Master's thesis

Unsupervised Speaker Identification of Quotes in Literary Text

（文学作品における発話の教師なし話者同定）

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

48-176420 Satoshi Tohda

Supervisor
Associate Professor Naoki Yoshinaga

January 31, 2019

Abstract

Understanding literary texts that contain multiple characters interacting with each other requires the identifying of speakers and their quotes. There have been many studies using this information to build and visualize a network of interactions between characters, which is useful in literary analysis. Furthermore, a set of quotes from the same speaker can be utilized in language generation tasks. Existing studies that perform speaker identification of quotes assumes the use of a manually created list of characters, which is costly to produce. Therefore, we proposed a novel task in which the entities of speakers of quoted speech are identified in an unsupervised manner. We then proposed a method of clustering quotes based on building a vector representation of each quote, training a distance metric, and clustering using k-means. Through experiments, we were able to evaluate the superior performance of our system over a baseline system.

Keyword: natural language processing, speaker identification

Acknowledgements

I would first like to express my deep and profound gratitude to my supervisor, Associate Professor Naoki Yoshinaga, who has been instrumental to my life as a researcher. I still remember the time I first walked into the laboratory in the senior year of my undergraduate degree, a fish out of water wondering if he could make it in this field for which all he had was an interest in. But Dr. Yoshinaga was there in the lab along with his students, inviting me to the wonderful world of natural language processing, and willing to be my mentor through my graduate school journey. Thanks to him, this became a journey I will never forget. He has provided me with his insight and wisdom, without which my research endeavours would have ended in vain. I am very thankful for the countless hours that he has spent with me discussing my research, and been patient with me every time I did not understand something, or could not articulate a point clearly. Whenever I would attend a conference with him, he would go out of his way in his free time to take me and my colleagues to many interesting places. Research is as challenging as much as it is rewarding, and that is a lesson I am proud to have been taught by Dr. Yoshinaga.

I would like to thank Professor Masaru Kitsuregawa and Professor Masashi Toyoda, for giving me valuable feedback on my research, encouraging me at every turn, and providing me with the wonderful environment in which I was able to conduct my experiments with ease. Even though their fields of expertise were not natural language processing, they took the time to understand my research and provide crucial advice. Thank you for supporting me throughout my time at the laboratory.

I would like to thank the rest of the students and staff at the Kitsuregawa, Toyoda, Nemoto, Yoshinaga Laboratory for their advice and for making me feel at home. I would especially like to thank Mr. Shonosuke Ishiwatari and Mr. Satoshi Akasaki for lending me a hand, especially during my research on machine translation, and for looking out for me and the state of my research. I very much enjoyed discussing various topics with every one of you.

For my colleagues, I would like to thank them for putting up with me and my quirks. I really enjoyed working with everyone, and thanks to them, I was able to get acquainted with this university in a short amount of time. I don't think I would have made it without you all.

Special thanks to my friends and family, who have helped me throughout my life. I would like to thank my parents in particular, for being accepting of my choices and supporting me thorough my journey in college.

January 31, 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

When we encounter texts with multiple characters such as news articles and novels, we further our understanding of the social structures that underlie the text through quoted speech. Specifically, by understanding the content of quoted speech and its speakers and listeners, as well as its speech activities and where it was conducted, we are able to understand interpersonal relationships as well as their social standing and the flow of the plot.

In literary works in particular, a large number of characters have complex webs of relations with each other, and so it is important to obtain data that shows the relationship between listeners and speakers of speech. This data can be used to build conversational networks in which characters and their communities are visualized and analyzed [1, 4–6], which is an active task in literary analysis. An example of a conversation network is shown in Figure 1.1. Furthermore, this data is well suited for use in other NLP tasks, such as generation of fluent responses in dialogue systems based on persona [7]. Therefore, by solving the task of automatically identifying

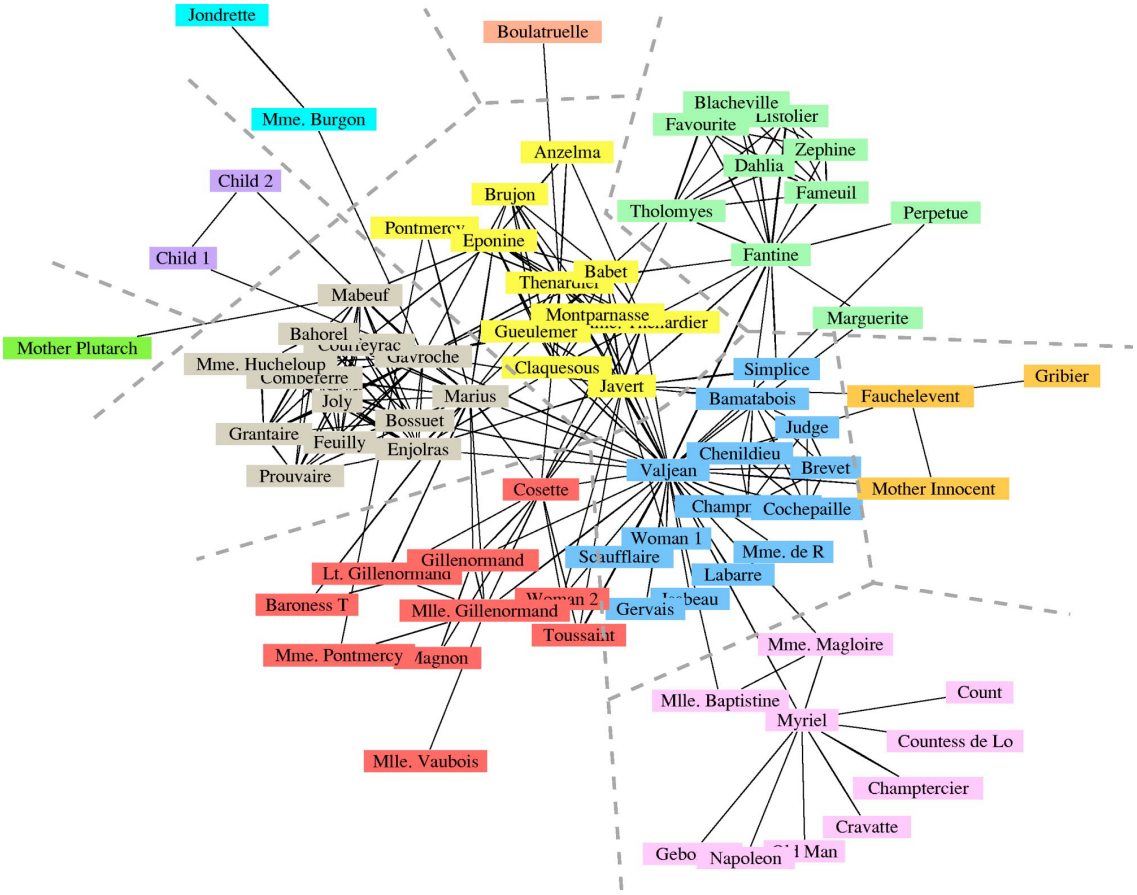

FIGURE 1.1: Network of interactions between major characters in the novel *Les Misérables* by Victor Hugo, as shown in Newman et al. [1] (Fig. 12). The nodes represent characters, the edges represent interactions between the characters, and the different colors represent different communities split by the proposed algorithm. The authors point out that the protagonist Valjean and his nemisis Javert are central to both the novel's plot and this graph.

speakers of each quoted speech in a work of literature, we can analyze a vast number of literary works or generate a large dataset for tasks such as dialogue generation.

In previous studies, the speaker identification task has been tackled under two different task settings. The first is a setting that associates each quote with a speaker entity, such as names. The other is a setting that associates each quote with a speaker mention, such as pronouns, in text near the quote. This can be seen as a subtask of the former.

While the first setting, in which quotes are linked to speaker entities, can be considered the most complete formulation of this task, studies that solved this used a predefined character list containing a list of all of the speakers in the work, as well as their genders, aliases, and a short description. Since this is a list produced from manual annotation of the dataset, using these systems in regards to unannotated works will necessitate a costly preprocessing step.

