Apertium
https://www.apertium.org

User-friendly lexical training

9th March 2021

Contact

Name	:	naan_dhaan
Email	:	xyz@gmail.com
IRC	:	vivekvelda*/naan_dhaan*
Git	:	https://github.com/xyz
Time zone	:	UTC+5:30

About Me

I am a fourth-year computer science and engineering undergraduate at IIT Kharagpur. Almost every field of computer science piques my interest, more specifically, security, systems, and AI derivatives. I have proficiency in programming languages such as C++, java, python, bash, x86 ASM, etc.

My academic curriculum includes:

- Theory of computation
- Algorithms
- Software Engineering
- OOPS
- Compilers
- Operating systems
- Information retrieval
- Probability and statistics

Apart from these, I have done Machine Learning and Deep Learning specializations of Coursera

I have done projects, spanning various disciplines, like implementing pintOS, Neural Machine Translation, Analyzing drug users from Reddit data, Extracting causalities of cancer from tweets, Object detection, etc. Most of them can be found on my git profile

Apertium

Aperium is an open-source machine translation system with a great motive to support low-resource languages. Last year, Natural Language Processing and Deep Learning caught my interest, and I have been working on various projects since then. Also, I have been longing to work in open source. With this project, I would like to kick-start contributing to apertium and become a part of this open-source community

User-friendly lexical training

The lexical selection module selects the right sentence in the context, based on lexical selection rules, from the multiple(ambiguous) sentences output by the transfer module. These rules can be written manually or inferred automatically by training on a corpus. But, the training process is a bit tedious with various tools like irstlm, fast-align, moses, etc, and various scripts like extract-sentences, extract-freq-lexicon, process-tagger-output, etc, involved, which require a lot of manual configs.

The goal of this project is to make this as simple and automated as possible with little involvement of the user. In a nutshell, there should be a single config file and the user does the entire training using a driver script. Finally, design regression tests on the driver script so that it works in the face of updates to the third-party tools. Also, train on different corpora and add lexical selection rules to the languages which have few to no lexical selection rules, thereby improving the quality of translation

coding challenges

As of April 12th, the training process on parallel corpora is complete:

- ➔ designed a simple TOML parser
- ➔ pre-processed the corpora
- ➔ generated the word-alignments using fast-align
- ➔ extracted the ambiguous sentences and freq-lexicon using the word-alignments and biltrans output
- ➔ extracted the lexical selection rules

Git repo: https://github.com/vivekvardhanadepu/user-friendly-lexical-training

Workplan

May 17 - June 14

Most of the typical community bonding period work is done. But, I would like to revisit and delve deep into them

- ❏ I read the documentation of the apertium until the lexical selection module. So, I will brush them up and complete the remaining
- ❏ Revisit the entire training scripts, apertium modules involved, tools used and look for any optimizations in code, using new tools, etc
- ❏ Also, look into helper scripts like extract-freq-lexicon, extract-sentences, process-tagger-output, etc
- ❏ Finally, test it extensively to find any bugs[Because, while doing coding challenges, I found that many scripts and tools had bugs and were outdated]
- ❏ Revise regression testing

June 14 - June 21

- ✓ Run the entire training process on a parallel corpora
- ❏ validate the rules thus produced

June 21 - June 28

Initialize the driver script:

- ✓ validate the config file
- ✓ ensure that the required tools are all installed and configured

Deliverable: driver script can validate if the required tools are setup

June 28 - July 5

Continue with the driver script:

- ✓ checking if the corpora are available, if not, download it(in case of parallel corpora training)
- ✓ pre-process the corpora
- ✓ do the lexical selection training on the pre-process corpora

Deliverable: driver script can train on a parallel corpus and produce an lrx file

July 5 - July 12

- ❏ test the produced lrx file on the held-out test corpus
- ❏ repeat this for some of the other corpora and other languages

July 12 - July 26

Repeat the above procedure for non-parallel corpora

Note: July 12-19 phase 1 evaluation

Deliverable: driver script can train on a non-parallel corpus and produce an lrx file

July 26 - Aug 2

- ❏ build regression suite
- ❏ design regression test
- ❏ Repeat the same for non-parallel corpora

Deliverable: regression tests are designed

Aug 2 - Aug 16

- ❏ train on language pairs that have few or no lexical selection rules
- ❏ check if it improves the quality of the translation
- ❏ if so, add those to the language pairs with the help of the maintainers

Deliverable: improving the quality of translation by adding the produced lexical translation rules

Aug 16 - Aug 23

- ❏ documenting the project(most of which will be done in the previous phase)
- ❏ re-structuring the code, if needed and adding it to apertium
- ❏ updating the wiki wherever required

Non-SoC plans

I will be graduating this year[semester ends on April 14]. So, I will be free this whole summer and I can devote my entire time to this project.