Computational Linguistics
Assignment 4
Word alignments

Sangeet Sagar
sasa00001@stud.uni-saarland.de

January 15, 2021

1 Introduction

This assignment implements the IBM Model 1, which is used in statistical machine translation (SMT) to train word alignment model. So IBM models in general are generative models, which break up the translation process into smaller steps and achieve better statistics with simpler models. **IBM Model 1** uses only lexical translation. It ignores any position information (order), resulting in translating multisets of words into multisets of words.

2 Description

The script:

- `main.py`: this is the main script that you should be running.
- `ibm_model1.py`: IBM Model 1 class file.

The alignment extraction has been performed in two ways. Lets discuss them:

- **One-to-one alignment**: For every source (English) token, we only take one target token corresponding to the maximum translation probability score.
- **One-to-many alignment**: We set a threshold score i.e. `alpha` and for every source token, we only take target tokens whose translation probability score is equal or greater than the threshold. This results in superior results which has been discussed in further section.

I also had a chance to compare results from IMB model 1 with an off-the-shelf aligner **MGIZA**. Already, having the compiled version for this library, I used it to generate alignments as given in `resultsresultsmgiza_out.txt`. These were further processed into an index-index format using the script `read_mgiza_alignmetns.py`, that is accepted by the evaluation script.

3 Requirements

The scripts have been tested on:

1. Python: `3.8.3`
2. Numpy: `1.19.2`
3. tqdm: `1.6.3`. Install: `pip install tqdm`

4 Project file structure

```
├── ibm_model1.py
├── main.py
├── README.md
├── read_mgiza_alignmetns.py
└── results
    ├── grid_alignment_one2many.txt
    ├── grid_alignment_one2one.txt
    ├── ibm1_one2many_alpha0.3.a
    ├── ibm1_one2one.a
    ├── mgiza.a
    └── mgiza_out.txt
```

5 Usage

- **Help**: for instructions on how to run the script with appropriate arguments.
  `python main.py -help`

```
python main.py --help
usage: main.py [-h]
               [--epochs EPOCHS]
               [--num_sents NUM_SENTS]
               [--alpha ALPHA]
               [--save_model SAVE_MODEL]
               eng_f
               foreign_f
               out_dir

Implementation of IBM Model 1, which is used in statistical
machine translation to train an alignment model.

positional arguments:
  eng_f                 path to source (eng) file
  foreign_f             path to target (foreign) file
  out_dir               output dir to save the obtained alignments

optional arguments:
  -h, --help            show this help message and exit
  --epochs EPOCHS       number of training epochs for EM
  --num_sents NUM_SENTS number of sentences to train from
  --alpha ALPHA         threshold score of translation probability
                        for alignment
  --save_model SAVE_MODEL save trained model
```

- **Run IBM Model 1**: Given 100K English ↔ French parallel sentences, run IBM model 1 and generate one-to-one word alignments
  `python main.py jhu-mt-hw/hw2/data/hansards.e jhu-mt-hw/hw2/data/hansards.f results/`

- Run IBM model 1 and generate **one-to-many** word alignments.
  `python main.py jhu-mt-hw/hw2/data/hansards.e jhu-mt-hw/hw2/data/hansards.f results/ -alpha 0.30`

6 Datatset

Trained on 100K parallel English ↔ French sentences from Hansard French/English dataset.

7 Runtime

- **Total** runtime: 1148.180 s
- **Aligner (IBM model 1)** runtime: 1142.530 s
- **Alignment extraction** runtime: 0.704 s

8 Results

- **Baseline**

  ```
  Precision = 0.243110
  Recall = 0.544379
  AER = 0.681684
  ```

- **IBM Model 1**
  - **one-to-one alignment**

    ```
    Precision = 0.904762
    Recall = 0.491124
    AER = 0.350365
    ```

  - **one-to-many alignment**: `alpha 0.30`

    ```
    Precision = 0.854103
    Recall = 0.677515
    AER = 0.235382382
    ```

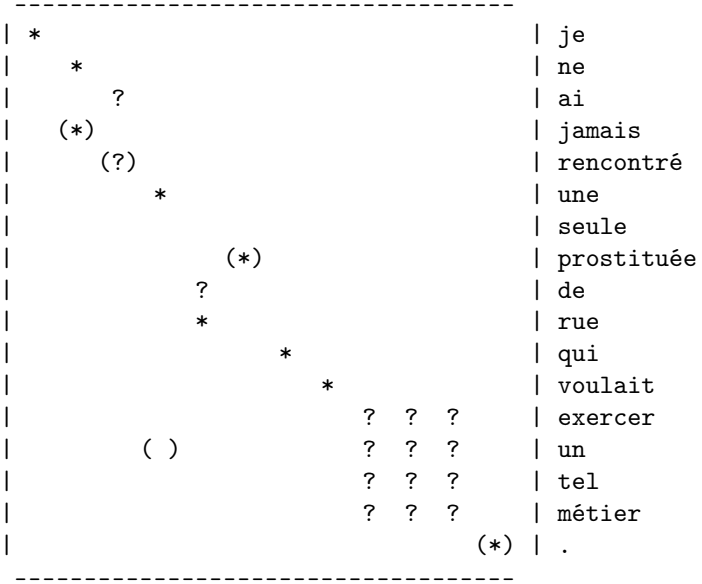
- Off-the-shelf aligner: **MGIZA**, already having the compiled version of MGIZA, I used it to generate alignments and results were:

  ```
  Precision = 0.639601
  Recall = 0.742604
  AER = 0.326923
  ```

9 Glimpse of results

While all alignments (`*.a` files) and alignment-grids (`*.txt` files) can be found in `results`, here is a glimpse of an alignment grid (one-to-one alignment):

Alignment 5 KEY: ( ) = guessed, * = sure, ? = possible



I	n e v e r	m e t	a	s t r e e t	h o o k	w h a t	w a n t	t o	b e	t h e r e	.