Studies in the second setting show improvements in linking quotes to character mentions. However, this is only a subset of the more general task of linking the character entities of speakers to quotes, since the quotes cannot be traced to characters, and vice versa.

1.2 Research Overview

We propose a new task in which speakers of quoted speech in literary texts are identified in an unsupervised manner, in the sense of not using preexisting information regarding the text, including but not limited to the aforementioned character list. This setting makes it difficult to apply the rule-based methods that 'narrow down' the speaker character from a preexisting list of speakers, and to identify which character entities are congruent, or aliases of, other entities.

By solving this task, we are able to extract quotes and their speakers from a large amount of data with minimal cost. This is very beneficial when we want a large dataset with tens of thousands of quotes attributed to a specific type of persona, as we may building a dialogue system, or when we want to use the result to analyze trends in character interactions in various novels over a period of time.

Our method of solving this task involves the linking of quotes to speaker mentions in text surrounding the quote, and the use of a clustering algorithm to cluster quotes based on speakers using the text within the quote as well as select contextual information. The features that are used in the clustering stage are combined to

form a vector that represents each data point, and a distance metric matrix that shrinks the distance between those of the same speakers and expands those between dissimilar speakers is calculated from quotes in which the extracted mentions are of entities. This metric is then used to calculate clusters of similar quotes, and the results are compared against the clustering of quotes by the same speaker in the annotated data.

During the experiments, we used the QuoteLi3 dataset that includes portions of three 19th-century English novels as annotated text [3]. We performed the following experiments to test the validity and performance of our method.

Experiment 1: we tested the performance of the quote-mention extraction stage of our system. Results demonstrated acceptable performance when compared to previous studies, especially in light of the inability to use the character list.

Experiment 2: we tested the performance of our proposed system. Results indicate better performance than the baseline system, in which only one feature, the mean of the word embedding in each sentence of the quote, was used.

Experiment 3~5: we tested the performance of our proposed system under various settings.

1.3 Structure of this paper

The remainder of this paper is organized thus.

In § 2, we introduces related studies, including previous work done on quote speaker identification in other domains and under different task settings.

In § 3, we define our task of unsupervised speaker identification of quotes in literary text.

In § 4, we present our proposed system in detail.

In § 5, we describe our experiments, as well as their results.

In § 6, we conclude the findings of our study.

In § 7, we discuss the potential direction of future research with regards to this task.

Chapter 2

Related Work

To the best of our knowledge, there have been no attempts to identify the speaker of quotes in a literary text by its entity, without the aid of a preprocessed list of characters and information regarding them. In this chapter, we will discuss studies that identify in some way a speaker of a quotes in various domains.

2.1 Speaker Mention Identification

Early studies that dealt with the task of identifying speakers of quotes [8, 9] used a simple rule-based method using speech verbs, such as 'said', to link speakers with mentions of quotes. While this approach fares well for domains such as news articles that contain more quotes that appear consecutively with speaker mentions and speech verbs, it encounters difficulty in capturing rare patterns, or cases in which the quote is in a paragraph of its own, for example, in conversation. In addition, these studies simply tackled the issue of identifying a mention of a speaker for every quote, and did not explicitly consider the issue of linking to speaker entities.

In the study conducted by Elson and McKeown (2010) [10], they applied supervised machine learning to solve the task of speaker identification in the literary domain.

First, all named entities and nominals were extracted from text before each quote, which became the list of candidate speakers. Next, the quote was classified into one of a few syntactic categories, which determined for each quote whether to attribute a speaker by a rule-based method or to undergo further processing. For quotes that did not have their speakers attributed, each candidate speaker was classified by a binary classifier into 'speaker' or 'not speaker' labels, using features of preceding quotes. Finally, the results were combined to output an identified speaker. Their overall accuracy of 83% is misleading, however, as gold speaker information for preceding quotes was used during test time. Furthermore, while their list of candidate speakers omits pronouns, it still does not identify which mentions are coreferent with which entities, and so the non-trivial task of identifying which quote was said by whom remains.

O'Keefe et al. (2012) [11] leveraged text sequence features without using the gold speaker information by decoding the sequence of a set of speaker attributions in a document. They compared their approach against a baseline, which was a simple modification of a speech verb focused rule-based system that added a final step of returning an entity mention nearest the quote. While their system achieved 92.4% and 84.1% accuracy on two different news article datasets, they failed to beat their baseline (53.5% accuracy) on the literary dataset featuring the 19th-century English novel Emma. This established the difficulty of the quote-speaker identification task on unstructured literary text.

2.2 Speaker Entity Identification

Studies in the News Domain

For the news domain, Almeida et al. (2014) [12] showed a system that jointly solved the task of coreference resolution on all mentions in the text and quote-speaker identification, measuring the performance of each task independently.

More recently, Pavllo et al. (2018) [13] exploited the existence of redundant news articles, which is unique to the news domain, to achieve state-of-the art results in the speaker entity identification task. In this domain, the same quotations tend to be seen in different articles, each in different contexts. Their method extracts a number of seed quotation-speaker pairs using a number of rule-based patterns, which are used to obtain new patterns in which the quotation-speaker pairs appear, and this pattern is used to identify a new quotation-speaker pair. This unsupervised bootstrapping method resulted in 90% precision and 40% recall.

While the obtained results are promising, application of this method to the literary domain is not possible, not only because the method relies on redundant news articles, but also because their method of coreference (anaphora) resolution relies on there being at least one quotation-speaker pair that maps the quotation to a speaker entity, that is, the speaker mention is a person name.

Studies in the Literary Domain

For the literary domain, He et al. (2013) [2] modified the task to link speaker entities with quotes. With a list of characters as additional input, they built a number of features including output from an actor-topic model [14] to predict the speaker using a supervised learner. They achieved an accuracy of 74.8% on Emma and 82.5% on Pride & Prejudice, a novel by the same author. However, their use of a comprehensive character list, as shown in Figure 2.1 necessitates a manual preprocessing step to annotate all aliases, gender, and descriptions for all characters, a step that is costly and requires deep human understanding of the source material.

