# A Stylometric Investigation of Character Voices in Literary Fiction

by

Krishnapriya Vishnubhotla

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

# Abstract

A Stylometric Investigation of Character Voices
in Literary Fiction

Krishnapriya Vishnubhotla
Master of Science
Graduate Department of Computer Science
University of Toronto
2019

Characters in a novel can be viewed as a set of distinct voices, whose distinguishing features could be a certain dialect, a preference for certain words or phrases, a tendency to talk about a specific set of topics, or any combination of these. In this work, we investigate whether, and how, characters in literary fiction can be distinguished from one another by means of their dialogue. We explore a number of approaches to the problem, including feature sets from authorship attribution studies, and show that Sparse Additive Generative (SAGE) models of text are especially good at picking up on character idiosyncrasies. The SAGE model, when combined with word embeddings, results in high classification scores for our dataset. We then use this model to build a semi-supervised model for quote attribution in literary texts, which achieves performance comparable to that of state-of-the-art systems for the task.

# Acknowledgements

I am deeply grateful to all those with whom I've had the opportunity to work with during the course of my Masters degree. I would like to thank my supervisor, Dr. Graeme Hirst, for his constant advice and support through the entire process. Our weekly meetings were invaluable in helping me find my footing during times of uncertainty. I am also indebted to Dr. Adam Hammond for presenting me with the opportunity to collaborate on such a wonderful project, and his guidance and enthusiasm in finding new research directions to probe.

My sincere thanks to the reviewers and conference chairs of the Digital Humanities Conference and the LaTech-CLfL workshop for their comments and feedback regarding our research.

I am especially grateful to the Natural Sciences and Engineering Research Council of Canada for their support in funding this work. I am grateful to the Department of Computer Science, University of Toronto, for their financial aid as well.

I would like to thank everyone in the Computer Science department, and the Computational Linguistics group in particular, for the many wonderful collaborations and conversations throughout that made this a much more enjoyable learning experience.

Finally, I am grateful for the wonderful friends I made along this journey, both within and outside the University, who made these a memorable couple of years. To my parents and my family back in India, thank you for all the love, support and strength from afar.

# Contents

# Chapter 1

# Introduction

Variation in language can be studied at various levels of granularity; between different languages, between dialects of a language, and within a language. We can look at language use at different historical points in time, or across geographical/social groups and demographics, etc. At a very fine-grained level, we can look at how language use varies between individuals themselves. The idea that each person has a unique stylistic fingerprint in their language use (also called the *the human stylome hypothesis*) is the basis of several fields of study in computational linguistics, such as plagiarism detection and authorship attribution. In the digital humanities, much research has focused on computationally analyzing writing styles of different authors. In this thesis, we focus on the phenomenon of dialogism, primarily investigating the characteristics of character quotes in literary fiction.

Identifying the idiosyncrasies of a person's language proves to be a computationally challenging task. Despite being an active research area for several decades in computational stylometry, it is hard to formulate a fixed set of features that can be used to conclusively pinpoint the writer of any text. One can imagine that one's style of writing varies noticeably based on the domain — whether it is an email, a formal report, or a greeting card. It is also possible that it varies based on whom the speaker is interacting with, in what social situation, and so on. Nevertheless, for certain domains, researchers have been able to identify the author of a text with fairly high precision — we take a closer look at such works in Section 2.1.

In the next few subsections, we will gradually introduce the research problem being addressed in this thesis. We start with a brief look at computational stylometry and its applications, followed by a look at studies in the digital humanities on the topic. We then formalize our research objective.

## 1.1 Stylometry

Stylometry, or computational stylistics, is the formal term for the set of methods that attempt to quantify linguistic style. Though here we will refer to it in the context of written texts, it has also been applied to other fields like music and painting. Stylometric techniques were famously used by Mosteller and Wallace (1963) to determine the authorship of twelve of the disputed Federalist Papers, and to resolve questions of Shakespearean authorship by several researchers (Mendenhall, 1887; Holmes, 1994). They have been used in forensic linguistics to identify writers of ransom notes, for plagiarism detection (Stein et al., 2007), and software and code misuse (Burrows et al., 2009).

Typically, these studies involve identifying certain linguistic properties, such as lexical or syntactic patterns, that can accurately distinguish between the different classes of texts. Authorship attribution refers to the task of identifying the writer of an unknown text $T_{unk}$ from a given candidate set of authors, $A = (A_1, A_2, ..., A_m)$. Let us make the concept clearer by describing a common way of attributing authorship. We first gather a representative set of texts for each author. For simplicity, let us assume we use one representative text $T_k$ for each author $A_k$. A common linguistic feature used in authorship attribution is the frequencies of stop words. Given a predefined set of $n$ stop words $S$, we extract for each author a feature vector $f_k \in R^n$ that consists of the frequencies of each stop-word in $S$. We now extract the feature vector $f_{unk}$ for the unknown text $T_{unk}$, and compare it to the feature vectors for each of our authors. This comparison operation can be Euclidean distance, cosine distance, or passed into a more complicated classifier. Again, for simplicity, let us use Euclidean distance. We can now rank each of our authors by the magnitude of the Euclidean distance between $f_k$ and $f_{unk}$, and declare the one with the least distance to be the most probable author.

This pattern of feature extraction and classification is the most common approach to attribution, and is described in more detail in Section 2.1. These methods are not without their problems, with several debates on their actual predictive power. Note that we are picking the author of the unknown text only from the pool of candidate authors - it would be impractical to compare with every person that has ever written a text. Successful feature sets, in some cases, have been shown to correlate more with genre or topic than the author. It is important that we justify our selection of candidate authors and representative texts, ensure that our methods are not picking up on spurious patterns, and perform simple sanity checks for false positives and negatives. An alternative formulation seeks to treat attribution as a one-class classification problem, wherein we only classify a text as being written by an author or not.

## 1.2 Stylometry in the Digital Humanities

In the digital Humanities, the use of computational stylometry extends beyond characterizing the style of an author. One can look at stylistic patterns that emerge within genres, time-periods, and author groups based on gender, ethnicity, age etc. A fair amount of research, both literary and computational, has looked into how author's fingerprint itself varies within a text. Do different characters in a novel have their own unique style of talking? Can these "character voices" be distinguished from the author's "narrative voice"? The Russian literary critic Mikhail Bakhtin theorized that a dialogic novel is one in which characters present "a plurality of independent and unmerged voices and consciousnesses, a genuine polyphony of fully valid voices" (Bakhtin, 2013). He cites Dostoevsky's works as a preeminent example of dialogic novels, arguing that they are "multi-accented and contradictory in [their] values," whereas the works of other novelists like Tolstoy are monologic or homogeneous in their style, with characters reflecting the prejudices as well as the distinctive mannerisms of their authors.

Perhaps an easily recognizable example of character dialogue uniqueness is that of Ebenezer Scrooge from Charles Dickens's *A Christmas Carol*, and his use of the phrase "Bah! Humbug!". If one were given a random quote from the book that contained this phrase, a reasonable conclusion to draw would be that the quote was uttered by Scrooge. These kinds of catchphrases form a large part of popular character quote idiosyncrasies. However, it is possible for it to extend to lexical and syntactic features, similar to the stylometric features used for attribution attribution.

## 1.3 Research Objectives

Are certain authors more adept than others at creating distinctive characters? What features and computational techniques are most effective in making these distinctions? Are certain syntactic categories of words, such as adjectives, used more by a certain character, or more interestingly, a certain group of characters? This thesis attempts to investigate, and answer, some of these questions.

Our primary objective is analyzing the stylistic distinctiveness of character quotes in literary fiction. Due to the unreliability of automated quote extraction from novels, we focus our experiments on plays written by six different authors, and published in the late 19th and early 20th centuries. Along with predictive performance, interpretability of our models is an important facet. We want to not only be able to distinguish these characters from one another through their quotes, but also pinpoint the features responsible for it.

We explore a variety of methods to determine whether, and how, character quotes in a

play are distinctive. These include authorship attribution models, generative Bayesian models and neural methods. We present a classification method that uses Sparse Additive GEnerative (SAGE) models of text (Eisenstein et al., 2011) to achieve relatively high accuracies on the problem, and has the added advantage of being quite interpretable. We also examine differences in character quotes at the semantic level, making use of previously defined style-based lexicons.

We also look into the problem of quote attribution for literary texts. A key first step towards any analysis of character dialogue is *extracting* this dialogue from the text, and then *attributing* it to the right character. While the former can be done with fairly simple pattern matching algorithms, the latter requires more complex analysis. Armed with our classification model, we propose a semi-supervised alternative that relies solely on the content of the quote to identify its speaker. Despite certain limitations, this method achieves accuracies comparable to those of state-of-the-art systems that utilize contextual information surrounding the quote.

# Chapter 2

# Background and Related Work

This section gives an overview of previous work in authorship attribution and computational approaches to dialogism, and describes the key algorithms used in our analyses.

