Decoding Authorial Style, Tone, and Mood in Poetic Translations through Natural Language Processing: an Analysis of *Beowulf*

Daniel Quigley*

Abstract. This paper explores NLP applications to analyzing literary translation, with a focus on poetry, by examining how different translations convey authorial style, tone, and mood. Translating literature involves a balance between preserving the original text's meaning and creating something evocative in the target language. This task is particularly challenging for poetry, where word choice, rhythm, and syllable structures are crucial to the work's impact. This paper analyzes translations of *Beowulf* using natural language processing techniques to identify authorial style, tone, and mood. This research demonstrates the use of computational tools in literary studies, and not least of all serves as a sort of primer for those interested in the basics of applied natural language processing. The findings contribute to developing computational tools sensitive to style, tone, and mood, and offers tools for computational analyses therein.

Keywords. literary analysis; natural language processing; translation; computational linguistics

1. Introduction. This project is the foundation for a literary analysis tool to be used for the cross-comparison of texts-in-translation to probe for the literary style of the translator. In literary analysis, one often cares for the style, tone, and mood of the work. This information is interpretable by a human reader; is such information interpretable by a machine, and can a machine's interpretation(s) be useful for literary analysis?

For the purpose of this project, style (diction and syntax), tone (attitude towards an utterance), and mood (overall impact of tone on the reader) of an author (Zhao & Wu 2022; Zhao 2013; Dewald 1987; Bondi 2014; Phelan 2014) are measured according to various metrics, including: relative frequencies of various kinds, including parts of speech frequencies, n-grams, and hapaxes (Azam & Yao 2012; Pierrehumbert & Granell 2018); relative "complexities" of syntax and morphology, here including word count per syntactic phrase (Pierrehumbert & Granell 2018; Green 2019; Pallotti 2015; Juola 1998); vocabulary choice, whether a chosen word belongs to a certain language family by its etymology, as well as the sentiment carried by those word choices (de Melo 2014; Mohammad & Turney 2010; Piantadosi 2014; Hare et al. 2020; Reeve 2016a).

1.1. METHODS. Certainly, such an analysis lends itself to any of the many translated texts that are available in English. This invites further application of the analyses present in this project to the interested reader's choice of translated work. This project, however, measures these features within and between translations of the *Beowulf* poem. The choice of this text is not without reason. Ethical artificial intelligence should be transparent about the data that it uses; because *Beowulf* is in the public domain, it makes itself an attractive source text. Furthermore, many translations of the poem are likewise in the public domain, or are otherwise available freely. Digital texts that are ethically available (some of which are already marked in XML format, for example) include the four used here: *Beowulf by All: Community Translation and Workbook* (Abbbot et al.

* University of Wisconsin–Milwaukee (quigleyd@uwm.edu)

2021), we will call ”All”; *Beowulf* (Gummere 1910), we will call ”Gummere”; *Beowulf* (Hall 1892), we will call ”Hall”; *Beowulf* (Slade 2022), we will call ”Slade”.

Other texts of *Beowulf* exist (*Beowulf: A Verse Translation* (Heaney 1999) and *Beowulf: A Translation and Commentary* (Tolkien & Tolkien 2014), to name a few famous examples), but are *not* necessarily ethically available, hence their exclusion here.

The usual NLP pipeline for analysis follows Jurafsky & Martin (2009); McRoy (2021) (throughout, the Natural Language Toolkit (NLTK) Bird et al. (2009) and scikit-learn Pedregosa et al. (2011); Buitinck et al. (2013) were used):

(1) NLP pipeline

1. Each text was acquired from an open online source in text file format and separated into comparable sections.
2. Each text was cleaned, which involved: lowering case, removing html tags, removing punctuation, replacing special characters, removing stopwords, removing extra spacing and applying uniform formatting, expanding contractions.
3. Sentences were broken into distinct words (tokens) and sentences (sentence tokens; done before removal of punctuation and saved separately)
4. Suffixes (and obvious grammatical inflection) were either removed or replaced to recover the base word (PorterStemmer was used)
5. Tokens were tagged with their appropriate part of speech
6. Lemmatize tagged words (similar to stemming, but words recover their dictionary form; WordNetLemmatizer was used)
7. Text converted into numerical data (TF-IDF: Term Frequence-Inverse Document Frequency)
8. Process data; here we check for the various stylistics of the translator and measure the various metrics with which we probe the author’s translations

All analysis was conducted in Python, with any of various libraries. All code and data is freely available. At the time of the analyses, the following were true:

(2) Technological tools

- OS: Windows 10 Home
- GPU: NVIDIA GeFORCE RTX 307
- Preliminary tests were done on the CPU: AMD Ryzen 7 3700X 8-Core Processor 3.59 GHz
- Installed RAM: 16.0 GB
- Python Version: 3.7.8
- Environment: Conda tf-gpu

2. Results and Discussion.

2.1. SIMILARITY. Similarity scores across texts are given in Table 1. Here, similarity checks for comparisons of content, structure, and meaning. Similarity was checked using TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity, which uses the conversion into numerical vectors from the NLP pipeline and assigns a weight to each word based on its frequency in the document and its rarity in the corpus (Kim & Gil 2019; Jurafsky & Martin 2009; McRoy 2021). The cosine similarity measures the angle between two non-zero vectors; scores range from 0 to 1, with higher scores indicating greater similarity between the documents.

Document Pair	Similarity Score
All - Gummere	0.7511
All - Hall	0.6743
All - Slade	0.9243
Gummere - Hall	0.7287
Gummere - Slade	0.7406
Hall - Slade	0.6537

Table 1. Similarity scores between document pairs

The similarity scores across the paired texts distills the comparison metrics into a quick numerical value. The extreme values can be interpreted in the following way: The Hall - Slade correspondence is weakest, indicating the least similarity in content, structure, and meaning, while the All - Slade correspondence is strongest, indicating elsewise.

As is convention in analyses of *Beowulf* (Walkden 2013; Sayers 2018), we inspect the introductory lines of the poem across the two extreme pairs, to illustrate the extent to which cosine similarity captures the authors' similarities.

Hall	Slade	All
Lo! the Spear-Danes' glory through splendid achievements The folk-kings' former fame we have heard of, How princes displayed then their prowess-in-battle. Oft Scyld the Scefing from scathers in numbers From many a people their mead-benches tore. Since first he found him friendless and wretched, The earl had had terror: comfort he got for it, Waxed 'neath the welkin, world-honor gained, Till all his neighbors o'er sea were compelled to Bow to his bidding and bring him their tribute: An excellent atheling! After was borne him A son and heir, young in his dwelling, Whom God-Father sent to solace the people. He had marked the misery malice had caused them, That reaved of their rulers they wretched had erstwhile Long been afflicted.	Listen! We –of the Spear-Danes in the days of yore, of those clan-kings– heard of their glory. how those nobles performed courageous deeds. Often Scyld, Scef's son, from enemy hosts from many peoples seized mead-benches; and terrorised the fearsome Heruli after first he was found helpless and destitute, he then knew recompense for that:- he waxed under the clouds, throve in honours, until to him each of the bordering tribes beyond the whale-road had to submit, and yield tribute:- that was a good king! To him an heir was born then young in the yards, God sent him to comfort the people; He had seen the dire distress that they suffered before, leader-less a long while	Heyla! We have a story about the Spear-Danes, from the old days when they were big and their kings showed their strength. There was one king, Shield Schefing, who stole many mead-benches from other tribes and terrified their leaders. At first, he was found weak and wandering, but was taken in and then grew under the comfort of the skies. He consumed honors until each of the other surrounding tribes over the whales road were forced to obey him and pay tribute. They say, that was a good king. After all this (when he was old), Shield had a sona young one in the courtyardwho had been sent by God as a comfort to the people because He had seen how they were distressed, left without a strong leader for a long while.