Muzny et al. (2017) [3] proposed a system that is the current SOTA for the task of identifying speaker entities for quotes in literary text, allowing for the use of manually preprocessed character lists. Their system has two stages, the first linking quotes to speaker mentions, and the second linking mentions to speaker entities. Both of these stages are composed of consecutive rule-based sieves that attribute

```
<characters>
<character aliases="Emma Woodhouse;Emma;Miss Woodhouse" description="The protagonist of the novel. A beautiful, high-spirited, intelligent, and 'slightly' spoiled young woman. Her
 mother died when she was young. She has been mistress of the house since her older sister got married." gender="female" id="0" name="Emma_Woodhouse"/>
<character aliases="Isabella Woodhouse;Isabella;Mrs. John Knightley;Mrs. Knightley" description="Emma's older sister, who lives in London with her husband, Mr. John Knightley, and
 their five children." gender="female" id="1" name="Isabella_Woodhouse"/>
<character aliases="Mr. John Knightley;John Knightley;Mr. Knightley;John" description="Emma's brother-in-law, and Mr. George Knightley's brother." gender="male" id="2"
 name="Mr_John_Knightley"/>
<character aliases="Miss Taylor;Mrs. Weston;Anne Taylor" description="Formerly Miss Taylor, Emma's beloved governess and companion. Known for her kind temperament and her devotion to
 Emma, Mrs. Weston lives at Randalls with her husband, Frank Churchill's father." gender="female" id="3" name="Miss_Taylor"/>
<character aliases="Mr. George Knightley;George" description="Emma's brother-in-law and the Woodhouses' trusted friend and advisor. Knightley is a respected landowner in his late
 thirties. He lives at Donwell Abbey." gender="male" id="4" name="Mr_George_Knightley"/>
<character aliases="Mr. Elton;Elton" description="A good-looking, initially well-mannered, and ambitious young vicar." gender="male" id="5" name="Mr_Elton"/>
<character aliases="Mr. Frank Churchill;Frank C. Weston Churchill;F. C. Weston Churchill;Weston Churchill;Frank Churchill;Frank" description="Mr. Weston's son and Mrs. Weston's
 stepson. Frank Churchill lives at Enscombe with his aunt and uncle, Mr. and Mrs. Churchill. He is considered a potential suitor for Emma." gender="male" id="6"
 name="Mr_Frank_Churchill"/>
<character aliases="Mr. Weston;Weston" description="The widower and proprietor of Randalls, who has just married Miss Taylor when the novel begins. Mr. Weston has a son, Frank, from
 his first marriage to Miss Churchill." gender="male" id="7" name="Mr_Weston"/>
<character aliases="Miss Harriet Smith;Harriet Smith;Harriet;Mrs. Robert Martin;Mrs. Martin" description="A pretty but unremarkable seventeen-year-old woman of uncertain parentage,
 who lives at the local boarding school." gender="female" id="8" name="Miss_Harriet_Smith"/>
<character aliases="Mr. Robert Martin;Mr. Martin;Robert Martin" description="A twenty-four-year-old farmer. Mr. Martin is industrious and good-hearted, though he lacks the refinements
  of a gentleman. He lives at Abbey-Mill Farm, a property owned by Knightley, with his mother and sisters." gender="male" id="9" name="Mr_Robert_Martin"/>
<character aliases="Miss Jane Fairfax;Jane Fairfax;Miss Fairfax;Jane" description="Miss Bates's niece, whose arrival in Highbury irritates Emma.  Later married to Frank Churchill."
 gender="female" id="10" name="Miss_Jane_Fairfax"/>
<character aliases="Lieutenant Fairfax;Lieut. Fairfax" gender="male" id="11" name="Lieutenant_Fairfax"/>
<character aliases="Miss Jane Bates;Jane Bates;Miss Bates;Bates" description="Friend of Mr. Woodhouse and aunt of Jane Fairfax, Miss Bates is a middle-aged spinster without beauty or
 cleverness but with universal goodwill and a gentle temperament." gender="female" id="12" name="Miss_Jane_Bates"/>
<character aliases="Miss Hawkins;Mrs. Elton" description="Formerly Augusta Hawkins, Mrs. Elton hails from Bristol and meets Mr. Elton in Bath." gender="female" id="13"
 name="Miss_Hawkins"/>
<character aliases="Mrs. Woodhouse" gender="female" id="14" name="Mrs_Woodhouse"/>
<character aliases="Mr. Henry Woodhouse;Mr. Woodhouse" description="Emma's father and the patriarch of Hartfield, the Woodhouse estate." gender="male" id="15"
 name="Mr_Henry_Woodhouse"/>
<character aliases="James" gender="male" id="16" name="James"/>
<character aliases="Hannah" gender="female" id="17" name="Hannah"/>
<character aliases="Miss Churchill" gender="female" id="18" name="Miss_Churchill"/>
<character aliases="Mrs. Churchill" description="Mr. Weston's ailing former sister-in-law and Frank Churchill's aunt and guardian." gender="female" id="19" name="Mrs_Churchill"/>
<character aliases="Mr. Churchill" description="Brother to the first Mrs Weston." gender="male" id="20" name="Mr_Churchill"/>
<character aliases="Mr. Perry;Perry" description="An apothecary and associate of Emma's father. Mr. Perry is highly esteemed by Mr. Woodhouse for his medical advice even though he is
```

FIGURE 2.1: Excerpt from a character list of the novel Emma, utilized in previous studies in the literary domain [2, 3]

speaker mentions or entities to each unattributed quote. The addition of a supervised component, a binary classifier that predicts the most confident of all candidate mentions in the paragraphs surrounding the quotes, results in a high system performance, with an average F-score of 87.5% across three novels, or an accuracy of 76.1% on Emma and 85.2% on Pride & Prejudice using the same test settings as He et al. [2]

Yeung et al. (2017) [15] rejected the task setting of using a character list and proposed a supervised approach with conditional random fields. They tagged all tokens near the quote as either 'speaker,' 'listener,' or 'neither,' and identified dialogue chains, performing better than their re-implementation of O'Keefe et al. (2012) [11]. However, they were only able to link quotes to gold coreference chains of speaker mentions, relying on this to link to speaker entities. We consider this work to not have solved the issue of linking quotes to speaker entities unsupervisedly, without manual preprocessing or other data external to the literary text.

Chapter 3

Task: Unsupervised Speaker Identification of Quotes in Literary Text

In this chapter, we will define the novel task of unsupervised speaker identification of quotes in literary text, and discuss the dataset that can be used in this task.

3.1 Task Description

In this task, we attempt to identify the speakers of quotes in a literary text without the use of information that can be used in a supervisory nature. This information includes, but is not limited to, information regarding the speaker characters such as aliases and gender built from a manually annotated character list, and quotes labeled with speaker information.

In terms of input and output:

Input The input is the text of a given work of literature or news text that contains multiple quotes with a speaker for each quote.

Output Our goal is to link each quote with the entity of the speaker. Thus, the output is a set of clusters of quotes, with each cluster containing quotes from one character.

To illustrate what this task entails, below is an example of a text in the literary domain.

> My name is John. I went to the store to buy some clothes, and there I saw my daughter, Emily.
> I said, "Hello, Emily. What brings you here?"
> She replied, "Father, I am just browsing."
> "I didn't realize you fancied wearing men's clothing."
> "Oh, it's a present for a friend," said Emily, smiling.

Here, we have the entire text as input. If we label the quotes as follows:

> Q1 = "Hello, Emily. What brings you here?"
> Q2 = "Father, I am just browsing."
> Q3 = "I didn't realize you fancied wearing men's clothing."
> Q4 = "Oh, it's a present for a friend,"

The task is to cluster each quote Q into a set of clusters K with clusters $K_1...K_i$, having one speaker for each cluster. For example, an output of the system might look like this:

$$K = \{Q1, Q3\}\{Q2, Q4\} \tag{3.1}$$

It should be noted that the extraction of quotes from text is trivial, achieving over 99% accuracy using a simple pattern matching algorithm by O'Keefe et al. [11]

Hence, the quotes and their respective spans within the text can be treated as having been given.

A significant difference between this task and the tasks put forward in other studies mentioned in § 2 is the unsupervised nature of this task. The implication of this difference is that existing methods that narrow down the appropriate speaker entity from a list of speakers are not usable or must be modified, and in some cases, there is an additional layer of complexity involved, by having to identify coreferent character aliases for proper labeling.

Gold Standard

Since we use the QuoteLi3 dataset provided by Muzny et al. [3], the gold standard adheres to their annotation guidelines as outlined in the same paper. The annotation guidelines are as follows.

For each quote in the text, annotaters are asked to identify its speaker from a list of characters, and to identify the mention that was the most helpful when doing so. When choosing mentions, those that are closer to the quote or closer to speech verbs are chosen over those that were not. In doing so, quotes are linked with both mentions of speakers and speaker entities.

An example of quote-mention-entity links annotated in this manner for the text above would be:

Q1: Mention = I, Entity = John
Q2: Mention = She, Entity = Emily
Q3: Mention = Father, Entity = John
Q4: Mention = Emily, Entity = Emily

The resulting quote-entity links are used to make a set of clusters of quotes, in which all quotes linked to one entity are grouped into a single cluster. This is treated as the gold data against which system outputs are evaluated.

	Emma	Pride & Prejudice	The Steppe
Paragraphs	817	1863	790
Quotes	734	1747	622

TABLE 3.1: Number of paragraphs and quotes in each novel portion of the QuoteLi3 dataset [3]

Unlike existing tasks, the output for this task will be evaluated using coreference resolution evaluation metrics to capture the clustering performance. Hence, entity labels will not be used during evaluation.

3.2 Dataset

In this study, the QuoteLi3 dataset [3] is used. This is a dataset of three novels written in the 19th century: Pride & Prejudice, Emma, and The Steppe. Of the three novels, Pride & Prejudice and Emma were written by Jane Austen, and The Steppe by Anton Chekhov. Information regarding the dataset is shown in Table 3.1.

All annotation is done according to the guidelines mentioned above, with each quote, mention, and entity written in an XML format. Figure 3.1 shows how each quote can be traced to a speaker mention using the connection id, and how they are in turn labeled with speaker entities.