## 2.1 Authorship Attribution Methods

One of the earliest works that extensively looked at quantifying writing style was Mendenhall (1887), who proposed analyzing the "characteristic curve" of a text to identify authors. Medenhall applied his model to the Shakespeare authorship question, comparing the word-length curves of works by Shakespeare with those of his contemporaries. The study found Shakespeare's characteristic curve to be significantly different from all but one of the rest, the exception being Christopher Marlowe. The validity of any conclusions drawn from these results was however criticized by other researchers, who pointed out that Mendenhall failed to account for significant differences in the genre of the texts.

In 1963, Mosteller and Wallace tackled the problem of authorship attribution for the disputed Federalist Papers (Mosteller and Wallace, 1963). They followed a naïve Bayes classification approach, where the posterior odds of a class is the product of the prior odds and the likelihood. The likelihood function used here, analogous to Medenhall's use of word-lengths, was the relative frequencies of a selected set of *function* words. Using only function words for classification gets rid of the problems introduced by non-homogeneity of text topics, to a certain extent. Since function words are largely used in an unconscious manner by writers, they are able to capture their innate stylistic characteristics across works. The Federalist Papers case was tackled by several other studies in stylometry; Bosch and Smith (1998) used Support Vector Machines (SVMs) with function word frequencies to find a separating hyperplane between the two candidates, Kjell (1994) used artificial neural networks with character n-gram frequencies. All of their findings confirmed Mosteller and Wallace's original conclusion that

Madison was the more likely author of the disputed papers (Demir, 2015).

### 2.1.1 Burrows's Delta

In his 2002 paper, John Burrows proposed the 'Delta' statistic for the authorship attribution problem, which has since been widely accepted as an effective measure for stylistic comparison (Burrows, 2002). Burrows's Delta, as it is popularly known, is a measure of distance between two texts, to be interpreted as the stylistic distance or dissimilarity between them. It is calculated as follows: all the texts written by an author are first combined into one document. We then build a feature vector $f_d$ for each author-document in the set of documents $D$, representing the relative frequencies of the $n$ most frequent words (across all documents) in that text. Each feature $i$ of the feature vector $f_d$ is then standardized using a z-transformation, as follows:

$$z_d^i = \frac{f_d^i - \mu^i}{\sigma^i}$$

where $\mu^i$ and $\sigma^i$ are the mean and standard deviation of feature $i$ across the corpus of documents $D$. Each feature now has a mean of 0 and a standard deviation of 1.

The Burrows's Delta, $\Delta_B$ between two texts corresponds to the Manhattan distance between the feature vectors:

$$\begin{aligned}\Delta_B(d,d') &= \|z_d - z_{d'}\|_1 \\ &= \sum_{i=1}^{n} |z_d^i - z_{d'}^i|\end{aligned}$$

To attribute an anonymous text, we calculate the delta between it and each candidate's sub-corpus, and pick the one with the least magnitude of distance.

Burrows's Delta has been shown to be a consistent measure of style across genres and languages (Evert et al., 2015). However, Burrows himself demonstrated that it is a not a fail-proof measure of the authorial fingerprint; the performance improves as the number of features $n$ increases, it works better for longer texts, and sometimes, the attributed sub-corpus of the author is simply a badly skewed example of their work.

### 2.1.2 Other feature sets

Mendenhall's method used word lengths, Mosteller and Wallace used function words, and Burrows used the most frequent words to build style vectors. Over the years, several other feature sets have been proposed for authorship attribution; these are referred to as *style markers*. This sections reviews some popular ones.

**Lexical features**

1. Surface features: Word lengths and sentence lengths.
2. Vocabulary richness: type-to-token ratio, number of *hapax legomena* (words occurring only once).
3. Bag-of-words models
   (a) Top-$k$ most frequent words
   (b) Frequencies of function words
   (c) Frequencies of punctuation
   (d) Frequencies of word n-grams, i.e, n contiguous word blocks
   (e) Types and frequencies of spelling errors or formatting errors can be interpreted as author-specific idiosyncrasies.
4. Character-level features
   (a) Character type (capitalized letters, digits, punctuation etc.) frequencies
   (b) Character n-gram counts. Character trigrams, in particular, are surprisingly effective at text classification (Sapkota et al., 2015).

**Syntactic features**

The syntactic structure of a text is considered to be a more subconscious choice of a writer than their lexical choices; hence syntactic features are considered more reliable than the latter. The text is first passed through a statistical tagger or parser to obtain part-of-speech (PoS) tags, syntax trees, and dependancy parses of each sentence. We can then extract any or all of the following features:

1. Part-of-speech n-grams; PoS trigrams, in particular, have been found to a good indicator of authorial style (Koppel et al., 2009).
2. Syntactic dependencies: From the output of the dependency parser, we extract relations of the type $(word_{parent}, word_{child}, relation)$ for each relation-arc. Relative frequencies of each triplet are used as features. The triplet is sometimes also represented as $(PoS_{parent}, PoS_{child}, relation)$.
3. Features from syntax trees

(a) Phrase structure rules, or rewrite rules, are obtained from the syntax tree of a sentence. They indicate how the different constituent phrases of the sentence are derived from one another in a top-down manner. For example, for the sentence "The cat jumped over the moon.", we would have the following phrase structure rules:

$$S \rightarrow NP\ VP$$
$$VP \rightarrow VB\ PP$$
$$PP \rightarrow IN\ NP$$
$$NP \rightarrow DET\ NN$$

where S, NP, VP, PP, IN, DET and NN stand for the start symbol, noun phrase, verb phrase, prepositional phrase, preposition, determiner, and noun, respectively. The frequencies of these rewrite rules are used to build the feature vector.

(b) Another approach proposed in Stamatatos et al. (2001) simply used the frequencies of the above constituents, obtained by chunking, as features. This information can be obtained with a higher accuracy when compared to rewrite rules.

(c) Other features that measure syntactic complexity, such as the depth and breadth of the syntax tree.

All of the features in this section rely on having a robust syntactic analyzer; any errors in the parsing algorithm will propagate into the attribution model as well.

**Semantic features**

In general, the more detailed the level of text analysis, the more robust and accurate our attribution system. However, extracting these detailed features requires having robust and accurate text processing tools. Semantic features in stylometry rely on the outputs of semantic parsers, tools like WordNet (Miller, 1995), and other specialized dictionaries.

1. Semantic Relations:
   Semantic role labelling of a natural language sentence, analogous to syntactic part-of-speech tagging, involves identifying the semantic roles of the words or constituents. Semantic roles indicate the semantic (or thematic) relation of a word in the sentence with its main verb — such as agent, actor, beneficiary, etc. Once these have been labelled, a possible feature set is building role-based triplets like with dependency parses, i.e, for the sentence "The boy kicked the ball", we have the triplet *("the boy", "kicked", agent)*.

2. Features from WordNet:
   WordNet is a lexical database for the English language. It groups entities, mostly noun

words and expressions, into clusters called synsets, with entities being linked by relations such as synonymy, hyponymy, anotnymy etc. McCarthy et al. (2006) used this information along with other lexical and syntactic features for authorship attribution; however, there is no detailed explanation on how feature vectors are formed. Tanguy et al. (2012) use WordNet to measure the lexical genericity and ambiguity of each noun, verb, adjective and adverb in the text. Genericity is measured by the depth of the word in the WordNet graph, and ambiguity by the number of synsets the word appears in.

3. Other Lexicons:

   (a) Goldstein-Stewart et al. (2009) used features from the Linguistic Inquiry and Word Count (LIWC) program. LIWC (Pennebaker et al., 2001) defines a range of syntactic and semantic properties for a dictionary of words; the latest version, LIWC2015, contains approximately 12000 entries. The program takes a text as input and outputs aggregated scores along these dimensions; syntactic features include number of pronouns, articles and negations. Some semantic dimensions are the authenticity of the text, its clout (how authoritative is the text?), emotional tone and formality.

   (b) Other lexicons of emotion and style relevant to our work are the sentiment and emotion lexicons from the National Research Council (NRC) Canada (Mohammad, 2018a,b), and the style lexicon from Brooke and Hirst (2013). These lexicons will be explained in more detail in the Section 2.3.4 of the thesis, where we discuss the features used in our analysis.

4. Application-specific features:
   As we noted before, the features used for authorship attribution vary depending on the genre and domain of the texts involved. Some of these features are domain-specific; for example, the type of salutation used in emails, or the tag distribution in HTML documents. A relevant example in recent times is the use of emoticons, abbreviations, and slang in identifying authors of Twitter posts (Schwartz et al., 2013).

## 2.2 Generative Text Models

In contrast to the above approaches of building feature vectors that can discriminate between classes, generative models attempt to directly build a model of the distributions of words in a class. For example, let us assume that each document (or class) has an associated multinomial

distribution over the vocabulary of words. The naïveBayes model, one of the most basic generative text models, models each word in that document as being generated independently by sampling from this multinomial distribution. Though the naïveBayes model makes unrealistic assumptions about conditional independence between words, it is often used as a reliable text classification baseline.

### 2.2.1 Latent Dirichlet Allocation

Other generative models associate latent variables with each document that govern the distribution it samples from. The most widely studied of these is the latent Dirichlet allocation (LDA) (Blei et al., 2003), which calls these latent variables *topics*. This section takes a brief look at how LDA and its variants have been applied to computational stylistics. The next section describes the Sparse Additive Generative model of text (Eisenstein et al., 2011), which is our generative model of choice in this work.