The content of each excerpt covers the same material: the great king (Shield Schefing, Scyld, or Scyld Scefing), his early hardships, his rise to power, and the birth of his heir. Each excerpt mentions the king's prowess in battle, his ability to seize mead-benches from other tribes, and his eventual command over neighboring tribes. All three excerpts also highlight the king's status as a paragon, a good or excellent leader, and note that his heir was sent by God to comfort the people.

The translation of the initial *hwæt* as any of *Lo!, Listen!, Heyla!* is worth mentioning, following convention in analyses of *Beowulf*, as each author has opted for some interrogative to evoke the notion of *Hey! Listen here!*, though each varies in its archaisms. The most obvious differences elsewise are apparent the language choice and style of each translation. Hall tends to use more archaic language, as in the choice of *Lo!*, but also in *welkin, atheling, reaved*, while Slade tends towards a more modern language choice while retaining few archaisms (*yore, throve, yards*). Slade mentions the Germanic tribe Heruli, while the other two excerpts do not. All uses the most modern and colloquial language (*Heyla, big, showed their strength*), and includes details about the king's old age and the people's distress before the heir's arrival. Proper names are not exempt from the authors' differences: All and Slade refer to the king as *Scyld the Scefing* or *Scyld, Scef's son*, while All calls him *Shield Schefing*, adopting a more modern spelling and word-choice for the name's translation.

2.2. COMPLEXITY. 'Complexity' is not well-defined, but we have some measurements with which to define a *kind* of complexity (Juola 1998; Pallotti 2015; Green 2019). Admittedly, our probe for complexity is mostly *ad hoc*, though we should be tempered by linguistic foundation: that is, do we say that *com´ı* is 'more complex' than *I ate*? The former may be *morphologically* more complex, while the latter may be *syntactically* more complex.

For this paper, we define complexity admittedly superficially, working within the usual NLP tools as our computational boundaries, and probe for it by measuring the following that *together* constitute a measure of complexity:

(3) Complexity metrics

1. (Average) Lengths relative to:
    - Words
    - Phrases (by type)
    - Sentences
2. Type frequencies
3. Deviancy from some otherwise expected syntax

When we process text, we observe the following in Table 2 for each translation (note that, for some metrics, using the cleaned text is nonsensical, and so are not represented). The items in the first four columns are for the cleaned and processed texts; the items in the last four columns are the raw texts prior to processing. Note that word length and words per sentence metrics are average values. The processed texts do not have metrics for average number of words per sentence, since sentence definition is lost in the processing procedure, hence an absence of a score. The number of words is exactly the number of space-separated tokens; this metric loses sensitivity to words-as-a-unit; *Spear-Danes* or *Shield Schefing* may be broken up.

Table 2. Some length metrics for each text

	Processed				Raw			
	All	Gummere	Hall	Slade	All	Gummere	Hall	Slade
Word Count	12429	12631	13057	12305	24770	23133	24308	24676
Unique Word Count	3485	3466	4227	3448	5784	5957	6837	5732
Word Length	5.84	5.53	6.32	5.91	4.66	4.61	5.01	4.71
Words/Sentence					20.47	23.46	34.73	70.10

Upon inspection, we notice the following. Going by sentence-length, Slade is noticeably the leader, while All has the fewest words-per-sentence. Hall has the longest words in sentences. All has the most words prior to cleaning, indicating more auxiliary and stopwords. Certainly, such information is readily available to a machine, but the question remains: what *effect* do these metrics have on the reader/machine; what is the extent to which length metrics correlate with the idea of complexity?

While sentence length is used in most readability formulae, short sentences are also agents in difficulty (Schlesinger 1968; Brookshire & McNeil 2015). Naively, longer sentences tend to be thought of as 'more difficult' to read than shorter sentences (here, both sentence length by word tokens and by word length per word tokens); this is further influenced by the bit rate of information passed or contained in a sentence (Halliday 1985; Gruber & Gibson 2004).

While single word tokens were the units of measurement in Table 2, the number and kind of clauses in a sentence are also used as metrics for complexity (Khudair & Owaid 2022). Simple, parataxis, and hypotaxis sentences induce variable degrees of complexity, in which a simple sentence is one independent clause, parataxis links like-elements, and hypotaxis is syntactic subordination (Halliday 1985; Collins 1991; Khudair & Owaid 2022). Such syntactic combinations can be exhausting to computationally extract (see below about syntactic deviancy), but clause types and type frequencies *can* be (relatively easily) extracted based on word tokens and POS tagging.

Clause types and type frequencies invite analysis of the following: term frequencies (Azam & Yao 2012); POS frequencies (Klavans et al. 2011; Azam & Yao 2012); Zipf-Mandelbrot distributions (Piantadosi 2014; Hare et al. 2020); hapax legomena (Pierrehumbert & Granell 2018). In particular, the slope from a Zipf-Mandelbrot distribution[1] in log space is known as the Zipf exponent or the scaling exponent, and it is exactly a measure of the degree of inequality in the distribution of word frequencies, hence a useful metric for checking relative complexity. A steeper slope indicates a more unequal distribution, *i.e.*, more complex and expressive text; a shallower slope indicates a more equal distribution, *i.e.*, simpler and less expressive text. Table 3 gives the slope and R^2 values from the frequency-rank distributions of each text.

Table 3. Slope and R^2 values from frequency-rank distributions

	All	Gummere	Hall	Slade
Slope	-0.8897	-0.9010	-0.8249	-0.8897
R^2	0.9483	0.9459	0.9471	0.9486

We interpret the following from Table 3. For All: the slope of -0.8897 is quite steep, indicating a relatively unequal distribution of word frequencies, and thus a relatively complex and expressive text. For Gummere: the slope of -0.9010 is even steeper than All, indicating an even more unequal distribution of word frequencies, and thus an even more complex and expressive text. For Hall: the slope of -0.8249 is less steep than All and Gummere, and thus a relatively simpler and less expressive text. For Slade: the slope of -0.8897 is the same as All, indicating a similar level of complexity and expressiveness. What follows: Gummere appears to be the most complex and expressive, since it has the steepest slope. Hall assumes the simplest and least expressive text, since it has the shallowest slope. Each has an R^2 value close to 1, indicating a good fit to the distribution.

Type frequencies (relative distributions of nouns, adjectives, verbs, and their associated phrases) and n-grams lend themselves best to data visualizations, which are given in the appen dices.

[1] A generalization of Zipf's law: in a given text, the frequency of a word is inversely proportional to its rank in the frequency table.

A more difficult metric to capture is deviancy from an expected syntax. Poetry is often recognizable by some ‘poetic style‘ and certain effects by some manipulation of the language, often syntactic (Cureton 1980; Panajoti 2013; Blockley 2001). This might be seen in the following (humorous criticism of Tolkien, original source unknown):

(4) Poetic language manipulation

a. *The clause that is not inverted is rare.*

b. *Rare the clause that not inverted is.*

The effect of this manipulation is variable, and may depend on the skill of the author, the audience readership, the context, or some combination therein (Stoehr 1964; Lord 1975; Kintgen 1978; Russom 2022). In lieu of ready results, we suggest an algorithm for checking this particular metric.