A comprehensive character list is also provided with the data, but it is not used in this task setting.

```
obliged to part with Miss Taylor too, and from his habits of gentle selfishness, and of being never able to suppose that other
people could feel differently from himself, he was very much disposed to think Miss Taylor had done as sad a thing for herself
as for them, and would have been a great deal happier if she had spent all the rest of her life at Hartfield.  <mention
connection="s261,s263,s265,s267" entity="person20" entitytype="PERSON" gender="female" id="s64" mentiontype="name" oid="named30"
 speaker="Emma_Woodhouse">Emma</mention> smiled and chatted as cheerfully as she could, to keep him from such thoughts; but
when tea came, it was impossible for him not to say exactly as <mention connection="s260" id="s310"
speaker="Mr_Henry_Woodhouse">he</mention> had said at dinner,

<quote connection="s310" id="s260" oid="0" speaker="Mr_Henry_Woodhouse">"Poor Miss Taylor!--I wish she were here again.  What a
 pity it is that Mr. Weston ever thought of her!"</quote>

<quote connection="s64" id="s261" mention="named30" oid="1" speaker="Emma_Woodhouse">"I cannot agree with you, <mention
 connection="s262,s264" id="s311" speaker="Mr_Henry_Woodhouse">papa</mention>; you know I cannot.  Mr. Weston is such a
 good-humoured, pleasant, excellent man, that he thoroughly deserves a good wife;-- and you would not have had Miss Taylor live
 with us for ever, and bear all my odd humours, when she might have a house of her own?"</quote>

<quote connection="s311" id="s262" mention="organism32" oid="2" speaker="Mr_Henry_Woodhouse">"A house of her own!-- But where is
 the advantage of a house of her own? This is three times as large.-- And you have never any odd humours, my dear."</quote>

<quote connection="s64" id="s263" mention="named30" oid="3" speaker="Emma_Woodhouse">"How often we shall be going to see them,
 and they coming to see us!-- We shall be always meeting! We must begin; we must go and pay wedding visit very soon."</quote>

<quote connection="s311" id="s264" mention="organism32" oid="4" speaker="Mr_Henry_Woodhouse">"My dear, how am I to get so far?
 Randalls is such a distance.  I could not walk half so far."</quote>

<quote connection="s64" id="s265" mention="named30" oid="5" speaker="Emma_Woodhouse">"No, <mention connection="s266" id="s312"
 speaker="Mr_Henry_Woodhouse">papa</mention>, nobody thought of your walking.  We must go in the carriage, to be sure."</quote>

<quote connection="s312" id="s266" mention="organism32" oid="6" speaker="Mr_Henry_Woodhouse">"The carriage! But James will not
 like to put the horses to for such a little way;-- and where are the poor horses to be while we are paying our visit?"</quote>

<quote connection="s64" id="s267" mention="named30" oid="7" speaker="Emma_Woodhouse">"They are to be put into Mr. Weston's
 stable, papa.  You know we have settled all that already.  We talked it all over with Mr. Weston last night.  And as for James,
 you may be very sure he will always like going to Randalls, because of his daughter's being housemaid there.  I only doubt
 whether he will ever take us anywhere else.  That was your doing, <mention connection="s268" id="s313"
 speaker="Mr_Henry_Woodhouse">papa</mention>.  You got Hannah that good place.  Nobody thought of Hannah till you mentioned
 her-- James is so obliged to you!"</quote>

<quote connection="s313" id="s268" mention="organism32" oid="8" speaker="Mr_Henry_Woodhouse">"I am very glad I did think of her.
  It was very lucky, for I would not have had poor James think himself slighted upon any account; and I am sure she will make a
 very good servant: she is a civil, pretty-spoken girl; I have a great opinion of her.  Whenever I see her, she always curtseys
 and asks me how I do, in a very pretty manner; and when you have had her here to do needlework, I observe she always turns the
 lock of the door the right way and never bangs it.  I am sure she will be an excellent servant; and it will be a great comfort
 to poor Miss Taylor to have somebody about her that she is used to see.  Whenever James goes over to see his daughter, you
 know, she will be hearing of us.  He will be able to tell her how we all are."</quote>
```

FIGURE 3.1: Portion of the annotated XML data from the Emma part of the QuoteLi3 dataset [3].

Chapter 4

Proposed Method

In this chapter, we will explain the specifics of our proposed method.

Our proposed method is a two-stage system, where mentions are linked to each quote in the first step, and a clustering of mention information is done in the second step.

The first step is centered on a reimplementation of the sieve-based system by Muzny et al. [3], however, it has been modified to comply with the requirements of this task. The results of this mention mapping stage will be used to extract a set of quotes with mentions that already contain character names, so that the learning of the distance metric in the second step can be done in a semi-supervised manner.

The second step involves the extraction of features from the first step, adding weights through training of a distance metric on entity-labeled quotes, and clustering the quotes with k-means clustering.

An overview of the system is presented in Figure 4.1.

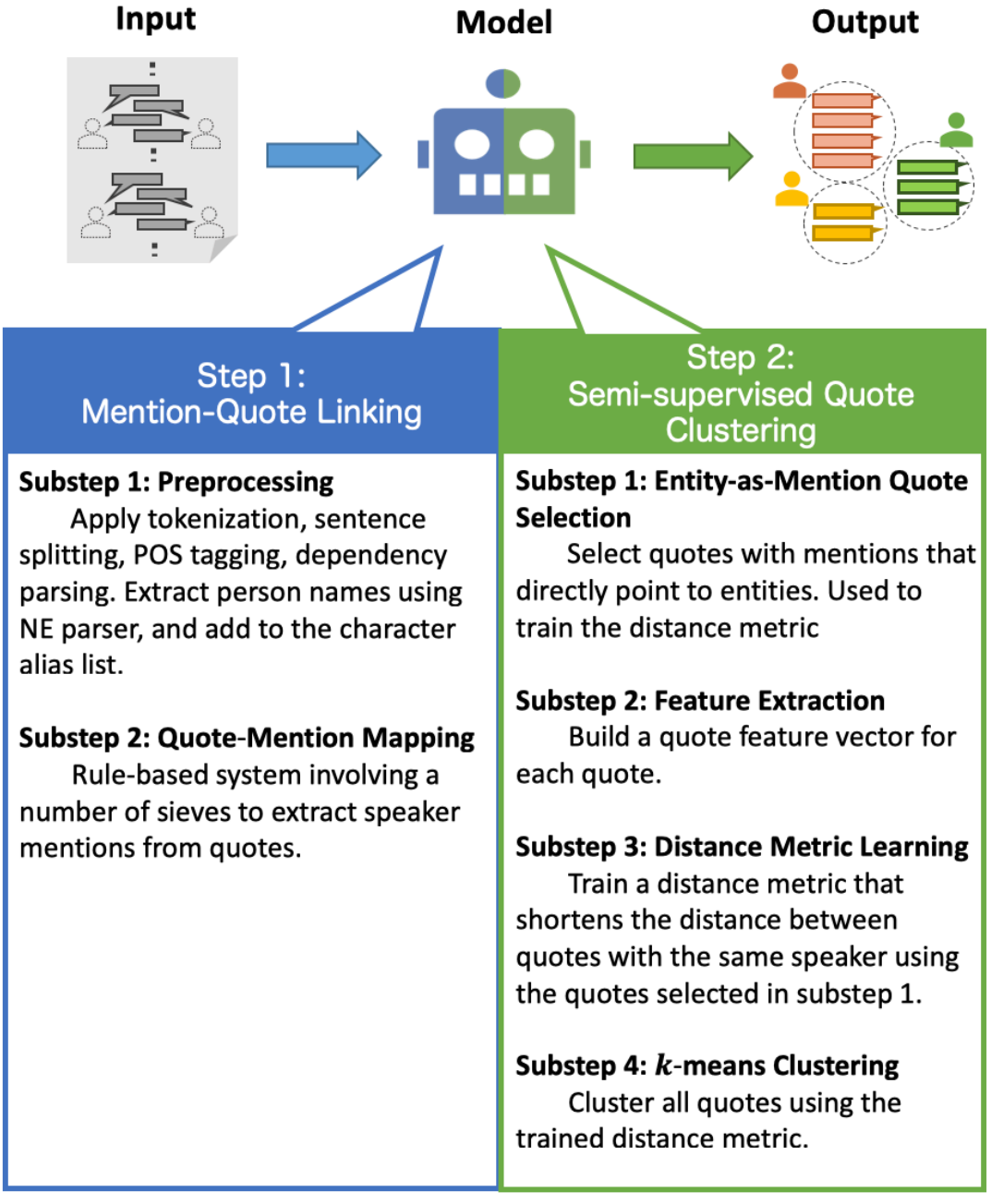

FIGURE 4.1: Overview of our system.

4.1 Preprocessing

For the preprocessing test, first, we extract the quotes and separate them from the rest of the text. Next, we use the integrated tools of the Stanford CoreNLP [16] toolkit to perform sentence splitting and tokenization, as well as POS tagging for each token and dependency parsing. Finally, we search for person names in the text using named entity recognition, and add these names to a character list.

4.2 Rule-based Quote-Mention Mapping

In the first step of our system, we implement the same rule-based sieves that were tested by Muzny et al. [3] in their method. For each unlabeled quote, a set of rules attempt to extract a mention from the surrounding text, and map this mention to the quote. If there is no mention that can be mapped to the quote, the quote stays unlabeled, and it is passed to the next sieve. Character lists used in this stage are obtained in the preprocessing stage, and do not reference gold speaker information.

The sieves are as follows.

Trigram Matching

This sieve uses trigrams of Quotes, Mentions, and Verbs, such that quotes are either at the front or the back (Quote-Mention-Verb, Quote-Verb-Mention, Mention-Verb-Quote, or Verb-Mention-Quote). This is similar to patterns used in Elson et al. [10] Mentions include characters in the character list, pronouns, or familial relation nouns.

Dependency Parsing

This sieve uses the dependency parser to identify the subjects of speech verbs that relate to the quote. In order to do so, we use the dependency parser to identify

Common speech verbs	say cry reply add think observe call answer
Family relation nouns	ancestor aunt bride bridegroom brother brother-in-law child children dad daddy daughter daughter-in-law father father-in-law fiancee grampa gramps grandchild grandchildren granddaughter grandfather grandma grandmother grandpa grandparent grandson granny great-granddaughter great-grandfather great-grandmother great-grandparent great-grandson great-aunt great-uncle groom half-brother half-sister heir heiress husband ma mama mom mommy mother mother-in-law nana nephew niece pa papa parent pop second cousin sister sister-in-law son son-in-law step-brother stepchild stepchildren stepdad stepdaughter stepfather stepmom stepmother stepsister stepson uncle wife

TABLE 4.1: Common speech verbs and relation nouns.

the dependent `nsubj` nodes of all common speech verbs. If the `nsubj` node is a character name, a pronoun, or a family relation noun, this is mapped to the quote as the speaker mention.

The list of common speech verbs and family relation nouns is provided in Table 4.1.

Single Mention Detection

This sieve identifies the paragraph that which contains the quote, searching for other mentions. If only a single mention exists in the non-quote text, the mention is mapped to the quote. Mentions include characters in the character list, pronouns, or familial relation nouns.

Vocative Detection

This sieve looks for certain vocative patterns that identify the speaker of the next quote. The patterns are listed in Table 4.2.

Pattern	Example
between , and !	, Hanako!
between , and ?	, Hanako?
between , and .	, Hanako.
between , and ;	, Hanako;
between , and ,	, Hanako,
between “ and ,	“Hanako,
between , and ”	, Hanako”
after the word “dear”	Dear Hanako

TABLE 4.2: Vocative patterns. 'Hanako' is the mention in all cases.

For every mention unlabeled quote, if the previous quote was in the same paragraph or the one preceding it, and contains a vocative pattern, the mention that is contained in that pattern is mapped to the unlabeled quote. The mentions can be characters or familial relation nouns, but cannot be pronouns.

Paragraph Final Mention Linking

This sieve maps the final mention in the sentence preceding the quote, if the quote occurs at the end of a paragraph. Mentions include characters in the character list, pronouns, or familial relation nouns.

Conversation Pattern

This sieve identifies sections of text where characters are in conversation with each other. Quotes are considered to be conversations if there are no additional text other than the quote. Within conversations, the speaker of a quote in paragraph n is considered to be the same as that of the quote in paragraph $n + 2$.

For every mention labeled quote, we look at the quotes in paragraphs $n + 1$ and $n + 2$, with n being the paragraph number of the labeled quote. If they are in a paragraph on their own, we consider these quotes to be in conversation. If the quote at paragraph $n+2$ is not labeled with a mention, we map the speaker mention of the

quote at paragraph n, to the quote at paragraph $n+2$. Mentions include characters in the character list, pronouns, or familial relation nouns.

Loose Conversation Pattern

This sieve is a looser form of the previous sieve that does not look if quotes are in a conversation, introduced to improve recall. For every mention labeled quote at paragraph n, if the quote at paragraph $n+2$ is not labeled with a mention, we map the speaker mention of the quote at paragraph n to the quote at paragraph $n+2$.

4.3 Semi-supervised Quote Clustering Based on Speakers

In the second step of our system, we use the mention information gained in the first step to cluster quotes. This step is comprised of three substeps, in which we perform the selection of quotes to be used in training of the distance metric, a building of a quote vector using features from the text, the training of a distance metric that shortens distances between quotes with the same speakers, and finally, k-means clustering using the learned metric.

Substep 1: Entity-as-Mention Quote Selection

For this substep, we extract quotes linked to mentions that are already characters for use in the training of the distance metric. For the set of all quotes Q_A, we choose all quotes q_n which have mentions that are entities included in the rudimentary character (alias) list obtained in the previous stage. They are added to the set Q_{EM}.

Substep 2: Feature Extraction

For this substep, we build a quote feature vector $\boldsymbol{V}_n$ for each quote q_n.

Here, $\boldsymbol{V}_n$ is a concatenation of six different feature vectors that are extracted from the text and the information gained in the previous steps of the system.

$$\boldsymbol{V}_n = \left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right] \tag{4.1}$$

Explanations for each feature vectors are as follows.

$\boldsymbol{F}_n^{\text{emb}}$: Sentence Embedding We can assume that there is some similarity in terms of content of speech that is produced by the same person in the text. This content is assessed by taking the average of the word embeddings of the words in the quote.

$\boldsymbol{F}_n^{\text{wc}}$: Quote Length This feature attempts to disambiguate speakers with shorter spoken content from those with longer spoken content. The length of the quote in tokens is used for this calculation.

$\boldsymbol{F}_n^{\text{gen}}$: Gender F-score This feature attempts to capture the gender of the speaker. We use the gender F-score originally proposed in Heylighen and Dewaele (2002) [17] , which Nowson et al. (2005) [18] has shown to be effective in identifying genders of authors of text in various domains.

F-score is a measure of the implicitness or explicitness of speech. Explicit speech is considered to have a higher frequency of nouns, adjectives, prepositions, and articles, whereas implicit speech is considered to have a higher frequency of pronouns, verbs, adverbs, and interjections. It has been observed that a higher F-score, implying implicit speech, is preferred by females, while a lower F-score, implying explicit speech, is preferred by males.

This feature is calculated as follows:

$$\begin{aligned} \boldsymbol{F}_n^{\text{gen}} = (&\alpha_{\text{freq.}} + \beta_{\text{freq.}} + \gamma_{\text{freq.}} + \delta_{\text{freq.}} - \epsilon_{\text{freq.}} \\ &-\zeta_{\text{freq.}} - \eta_{\text{freq.}} - \theta_{\text{freq.}} + 100) \cdot 0.5 \end{aligned} \tag{4.2}$$

where $\alpha_{\text{freq.}}$ is the noun frequency, $\beta_{\text{freq.}}$ is the adjective frequency, $\gamma_{\text{freq.}}$ is the preposition frequency, $\delta_{\text{freq.}}$ is the article frequency, $\epsilon_{\text{freq.}}$ is the pronoun frequency, $\zeta_{\text{freq.}}$ is the verb frequency, $\eta_{\text{freq.}}$ is the adverb frequency, and $\theta_{\text{freq.}}$ is the interjection frequency.