LDA is a latent variable generative model of text that views each document as a Dirichlet distribution over topics, and each topic as a Dirichlet distribution over words. Each word in a document is therefore generated by first sampling from the associated document-topic distribution and then from the topic-word distribution.

Arun et al. (2009) applied LDA to characterize the works of 12 different authors, and in three different scenarios: retaining only stop-words, only content words, and using the complete texts. They observed the best results when LDA was applied to stop-words only - however, they failed to provide a comparison with simple count-based lexical methods. Seroussi et al. (2011) use the topic-distributions obtained for the documents as their representations, and train a classification model. Their results demonstrated that LDA-features were as effective as token-based ones at the task, and had the added advantage of reducing feature dimensionality. A variant of the LDA model, called the disjoint author-document topic model, was used by Seroussi et al. (2012). The authors build their model around the assumption that when authors write texts on the same issue, they share topic-words but have different frequencies of non-topic words (also referred to as author-words). They accordingly propose viewing a document as being generated from two disjoint sets of topics: document topics and author topics.

When applied to large text corpora, LDA associates semantically coherent and meaningful topic words with each document. However, its behaviour is less certain when the training dataset is small, leading to significant noise in the feature set. Further, the number of topics in the model is assumed to be *a priori* knowledge; in experiments, we usually need to try out a set of possibilities within an estimated range and pick the most coherent output. Studies in topic-modeling–based approaches to authorship attribution have nevertheless resulted in one

important observation: *the distributions of function words and topic-words in a text are not completely independent of one another*. Thus, while we must seek to minimize the reliance on topic, we cannot completely ignore it.

### 2.2.2 SAGE

The Sparse Additive Generative (SAGE) model of text proposed by Eisenstein et al. (2011) models text by estimating the log deviations of its word frequencies from a background lexical distribution. In its simplest form, the background lexical distribution is an average of the word frequencies across all the classes. Each class $k$ can then be represented by a vector $\eta_k \in R^V$, where $V$ is the size of the vocabulary, and $\eta_k$ is the log of the frequency deviations of class $k$ from the background. We can write down the probability of a word $w$ given its document $d$ as follows:

$$P(w|d,m,\eta) = \frac{exp(m_w + \eta_{d,w})}{\sum_{i \in vocab} exp(m_i + \eta_{d,i})}$$

Thus, if the document index $d$ is observed, the above equation is equivalent to a naïveBayes model with the multinomial distribution replaced by $\eta_d$. An alternative also to LDA-like models of text generation, the SAGE model enforces a sparse prior on its parameters, which biases it towards rare and infrequent terms in the text. This proves useful for our case, where character distinctiveness can be caused by the use of a few idiosyncratic words or phrases.

## 2.3 Dimensionality Reduction

The LDA model described above helps us to group documents together into soft clusters based on the similarity of their topic distributions. Burrows's method did something similar by measuring the Manhattan distance between feature vectors. While the latter was based only on word co-occurrence statistics, LDA allowed us to discover latent similarities via the concept of topics. Dimensionality reduction techniques seek to minimize the information lost when a high-dimensional vector is transformed, or projected, to a space with fewer dimensions. While getting rid of the problems of large and sparse feature vectors, they are also an excellent way of discovering latent dimensions of textual data. This section documents the dimensionality reduction and visualization techniques we use in our analyses.

### 2.3.1 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) decomposes a matrix $M$ into the product of two, lower-rank matrices, say $W$ and $H$. The sub-matrix W contains the NMF basis; the sub-matrix

H contains the associated coefficients (weights). Suppose we take $M$ to be the document-term matrix with dimensionality $d \times V$, and we wish to obtain $k$-dimensional latent representations for each document. The matrix $W$ with dimensions $d \times k$ now holds the latent vectors for each topic, and $H_{k \times V}$ gives us the relative importance of each term to each latent dimension. NMF is preferred for topic-modeling instead of LDA when the corpus size is small and relatively noisy. It also, however, suffers from the same problems as LDA in selecting the number of topics beforehand.

### 2.3.2 Principal Component Analysis (PCA)

Similar to NMF, principal component analysis (PCA) is extensively used for dimensionality reduction of document-term matrices. The two algorithms, however, minimize different objective functions. NMF finds **non-negative** coefficients, which means that each point in the transformed space can be viewed as a linear, additive combination of features. This results in easy interpretability, especially for document-term matrices like ours. PCA, on the other hand, aims to maximize the variance in the original data that can be explained by the latent vectors. It doesn't impose any positivity constraints; this is intuitively not very explainable. It does, however, lend itself nicely to the task of 2-dimensional visualization. This is a popular use of PCA in NLP applications, wherein one reduces the dimensionality of large document frequency vectors to 2 dimensions, and visualizes them on a graph. This gives one an intuitive idea of how documents might relate to one another; and going back to the authorship attribution problem, to achieve a rough clustering between them. One must take care to check the degree of variance explained by the two principal components before drawing any decisive conclusions.

### 2.3.3 Word Embeddings

Low-dimensional representations of words have been heavily researched for NLP tasks in general. In 2013, Mikolov et al. introduced a set of methods called word2vec, based on the distributional hypothesis [1] that were both computationally efficient and better than previous methods at capturing semantic and syntactic relations between words (Mikolov et al., 2013a). Word2vec can be trained on any text corpus in an unsupervised fashion; the authors also released models pretrained on the 100-billion word Google News corpus. A related, alternative approach to building distributional embeddings was proposed by Pennington et al. (2014) with the GloVe model. These pretrained vectors remain the most popular word representations to date, usually employed as the initial embedding layer of a task-specific network.

[1] Words that are used and occur in the same contexts tend to have similar meanings. As famously quoted by John Firth, a linguist, "a word is characterized by the company it keeps" (Firth, 1957).

Subsequently, there has been a lot of work done on exploring what kinds of linguistic information is encoded in these embeddings. They perform very well on word analogy tasks, a well-popularized example being the following:

$$vector_{king} - vector_{man} + vector_{woman} \simeq vector_{queen}$$

These analogies extend to a variety of semantic and syntactic relations (Mikolov et al., 2013b), such as part-of-speech tags, singular and plural forms of nouns, verb tenses, countries and their capitals, etc.

### 2.3.4 Vectors from lexicons

A glaring disadvantage of the above embedding methods is their lack of interpretability. In 300-dimensional word2vec embeddings, say, it is unclear what the magnitudes of each of those dimensions actually represent in terms of actual linguistic properties. In our experiments, we use certain lexicons that numerically encode words along a defined, specific set of semantic dimensions. This section introduces these lexicons.

1. **NRC Lexicons**
   Researchers at the National Research Council (NRC) Canada have developed a set of lexicons that list the associations of words with a range of sentiments and emotions. We use two of these: the NRC Emotion Intensity Lexicon (EmoLex) (Mohammad, 2018a) and the The NRC Valence, Arousal, and Dominance Lexicon (VAD Lexicon) (Mohammad, 2018b). The former provides real-valued intensity scores for four basic emotions — anger, fear, sadness and joy, and the latter for three dimensions of word meaning — valence, arousal and dominance. The scores for each word and dimension in both lexicons range from 0 to 1, with 0 indicating no association and 1 indicating the highest.

2. **Style Lexicon**
   Brooke and Hirst (2013) adapt the Bayesian LDA model to induce representations for words along multiple stylistic dimensions. Based on previous theoretical studies on literary style, they pick three dimensions to rate words along, the opposing polarities of which give us six styles: colloquial vs. literary, concrete vs. abstract, and subjective vs. objective. The scores along each dimension are normalized to again give us a set of values ranging from 0 to 1.

## 2.4 Experiments on Dialogism

In the above sections, we have described some popular approaches to stylometric text analysis in computational linguistics, as well as the algorithms that we explored in our experiments. Next, we take a brief look at work done more specifically on dialogism in literary works, the premise of this thesis. We conclude the background review with a look at the quote attribution problem, and move towards describing the different components of our system in subsequent chapters.

While the bulk of computational research in stylometry for literature has focused on authorship attribution problems, there has been work that explores differences in character dialogue as well. John Burrows applied stopword-frequency analysis to character dialogue from six novels by Jane Austen, and demonstrated the presence of perceptible differences between them, as well as between character dialogue and narration in the text. Despite these microvariations in character style, the overall style of Austen herself remains distinct from other authors. Thus, these two facets — authorial style and character styles - though seemingly contradictory, have been studied in parallel in the digital humanities.

In their work title "Authors and Characters", Burrows and Craig (2012) analyze dramatic works by Shakespeare and his contemporaries with principal component analysis (PCA) as their primary tool. Once again using the set of frequent function words as their feature set, the authors create 2-dimensional PCA plots for characters from all the plays, across authors. They show that the characters of each author are, with very few exceptions, placed in separate sections of the graph and are linearly separable by an SVM. Within the author-cluster itself, the characters still show range. Thus, while characters of different authors are distinct from one another, they form distinct envelopes within the authors' work as well. An analysis of the most weighted words in the PCA showed some of these character distinctions were due to idiosyncrasies reflecting an older style of dialogue; Shakespeare's characters were marked by their use of "hath" and "thy", while those from plays by John Fletcher tended to use "has" and "your". Other interesting characteristics emerge as well - some Shakespearean characters are more concerned with individuals and questioning (use of second-person pronouns) than Fletcher's ever are. Their results also demonstrated a dependence between character separability and the genre of the works.

A hierarchical clustering approach was taken by Hoover (2017), who used the set of most frequent words (MFW) as his feature set. Hoover first annotated and analyzed *The Hound of Baskervilles* by Arthur Conan Doyle. Each character's dialogue was split into chunks of 1500 words each and run through a clustering algorithm. Dialogue chunks uttered by the same character clustered together in the resultant dendrogram, apart from a few explainable

anomalies. This result also held for a separation of the narration from dialogue; however, this can be mostly attributed to the greater use personal pronouns in dialogue. Hoover also demonstrated that similar results did not hold for novels by other authors; distinctive character dialogue is therefore not a universal property.

Lastly, Muzny et al. (2017a) develop a new metric to quantify the dialogic context of a text, and call it "dialogism". Their corpus consisted of 1100 novels published between 1782 and 2011, allowing them to also look at how this measure varied with the time period. In order to remain agnostic to language changes, Muzny et al. focus solely on syntactic features; here, the distribution of part-of-speech tags. This distribution is calculated for each span of dialogic and non-dialogic (or narrative) text and passed to an algorithm called multi-dimensional analysis (MDA). MDA groups together PoS tags that are positively or negatively correlated with one another, and scores these groups based on the strength of the correlation. The authors pick 7 groups of these tags and combine their MDA scores to ultimately arrive at the dialogism score for the span. Their analysis, however, focused only on the variation between dialogic and non-dialogic text - it doesn't seek to distinguish within the dialogic text itself, a key part of the theory of dialogism we are interested in.

## 2.5 Quote Attribution

An interesting subproblem addressed by the above work is that of quote attribution in novels. With literary texts employing a wide variety of conversational patterns to convey dialogue between characters, automatically extracting quotes and their speakers from them becomes a non-trivial task. We take a more detailed look at this problem in this section, along with a review of previous systems that tackle it.

Quote attribution is the task of identifying the speaker of a quote in a text. It can be decomposed into two components: attaching a quote to an immediate associated mention (quote-to-mention resolution) and attaching the mention to a known speaker (mention-to-entity resolution). A mention can be a proper noun, a pronoun, or an alternative title (for example, *the judge*) that is explicitly used in the vicinity of the quote and is indicated as the speaker. Once identified, the mention must be linked to a known entity from, say, the complete list of character names in a novel or entities named in a news article.

While many quotes can be attributed with simple pattern-matching, others require making use of contextual clues and world knowledge. For example, the very first quote in Jane Austen's *Pride and Prejudice* is as follows:

> "My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

Identifying the speaker of this quote first requires identifying that the addressee is *Mr. Bennet*, and that the speaker is *his lady*. We then need to infer that the term *his lady* refers to *his* wife, where *his* refers to the addressee, Mr. Bennet. The speaker is therefore Mrs. Bennet, a character whose name is not even mentioned until several paragraphs later.

Previous approaches to quote attribution include rule-based systems (Glass and Bangay, 2007; Sarmento and Nunes, 2009), and methods that treat it as a machine learning problem (Elson and McKeown, 2010; O'Keefe et al., 2012). Some methods deal only with quote-mention attribution, while others seek to do it at the entity level.

Early work on quote attribution for literary texts used rule-based systems that looked for certain patterns containing known speech verbs, such as *said* or *interrupted*. Glass and Bangay (2007) used sentence parses to first identify the 'actor' associated with a quote, and then resolved the actor to a speaker using hand-coded decision rules that relied on the local context of the quote. Similar methods were used in other domains, such as news texts, by Sarmento and Nunes (2009).

Elson and McKeown (2010) introduced the Columbia Quoted Speech Corpus (CQSC), containing crowdsourced annotated data for novels by six different authors. Elson and McKeown classified quotes into a set of syntactic categories. They then built a feature vector for each candidate-quote pair, using hand-engineered features such as distance between the candidate and quote, types of punctuation between them, and length of quote. Each feature vector was then passed to a binary classifier that predicted whether the candidate was the speaker of the quote or not. The prediction with the highest confidence was chosen.

O'Keefe et al. (2012) took a different approach, treating the task as sequence labelling rather than classification. This allowed the model to consider the sequential nature of dialogues in a text. Using a feature encoding scheme similar to that of Elson and McKeown, they experimented with three sequence decoding models — greedy, Viterbi, and a linear chain conditional random field. They evaluated both their model and Elson and McKeown's on three corpora: the CQSC and two more from the news domain. Both systems performed poorest on the literary corpus, demonstrating the challenges posed by implicit speaker mentions and varied dialogue patterns in this domain.

The systems described above make minimal use of the actual text of quote, relying largely on surrounding contextual information. He et al. (2013) used the actor-topic model of Celikyilmaz et al. (2010) to model the content of a quote as a distribution over topics, which in turn are distributed across speakers. The complete model used these features, along with contextual clues similar to those of O'Keefe et al., as input to a ranking algorithm that determined who the speaker is.

More recently, Muzny et al. (2017b) presented a state-of-the-art deterministic system that

goes back to the two-stage approach to attribution: quote-to-mention resolution, and mention-to-entity resolution. Both of these subsystems are modelled as a series of deterministic sieves that operate with decreasing levels of confidence. For quote-to-mention resolution, the sieves include pattern matching, analyzing dependency relations, and conversation patterns. The mention-to-entity resolution used, among others, exact name matching and majority speaker attribution. Muzny et al. also introduced a new dataset for quote attribution, the QuoteLi[2] corpus, consisting of three novels annotated with quotes, speakers, and their aliases.

In this work, we use the techniques developed to model character dialogue to build a semi-supervised quote attribution system. Our classification model uses features based on weights from the Sparse Additive Generative model of text (SAGE), along with pretrained word embeddings (GloVe).

[2]http://nlp.stanford.edu/∼muzny/quoteli.html

# Chapter 3

# Datasets

## 3.1 Literary Texts

In the literary domain, datasets used for stylometric analyses mostly depend on the researchers' own interests in analyzing certain works or authors. Shakespeare and his contemporaries have been a fixture for many (Mendenhall, 1887; Craig and Kinney, 2009; Tearle et al., 2008), others have looked at specific time periods (Holmes et al., 2001; Muzny et al., 2017a), and yet others at the evolution of an author's style with time (Lancashire and Hirst, 2009).

For experiments in dialogism, there's an important precursory step that influences our choice of corpus — the availability of attributed quote data. As discussed in Section 2.5, there are a few datasets available for quote attribution that contain speaker-annotated texts. These are, however, limited in number; and the related quote attribution systems themselves are far from perfect.

With statistical classification algorithms being sensitive to noisy data, especially in low-data scenarios, our intention was to avoid introducing any avoidable inconsistencies into our dataset. Therefore, we focus our dialogism experiments on plays rather than novels. The advantages of doing so are twofold: first, quote attribution becomes a largely trivial task. Dialogues in a play are normally presented in a fixed format, as below:

*Character A:* Quote A
*Character B:* Quote B

The only variance we observed was in the choice of the separating character, as the colon was replaced in some cases by a newline character. Secondly, the dialogue-only nature of plays provides us with an excellent avenue to test Bakhtin's theory of distinctive character voices discussed in Section 1.2.

| Author | #Plays | #Characters |
|---|---|---|
| Shaw | 29 | 252 |
| Wilde | 6 | 49 |
| Maugham | 8 | 76 |
| Grundy | 4 | 30 |
| Pinero | 5 | 60 |
| Sudermann | 5 | 53 |
| Rice | 6 | 52 |

Table 3.1: Number of plays and characters for each author.

Our corpus, therefore, consists of plays published in the late 19th and early 20th centuries by six playwrights: George Bernard Shaw, Oscar Wilde, Cale Young Rice, Sydney Grundy, Somerset Maugham, Arthur Wing Pinero, and Hermann Sudermann (whose plays are translated from German). We restrict our dataset to plays written by these authors that were published between 1880 and 1920, to roughly capture the literary period from which Bakhtin developed his theory of dialogism. The plays are extracted using GutenTag (Brooke et al., 2015), a tool that, among other things, functions as a corpus reader for the Project Gutenberg text corpus[1]. Specifying the above constraints in the GutenTag tool gives us a total of 63 plays. Statistics on the number of plays and characters for each author are shown in Table 3.1.

## 3.2 Quote Attribution

We briefly mentioned the Columbia Quoted Speech Corpus (Elson and McKeown, 2010) in Section 2.5. It consists of annotated texts written by six authors who published in the 19th century, with a mix of short story collections and complete novels; some of the novels only have certain excerpts annotated. The annotation task was crowdsourced on Amazon's Mechanical Turk platform, a consequence of which is certain inaccuracies in the dataset. Muzny et al. (2017b) found that 57.8% of the quotes in the corpus either do not contain a speaker label or are annotated with a speaker label that is not linked to an entity. The corpus suffers from labelling errors as well — 8% of the quotes with resolvable speaker labels are incorrect (O'Keefe et al., 2012).