(5) Algorithm for syntactic deviancy

1. Define a function CHECKCOMPLEXITY that takes a sentence as input.
2. Initialize a variable score to 0.
3. Tokenize the sentence and perform POS tagging. In the algorithm, this is represented by TOKENIZEANDPOSTAG, a function that tokenizes the sentence and performs POS tagging.
4. Perform syntactic parsing with a syntactic parser to generate a syntactic tree or dependency graph of the sentence; PARSESYNTAX is a function that performs syntactic parsing on the tokens and POS tags.
5. From the syntactic tree or dependency graph, extract the constituents of the sentence and their order; EXTRACTCONSTITUENTS is a function that extracts the constituents and their order from the syntax tree.
6. Define the expected syntax (this should be a predefined set of rules that define the expected constituent order for the language in question: SVO, for example, for a prototypical transitive sentence in English); DEFINEEXPECTEDSYNTAX is a function that returns the expected constituents and their order for the language.
7. Compare the order of constituents extracted from the sentence to the expected order defined in the previous step.
8. Calculate score: for each deviation from the expected syntax, subtract a value from the score; CALCULATEDEVIATION is a function that calculates the deviation of a constituent from the expected syntax.
9. Return the total score as the output of the function.

What makes this a particular laborious task is the tokenization and constituent checking, which can be computationally expensive; indeed, if not using a pretrained parser, then hard-coded rules need to search and check an entire text, which takes exponential time.

Algorithm 1 Check Phrase Complexity

```
function CHECKCOMPLEXITY(sentence)
    score ← 0
    tokens, posTags ← TOKENIZEANDPOSTAG(sentence)
    syntaxTree ← PARSESYNTAX(tokens, posTags)
    actualConstituents ← EXTRACTCONSTITUENTS(syntaxTree)
    expectedConstituents ← DEFINEEXPECTEDSYNTAX
    for each constituent in actualConstituents do
        if constituent not in expectedConstituents then
            deviation ← CALCULATEDEVIATION(constituent, expectedConstituents)
            score ← score − deviation
        end if
    end for
    return score
end function
```

2.3. SENTIMENT. To compare with the sections of the poem and the sentiment trends, in brief, an outline of the poem is given below (spoiler warning for the poem's content).

(6) Outline of the poem

1. Introduction
   (a) The poem begins with a brief history of the Danish kings, starting with Scyld Scefing, a great king who ruled the Danes. His son, Beow, succeeded him, followed by his grandson, Hrothgar.
   (b) Hrothgar, a successful warrior, builds the grand mead-hall Heorot, where he and his men can celebrate their victories and enjoy life.
2. Grendel's Attacks
   (a) The monstrous creature Grendel, a descendant of the biblical Cain, becomes envious of the joy and festivities at Heorot and attacks the hall at night, killing Hrothgar's warriors.
   (b) Grendel's attacks continue for twelve years, leaving the Danes in a state of despair and helplessness.
3. Beowulf Arrives
   (a) Beowulf, a noble warrior from Geatland (Sweden), learns of the troubles of the Danes and decides to help them. He sets sail for Denmark with a small band of warriors.
   (b) Beowulf offers his services to Hrothgar, reminding him that he once helped his father. Hrothgar gratefully accepts Beowulf's offer.
4. Beowulf's Battle with Grendel

(a) Beowulf and his men stay in Heorot, waiting for Grendel to attack. When Grendel appears, Beowulf fights him unarmed, ripping off the monster's arm.

(b) Grendel flees back to his lair and dies from his wounds. Beowulf is hailed as a hero, and a grand feast is held in his honor.

5. Grendel's Mother's Revenge

    (a) Grendel's mother, another monstrous creature, seeks revenge for her son's death and attacks Heorot, killing one of Hrothgar's trusted warriors.

    (b) Beowulf follows her to her lair beneath a lake and fights and kills her with a sword he finds in her lair.

6. Return to Geatland

    (a) After defeating Grendel's mother, Beowulf receives many gifts from Hrothgar and returns to Geatland.

    (b) He is warmly welcomed by his king, Hygelac, and presents him with the gifts he received. Beowulf becomes the king of the Geats after Hygelac's death.

7. Beowulf's Final Battle

    (a) After ruling peacefully for fifty years, Beowulf's kingdom is threatened by a dragon awakened by a thief who stole a cup from its treasure hoard.

    (b) Beowulf, now an old man, decides to fight the dragon himself. He manages to kill the dragon with the help of a warrior named Wiglaf, but he is mortally wounded in the battle.

8. Beowulf's Death

    (a) Beowulf names Wiglaf as his successor and dies from his wounds. His body is cremated, and a large burial mound is built in his honor.

9. Conclusion

    (a) The poem reflects on the heroic ideals of the Germanic society: the importance of loyalty and bravery and the inevitability of death.

Sentiment analysis is a 'black box' in NLP (a model that takes an input and produces an output without revealing much about its internal logic or decision-making). Additionally, we are limited in the actual applicability of sentiment analysis for translations of tenth century literature. It was reasoned, then, that instead of using *one* model to give misleading results, *three* will be used, and their trends in parallel could help a machine interpret the sentiment of our texts[2]. Three models are used: VADER; TextBlob; NRC.

[2] Technically, this is a forest method, wherein the logic is that 'two heads are better than one' for analysis; see appropriate sections in Jurafsky & Martin (2009); Russell et al. (2010); Kelleher et al. (2020) for further discussion.

VADER (Hutto & Gilbert 2014) (Valence Aware Dictionary and sEntiment Reasoner) specifically examines sentiments expressed in social media, and is sensitive to polarity and intensity. The Neutral sentiment (neu; avg = 0.57, SD = 0.02) scores higher above the other sentiments consistently across all sections of the poem: neg (0.19, SD = 0.03); pos (0.24, SD = 0.03). This may follow from the author's style, which is certainly a formal register, or it may be a limitation of the analytical application, given that the language of *Beowulf* is not the intended target for VADER. An example visualization for the All text is given in Figure 1.

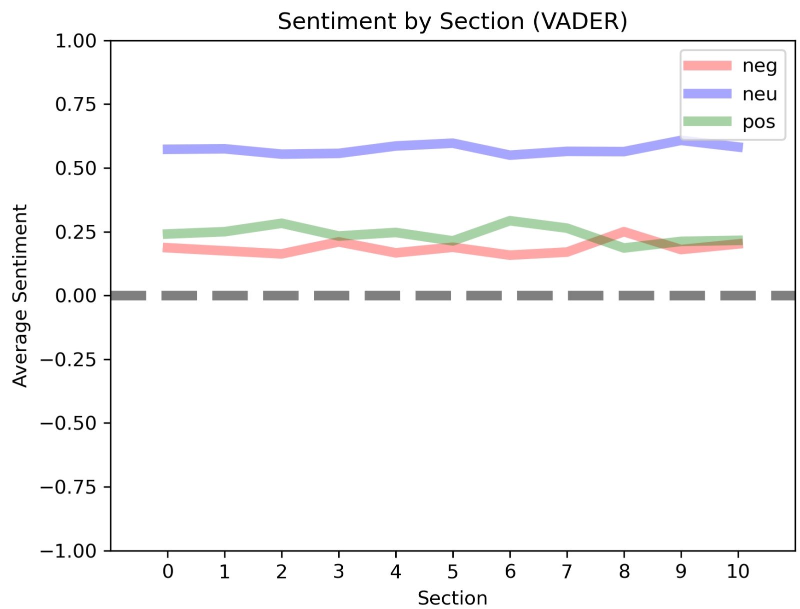

Figure 1. VADER analysis for All

TextBlob (Loria 2018) measures a text for its polarity (the extent to which sentiment is positive or negative on a [-1,+1] scale) and subjectivity (measured on a [0,+1] scale in which 0 refers to an objective statement and 1 refers to a subjective statement; subjectivity refers to personal opinions, emotions, or judgments, while objectivity refers to factual information). We see that the Subjectivity score (0.59, SD = 0.04) is significantly higher than the Polarity score (0.16, SD = 0.05). The higher scores for the Subjectivity metric indicate a more personal narrative rather than an objective account. An example visualization for the All text is given in Figure 2.