$\boldsymbol{F}_n^{\text{near}}$: Bag-of-Entities in Nearby Sentences This feature captures the candidates for speakers in text near the quote. We design this feature as a Bag-of-Entities limited to entities in the character list, considering two sentences before and in front of the quote, in the same paragraph as the quote.

$\boldsymbol{F}_n^{\text{in}}$: Bag-of-Entities in the Quote This feature attempts to look for listeners within the quote, which can be used to rule out the person from the candidates. We design this feature as a Bag-of-Entities limited to entities in the character list, considering the quoted text as well as the mention.

$\boldsymbol{F}_n^{\text{men-emb}}$: Mention Embedding This feature captures the semantic similarity of mentions. When we have X as the pretrained word embedding and the mention as m_n, we take the average of the word embeddings of the tokens in the mention.

Substep 3: Distance Metric Learning

For this substep, we train a distance metric that is able to map the feature vector to a vector space in which quotes with the same speaker can be clustered together. Essentially, this distance metric will give weights to certain features that are effective at disambiguating speakers, such that the similarity between vectors of quotes with the same speakers will be high, and those between vectors of quotes with different speakers will be low.

We will take the quotes and their speaker entities that were extracted in Substep 1 as the training data, and use this to train a distance matrix $\boldsymbol{L}$ with Mahalanobis Metric Learning for Clustering (MMC) [19].

Substep 4: Clustering

For this substep, we cluster together quotes that have the same speaker using k-means clustering. We determine the number of clusters by the number of entities in the character list.

The input of the clustering algorithm is a vector $\boldsymbol{V_n'}$, such that for a Mahalanobis metric matrix $\boldsymbol{M}$:

$$\boldsymbol{V_n'} = \boldsymbol{M}^{1/2}\boldsymbol{V_n} \tag{4.3}$$

The output is a set of clusters of quotes.

Chapter 5

Experiments

In this chapter, we will explain the experiments we conducted to test the performance of our proposed system.

Since our system is comprised of two steps, with quotes being linked to speaker mentions in the first step, and this information being leveraged in the second step to cluster quotes, a poorly performing first step will have an adverse effect on the output of the second step. Therefore, we will evaluate the performance of the mention linking step in the first experiment.

For the second experiment, we will evaluate our system against a baseline, as well as showing the additive effect of the features used in the quote vectors on system performance.

For the third experiment, we will evaluate in detail the effective features for our system.

For the fourth experiment, we will evaluate the additive effect of the features used in our system, with ineffective features removed.

For the fifth experiment, we will evaluate the performance of the system when a diagonal distance metric is used.

For the sixth experiment, we will test the performance of this system for various number of clusters.

5.1 Experiment Settings

Unless noted otherwise, we will use the following settings for each experiment.

Word embeddings used in Step 2 of our system are GloVe [20] embeddings trained on the Wikipedia 2014 + Gigaword 5 dataset. The pretrained embeddings were obtained from https://nlp.stanford.edu/projects/glove, and the dimension is 50.

Distance metrics were calculated using the metric-learn library for Python. (http://metric-learn.github.io/metric-learn/index.html) Clustering was done using the scikit-learn library for Python.

5.2 Evaluation

Since the output of our task is a set of clusters with identical speakers, we can evaluate the clustering performance of the systems with cluster-based coreference resolution evaluation metrics. Namely, we utilize the MUC [21], B^3 [22], and $CEAF_e$ [23] metrics via the CoNLL system scorer v8.01, available on http://conll.cemantix.org/2012/software.html.

5.3 Experiment 1: Quote-Mention Mapping

For our first experiment, we evaluated the performance of the first step of our system against existing methods. The results are shown in Table 5.1.

System	Precision	Recall	F_1	Accuracy
Muzny et al.[a] [3]	84.6	68.3	75.6	-
Yeung et al.[bc] [15]	-	-	-	52.5
O'Keefe et al.[b] [11]	-	-	-	43.7
Proposed Method	71.5	56.7	62.5	56.7

a: Uses gold character list
b: Different annotation range on the same book
c: Supervised training

TABLE 5.1: Performance of quote-mention mapping step

The lower performance of our implementation compared to the SOTA method by Muzny et al. is to be expected, because our task setting does not allow for the use of gold character lists, which they have utilized when matching for mentions at every sieve in the system. An incorrect character list with omitted characters or their aliases will fail to label the omitted strings as a mention of a quote. Since the performance is otherwise close to Muzny et al.'s system by all metrics, and does not exhibit a particularly large drop in precision or recall, we believe that any errors introduced during the reimplementation of the Muzny et al. sieves are minimal.

When compared to other systems that do not use the gold character list, however, our system performs slightly better. The system shows a higher accuracy than the most recent supervised method by Yeung et al. [15], which requires at least 200 labeled quotes as training data to significantly outperform the baseline by O'Keefe et al. It should be noted however, as the portion of the dataset that was annotated in their experiments was differs from ours, a direct comparison is not completely appropriate.

To summarize, our method achieved SOTA performance in the quote-mention linking subtask by our task setting, beating the previous SOTA using a supervised method.

$\boldsymbol{V}_n$	MUC F1	B^3 F1	CEAF_e F1	Avg. F1
$\boldsymbol{F}_n^{\text{emb}}$	70.59	13.32	6.48	30.13
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	62.83	8.23	5.38	25.48
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}]$	65.52	9.08	5.92	26.84
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}]$	77.01	12.86	11.86	33.91
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}]$	**78.35**	15.60	11.91	35.29
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}]$	77.31	**16.60**	**13.62**	**35.84**

TABLE 5.2: Results of the holistic system performance evaluation. Avg. F1 is an average of the F1 score for MUC, B^3, and CEAF_e.

$\boldsymbol{V}_n$	MUC Precision	MUC Recall	MUC F1
$\boldsymbol{F}_n^{\text{emb}}$	72.54	68.75	70.59
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	64.57	61.19	62.83
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}]$	67.33	63.80	65.52
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}]$	79.14	75.00	77.01
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}]$	**80.52**	**76.30**	**78.35**
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}]$	79.44	75.29	77.31

TABLE 5.3: Results of the holistic system performance evaluation in MUC.

5.4 Experiment 2: Holistic System Performance

In this experiment, we performed an evaluation of our entire system against a single feature baseline, as well as evaluating whether the features have an additive effect on system performance. We prepared six variations on the quote vector $\boldsymbol{V}_n$, starting with $\boldsymbol{V}_n = \boldsymbol{F}_n^{emb}$. Features are added in the order of $\boldsymbol{F}_n^{\text{wc}}$, $\boldsymbol{F}_n^{\text{gen}}$, $\boldsymbol{F}_n^{\text{near}}$, $\boldsymbol{F}_n^{\text{in}}$, $\boldsymbol{F}_n^{\text{men-emb}}$. The results are shown in Table 5.2, Table 5.3, Table 5.4, and Table 5.5.

Results show promising results, with an increase in system performance across metrics for every feature added. This is potentially due to the increased amount of information contained in each quote vector, enabling easier disambiguation of speakers of quotes. The final system with all features added achieved an average F1 score of 35.86, which was 5.71 points above the baseline system using only $\boldsymbol{F}_n^{\text{emb}}$.

$\boldsymbol{V}_n$	B^3 Precision	Recall	F1
$\boldsymbol{F}_n^{\text{emb}}$	25.34	9.03	13.32
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}\right]$	24.89	4.93	8.23
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}\right]$	25.66	5.51	9.08
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	37.80	7.75	12.86
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	38.89	9.75	15.60
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	**44.70**	**10.19**	**16.60**

TABLE 5.4: Results of the holistic system performance evaluation in B^3.

$\boldsymbol{V}_n$	CEAF_e Precision	Recall	F1
$\boldsymbol{F}_n^{\text{emb}}$	4.11	15.28	6.48
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}\right]$	3.41	12.68	5.38
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}\right]$	3.76	13.97	5.92
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	7.52	27.96	11.86
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	7.55	28.07	11.91
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	**8.64**	**32.12**	**13.62**

TABLE 5.5: Results of the holistic system performance evaluation in CEAF_e.

