Summary

October 29, 2023

[1]:
```
print("""
This is demonstrating various Natural Language Processing techniques we've␣
 ↪learned so far in this class.

I got a small sample of some of my friend Conor's poetry, taken from some␣
 ↪portfolio submission he did a bit ago. It's 8 pages total, though I had to␣
 ↪skip the 7th since it's doesn't have any real text on it.

I read the text from a PDF directly via pypdf, including the titles for each␣
 ↪poem & the name at the start, and then do a little immediate clean-up for␣
 ↪some of the punctuation. It's mostly just putting periods at the end of␣
 ↪stanzas without punctuation, since otherwise the tokenizer gets a little␣
 ↪confused on what to break on - I didn't have to do this, but it seemed more␣
 ↪conducive to interesting data.

After, the sentences are tokenized and some basic exploratory processing is␣
 ↪done. Following that, the text is cleaned up - non-alpha words are purged,␣
 ↪all words are cast to lowercase, etc.

Then, TF-IDF vectorization is done to get some of the most "important"␣
 ↪sentences in the text - the top 3 are printed. It's a little biased towards␣
 ↪longer sentences but it's not like the text isn't full of some of those.

Following the TF-IDF, less technical things are done - a WordCloud on both the␣
 ↪original and cleaned up text, translating the original text to spanish,␣
 ↪using displacy to identify important words in the original/cleaned up text,␣
 ↪and then finally POS tagging & FreqDist counting of words in the original/
 ↪cleaned up texts.
""")
```

```
This is demonstrating various Natural Language Processing techniques we've
learned so far in this class.

I got a small sample of some of my friend Conor's poetry, taken from some
portfolio submission he did a bit ago. It's 8 pages total, though I had to skip
the 7th since it's doesn't have any real text on it.
```

I read the text from a PDF directly via pypdf, including the titles for each poem & the name at the start, and then do a little immediate clean-up for some of the punctuation. It's mostly just putting periods at the end of stanzas without punctuation, since otherwise the tokenizer gets a little confused on what to break on - I didn't have to do this, but it seemed more conducive to interesting data.

After, the sentences are tokenized and some basic exploratory processing is done. Following that, the text is cleaned up - non-alpha words are purged, all words are cast to lowercase, etc.

Then, TF-IDF vectorization is done to get some of the most "important" sentences in the text - the top 3 are printed. It's a little biased towards longer sentences but it's not like the text isn't full of some of those.

Following the TF-IDF, less technical things are done - a WordCloud on both the original and cleaned up text, translating the original text to spanish, using displacy to identify important words in the original/cleaned up text, and then finally POS tagging & FreqDist counting of words in the original/cleaned up texts.

[2]:
```
print("""

Overall, this was pretty interesting data to look at. I was curious to see␣
 ↪which words were most common here and what that says about the poetry, but␣
 ↪also curious from a technical standpoint what happens with really long␣
 ↪'sentences' like in Conor's poetry.

It's especially interesting the things that spacy highlights - like "grey␣
 ↪forecast" as a person, for some reason, which is obviously not right. I␣
 ↪guess it doens't really like very figurative/flowery language.

The WordClouds were also pretty fun to look at - I didn't realize "back" was␣
 ↪such a common word in general let alone in this poetry.

One downside I did notice - TF-IDF seems very awkward for very varying sentence␣
 ↪lengths. It really liked the longest sentences, presumably since they share␣
 ↪a lot of similarities etc. with the other ones, but that's something to␣
 ↪consider when using TF-IDF with future datasets.
""")
```

Overall, this was pretty interesting data to look at. I was curious to see which words were most common here and what that says about the poetry, but also curious from a technical standpoint what happens with really long 'sentences' like in Conor's poetry.

It's especially interesting the things that spacy highlights - like "grey forecast" as a person, for some reason, which is obviously not right. I guess it doens't really like very figurative/flowery language.

The WordClouds were also pretty fun to look at - I didn't realize "back" was such a common word in general let alone in this poetry.

One downside I did notice - TF-IDF seems very awkward for very varying sentence lengths. It really liked the longest sentences, presumably since they share a lot of similarities etc. with the other ones, but that's something to consider when using TF-IDF with future datasets.

[ ]: