

Machine Translation: Practical Work 2

Sharid Loáiciga

1 Introduction

In this session, you will use two different tools for sentence alignment: Gale & Church and Hunalign. You will test them with the text *The Little Prince* in four languages: English, French, Spanish and German.

2 Preprocessing

1. Create a folder `test_littleprince`.
2. Choose and download any two languages from the four available versions in data into your folder `test_littleprince`.
3. Download and unpack the folder `preprocessing-tools.zip` into your folder `test_littleprince`.
4. Divide the sentences. For this, use the script `split-sentences.perl` to create two new files. This is an example for the English file:

```
$ perl split-sentences.perl -l en < petit-prince.en.txt \
> petit-prince.split.en.txt
```

The `-l` parameter can take the values `de`, `en`, `fr`, `es`.

- (a) What does this script do?
 - (b) What are the files in the `nonbreaking_prefixes` directory for?
5. Tokenize the sentences. To do this, use the `tokenizer.perl` script to create two new files. Example for the English file:

```
$ perl tokenizer.perl -l en < petit-prince.split.en.txt > \
petit-prince.tok.en.txt
```

- (a) What does this script do?
6. Convert the files to the format required by the Gale & Church aligner. To do this, use the `convert-gch.perl` script to create two new files. Example for the English file:

```
$ perl convert-gch.perl < petit-prince.tok.en.txt > petit-prince.gch.en.txt
```

- (a) What does this script do?
7. Convert the files to the format required by Hunalign. To do this, use the `convert-hunalign.perl` script to create two new files. Example for the English file:

```
$ perl convert-hunalign.perl < little-prince.tok.en.txt > \
little-prince.hun.en.txt
```

(a) What does this script do?

3 Gale & Church

1. Create a `test_gch` folder.
2. Copy the two versions `petit-prince.gch.*` to this folder.
3. Download the file `sentence_align.py` and place it in `test_gch`.
4. Open a terminal, position yourself in `test_gch`, and align the two texts: Example for English-French:

```
$ python sentence_align.py petit-prince.gch.en.txt \
petit-prince.gch.fr.txt > petit-prince.gch.en-fr.txt
```

5. Browse the file created by the aligner to find:
 - misalignments 0—1 and 1—0
 - misalignments 2—1 and 1—2

4 Hunalign

1. `test_hunalign` folder.
2. Copy the two versions of `petit-prince.hun.*` to this folder.
3. Download Hunalign from <http://mokk.bme.hu/resources/hunalign/> (source package) and extract the archive in `test_hunalign`. Alternatively, download from Moodle.
4. With a terminal, go to the `hunalign-1.2/src/hunalign` directory (the very last one in the file tree), and compile the program by typing `make`.
5. This creates a `hunalign` file (without extension) in `src/hunalign/`. To simplify the following operations, output this file and place it directly in the `test_hunalign` directory.
6. To use `hunalign`, you must indicate a file containing a dictionary. Since we want to use `hunalign` without a dictionary, we will simply create an empty file, using the following command:

```
$ touch emptyfile.dic
```

7. Launch the aligner. Example for German-French:

```
$ ./hunalign -text -utf emptyfile.dic \
petit-prince.hun.de.txt petit-prince.hun.fr.txt > result1.txt
```

8. Look at the file containing the results. Note the quality score indicated in the order line!
9. To use `hunalign` with a temporary dictionary created automatically, you must add the `-realign` option:

```
$ ./hunalign -text -utf -realign -autodict=temp.dic emptyfile.dic \
petit-prince.hun.de.txt petit-prince.hun.fr.txt > result2.txt
```

10. Look at the file containing the results (`result2.txt`), as well as the file containing the temporary dictionary (`temp.dic`).

(a) Has the quality score changed?

(b) How do you rate the quality of the temporary dictionary?

5 Submission

Submit a PDF file in Moodle by Wednesday May 20th, 11:00pm. You should document your progress in the assignment. Answer the given questions but also note your observations and impressions.