He et al. (2013) provide a version of Jane Austen's *Pride and Prejudice* in which each quote is annotated with the speaker entity. Unlike the CQSC, which has quote-mention labels with labels linked to entities, He et al.'s corpus directly provides with quote-entity labels. He et al. assume that all quotes within a paragraph are by the same speaker — an assumption that is not always true, but rarely violated.

[1] http://www.gutenberg.org

| Novel | Explicit | Other | Total |
|---|---|---|---|
| *Pride and Prejudice* | 555 | 1192 | 1747 |
| *Emma* | 240 | 494 | 734 |
| *The Steppe* | 278 | 344 | 622 |

Table 3.2: Numbers of quotes of each type in the QuoteLi dataset. Explicit quotes have the speaker's name in the vicinity of the quote, while other mentions include anaphoric and implicit conversational patterns.

Finally, Muzny et al. (2017b) released the QuoteLi corpus with annotated quotes for three novels. Each quote is linked to both the associated mention and the character entity, resolving conflicts between the previous two datasets. It is annotated by experts and not crowdsourced, so there are far fewer avenues for incorrect attributions[2]. Statistics for this corpus are presented in Table 3.2. In our experiments on quote attribution, we focus solely on the QuoteLi corpus.

[2]We did not come across any significant errors in the corpus during our experiments, apart from a few in mention-entity linking; for example, one mention *Miss Bennet* was incorrectly linked to the character Elizabeth Bennet instead of Jane Bennet.

# Chapter 4

# Models and Methods

In this section, we will describe the different feature sets that we experimented with, the classification pipeline, and evaluation methods used for both analyzing character quotes and the quote attribution task.

## 4.1 Character-quote classification

The primary question we are trying to answer is whether character voices are stylistically distinct. We take the classification approach to this problem: our task is to build a classifier that is able to correctly discriminate between the speech of different characters. For each text, the set of characters is the set of classes for our classifier. The set of quotes extracted for each character comprise our dataset; we keep each individual quote as one datapoint. These are split into training and test sets in a 7:3 ratio. In almost all cases, we build our feature vector for each datapoint and pass them into a classifier for evaluation.

In line with previous work on authorship attribution, we start with the most commonly used style markers: surface features and function words.

### 4.1.1 Feature Set 1: Surface and Lexical

For each datapoint, we extract the following features:

1. Distribution of word lengths
2. Sentence length
3. n-grams of only stop-words, where n $\in \{1,2,3\}$.

Since word and PoS n-grams are very sparse features, the resulting feature vector has a relatively high dimensionality. We therefore pass it through a feature selection pipeline before classification. Two main feature selection algorithms are used: variance threshold and k-best selection. The former removes all features with a zero variance across samples — i.e, features that have the same value at each datapoint. The k-best selection algorithm then picks the top-k features according to some correlation measure. Here, we use the chi-squared statistic, which gets rid of the features that are the most likely to be independent of class and therefore irrelevant for classification.

We use this as our default feature selection pipeline in all experiments.

### 4.1.2 Feature Set 2: Lexical and Syntactic

Our second feature set adds on to Feature Set 1, where we use all words instead of only function words. We also extract syntactic features for each sentence: part-of-speech tags and dependency relations.

1. Word n-grams, where n $\in \{1,2,3\}$

2. PoS n-grams, where n $\in \{1,2,3\}$

3. Dependency relation triples:
   Dependency relations between words in a sentence are obtained by using a dependency parser. We then extract triples of the form $(PoS_{parent}, PoS_{child}, relation)$. For example, consider the sentence "I went to the park to get some air.". The output of the dependency parser gives us the following:

   went [(I, nsubj), (to, prep), (get, advcl), (., punct)]
   to [(park, pobj)]
   park [(the, det)]
   get [(to, aux), (air, dobj)]
   air [(some, det)]

   For each parent-child arc, we now build our triples as follows:

   (VERB, PRON, nsubj), (VERB,ADP,prep), (VERB,VERB,advcl), (VERB,PUNCT,punct)
   (ADP,NOUN,pobj)
   (NOUN,DET,det)
   (VERB,PART,aux), (VERB,NOUN,dobj)
   (NOUN,DET,det)

For each of the above feature sets, we create feature vectors based on either simple frequency counts, or TF-IDF counts, and pass them through the feature selection algorithm.

### 4.1.3 Feature Set 3: Weighted Sentence Vectors

Similar to the way in which word embeddings are used as representations of word meaning, sentence embeddings attempt to capture salient properties of a sentence or paragraph in a single vector. The current state-of-the-art in sentence embeddings uses large neural networks trained on massive datasets, on tasks such as language modeling. Concurrently, there have been several studies that prove simpler methods like weighted averages achieve performance on par with, if not surpassing, these neural models (Arora et al., 2017; Shen et al., 2018). The simplest approach to constructing sentence vectors involves averaging the embeddings of the words present in it. Another alternative is to take a weighted average of the constituent word vectors; these weights can be obtained as TF-IDF counts or via topic modeling techniques.

We use the SAGE model to derive weights for each quote in our dataset. Sentence vectors are obtained by averaging the vector representation of each word in the sentence with its corresponding SAGE weight. We use GloVe embeddings pretrained on Common Crawl data as our word representations, obtained from `https://nlp.stanford.edu/projects/glove/`. These sentence vectors are now our feature vectors for each quote, to be passed into the classifier.

### 4.1.4 Classifiers

Once we have the feature vector for each quote, we train a supervised classification algorithm on the dataset. Each play is treated as a distinct dataset, and each character as a separate class. We split the dataset into a training set and a test set; character distinctiveness is measured by the $F_1$ score of the classification algorithm for that particular class.

We test our methods with two main classification algorithms: support vector machines (SVMs) and logistic regression (LR). Both of the models come with their own set of advantages. SVMs have been shown to perform well with high-dimensional data, and have the added advantage of easy feature interpretability. Thus, the coefficient matrix returned by the SVM algorithm can be used to identify the top positive and negative contributing features for each class.

Logistic regression works well with low-dimensional data, and returns a probability distribution over the classes for each datapoint. This lets us identify the confidence with which each datapoint was assigned to a class, and can be used as a measure of the uniqueness of a quote to its character.

In our experiments, we use an SVM for classifying feature vectors that use lexical and syntactic features (Feature Sets 1 and 2), and a logistic regression classifier for the vector-based representations (Feature Set 3).

## 4.2 Exploratory Work

While the classification approach provides us with a concrete, numerical measure of distinctiveness, we also glean useful insights from unsupervised methods such as LDA, NMF, and PCA. We perform a second set of experiments applying these techniques to our dataset.

1. **Topic Modeling:**
   We consider the complete set of quotes of each character to be one document. We run two topic-modeling algorithms, LDA and NMF, on the collection of documents to obtain a document-topic distribution for each document, and a topic-word distribution for each topic. This allows us to retrieve the top terms associated with each document, i.e, the distinguishing lexical features associated with each character. We begin by setting the number of topics $k$ to be equal to the number of characters, and gradually decrease it to the least possible number, $k = 2$. This allows us to observe which character(s) remain(s) the most distinct in their vocabulary usage when compared to the rest.

2. **Visualizing Style Vectors**
   Analogous to word representations obtained with GloVe, we build *style vectors* for each word by using the NRC and style lexicons described in Section 2.3.4. The scores along each dimension in both dictionaries are first normalized to lie between 0 and 1. We construct three sets of vectors based on the lexicons used for the task: only the style lexicon (from Brooke and Hirst (2013)), only the NRC lexicons (from Mohammad (2018a,b)), and a combination of the two. If a word does not have an entry in either one, or both, of the lexicons, we assign a score of zero to the missing dimensions.

   Once we have style vectors for all of the words in both lexicons, we construct a vector representation of each character following a procedure similar to that of sentence vectors. We concatenate the all of the character's quotes and average the style vectors of each of the constituent words. These vectors are then projected to a 2-dimensional space using PCA; the first two principal components are plotted to visualize the relations between different authors as well as characters.

## 4.3 Quote Attribution

We take the best performing classification system from the above approaches to build a semi-supervised model for the quote attribution task. Semi-supervised classification algorithms typically use a small amount of labelled data, along with a large amount of unlabelled data, to iteratively build an accurate system. We follow the self-training procedure of Yarowsky (1995) to classify the rest of the quotes. Assuming that we have an initial labelled set $S = (X_l, Y_l)$, and a set of unlabelled points $U = (X_u)$, self-training uses the predictions of a classifier trained on the labelled set to iteratively add points from $U$ to $S$.

### 4.3.1 Seed Set Extraction

For our semi-supervised method, we need to extract a seed set of attributed quotes for the initial classification round. Since it is important for this seed set to be as accurate as possible, we use the first of the high confidence sieves described by Muzny et al. (2017b), which is trigram matching. This is also used by Elson and McKeown (2010) and O'Keefe et al. (2012) as a feature for their models. We first define a set of speech verbs, along with all their inflected forms. Quotes are identified as any text enclosed between opening and closing quotation marks, and proper nouns are tagged by using the character names and aliases defined in the annotated text. We then search for all trigram permutations of *Quote–Person–Verb* in our text, and attribute the *Quote* in each case to the corresponding *Person*.