On the other hand, we also observed a 4.65-point drop when adding the Word Count feature to the baseline. This result suggested that not all features may be effective, and some may even be harmful, contrary to intuition.

5.5 Experiment 3: Evaluation of Effective Features

For this experiment, we evaluated the base effectiveness of each feature, to determine which ones have a negative effect and should be removed. We prepared six variations on the quote vector $\boldsymbol{V}_n$, starting with $\boldsymbol{V}_n = \boldsymbol{F}_n^{emb}$. Each feature, in the order of $\boldsymbol{F}_n^{\text{wc}}$, $\boldsymbol{F}_n^{\text{gen}}$, $\boldsymbol{F}_n^{\text{near}}$, $\boldsymbol{F}_n^{\text{in}}$, $\boldsymbol{F}_n^{\text{men-emb}}$, is concatenated individually with $\boldsymbol{F}_n^{emb}$ to determine whether or not they have a negative effect. If a combination performs worse than

$\boldsymbol{V}_n$	MUC F1	B^3 F1	CEAF_e F1	Avg. F1
$\boldsymbol{F}_n^{\text{emb}}$	70.59	13.32	6.48	30.13
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	64.77	8.42	6.14	26.44
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}]$	72.23	13.21	6.36	30.60
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{near}}]$	80.74	17.23	14.61	37.53
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{in}}]$	79.10	17.08	10.11	35.43
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{men-emb}}]$	76.11	16.62	8.04	33.59

TABLE 5.6: Results of the evaluation of effective features. Avg. F1 is an average of the F1 score for MUC, B^3, and CEAF_e.

$\boldsymbol{V}_n$	MUC Precision	MUC Recall	MUC F1
$\boldsymbol{F}_n^{\text{emb}}$	72.54	68.75	70.59
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	66.56	63.08	64.77
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}]$	74.23	70.34	72.23
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{near}}]$	82.97	78.63	80.74
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{in}}]$	81.28	77.03	79.10
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{men-emb}}]$	78.22	74.12	76.11

TABLE 5.7: Results of the evaluation of effective features in MUC.

$\boldsymbol{V}_n$	B^3 Precision	B^3 Recall	B^3 F1
$\boldsymbol{F}_n^{\text{emb}}$	25.34	9.03	13.32
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	26.12	5.02	8.42
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}]$	25.90	8.86	13.21
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{near}}]$	40.02	10.98	17.23
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{in}}]$	37.33	11.07	17.08
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{men-emb}}]$	30.64	11.41	16.62

TABLE 5.8: Results of the evaluation of effective features in B^3.

the baseline, we can assume that the combined feature is adversarial to performance gains.

The results are shown in Table 5.6, Table 5.7, Table 5.8, and Table 5.9.

$\boldsymbol{V}_n$	CEAF_e Precision	Recall	F1
$\boldsymbol{F}_n^{\text{emb}}$	4.11	15.28	6.48
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}]$	3.89	14.47	6.14
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}]$	4.04	15.00	6.36
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{near}}]$	9.27	34.45	14.61
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{in}}]$	6.41	23.83	10.11
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{men-emb}}]$	5.10	18.95	8.04

TABLE 5.9: Results of the evaluation of effective features in CEAF_e.

Results show that most features have a beneficial effect on the baseline system. As expected, the word count feature $\boldsymbol{F}_n^{\text{wc}}$ was shown to have the greatest negative effect.

While it is not known exactly why this is the case, we may suppose that the distribution of speech lengths for all spoken text by a given speaker is consistent, and that most characters will utter a combination of short phrases and long sentences. It remains to be seen whether this feature is detrimental only for this dataset, or for the literary domain in general.

5.6 Experiment 4: Feature Pruned System Performance

For this experiment, we reflect the findings of the previous experiment and prune features accordingly. Namely, we remove the word count feature $\boldsymbol{F}_n^{\text{wc}}$ and test the performance of our newly pruned system in the same manner as Experiment 2.

We prepared five variations on the quote vector $\boldsymbol{V}_n$, starting with $\boldsymbol{V}_n = \boldsymbol{F}_n^{emb}$. Features are added in the order of $\boldsymbol{F}_n^{\text{gen}}$, $\boldsymbol{F}_n^{\text{near}}$, $\boldsymbol{F}_n^{\text{in}}$, $\boldsymbol{F}_n^{\text{men-emb}}$. The results are shown in Table 5.10, Table 5.11, Table 5.12, and Table 5.13.

$\boldsymbol{V}_n$	MUC F1	B^3 F1	CEAF_e F1	Avg. F1
$\boldsymbol{F}_n^{\text{emb}}$	70.59	13.32	6.48	30.13
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}\right]$	72.23	13.21	6.36	30.60
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	82.83	20.79	16.34	39.99
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	84.32	25.25	15.67	41.75
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	88.80	31.78	18.19	46.26

TABLE 5.10: Results of the feature-pruned system performance experiment. Avg. F1 is an average of the F1 score for MUC, B^3, and CEAF_e.

$\boldsymbol{V}_n$	MUC Precision	MUC Recall	MUC F1
$\boldsymbol{F}_n^{\text{emb}}$	72.54	68.75	70.59
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}\right]$	74.23	70.34	72.23
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	85.12	80.66	82.83
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	86.65	82.12	84.32
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	91.25	86.48	88.80

TABLE 5.11: Results of the feature-pruned system performance experiment in MUC.

$\boldsymbol{V}_n$	B^3 Precision	B^3 Recall	B^3 F1
$\boldsymbol{F}_n^{\text{emb}}$	25.34	9.03	13.32
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}\right]$	25.90	8.86	13.21
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	44.15	13.60	20.79
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	41.36	18.17	25.25
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	44.22	24.80	31.78

TABLE 5.12: Results of the feature-pruned system performance experiment in B^3.

Results show an impressive improvement over both the baseline and the results in Experiment 1. Not only does all of the features newly concatenated show an additive effect over the previous vector (although this may have been a minor coincidence with $\boldsymbol{F}_n^{\text{gen}}$), the final system using the quote vector

$$\boldsymbol{V}_n = \boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}$$

$\boldsymbol{V}_n$	CEAF_e Precision	Recall	F1
$\boldsymbol{F}_n^{\text{emb}}$	4.11	15.28	6.48
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}\right]$	4.04	15.00	6.36
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}\right]$	10.37	38.53	16.34
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}\right]$	9.94	36.94	15.67
$\left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]$	11.54	42.88	18.19

TABLE 5.13: Results of the feature-pruned system performance experiment in CEAF_e.

achieved a 10.42 gain in average F1 over the original system with the quote vector

$\boldsymbol{V}_n = \boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{wc}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}$.

Such a large increase in performance holds promise for the discovery of better features to build quote vectors out of.

5.7 Experiment 5: Diagonal Matrix Utilization

For this experiment, we consider the runtime of our system, and attempt to improve by training a diagonal matrix for the MMC model.

All of the experiments so far have trained a square matrix as the distance metric for use in the clustering stage of the algorithm. Although the clustering performance has been satisfactory, updating the values of a square matrix, as opposed to those of a diagonal matrix, takes time in the order of $O(N^2)$ as opposed to $O(N)$. This meant that the amount of time it took to train these models for the largest quote vectors, which are around 200 dimensions in our experiment setting, took hours.

Since one of the merits of solving the task of unsupervised speaker identification of quotes is that systems do not need a costly preprocessing step, and thus, can be applied to large amounts of literary texts. It would not make much practical sense if processing a single book took a system several hours.

$\boldsymbol{V}_n$	Avg. F1		Runtime	
	Square	Diagonal	Square	Diagonal
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}]$	30.60	30.31	0:13:06	0:41
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}]$	39.99	40.61	0:52:21	2:08
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}]$	41.75	40.40	6:37:38	2:09
$[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}]$	46.26	41.13	9:21:30	2:07

TABLE 5.14: Performance of systems with different matrices for the distance metric. The clustering performance is measured by Avg. F1, an average of the F1 score for MUC, B^3, and CEAF_e.

