L02_Journal_CodieMunos_ITAI_2373.docx

Reflection

While working through this lab, I was able to build more relationships in the sense of real world scenarios that benefit from preprocessing. I had so many questions being on the user side of technology, and now seeing it from the inside I have answers. For example, I wondered how social media apps can filter hashtags that are only followed with an emoji and filter results associated with the emoji being used. Now I know that lemmatization allows the meaning to be kept and associated with the character being used.

The lab also explored the differences in SpaCy and NLTK. Both of these function, but the results differ due to the different grammatical extractions. SpaCy uses its own pretrained NLP library which gives it the advantage in certain situations. Honestly, I would probably use SpaCy for more AI related projects such as chatbots or social media applications. Although, I would use NLTK more for school based projects as it is better for more formal work as it will not filter too many important tokens since there will be less noise. I would also use Lemmatization in AI related projects and social media usage as it is better for these situations. Stemming is better for research which means it is most likely better for school related projects as well.

Honestly, I did not know that so many words were removed during preprocessing but it makes sense due to not every word being predefined in technology. There are so many different languages and sentence structures which can overload a system if it has to define every bit of information. There was a bit of confusion for me at the beginning of the lab due to the amount of stages that are used in preprocessing. There were words that my brain associated with different meanings such as tokens and text cleaning. Tokens before this meant a coin or a valuable, but it means characters, sentences, and words in coding. It was insightful to see how SpaCy and NLTK break down their tokens differently.

Overall, preprocessing is all around us and seeing it from both sides, it is crucial for us in our day to day. Fifty years ago, it was not as apparent, but today’s society relies on it more than they think. Nearly everyone has some form of social media or search engine access, and I guarantee you they have encountered errors that did not make sense on the user side of the program. Especially with autocorrect, or changing one word in a question on google and a whole different array of sites popping up. Technology is always advancing, and work continues to be done to avoid having these issues occur on the customer’s end. Now I am wondering which techniques are actively being used on platforms I use daily and if I could build connections between issues I face when using them since I have more information on the impacts of the prepossessing techniques that are used. I am excited to continue to learn more about small differences on the coding side that make all the difference on the user side.