### 4.3.2 Semi-supervised Attribution

At each iteration, we extract datapoints that are classified with a confidence score greater than a predefined threshold value and add them to the set of classified quotes. We keep our seed set constant in the labelled set throughout; quotes that were added in one iteration may be removed in another round if their classification confidence falls below the threshold. We halt the procedure when no new datapoints are added to the classified set in a round, capped by an upper bound on the number of iterations.

For the classification, we represent sentences by weighting word vectors with SAGE coefficients, and perform classification with a logistic regression model. For each quote, we obtain the weighting coefficients by finding its SAGE coefficients with respect to the entire data distribution, i.e, both the test and training data contribute to the background lexical distribution. The word vectors are pre-trained GloVe embeddings Pennington et al. (2014). We set the threshold for our semi-supervised training procedure to be the average of the confidence scores on the training set.

## 4.4 Experimental Details

### 4.4.1 Dialogism

For all plays, we filter out those characters with less than 20 attributed quotes. Our initial exploratory experiments using topic modeling and clustering are limited to plays by Oscar Wilde and George Bernard Shaw; we later extend it to all six playwrights.

All of our models were implemented using Python's scikit-learn library[1]. Syntactic features were extracted using the spaCy[2] text processing library. For our lexical features, all proper nouns in the text are masked, i.e, replaced with an UNK token. This is done to avoid lazy classification based on character names — the presence of a proper noun in a quote is often a strong indicator of who the speaker is, or more appropriately, is not. We observe that masking results in more informative discriminating features.

For classification experiments, the dataset is split into training and test sets in a 7:3 ratio. Hyperparameters of the classifiers are tuned using grid search, along with 5-fold cross validation. The SVM classifier can be used either with a linear kernel or an RBF kernel. Though the RBF kernel achieves slightly (1-2%) higher accuracies, we restrict ourselves to the linear kernel for better feature interpretability. Model performance is measured using the $F_1$ score, which strikes a balance between precision and recall. We also report a baseline scores for these experiments, where the baseline randomly generates predictions that respect the class distributions of the training data.

The class sizes for all of our plays are also heavily imbalanced. We use the Synthetic Minority Oversampling Technique (SMOTE) to over-sample the minority classes and balance the classification problem (Chawla et al., 2002). Though this ensures a more balanced precision-recall score across classes, it doesn't entirely solve the problems of majority classes overpowering the classifier.

#### Artificial Plays

In addition to running our classification on the set of original plays, we create a set of "artificial plays" using our complete list of characters. These artificial plays are constructed by means of two strategies — by sampling a random subset of characters either across plays (strategy 1) or across authors (strategy 2). Intuitively, we expect the character speech in these artificial plays to be more readily distinguishable than in actual plays, because the characters are likely to discuss a wider variety of topics and to come from a wider variety of classes, professional

[1] https://scikit-learn.org/stable/
[2] https://spacy.io/

| Novel | % of total | Seed set | Unassigned |
|---|---|---|---|
| *P&P* | 77.6 | 372 | 952 |
| *Emma* | 91.4 | 513 | 526 |
| *The Steppe* | 71.3 | 199 | 207 |

Table 4.1: Proportion of total quotes we consider, the seed set sizes, and number of unassigned quotes for each novel. *P&P* = *Pride and Prejudice*.

milieus, and dialect communities than a group of characters in any actual play (strategies 1 and 2), and because the characters are the creations of different authors, each with their own distinct stylistic fingerprints (strategy 2). We fix the number of characters for each artificial play to be 7 — a natural constraint in the case of strategy 2, and one we keep to ensure uniformity. All characters with less than 20 utterances are, again, not considered. We also constrain the maximum difference in the class sizes to be less than 100 to mitigate the class imbalance problem. We generate a maximum of 50 artificial plays for each author by sampling 7 characters from the complete set of characters, without repetition.

### 4.4.2 Quote Attribution

Because of the limitations of the CQSC discussed in Section 3.2, we use the QuoteLi corpus, with minor corrections, as our gold standard, even though this precludes comparing our results with previous work that used the CQSC for evaluation.

We observe that, unsurprisingly, the character-quotes distribution in all the novels is heavily skewed. Few example, the central character in *Emma* has almost twice as many quotes as the next character. Further, the distribution has a long tail, with several minor characters having very few attributed quotes in the seed set. To avoid these issues affecting the performance of our classifier, we restricted our experiments here to those characters with at least 15 attributed quotes in the seed set. The portion of total quotes covered by this method, and their statistics, are reported in Table 4.1.

Our word vectors are 300-dimensional GloVe embeddings,[3] pretrained on Common Crawl data. Sentence vectors are constructed as a weighted average of the individual word vectors, where the weights are the corresponding SAGE coefficients. The sentence vectors are passed to a logistic regression classifier trained using stochastic gradient descent, with all hyperparameters tuned using grid search. We keep all words from the text in our vocabulary, including stop words and punctuation. Running the text through our trigram pattern-matching algorithm gives us a seed set of attributed quotes. We then run the semi-supervised algorithm until no

[3] https://nlp.stanford.edu/projects/glove/

new quotes are classified, with an upper cap of 20 iterations. In each iteration, the threshold is set to the average of the confidence scores of the training data; the seed set quotes are left untouched.

# Chapter 5

# Results and Discussion

## 5.1 Dialogism

Recall from Section 4.1 the three different feature sets we experimented with for classification of character quotes. For all six authors, and all feature sets, we report the average $F_1$ score of classification across all of their plays in Table 5.1.

The scores for all authors are also consistently above the baseline, though the absolute values themselves are not very high. The numbers for Feature Set 1 suggest that traditional markers of style (as used in authorship attribution) do not serve as reliable indicators of character-style — at least, not at the quote level. By not restricting ourselves to function words, we achieve better classification. Equivalently, we can view this as including topical features in the model, i.e, *what* characters talk about, along with stylistic features. We achieve the best classification results, however, using the SAGE model for text, which models the log frequency deviations of each characters text from a background distribution, along with word2vec vectors and a logistic regression classifier.

There is a high amount of variation in the $F_1$ scores of different authors, characters and plays. Figure 5.1 plots these scores for the plays of each author, sorted in increasing order of the $F_1$ scores, for Feature Set 2; we see similar trends for the other feature sets as well. Shaw and Wilde consistently achieve the highest classification scores; this trend persists among all models.

At the character level, looking at the top features from the SAGE algorithm provides insights into the easiest types of stylistic distinction one can make while creating characters. Servants and butlers are easily recognizable by their use of words such as "sir", "yes" and "please". In Shaw's *Pygmalion*, the character of The Flower Girl is distinguished by her unique vocabulary of words like *'ow', 'ai', '–', "'m", 'ah', 'oo'* etc. These kinds of lexical, dialectal features seem to be the most popular way of creating unique character voices; however, we

| Author | Baseline | FS 1 | FS 2 | FS 3 |
|---|---|---|---|---|
| *Shaw* | .153 | .287 | .400 | **.635** |
| *Wilde* | .116 | .279 | .376 | **.641** |
| *Maugham* | .137 | .219 | .318 | **.622** |
| *Pinero* | .090 | .153 | .272 | **.458** |
| *Grundy* | .107 | .216 | .283 | **.517** |
| *Sudermann* | .084 | .150 | .253 | **.538** |
| *Rice* | .151 | .190 | **.234** | .181 |
| Weighted Avg. | .133 | .242 | .342 | **.561** |

Table 5.1: Average $F_1$ scores of classification for each author, with the feature sets from Section 4.1. The final row reports the weighted average of the scores for each author, where the weights are proportional to the number of their plays in our dataset. Baseline is stratified random classification.

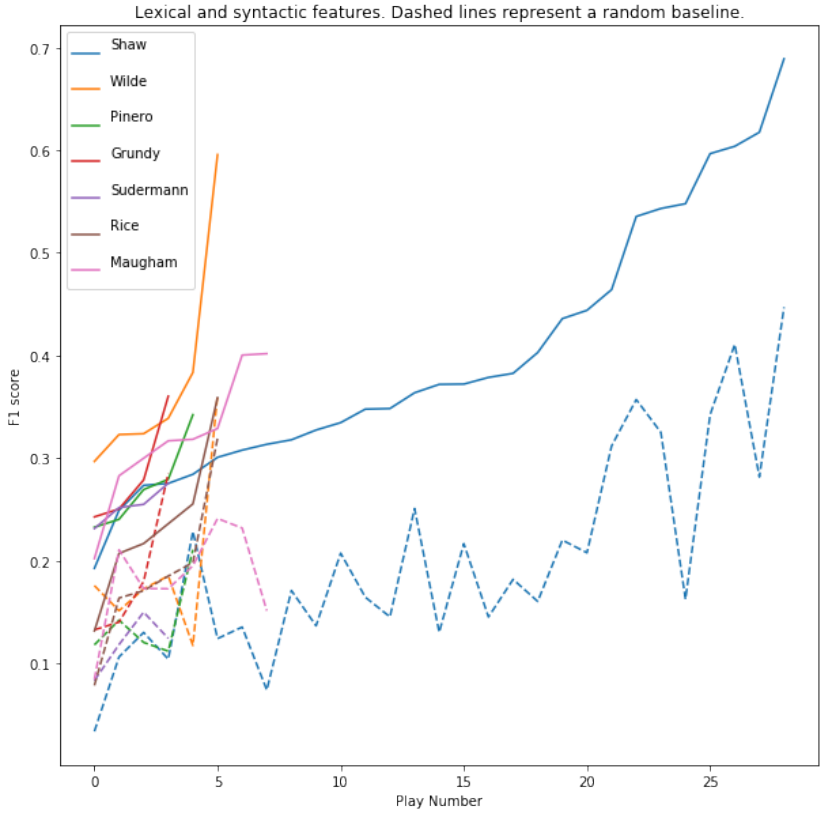


Figure 5.1: $F_1$ scores of classification for the plays of each playwright for Feature Set 2. Dashed lines represent the baseline.

