

Project Proposal
ON
Machine Translation of Noisy Text (MTNT)
Stevens Institute of Technology



STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

Project Guide:

Professor Abdul Rafae Khan

Submitted by:

Janmejay Mohanty

jmohanty@stevens.edu

Course Name:

CS 541-A Artificial Intelligence

Abstract

In most modern Machine Translation (MT) systems, noisy or non-standard input text can result in devastating mistranslations, and there has been a growing research interest in developing noise-resistant MT systems. However, there are currently no publicly available parallel corpora of naturally occurring noisy inputs and translations, therefore past research has had to rely on synthetically constructed datasets for evaluation. The author presents a benchmark dataset for Machine Translation of Noisy Text (MTNT) in this study, which includes noisy Reddit comments and professionally supplied translations. On the huge number of sentences per language pair, we commissioned translations of English comments into French and Japanese, as well as French and Japanese comments into English. We look at the many types of noise in this dataset both qualitatively and quantitatively.

Git-hub Link:

<https://github.com/Janmejaya1998/mtnt>

Updated Git-hub Link:

https://github.com/Janmejaya1998/MTNT_AI

Data

For each different language, we are selecting a set of groups ("subreddits") that we know contains many comments in that language such as English: Since a huge majority of the discussions on Reddit took place in the medium of English, so we don't restrict our collection to any community in particular.

English: As huge majority of the discussions on Reddit are conducted in English, we are not restricting our collection to any specific community.

French: The total of the French side of the parallel training data which is provided for the English-French WMT 2015 translation procedure. This amount is to be approximately 40.86 million sentences.

Japanese: We had collected three small/medium sized MT datasets: KFTT (Neubig, 2011), JESC (Pryzant et al.) and TED talks (Cettolo et al., 2012), amounting to be approximately ≈ 4.19 million sentences.

The data files for MTNT can be found in this link:

https://github.com/Janmejy1998/MTNT_AI/tree/main/mtnt-master/config

Updated

Most of the data we are dealing with are from Reddit based data.

Tools & Technologies

Software Requirements:

Currently running on Windows Operating System:

Version = Windows 10 (Latest)

To run this project code, you are required the following python modules to be installed:

1. kenlm
2. langid
3. numpy
4. pickle
5. praw
6. sentencepiece >=0.1.6 (Version)
7. yaml

Also, you are required to use Ubuntu Terminal as it contains both shell codes and programs:

Ubuntu Version = 18.04 LTS

Hardware Requirements:

Using personal CPU and GPU which are:

1. CPU = AMD RYZEN 7 5800H
2. GPU = NVIDIA GEFORCE RTX 3060

Data

For pre-processing the data, you are required to perform following things:

1. Moses: For tokenization, clean-up, etc... (<https://github.com/moses-smt/mosesdecoder>)
2. Sentencepiece: For subwords. (<https://github.com/google/sentencepiece>)
3. KenLM: For n-gram language modelling. (<https://kheafield.com/code/kenlm/>)

If you want to work on Japanese language data you can install Kytea (for word segmentation)

Reference link : <http://www.phontron.com/kytea/>

For preparing and downloading the data use the given below commands:

```
# Monolingual en data from WMT17
```

```
bash scripts/download_en.sh config/data.en.config
```

```
bash scripts/prepare_model config/data.en.config
```

```
# Monolingual fr data from WMT15
```

```
bash scripts/download_fr.sh config/data.fr.config
```

```
bash scripts/prepare_model config/data.fr.config
```

```
# Prepare en<->fr parallel data
```

```
bash scripts/prepare-en-fr.sh config/data.fr.config path/to/moses/scripts
```

```
# Download and prepare the en<->ja monolingual and parallel data
```

```
bash scripts/download_ja.sh config/data.ja.config path/to/moses/scripts
```

```
# Download and extract MTNT
```

```
wget http://www.cs.cmu.edu/~pmichel1/hosting/MTNT.1.0.tar.gz && tar xvzf MTNT.1.0.tar.gz &&  
rm MTNT.1.0.tar.gz
```

```
# Split the tsv files
```

```
bash MTNT/split_tsv.sh
```

After completing the above command execution, for Running the Scraper you can go through the below description and some commands are mentioned below:

For editing the config/{en,fr,ja}_reddit.yaml to include the appropriate credentials for your own bot. You can modify some of the parameters and settings such as subreddits, etc...

For this you have to run this command which I have mention below:

```
bash scripts/start_scraper.sh [config_file]
```

Note: When you are running this scraper, please go through the Reddit API terms on their official website. (<https://www.reddit.com/wiki/api>)

After completing all these procedures, for analysis the data, please use the given below commands for project execution:

Count the number of profanities (should return 38)

```
cat MTNT/test/test.en-fr.en | python3 analysis/count_keywords.py resources/profanities.en
```

Count the number of emojis (should return 46)

```
cat MTNT/test/test.en-fr.en | python3 analysis/count_emojis.py
```

Check the ration US/UK spelling (for ise/ize which is a good indicator) (should return 35.7% 64.3%)

```
cat MTNT/test/test.en-fr.en | python3 analysis/uk_us_ratio.py
```

Count the number of informal pronouns (in japanese) (should return 35)

```
kytea -model /path/to/kytea/data/model.bin -out tok MTNT/test/test.ja-en.ja | python3  
analysis/count_keywords.py resources/informal_pronouns.ja
```

