(T)}/Word}

Model	Logprob		ppl		ppl1	
	Train	Test	Train	Test	Train	Test
1-gram	-4.308914e+07	-1.30408e+07	18.42742	18.41121	18.91007	18.89719
2-gram	-3.748992e+07	-1.133727e+07	12.61896	12.58415	12.90604	12.87244
3-gram	-3.269938e+07	-9886926	9.127097	9.101782	9.307942	9.283356
4-gram	-2.685506e+07	-8144298	6.147431	6.167013	6.24729	6.268178
5-gram	-2.199247e+07	-6750760	4.424727	4.517398	4.483502	4.578738
6-gram	**-1.894724e+07**	**-5977233**	**3.601245**	**3.800571**	**3.64242**	**3.846228**

Table 4: Perplexity results of different language models on character-based datasets. Logprob = logP(T); ppl=10^{-{logP(T)}/{Sen+Word}}; ppl1=10^{-{logP(T)}/Word}

5.2 Additional Data Preprocessing

As same as task1 and task2.

5.3 Experiment Procedure

The experiment is divided into two parts: word-based language evaluation and character-based language evaluation. The parameter settings are aligned with task1 and task2, which are 1-3 gram and 1-6 gram respectively.

The experimental steps are the same as task1 and task2, which will not be described redundantly here.

5.4 Experiment Results and Conclusion

From the statistical results of the experiments in Table 5 and Table 6, it can be seen that after adding the Europarl v7 corpus as a training set, the number of terms in each order(N) of the language model increases, and the larger the model length, the more it increases. In addition, compared with the results of task1 and task2, it can be found that although the training data of the model is greatly increased, the model achieves a worse effect on the test set and a better effect on the training set, which also verifies the conjecture of the conclusion of task1: when the text gap between the training set and the test set is too large, the performance of the model will be degraded. In addition, it can be further inferred that when the training data and test data belong to the same field, the performance of the model will get better and better as the volume of training data increases; when the training data and test data do not belong to the same field, as the volume of training data increases, the performance of the model gets worse.

Model	Logprob		ppl		ppl1	
	Train	Test	Train	Test	Train	Test
1-gram_NC	-2.104566e+07	-6307528	1199.432	1121.535	1659.787	1551.659
1-gram_NE	-1.821581e+08	-6547387	726.466	1445.313	951.7584	2021.884
2-gram_NC	-1.48957e+07	-5062321	151.0908	280.3696	190.147	363.8163
2-gram_NE	-1.305573e+08	-5300090	112.3841	361.3927	136.3907	474.2414
3-gram_NC	-1.362508e+07	-4860715	98.47958	224.0012	121.5291	287.6705
3-gram_NE	-1.13939e+08	-5046690	61.6135	272.6976	72.95475	353.2309

Table 5: Comparison of word-based language model perplexity results based on **NC (News Commentary)** and **NE (News Commentary and Europarlv7).**
Europarl v7: https://www.statmt.org/wmt14/training-monolingual-europarl-v7/europarl-v7.en.gz

Model	Logprob		ppl		ppl1	
	Train	Test	Train	Test	Train	Test
1-gram_NC	-4.308914e+07	-1.30408e+07	18.42742	18.41121	18.91007	18.89719
1-gram_NE	-3.812808e+08	-1.304891e+07	18.12441	18.44459	18.5679	18.93176
2-gram_NC	-3.748992e+07	-1.133727e+07	12.61896	12.58415	12.90604	12.87244
2-gram_NE	-3.278427e+08	-1.139523e+07	12.07572	12.74812	12.32935	13.04167
3-gram_NC	-3.269938e+07	-9886926	9.127097	9.101782	9.307942	9.283356
3-gram_NE	-2.792632e+08	-1.010249e+07	8.348256	9.550775	8.497389	9.745501
4-gram_NC	-2.685506e+07	-8144298	6.147431	6.167013	6.24729	6.268178
4-gram_NE	-2.238575e+08	-8504895	5.479634	6.684305	5.557963	6.798853
5-gram_NC	-2.199247e+07	-6750760	4.424727	4.517398	4.483502	4.578738
5-gram_NE	-1.806291e+08	-7117313	3.945423	4.902837	3.990868	4.973051
6-gram_NC	-1.894724e+07	-5977233	3.601245	3.800571	3.64242	3.846228
6-gram_NE	-1.551532e+08	-6277203	3.251026	4.063954	3.283166	4.115241

Table 6: Comparison of character-based language model perplexity results based on NC and NE.