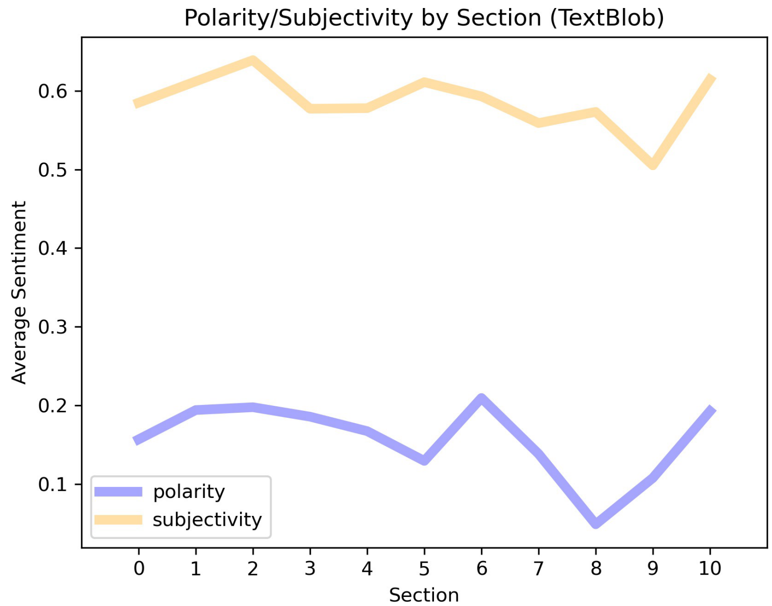

Figure 2. TextBLOB analysis for All

NRC (Mohammad & Turney 2010, 2013) measures English words and their associations with eight basic emotions and two sentiments, which combine to yield complex emotions. This follows from the work of psychologist and professor Robert Plutchik. NRC considers attributes in diametric pairs (positive group (anticipation, joy, surprise, and trust) relative to negative group (anger, disgust, fear, and sadness)). The NRC scores indicate that the poem tends towards a higher overall positive score (0.2, SD = 0.02) than negative score (0.15, SD = 0.02). Strongest emotions are: trust (0.11, SD = 0.01); fear (0.11, SD = 0.01); anticipation (0.09, SD = 0.01); anger (0.09, SD = 0.01). The lowest scores are: joy (0.08, SD = 0.01); sadness (0.06, SD = 0.01); disgust (0.06, SD = 0.01); surprise (0.04, SD = 0.01). An example visualization for the All text is given in Figure 3.

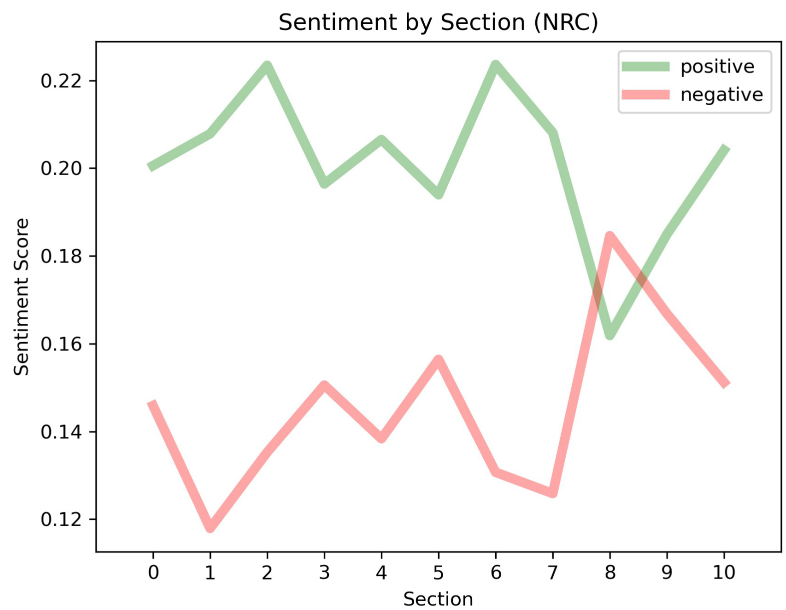

Figure 3. NRC analysis for All

We see the following trend: polarity extrema in TextBLOB relate as crossover points in VADER and NRC for positivity and negativity metrics, and do so such that minima of polarity scores in TextBLOB correlate with increasing negative scores and decreasing positivity scores in VADER and NRC, and vise versa. This is congruent with a similar sentiment analysis of John Milton's *Paradise Lost* (Quigley 2022) and T. E. Lawrence's *Seven Pillars of Wisdom* (in progress). These behaviors may be more indicative of the sentiment models themselves, and not necessarily of the texts upon which they analyze. Generally, however, the intuition follows for sentimentality of the poem is captured by these metrics relative to the subject matter.

2.4. Etymological Choice. Etymological choice is useful for studying an author's choice of vocabulary. Indeed, there is an established precedence for this. J.R.R. Tolkien very deliberately decided to prefer words of Germanic origin over those of Latinate origin (Jeffrey 1980; Croft et al. 2007); this is why a reader will not find 'human' when referring to people, but instead you will find 'man' (usually capitalized 'Man'). John Milton was deliberate in his choice of words as Latinate or Germanic, and used those choices to affect the speaking style of his narrator-voice and characters in his works (Considine 1998; Reeve 2016b; Quamen 2021).

The NLP process for extracting the etymological information of a text is relatively straightforward; however, at the time of this paper, the tool used for it has many holes, and more work is being done to include a greater source of etymological information and search optimization. Following Reeve (2016a), tracing the etymologies follows:

(7) Measuring etymological choice

1. For each word in a processed text, trace its etymology

2. Categorize the traced word origin into appropriate bins (language families; granularity is user's choice), from which we may get a distribution of vocabulary.

To get etymological information, we used the Etymological Wordnet (de Melo 2014; Reeve 2016a) as a dataset that is acquired from Wiktionary. For each word in the text, its origin was tracked and grouped according to: Germanic, Latinate, Hellenic, Semitic, Celtic, Indo-Iranian, Turkic, Uralic, Other. Preliminary results are given in Table 4.