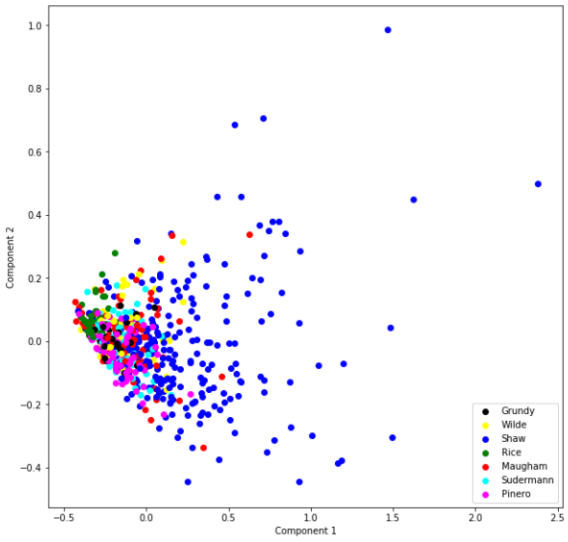


Figure 5.2: PCA Plot of the concatenated style+NRC vectors for all characters.

keep in mind that the word embedding features don't lend themselves to such interpretability.

We did attempt an initial clustering experiment with the word embeddings, which resulted in some insightful clusters. Proper nouns were grouped into one, another had words associated with tragedy (*[sad, dreadful, miserable, awful, horrible, terrible, unfortunate]*), and yet another with *[duty, servants, rank, ideals]*. These are indicative of some stylistic aspect of words being captured by the embeddings which, when combined with the SAGE weights, boosts our classification performance. However, we reiterate that quantifying this is a hard-to-solve problem.

Our experiments with topic-modeling return similar results, and are again confined to lexical features only. Let us again take Shaw's *Pygmalion* as our case study, which has a total of 10 characters. At the lowest possible value of $k = 2$ for the number of topics, we have *The Flower Girl* separate from the rest of the characters; this is followed by *Mrs Pearce*, the housekeeper, and then the rest of the characters. Similarly in Wilde's *The Importance of Being Earnest*, the butler (Lane) emerges as the most distinctive character.

The PCA plots we generate from the NRC and style lexicons allow us to examine character distinctiveness at the author level. Figure 5.2 plots the first two principal components obtained from the PCA of the concatenated style+NRC lexicons for each character. Each dot corresponds to a character in a play, and wider spacing between them indicates a wider range of styles and emotions. It is immediately obvious that Shaw's characters are the most stylistically diverse, followed by Maugham and Wilde. These two components explain approximately 75% of the variance in the data.

We can also look at the contribution of each of the style-dimensions towards each PCA

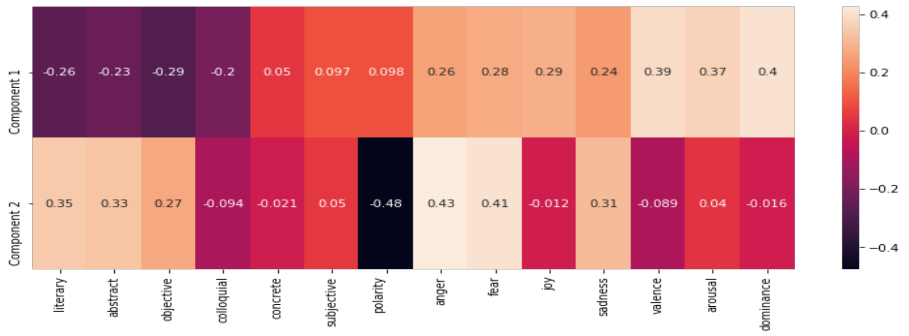


Figure 5.3: Contribution of each stylistic dimension towards the principal components.

| | Average $F_1$ | | |
|---|---|---|---|
| | **Baseline** | **FS 2** | **FS 3** |
| Wilde | .194 | .569 | **.669** |
| Shaw | .148 | .573 | **.630** |
| Maugham | .182 | .545 | **.645** |
| Sudermann | .119 | .462 | **.574** |
| Grundy | .184 | .402 | **.517** |
| Pinero | .140 | .537 | **.543** |
| Rice | .186 | **.258** | .208 |

Table 5.2: $F_1$ scores for classification of characters in strategy 1 artificial plays using feature set 2 (lexical and syntactic) and feature set 3 (SAGE classification model).

component. These numbers are presented in Figure 5.3. The valence, arousal, and dominance dimensions are the most influential; these three are also highly correlated with one another.

Table 5.2 presents the classification results for strategy 1 artificial plays. These are generated by sampling a random subset of characters from across all of an author's plays. The scores are on average higher than those of the original plays — but surprisingly, not very much higher. We expected that characters coming from different plays would be more easily distinguishable from one another merely because of topical differences in the dialogue. We note, however, that the gradient among authors in their ability to write unique characters follows a pattern similar to that in the original plays. The average $F_1$ score for strategy 2 artificial plays is even higher, at 0.605.

The number of sources of variance in our artificial plays make it hard to interpret these results. Even with the number of characters being fixed, the lengths of the plays range from as low as 300 quotes to as high as 1000. The distributions of character-quote lengths within these plays can fluctuate as well. We could, in certain cases, be picking a biased mixture of "main" and "side" characters, each with their own varying degrees of stylistic uniqueness. Investigating these scores requires performing more controlled experiments regarding our choice

| | Munzy | Our work | | | |
|---|---|---|---|---|---|
| **Novel** | **Acc.** | **P** | **R** | **$F_1$** | **Acc.*** |
| *P&P* | **.851** | .77 | .70 | .68 | .703 |
| *Emma* | .759 | .83 | .82 | .81 | **.817** |
| *The Steppe* | .727 | .81 | .80 | .80 | **.801** |
| Average | **.779** | .80 | .77 | .76 | .773 |

Table 5.3: Precision (P), Recall (R), $F_1$ score and accuracy scores of our system on the quote attribution task. The accuracy scores reported by Muzny et al. (2017b) on the complete corpus are shown for comparison. Best accuracy on each novel is shown in boldface. *Evaluated only on a subset of the complete dataset.

of characters for each artificial play.

## 5.2 Quote Attribution

Table 5.3 presents the accuracy scores achieved by our semi-supervised quote attribution model on the Stanford QuoteLi dataset. As explained in Section 4.4.2, we evaluate our model on only a subset of this dataset, taking into consideration only those characters with more than 15 attributed quotes.

Our method achieves an average accuracy almost exactly equal to that of the current state-of-the-art Muzny et al. (2017b), showing that the distinctiveness of character dialogues in these texts is a strong indicator of speaker identity. Further, the average classification probability of the correctly attributed quotes in each iteration remained consistently above 0.8, across all novels. However, despite the similar averages of the two methods, they vary dramatically in their accuracy on each of the three novels.

Our initial set of unassigned quotes is, for the most part, implicit and anaphoric quotes, which context-based systems perform poorly on, but which we were able to attribute with fair accuracy. Analysis of the results shows that certain characters are stylistically more distinct than others. For example, in Jane Austen's *Emma*, the central character, Emma Woodhouse, has a final $F_1$ score of 0.9. Conversely, some major characters fared quite badly in our system. Stylistic distinctiveness of characters is the consequence of a conscious choice and effort on the author's part, and our system is reliant on this. Nevertheless, we capture topical diversity between characters as well, allowing us to distinguish between them also by the social settings in which they appear and their preferred subjects of conversation.

The different performance of the two methods on each the three novels suggests that they each have a very different pattern of correct and erroneous classifications (although we did not have Munzy et al.'s raw results to compare with ours). This, in turn, suggests that the two

methods have complementary strengths and weaknesses, and that a hybrid method that draws on both would have a still higher accuracy.

