Webpage Translation Using Machine Learning | AI

Omnibond HackHPC 2018

Paul Devlin
Peter Lieblang
Yeabkal Wubshit

November 14, 2018

Motivation

- Improve and vary current machine learning models used in TensorFlow to translate between languages.
- Create a useful application with our newly trained model.

Machine Translation

- Used *TensorFlow* and *AWS* to train a model to translate between English and Vietnamese in 2 layers.

Webpage Data Retrieval

- Used *Java* to fetch the source code of webpages and separate the HTML code from the text that needs to be presented on the page.
  - Implemented **`SourceCodeFetcher`**, **`Translator`**, **`FileExtractor`**, and **`HTMLParser`** tools.
  - Implemented an efficient separation of HTML code from body text that runs in linear time.

Original Plan to Link Our Machine Translator to our Java Translator Program

- Wrote a Python script that queries our machine learning model to get translations of relevant parts of our target webpages without distorting the HTML code structure.
- Wrapped up our work in a simple GUI that takes
  - The URL of the webpage that a user needs to translate
  - The language to translate it to
- GUI generates a new translated version of the webpage.

Changes of Plan

- Process of training our model took long and became infeasible in relation to the time we were left with.
  - Translation models trained in Vietnamese had only two layers, meaning that they were far from reliable to be integrated in a usable product.
  - Issues with AWS GPU instance creation
  - Issues with slow Wifi to download large datasets (of > 4.5M sentences)
- So, we had to make a quick change of plan.

Google AI Translation API

Benefits of using Google's API

- **Flexibility**:
  - We can translate web pages from and to any of the 100s of languages supported by the **Google Translation API**.
  - Adopting TensorFlow would have bound our language choice to the ones with which we have run the training.
- **Reliability**:
  - Google's Translation API is a well-tested, thorough API. Using it in our product ensures high quality so long as we handle the parsing of source code correctly.

Our Final Product

- Developed a functional webpage translator that has a user interface from which a user can choose from several languages to translate to and from.
- The translated webpage automatically pops up without the user having to navigate to the new HTML source code on-prem.

TechStack

- *TensorFlow*
- *AWS*
- *Java*
- *Python*
- *Google Translation API*
- *Maven*

Real world applications of our project

- The machine translation section of this project can be expanded, and with further research, can possibly lead to a better machine translation model that can be applied to real world scenarios, like the one we tried to demonstrate (i.e. webpage translation).

Real world applications of our project Developer Side:

- Our **HTMLParserTool** can be used in a variety of contexts by developers who wish to play with HTML source codes:
- **Advantages over common Java HTML Parsers like *JSoup*:**
  - An easy to use API not only to fetch the whole body text into a single `String` (or `Document` object), but also to get a block by block access to parts of the body text.
  - This way, a developer can access/modify contents of webpages without distorting the HTML code structure.

Thank You

Thank you for providing us this awesome opportunity to learn and develop something useful! It was great learning from all who helped us along the way!!!