Tuhina Das

ISM II

Research Assessment #5

Date: September 22, 2024

Subject: Full-Stack Software Development

Assessment: Research Assessment #5 - Natural Language Processing with SpaCy

My prior research on web scraping basics with Python and software's potential applications in mitigating mental health both pointed to artificial intelligence having a larger role with managing complex contextual data provided by users. Most mentionings of artificial intelligence bring up textual analysis, be it sentiment analysis for chatbot capabilities or tokenizing strings for complex web scraping. So, I decided to investigate the basics of Natural Language Processing (NLP) by reading the documentation of spaCy, a popular open-source Python library used to process large amounts of text.

Overall, this documentation gave me a fairly thorough overview of not only how NLP works, but also how spaCy's features work. spaCy's second section gives a brief overview of its features, which are covered in the following sections of the documentation, including tokenization (segmenting text into words and punctuation), dependency parsing (describing the relationships between words), and text classification (assigning categories to parts of the input). The last two aforementioned features are particularly interesting to me – it is simple to delimit, or break apart, text into words, but I have been wondering how we can contextualize those words to enable a machine to "understand" what the user is saying and respond in kind.

To address the issue of establishing a machine's contextual understanding of text, there are a few features that spaCy's documentation covers. In the pipeline of processing language, the library starts by tokenizing the text grammatically – this makes sense to me, as in order to understand language, one must first be capable of identifying words and sentences. By tokenizing given text and assigning syntactic dependencies, the computer can then further analyze each token (or piece of text) to build a given context for it. This does, however, make me curious: Is there any difference in memory usage or runtime between using a regular delimiting function and spaCy's? spaCy's documentation then goes on to describe a few features building on tokenization that may help me process language. For example, the library has an entity linking feature that associates objects in the text with data in a KnowledgeBase, an object that stores contextual data. I am curious to know what algorithms run to create this relationship between the two, as I imagine a database's size may be a hindrance if it becomes too large.

spaCy trains models through weight values assigned to text and their associated labels. However, the library's documentation warns against training models with data from a given context to preserve statistical accuracy. The documentation also mentions using backpropagation to calculate the gradient of annotation weights provided by the aforementioned tagging system – I am currently unfamiliar with both concepts, so this may be a potential area of interest for future investigation should I continue researching machine learning.

Reading spaCy's introductory documentation, I now better understand not only how the library itself works, but also the different techniques involved in NLP. The ways in which a machine can interpret strings of text are several, and there's a means to connect language to numbers to bridge an otherwise seemingly context gap. I previously understood little about how the logic behind NLP worked, but it truly seems to boil down to efficient data handling and algorithms – something that I have experience in, and a skill I will certainly continue to exercise as I continue my exploration of software development in relation to machine learning.

APA citation(s):

spaCy 101: Everything you need to know · spaCy Usage Documentation. (2024). spaCy. Retrieved September 22, 2024, from https://spacy.io/usage/spacy-101