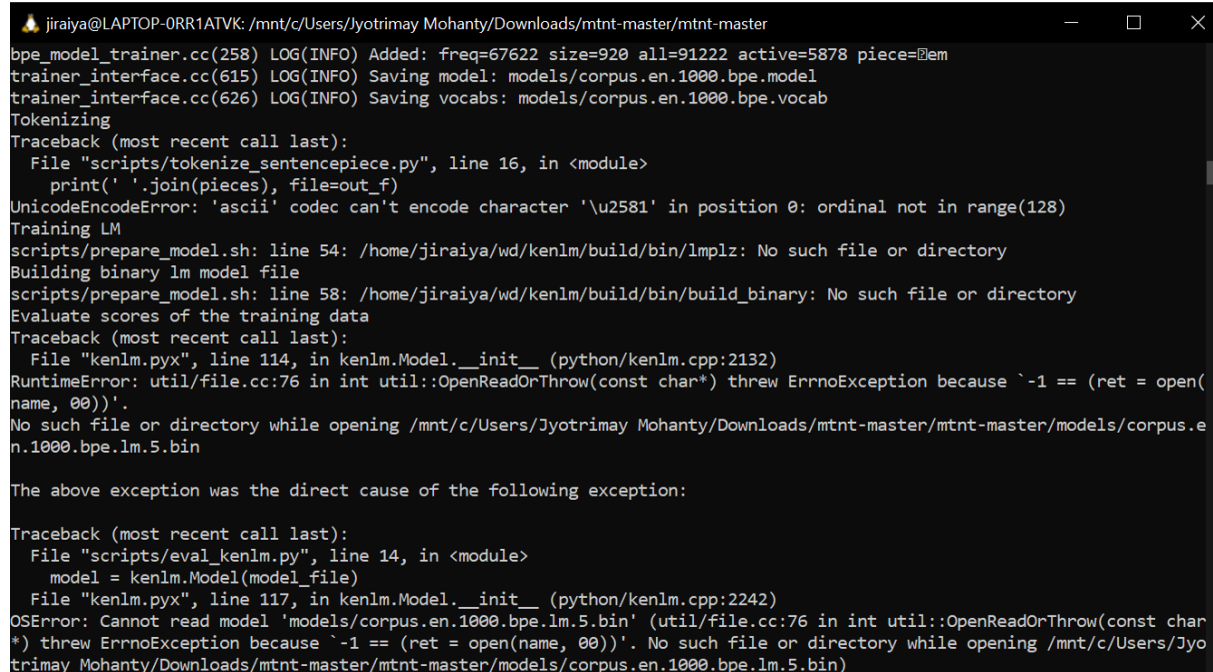
Results:

Right now, due to some technical issue which I had mentioned in the **Problems and Issues sections**, the results are not generating.

Problems and Issues:

As I had executed the above steps, which I had mentioned in the **Data section**, I am getting this error right now.

I had attached the screen shots right below so you can have a look.



```
jiraiya@LAPTOP-0RR1ATVK: /mnt/c/Users/Jyotrimay Mohanty/Downloads/mtnt-master/mtnt-master
bpe_model_trainer.cc(258) LOG(INFO) Added: freq=67622 size=920 all=91222 active=5878 piece=em
trainer_interface.cc(615) LOG(INFO) Saving model: models/corpus.en.1000.bpe.model
trainer_interface.cc(626) LOG(INFO) Saving vocabs: models/corpus.en.1000.bpe.vocab
Tokenizing
Traceback (most recent call last):
  File "scripts/tokenize_sentencepiece.py", line 16, in <module>
    print(' '.join(pieces), file=out_f)
UnicodeEncodeError: 'ascii' codec can't encode character '\u2581' in position 0: ordinal not in range(128)
Training LM
scripts/prepare_model.sh: line 54: /home/jiraiya/wd/kenlm/build/bin/lmplz: No such file or directory
Building binary lm model file
scripts/prepare_model.sh: line 58: /home/jiraiya/wd/kenlm/build/bin/build_binary: No such file or directory
Evaluate scores of the training data
Traceback (most recent call last):
  File "kenlm.pyx", line 114, in kenlm.Model.__init__ (python/kenlm.cpp:2132)
RuntimeError: util/file.cc:76 in int util::OpenReadOrThrow(const char*) threw ErrnoException because '-1 == (ret = open(name, 00))'.
No such file or directory while opening /mnt/c/Users/Jyotrimay Mohanty/Downloads/mtnt-master/mtnt-master/models/corpus.en.1000.bpe.lm.5.bin

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "scripts/eval_kenlm.py", line 14, in <module>
    model = kenlm.Model(model_file)
  File "kenlm.pyx", line 117, in kenlm.Model.__init__ (python/kenlm.cpp:2242)
OSError: Cannot read model 'models/corpus.en.1000.bpe.lm.5.bin' (util/file.cc:76 in int util::OpenReadOrThrow(const char*) threw ErrnoException because '-1 == (ret = open(name, 00))'. No such file or directory while opening /mnt/c/Users/Jyotrimay Mohanty/Downloads/mtnt-master/mtnt-master/models/corpus.en.1000.bpe.lm.5.bin)
```

I had done several times the execution and tries to resolve it by changing Python version, but still this issue hadn't resolved.

After doing self-analysis and debugging I had noticed that there are some issues in `tokenize_sentencepiece.py` code.

Right now, we can't proceed any further till the `tokenize_sentencepiece.py` is fixed.