Table 4. Language families and their percentages in each dataset

	All		Gummere		Hall		Slade	
Language Family	%	Count	%	Count	%	Count	%	Count
Germanic	31.85	1110	33.50	1161	28.25	1194	29.61	1021
Latin	7.26	253	5.57	193	5.84	247	7.28	251
Hellenic	0.17	6	0.09	3	0.09	4	0.15	5
Semitic	0.06	2	0.03	1	0.02	1	0.06	2
Celtic	0.03	1	0.03	1	0.00	0	0.03	1
Indo-Iranian	0.03	1	0.00	0	0.00	0	0.00	0
Turkic	0.00	0	0.00	0	0.00	0	0.00	0
Uralic	0.00	0	0.00	0	0.00	0	0.00	0
Other	60.60	2112	60.79	2107	65.79	2781	62.88	2168

A significant amount of data is not captured by the tool as used here as a result of nascent development of the tool; the data in Table 4 is too sparse for a meaningful discussion here. Continued work on building an etymology tool is naturally invited and in progress. Work needs to be done on optimizing search through etymological roots and recording their language family origins. This tool would likewise benefit from the forest method as well, using not just the Etymological Wordnet, but also databases such as the Online Etymology Dictionary (Harper & McCormack 2003).

3. Conclusion. The purpose of this project was to develop and share tools from NLP that can be used in literary analysis. Texts in translation are interesting subjects for analysis since the translator must make deliberate stylistic decisions when translating a work. Poetry is a highly expressive and creative form of language and language manipulation, and translating poetic conventions is often a unique choice by the translator. Using tools from NLP, we may probe the extent to which a machine is sensitive to these deliberate authorial stylistics. Authorial style, tone, and mood as interpretable by a machine were here given as measuring similarity, complexity, sentiment, and etymological choice. We were able to define each metric, and build a learning tool for exploring these tasks in NLP.

The similarity scores show a weak correspondence between the Hall and Slade texts, while All and Slade show the strongest. Intuitively, this coincides with a reading of select passages. The similarities can be seen in the content and some word choice, but word choice is a better diagnostic for dissimilarity, in which the Hall and Slade texts both prefer greater archaisms compared to All.

Complexity is defined here according to: length (of words, phrases, sentences), type frequencies, syntactic deviancy. Length metrics capture word, phrase, and sentence level token lengths. The extent to which lengths correlate with complexity is not entirely straightforward, so an alternative metric in type frequencies that capture expressiveness is readily interpretable. Indeed, by frequency-rank distributions, Gummere is most complex, while Hall is simplest. Syntactic deviancy remains to be explored, but an algorithm with which to approach the problem is suggested.

Sentiment is measured relative to VADER, TextBLOB, and NRC. The TextBLOB extrema tend to correlate with intersection points or change in slope in VADER and NRC, in which minima of polarity scores in TextBLOB correlate with increasing negative scores and decreasing positivy scores in VADER and NRC, and vice versa. It can probably be said that these behaviors expected of the sentiment models themselves, rather than a statement about the sentimentality of the texts. Generally, however, the intuition follows for sentimentality of the poem is captured by these metrics: the extreme behavior of negative and positive sentiment correlate with the subject matter (Beowulf's death, for example, towards the end of the poem).

Not much meaningful interpretation can follow from the etymology analysis at this time. More work is needed to expand upon the base model so far constructed.

What remains is the following:

1. Proceed with a between-subject analysis across more texts for our various measurements. The choice of these four texts was a practical one: these texts were readily available. More certainly exist with which we may apply our various analysis tools to probe authorial stylistics.

2. Build an appropriate etymology machine. The one used here is *very* preliminary, but invites innovation. Indeed, etymological datasets like the Online Etymology Dictionary (Harper & McCormack 2003) can be a good complementary resource to the one scraped from Wiktionary used here. Furthermore, we can be granular with the etymologies in the same way that Reeve (2016b) is, checking the etymological choices of the narrator, deliberate characters, dialogue, *etc.*.

3. Develop an efficient way to track syntactic deviancy, given the computational expense described here.

4. Adapt a sentiment analyzer appropriate for text type. Good work has gone into the NLP of older forms of English (see Johnson et al. (2021)); adapting and including such greater NLP tools for older forms of language, for poetics, and for translated and cultural works is a healthy field of NLP.

Finally, a word should be said about the ethical use of data for tasks in artificial intelligence. This represents ongoing work, and is a part of larger collaboration efforts in open source AI, contributing to open and explanatory artificial intelligence (Ghassemi et al. 2021; Ehsan et al. 2021; Vilone & Longo 2020). All text acquired is publicly available. Powerful tools of language analysis in NLP should be cited appropriately, and should be transparent about the data used in building them (Veres 2021; Abdullah et al. 2022). To this end, effort was made to be transparent in this project, and all forthcoming and presented figures, code, and documentation are to be available as free and open source resources.

References

Abbot, Jean, Elaine Treharne & Mateusz Fafinski (eds.). 2021. *Beowulf by all: Community translation and workbook*. Arc Humanities Press.

Abdullah, Malak, Alia Madain & Yaser Jararweh. 2022. Chatgpt: Fundamentals, applications and social impacts. In *2022 ninth international conference on social networks analysis, management and security (snams)*, 1–8. https://doi.org/10.1109/SNAMS58071.2022.10062688.

Azam, Nouman & JingTao Yao. 2012. Comparison of term frequency and document frequency based feature selection metrics in text categorization. *Expert Systems with Applications* 39(5). 4760–4768. https://doi.org/https://doi.org/10.1016/j.eswa.2011.09.160.

Bird, Steven, Edward Loper & Ewan Klein. 2009. *Natural language processing with python*. OReilly Media Inc.

Blockley, Mary Eva. 2001. Aspects of old english poetic syntax: Where clauses begin, .

Bondi, Marina. 2014. Changing voices: Authorial voice in abstracts 187. 243–269.

Brookshire, Robert & Malcolm McNeil. 2015. *Introduction to neurogenic communication disorders*. Mosby 8th edn.

Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt & Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *Ecml pkdd workshop: Languages for data mining and machine learning*, 108–122.

Collins, Christopher. 1991. *The poetics of the mind's eye: literature and the psychology of imagination*. University of Pennsylvania Press.

Considine, John. 1998. John milton and the uses of etymology. *Henry Sweet Society for the History of Linguistic Ideas Bulletin* 31(1). 5–13.

Croft, Janet Brennan, Donald E Palumbo & CW Sullivan III. 2007. *Tolkien and shakespeare: essays on shared themes and language*, vol. 2. McFarland.

Cureton, Richard. 1980. Poetic syntax and aesthetic form. *Style* 14(4). 318–340. http://www.jstor.org/stable/42946080.

Dewald, Carolyn. 1987. Narrative surface and authorial voice in herodotus' "histories" 20(1). 147–170.

Ehsan, Upol, Samir Passi, Q. Vera Liao, Larry Chan, I.-Hsiang Lee, Michael Muller & Mark O. Riedl. 2021. The who in explainable AI: How AI background shapes perceptions of AI explanations. http://arxiv.org/abs/2107.13509.

Ghassemi, Marzyeh, Luke Oakden-Rayner & Andrew L Beam. 2021. The false hope of current approaches to explainable artificial intelligence in health care 3(11). e745–e750. https://doi.org/10.1016/S2589-7500(21)00208-9.

Green, Clarence. 2019. A multilevel description of textbook linguistic complexity across disciplines: Leveraging nlp to support disciplinary literacy. *Linguistics and Education* 53. 100748. https://doi.org/https://doi.org/10.1016/j.linged.2019.100748.

Gruber, Jeff & Edward Gibson. 2004. Measuring linguistic complexity independent of plausibility. *Language* 80(3). 583–590.

Gummere, Francis Barton. 1910. *Beowulf*. https://www.gutenberg.org/ebooks/981. Hall, J.

Halliday, M. A. K. 1985. *An introduction to functional grammar*. London.