# Chapter 6

# Conclusions and Future Work

In this thesis, we investigate a series of fundamental questions related to the phenomenon of literary dialogism and its tractability for computational analysis. The most fundamental is whether the voices of individual characters can be distinguished at all in literary texts. Our primary method of measuring this distinguishability of character voices is classification. Our results demonstrate that certain characters can be classified with very high accuracy, and that authors themselves lie on a spectrum regarding their ability to create these distinctive characters. Within our dataset, the relatively higher scores of well-known playwrights like Shaw and Wilde indicate that this ability is perhaps the property of more canonical writers. However, a bigger sample size would aid in drawing a more decisive conclusion, as well as inspecting the dependence of the ease of classification on other factors such as genre. This conclusion is also supported by our experiments with unsupervised topic-modeling and visualization models.

Our experiments with different feature sets also provide insights into how these characters are distinguishable from one another. SAGE, as an alternative to TF-IDF and naive Bayes measures of vocabulary usage, proves to be a very good indicator of which words are most distinctive for a particular character. The semantic and syntactic information captured by word2vec vectors forms the other key component of our analysis. While these dense vectors are not directly interpretable, our analysis with lexicon-based vectors illustrates some of the stylistic dimensions along which characters and authors differ. An interesting observation we make is that the artificial plays do not achieve a significantly higher score when compared to the original ones, despite the intuition that they must deal with more disparate topics.

The sparse prior enforced by the SAGE algorithm on its parameters allows it assign higher weights to the features most useful in distinguishing a particular character from the rest. With many characters in our dataset having very few datapoints (in the context of machine learning algorithms), this proves to be very beneficial. Whether these top-weighted features are purely 'stylistic' or 'topical' is an active area to explore, and made difficult by the fact it is hard to

draw a hard boundary between the two. This is reinforced by looking at the top $\eta$ coefficients for different characters in our experiments: we find a mix of function words ('very', 'oh'), punctuation, and other, more topical words ('mother', 'damned'). Our second component, the word embeddings, are perhaps picking up on syntactic and semantic similarities between words that correlate with aspects such as their parts of speech, or some stylistic dimension similar to the ones we obtained from the lexicons. This is hard to quantify, though our clustering experiments are indicative of it.

There is also a high amount of variation in the results: lexical and syntactic features perform better than the SAGE model for Cale Young Rice, while the scores for certain by Shaw and Wilde are also quite low. We have not yet investigated the specific reasons for this variation — whether it is the genre of the play, the distributions of different character types (for example, lords and ladies, butlers and maids), or indeed a difference in how character dialogue is crafted. Our classification approach means that we analyze each character quote separately from the rest. Certain quotes may simply not contain any character-specific idiosyncrasies of speech; examining individual classes for such phenomena, however, becomes a tedious exercises. The PCA results nevertheless do demonstrate that the stylistic spread of authors like Grundy and Rice is lower than those of Shaw and Wilde, with style here being defined by our choice of the lexicons in 2.3.4.

Our best performing classification model also allows us to formulate a novel semi-supervised model for quote attribution. We achieve performance comparable to, if not exceeding, that of current state-of-the-art systems. We are again limited by the scope of our corpus and the errors in existing quote attribution datasets. Extending our evaluation to a diverse set of authors and genres would provide a better indication of the generalizability of our semi-supervised approach. For now, we have demonstrated that a content-based approach is a viable and useful addition to existing quote attribution systems.

A more extensive corpus would also allow us to answer some of the other questions posed at the beginning of this thesis. Can characters be clustered into groups based on their stylistic similarity, and do these groups correlate with facets such as their social class or gender? Can they be clustered across genres, authors and time-periods? All of these are interesting avenues for exploration, and computationally tractable with reliable annotations.

# Bibliography

S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

R. Arun, R. Saradha, V. Suresh, M. Murty, and C. Madhavan. Stopwords and stylometry: a latent Dirichlet allocation approach. In *NIPS workshop on Applications for Topic Models*, 2009.

M. Bakhtin. *Problems of Dostoevsky's Poetics*, volume 8. U of Minnesota Press, 2013.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

R. A. Bosch and J. A. Smith. Separating hyperplanes and the authorship of the disputed federalist papers. *The American Mathematical Monthly*, 105(7):601–608, 1998.

J. Brooke and G. Hirst. A multi-dimensional Bayesian approach to lexical style. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 673–679, 2013.

J. Brooke, A. Hammond, and G. Hirst. GutenTag: an NLP-driven tool for digital humanities research in the Project Gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47, 2015.

J. Burrows. 'Delta': A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287, 2002.

J. Burrows and H. Craig. Authors and characters. *English Studies*, 93(3):292–309, 2012. doi: 10.1080/0013838X.2012.668786. URL `https://doi.org/10.1080/0013838X.2012.668786`.

S. Burrows, A. L. Uitdenbogerd, and A. Turpin. Temporally robust software features for authorship attribution. In *2009 33rd Annual IEEE International Computer Software and Applications Conference*, volume 1, pages 599–606. IEEE, 2009.

A. Celikyilmaz, D. Hakkani-Tur, H. He, G. Kondrak, and D. Barbosa. The actortopic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*, 2010.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

H. Craig and A. F. Kinney. *Shakespeare, computers, and the mystery of authorship*. Cambridge University Press, 2009.

N. M. Demir. A study in authorship attribution: The federalist papers. *Southeast Europe Journal of Soft Computing*, 4(1), 2015.

J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1041–1048. Omnipress, 2011.

D. K. Elson and K. R. McKeown. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1013–1019. AAAI Press, 2010.

S. Evert, T. Proisl, T. Vitt, C. Schöch, F. Jannidis, and S. Pielström. Towards a better understanding of Burrows's Delta in literary authorship attribution. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 79–88, 2015.

J. R. Firth. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, 1957.

K. Glass and S. Bangay. A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA07)*, pages 1–6, 2007.

J. Goldstein-Stewart, R. Winder, and R. E. Sabin. Person identification from text and speech genre samples. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344. Association for Computational Linguistics, 2009.

H. He, D. Barbosa, and G. Kondrak. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1312–1320, 2013.

D. I. Holmes. Authorship attribution. *Computers and the Humanities*, 28(2):87–106, 1994.

D. I. Holmes, L. J. Gordon, and C. Wilson. A widow and her soldier: Stylometry and the american civil war. *Literary and Linguistic Computing*, 16(4):403–420, 2001.

D. L. Hoover. The microanalysis of style variation. *Digital Scholarship in the Humanities*, 32: ii17–ii30, 04 2017. ISSN 2055-7671. doi: 10.1093/llc/fqx022. URL `https://dx.doi.org/10.1093/llc/fqx022`.

B. Kjell. Authorship attribution of text samples using neural networks and Bayesian classifiers. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1660–1664. IEEE, 1994.

M. Koppel, J. Schler, and S. Argamon. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, 2009.

I. Lancashire and G. Hirst. Vocabulary changes in Agatha Christies mysteries as an indication of dementia: a case study. In *19th Annual Rotman Research Institute Conference, Cognitive Aging: Research and Practice*, pages 8–10, 2009.

P. M. McCarthy, G. A. Lewis, D. F. Dufty, and D. S. McNamara. Analyzing Writing Styles with Coh-Metrix. In *FLAIRS Conference*, pages 764–769, 2006.

T. C. Mendenhall. The characteristic curves of composition. *Science*, 9(214):237–249, 1887.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013a.

T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013b.

G. A. Miller. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41, 1995.

S. M. Mohammad. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan, 2018a.

S. M. Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, 2018b.

F. Mosteller and D. L. Wallace. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309, 1963.

G. Muzny, M. Algee-Hewitt, and D. Jurafsky. Dialogism in the novel: A computational model of the dialogic nature of narration and quotations. *Digital Scholarship in the Humanities*, 32 (suppl_2):ii31–ii52, 2017a.

G. Muzny, M. Fang, A. Chang, and D. Jurafsky. A two-stage sieve approach for quote attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 460–470, 2017b.

T. O'Keefe, S. Pareti, J. R. Curran, I. Koprinska, and M. Honnibal. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 790–799. Association for Computational Linguistics, 2012.

J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic Inquiry and Word Count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

U. Sapkota, S. Bethard, M. Montes, and T. Solorio. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 93–102, 2015.

L. Sarmento and S. Nunes. Automatic extraction of quotes and topics from news feeds. In *DSIE'09 — 4th Doctoral Symposium on Informatics Engineering*, pages 1–12, Porto, 2009.

R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, 2013.

Y. Seroussi, I. Zukerman, and F. Bohnert. Authorship attribution with latent Dirichlet allocation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 181–189. Association for Computational Linguistics, 2011.

Y. Seroussi, F. Bohnert, and I. Zukerman. Authorship attribution with author-aware topic models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 264–269. Association for Computational Linguistics, 2012.

D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 440–450, 2018.

E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35(2):193–214, 2001.

B. Stein, M. Koppel, and E. Stamatatos. Plagiarism analysis, authorship identification, and near-duplicate detection (pan'07). In *SIGIR Forum*, volume 41, pages 68–71, 2007.

L. Tanguy, F. Sajous, B. Calderone, and N. Hathout. Authorship Attribution: Using Rich Linguistic Features when Training Data is Scarce. In *PAN Lab at CLEF*, 2012.

M. Tearle, K. Taylor, and H. Demuth. An algorithm for automated authorship attribution using neural networks. *Literary and Linguistic Computing*, 23(4):425–442, 2008.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA, 1995.