Hare, Adam, Yu Chen, Yinan Liu, Zhenming Liu & Christopher G. Brinton. 2020. On extending NLP techniques from the categorical to the latent space: KL divergence, zipf's law, and similarity search. *CoRR* abs/2012.01941. https://arxiv.org/abs/2012.01941.

Harper, Douglas & Dan McCormack. 2003. Online etymology dictionary. https://www.etymonline.com/.

Heaney, Seamus. 1999. *Beowulf: A verse translation*. Faber London.

Hutto, C. & Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media* 8(1). 216–225.

Jeffrey, David. 1980. Tolkien as philologist. *VII: Journal of the Marion E. Wade Center* 1. 47–61.

Johnson, Kyle P., Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier & William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: System demonstrations*, 20–29. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-demo.3.

Juola, Patrick. 1998. Measuring linguistic complexity: The morphological tier. *Journal of Quantitative Linguistics* 5. 206–213. https://doi.org/10.1080/09296179808590128.

Jurafsky, Dan & James H. Martin. 2009. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition* Prentice Hall series in artificial intelligence. Pearson Prentice Hall 2nd edn. OCLC: 213375806.

Kelleher, J.D., B.M. Namee & A. D'Arcy. 2020. *Fundamentals of machine learning for predictive data analytics, second edition: Algorithms, worked examples, and case studies*. MIT Press.

Khudair, Raghda & Manahil Owaid. 2022. Sentence length and complexity in ernest hemingway's "the doctor and the doctor's wife". *Uruk for Humanities* 15.

Kim, Sang-Woon & Joon-Min Gil. 2019. Research paper classification systems based on TF-IDF and LDA schemes 9(1). 30. https://doi.org/10.1186/s13673-019-0192-7. https://doi.org/10.1186/s13673-019-0192-7.

Kintgen, Eugene R. 1978. Perceiving poetic syntax. *College English* 40(1). 17–27.

Klavans, Judith, Ral Guerra, Rebecca Laplante, Robert Stein & Edward Bachta. 2011. Taming social tags: Computational linguistic analysis of tags for images in museums .

Lesslie. 1892. *Beowulf: An anglo-saxon epic poem*. https://www.gutenberg.org/ebooks/16328.

Lord, John B. 1975. Syntax and phonology in poetic style. *Style* 9(1). 1–31.

Loria, Steven. 2018. textblob documentation. *Release 0.15* 2.

McRoy, Susan. 2021. *Principles of natural language processing*. Susan McRoy. https://uwm.pressbooks.pub/naturallanguage/.

de Melo, Gerard. 2014. Etymological Wordnet: Tracing the history of words. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014)*, 1148–1154. Paris, France: European Language Resources Association (ELRA).

Mohammad, Saif & Peter Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 26–34. Los Angeles, CA: Association for Computational Linguistics.

Mohammad, Saif M. & Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence* 29(3). 436–465.

Pallotti, Gabriele. 2015. A simple view of linguistic complexity. *Second Language Research* 31(1). 117–134. https://doi.org/10.1177/0267658314536435.

Panajoti, Armela. 2013. Poetic syntax: Meaning and structure. *Bulletin of the Transilvania University of Brasov, Series IV: Philology and Cultural Studies* .

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot & E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12. 2825–2830.

Phelan, James. 2014. Voice, tone, and the rhetoric of narrative communication 23(1). 49–60. https://doi.org/10.1177/0963947013511723.

Piantadosi, Steven T. 2014. Zipfs word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin and Review* 21. 1112–1130.

Pierrehumbert, Janet & Ramon Granell. 2018. On hapax legomena and morphological productivity. In *Proceedings of the fifteenth workshop on computational research in phonetics, phonology, and morphology*, 125–130. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5814.

Quamen, Harvey. 2021. Stylometry without words: Analyzing john miltons grammatical style. *Renaissance and Reformation* 44(3). 111–149.

Quigley, Daniel. 2022. Paradise lost data visualization and sentiment analysis. https://dquigley.dev/PL/.

Reeve, Jonathan. 2016a. *A macro-etymological analysis of james joyce's a portrait of the artist as a young man* 203–222. London: Palgrave Macmillan UK. https://doi.org/10.1057/978-1-137-59569-09.

Reeve, Jonathan. 2016b. A macro-etymological analysis of miltons paradise lost. https://jonreeve.com/2016/07/paradise-lost-macroetymology/.

Russell, Stuart J., Peter Norvig & Ernest Davis. 2010. *Artificial intelligence: A modern approach*. Upper Saddle River, NJ: Prentice Hall 3rd edn.

Russom, Geoffrey. 2022. Metrical evidence for the evolution of english syntax. *English Language and Linguistics* 26(3). 583–601. https://doi.org/10.1017/S136067432200017X.

Sayers, William. 2018. Hwaet: The first word of the beowulf poem revisited. *ANQ: A Quarterly Journal of Short Articles, Notes and Reviews* 31(4). 213–217. https://doi.org/10.1080/0895769X.2017.1422427.

Schlesinger, I. M. 1968. *Sentence structure and the reading process*. Berlin, Boston: De Gruyter Mouton. https://doi.org/doi:10.1515/9783111353531.

Slade, Benjamin. 2022. Beowulf, by benjamin slade. http://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp11990.

Stoehr, Taylor. 1964. Syntax and poetic form in milton's sonnets. *English Studies* 45(4). 289–301. https://doi.org/10.1080/0013838X.1964.9709580.

Tolkien, J. R. R. & Christopher Tolkien. 2014. *Beowulf: A translation and commentary*. HarperCollins.

Veres, Csaba. 2021. Language models are not models of language. *CoRR* abs/2112.07055. https://arxiv.org/abs/2112.07055.

Vilone, Giulia & Luca Longo. 2020. Explainable artificial intelligence: a systematic review. http://arxiv.org/abs/2006.00093.

Walkden, George. 2013. The status of hwaet in old english. *English Language and Linguistics* 17(3). 465488. https://doi.org/10.1017/S1360674313000129.
Zhao, Cecilia Guanfang. 2013. Measuring authorial voice strength in l2 argumentative writing: The development and validation of an analytic rubric 30(2). 201–230. https://doi.org/10.1177/0265532212456965.
Zhao, Cecilia Guanfang & Jincheng Wu. 2022. Perceptions of authorial voice: Why discrepancies exist 53. https://doi.org/10.1016/j.asw.2022.100632.

4 Appendices

Various example visualizations are given here. Visualizations include: term frequencies; logspace distribution; wordcloud; n-grams; type frequencies for each text.

Term frequencies follow with the Zipf-Mandelbrot, which are given in Figure 4.

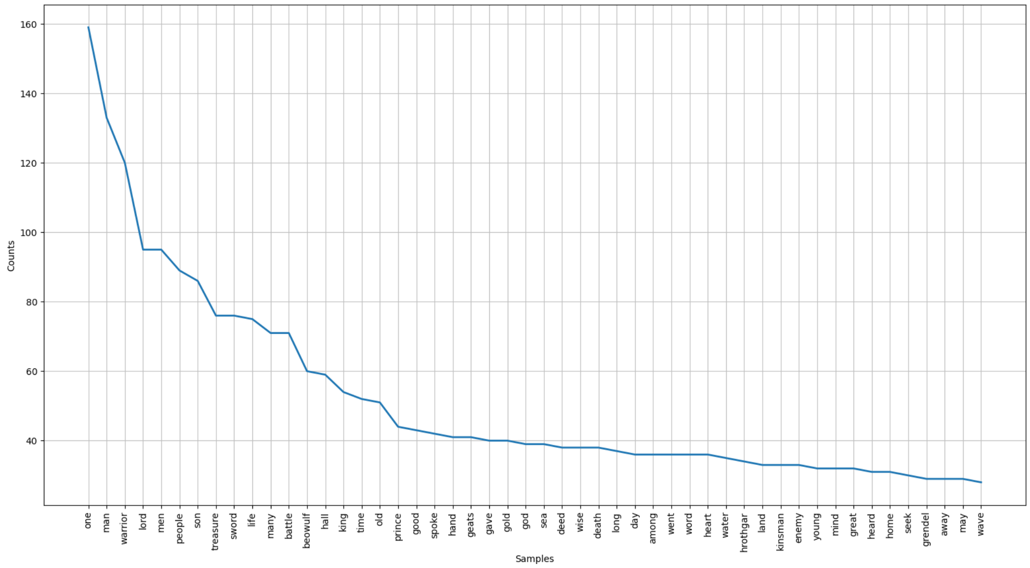

Figure 4. Zipf distribution for All

Term frequencies follow with the Zipf-Mandelbrot, which are given in Figure 5.

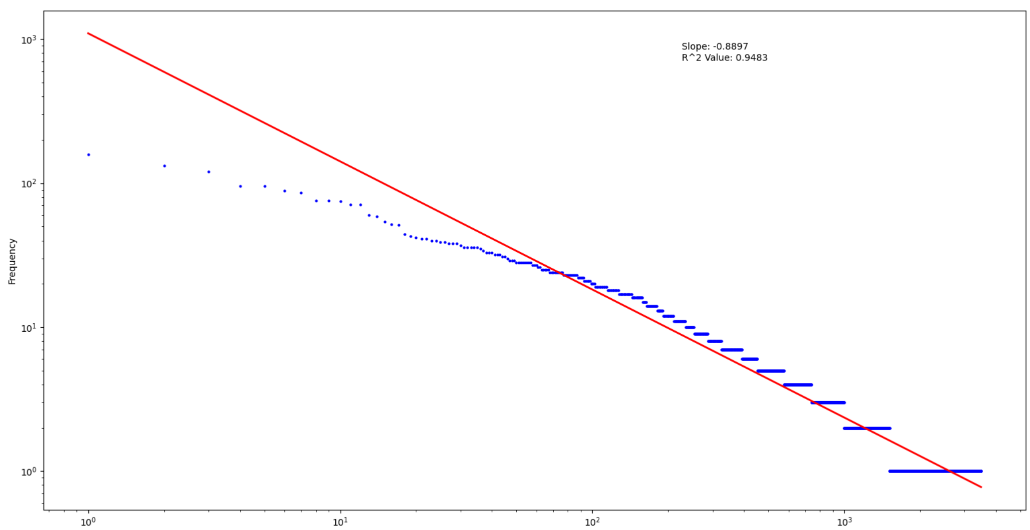

Figure 5. Logspace distribution for All

A useful visualization follows in Figure 6: wordcloud.

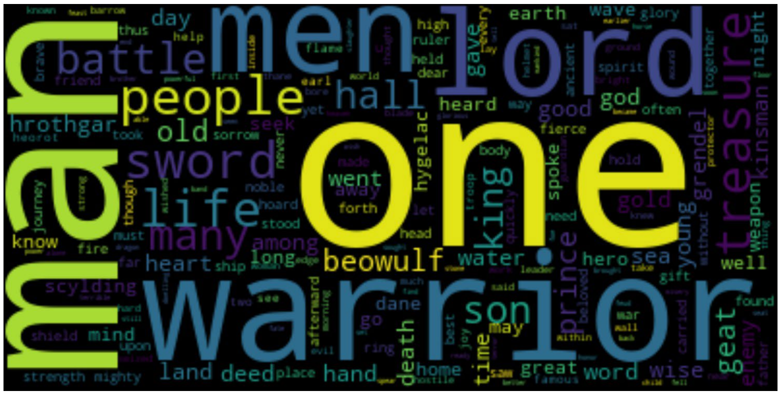

Figure 6. Wordcloud for All

A useful visualization follow in Figure 7: *n*-grams.

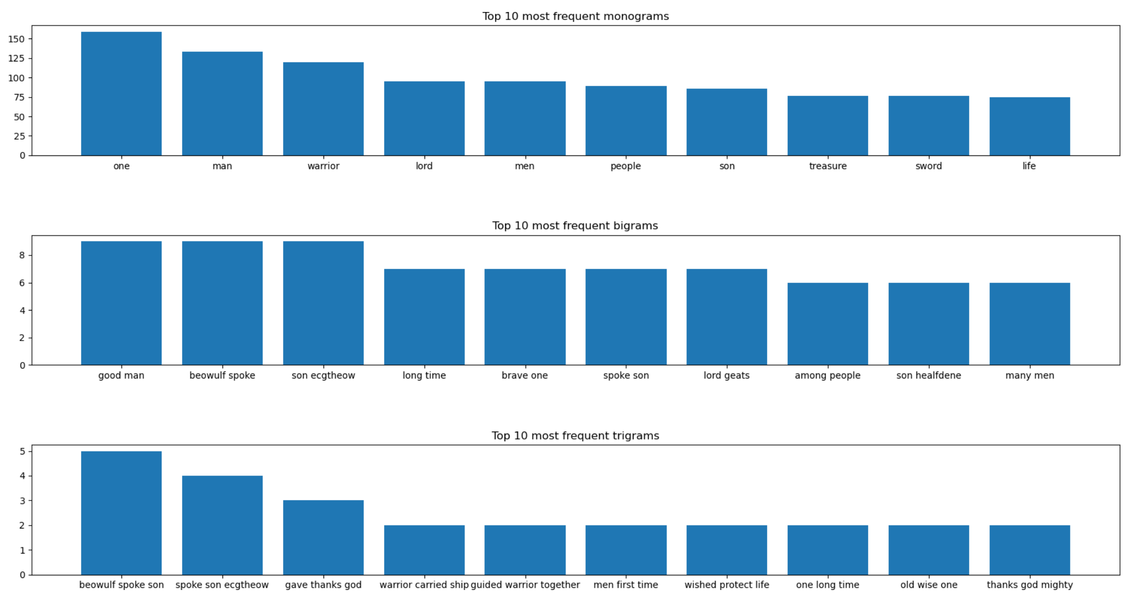

Figure 7. n-grans for All

To be thorough, type frequencies are: nouns, verbs, and adjectives as macro groups, and the different kinds of nouns, verbs, and adjectives as micro groups. Figures 8 and 9 are for All; Gummere, Hall, and Slade follow with Figures 10, 11, and 12, respectively.

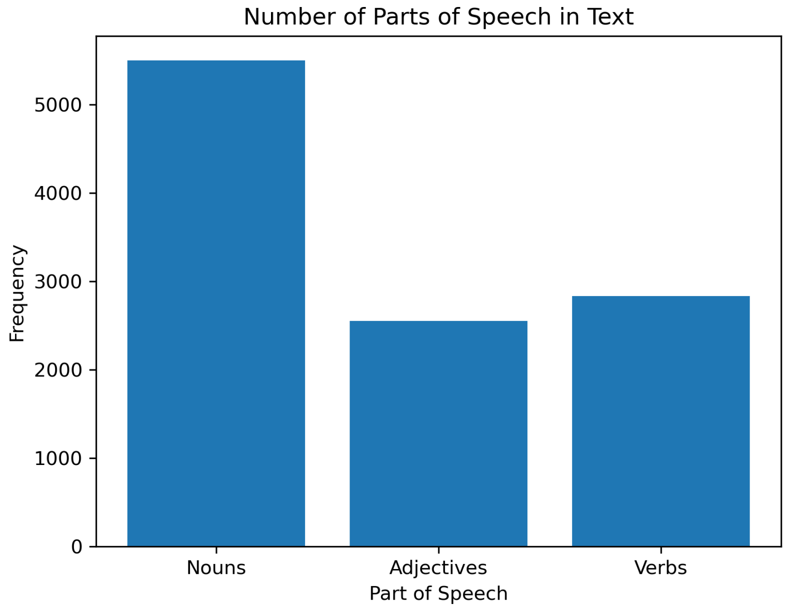

Figure 8. POS frequencies (macro) for All

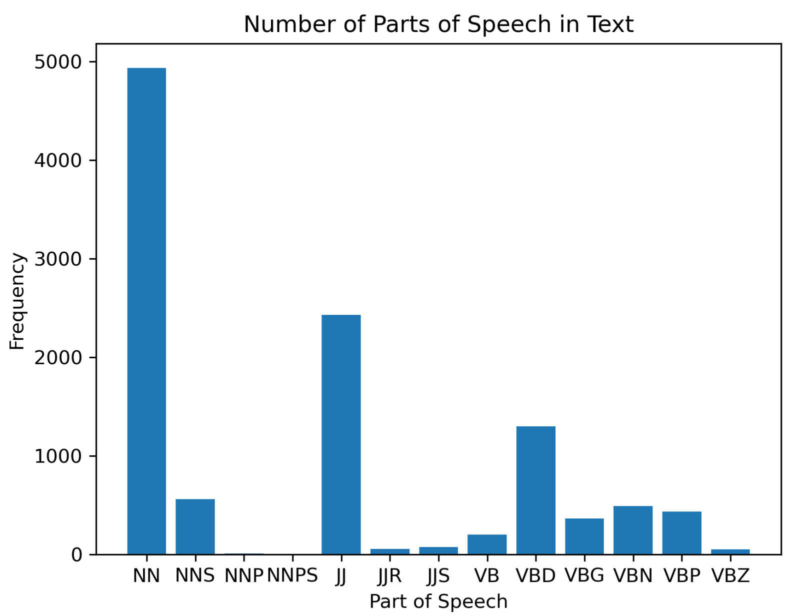

Figure 9. POS frequencies (micro) for All

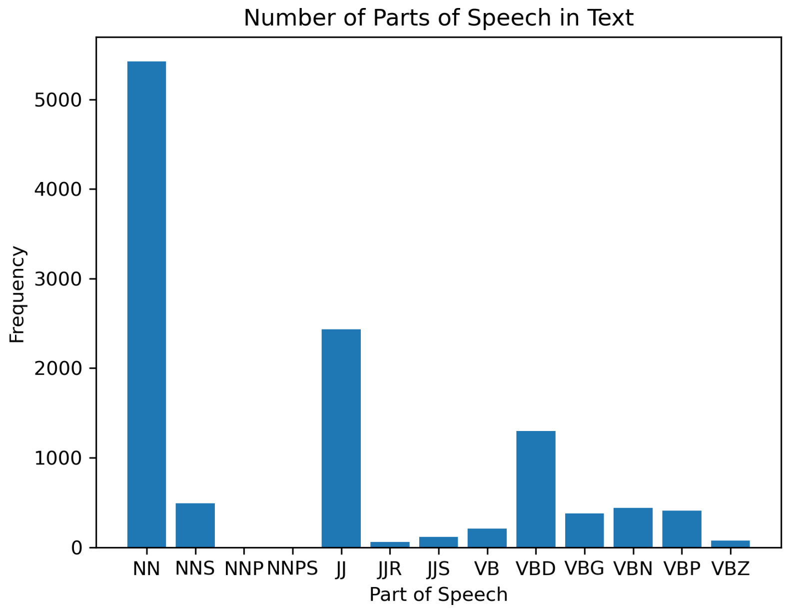

Figure 10. POS frequencies (micro) for Gummere

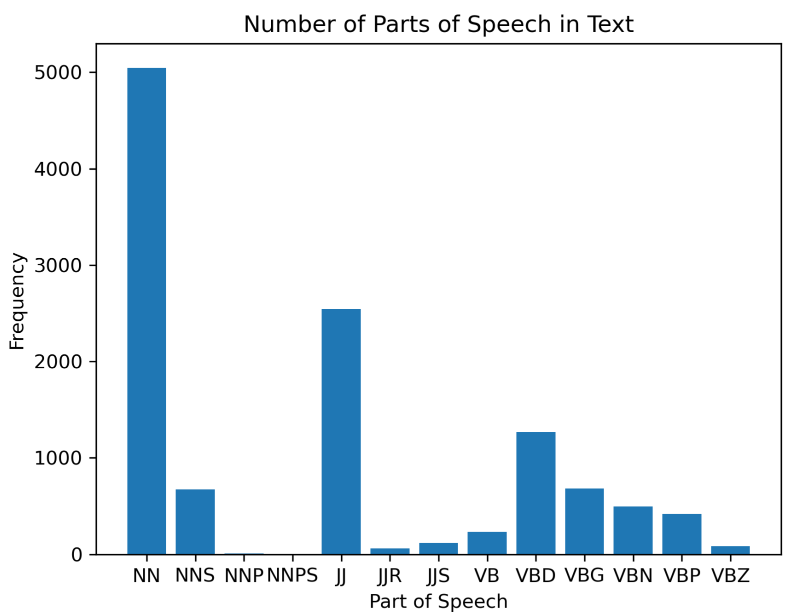

Figure 11. POS frequencies (micro) for Hall

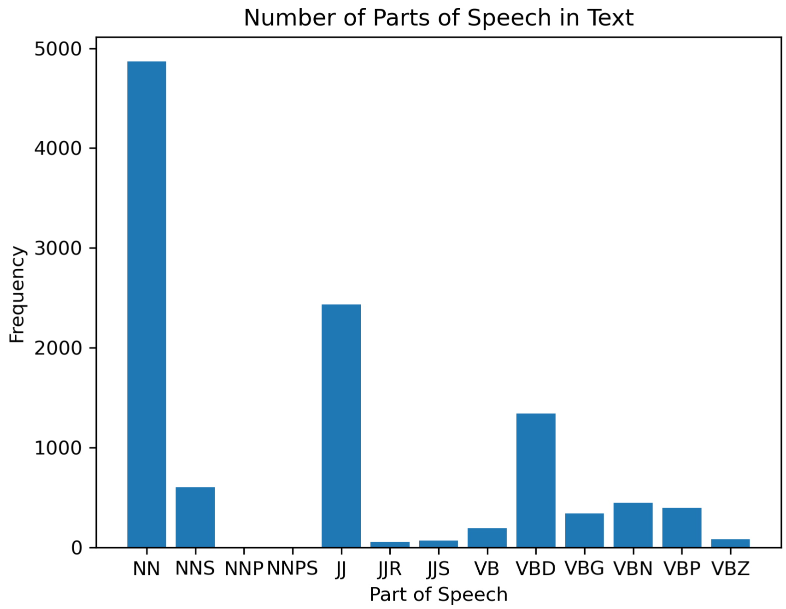

Figure 12. POS frequencies (micro) for Slade