Hence, we attempt to cut back on the time that is required for a good training of a distance metric, by only training the diagonal elements. Essentially, this means that the features in the input quote vector will be weighted according to corresponding elements in the distance matrix, with salient features being given more weight.

For this experiment, runtimes were calculated on only the second step of the proposed system, which includes processes other than training the diagonal matrix, including feature extraction and k-means clustering. However, as we want to evaluate the performance of the system and not just that of the metric training, we believe this setting to be appropriate.

We prepared five variations on the quote vector $\boldsymbol{V}_n$, starting with $\boldsymbol{V}_n = \boldsymbol{F}_n^{emb} + \boldsymbol{F}_n^{\text{gen}}$. Features are added in the order of $\boldsymbol{F}_n^{\text{near}}$, $\boldsymbol{F}_n^{\text{in}}$, $\boldsymbol{F}_n^{\text{men-emb}}$. We used a diagonal matrix for the distance metric, and their average F-scores and runtimes were compared to those in the previous experiment.

The results are shown in Table 5.14.

In terms of quote clustering performance, the average F1 of both Square and Diagonal systems are comparable, except the last system, in which the Square system is around 5 points ahead of the Diagonal system.

In terms of runtime, the Diagonal systems were much faster than their Square counterparts. While the difference in runtimes started off at a difference of 12 minutes or so, which could be a somewhat insignificant difference when dealing

with individual novel datasets, the difference in runtime expanded significantly with the introduction of additional features. Ultimately, of the systems using the quote vector $\boldsymbol{V}_n = \boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}$, the Square system took over 9 hours to complete the program, while the Diagonal system only took 2 minutes.

To summarize, although using a diagonal matrix for the distance metric results in a marginally lower clustering performance, not only is this difference negligible when the dimension of the quote vector is low, this difference is overshadowed by vast decrease in runtime that is possible.

5.8 Experiment 6: Cluster Size Adjustment

In this experiment, we investigate the effect changing the cluster size of the k-means clustering algorithm has on system performance. So far in this study, the cluster size K was determined to be the number of entities that were in the character list, which was built in the preprocessing stage. Optimally, this number should be the number of characters who have quoted speech, since our goal is to produce a set of clusters in which one cluster contains all quotes from one speaker. Although our list can be a starting point for counting the number of characters, it is not accurate, as there will most likely be entities which are duplicates (e.g. 'John Smith' appearing as 'Mr. Smith' and 'John' will be added to the list each time) as well as entities that failed to be captured by the named entity recognition parser.

We tested the cluster number K with values from 5 - 100. Results are shown in Table 5.15.

Results indicate that the best average F_1 score was achieved when $K = 10$, and gradually decreased as the value of k- went down. Since the correct number of speaker entities in this dataset is 41, we may interpret the results as showing that if K = 40 or below, some speakers were joined together, and that if K was greater than 40, clusters of quotes with the same speaker were further broken up. This

K	MUC F_1	B^3 F_1	$CEAF_e$ F_1	Avg. F_1
5	95.45	34.31	15.08	48.28
10	92.76	32.67	27.08	50.84
20	89.65	28.18	23.32	47.05
30	86.63	25.62	22.28	44.84
40	85.20	24.34	18.61	42.72
50	83.01	21.61	17.82	40.81
60	83.63	21.37	14.37	39.79
70	81.99	19.81	14.08	38.63
80	80.64	20.46	11.39	37.50
90	79.87	18.74	11.31	36.64
100	77.70	17.80	8.89	34.80

TABLE 5.15: Evaluation results for each cluster number in k-means clustering.

explains the general trend of score decreasing after 40, but does not explain why $K = 40$ did not have the best results. Further investigation may be needed.

5.9 Experiment 7: Testing on Other Datasets

For this experiment, we investigate the robustness of our system by evaluating the performance of our system on the Emma, Pride & Prejudice, and The Steppe portions of the QuoteLi3 dataset. We prepared three systems to be tested on each dataset.

Baseline For the baseline, as in the other experiments, we implement our system with the quote vector $\boldsymbol{V}_n = \boldsymbol{F}_n^{emb}$.

Dumb Clustering This is an alternative baseline in which quotes are clustered using only the information gained in the first part of the system. Namely, for every quote with a direct entity mention, we place every quote in a cluster with other quotes containing the same mention. All quotes with mentions that are not person names (such as pronouns), or with no identified mentions, will be excluded from clustering.

System	MUC F_1	B^3 F_1	$CEAF_e$ F_1	Avg. F_1
Baseline	70.59	13.32	6.48	30.13
Dumb Clustering	60.15	25.57	41.66	42.46
Proposed	88.80	31.78	18.19	46.26

TABLE 5.16: Evaluation results for the Emma dataset.

System	MUC F_1	B^3 F_1	$CEAF_e$ F_1	Avg. F_1
Baseline	62.56	7.11	4.35	24.67
Dumb Clustering	54.04	20.22	34.03	36.10
Proposed	88.79	29.43	14.90	44.37

TABLE 5.17: Evaluation results for the Pride & Prejudice dataset.

System	MUC F_1	B^3 F_1	$CEAF_e$ F_1	Avg. F_1
Baseline	52.81	11.19	10.87	24.96
Dumb Clustering	62.97	32.15	41.31	45.48
Proposed	81.62	37.69	23.28	47.53

TABLE 5.18: Evaluation results for the The Steppe dataset.

Proposed System Our proposed system. We utilize the quote vector

$$\boldsymbol{V}_n = \left[\boldsymbol{F}_n^{\text{emb}}; \boldsymbol{F}_n^{\text{gen}}; \boldsymbol{F}_n^{\text{near}}; \boldsymbol{F}_n^{\text{in}}; \boldsymbol{F}_n^{\text{men-emb}}\right]. \tag{5.1}$$

The full distance matrix was trained.

The results are shown in Table 5.16, Table 5.17, and Table 5.18.

Results indicate that our proposed system performed the best across all datasets in terms of Average F_1, with the largest gain from the baselines being observed in the Pride & Prejudice portion. This shows that our system is robust, and can be applied to data other than the one that is used to prune the features.

On the other hand, our proposed system performed notably worse than the dumb clustering baseline in the $CEAF_e$ F_1 metric. We believe this is due to the fact that

$CEAF_e$ involves an alignment of key and response clusters, which is very accurate when the clusters are built from speakers confidently predicted using the first stage of our system. The relatively low-precision, high-recall prediction of clusters in our second stage may work to lower the similarity of aligned clusters.

Chapter 6

Conclusion

The contributions of this research are twofold. First, we introduced a novel task, unsupervised speaker identification of quotes in literary text. Second, we proposed and evaluated a clustering method using vector representations of each quote and a distance matrix.

Proposal of a novel task: unsupervised speaker identification of quotes in literary text

We proposed a new task in which speakers of quoted speech in literary works are identified in an unsupervised manner, in terms of not using manually created character lists. The significance of this task lies in its potential to be readily applied to literary analysis and for generating a dataset that can be used in natural language generation tasks, because it does not require training data with labeled quotes or character lists, both of which need to be made by hand. This also means that existing methods that leverage this information to identify speaker entities cannot be used.

In terms of the dataset that can be used, we confirmed that the existing QuoteLi3 dataset for the (supervised) speaker entity identification task for the literary domain, was appropriate for this task.

Proposal and evaluation of a quote clustering system

Our proposed method for this task has two steps. The first step linked speaker mentions to quotes using multiple rule-based sieves. In the second step, we clustered quotes with the same speaker identity: various features were combined into a quote vector, a distance metric was trained, and the quotes clustered using K-means clustering.

We first conducted an experiment to test the effectiveness of the first step in linking mentions to quotes. Our system achieved the state-of-the-art accuracy for unsupervised speaker mention identification in the literature domain. Although it did not reach the same performance as a previous supervised method, this result showed the first step to provide reliable output for building features in the speaker entity identification stage.

In the following experiments, we evaluated the performance of our system in various settings. Against a baseline that used only word embeddings, the proposed system had a 5.71 higher F_1, and ultimately a 16.13 higher F_1 after detrimental features were identified and removed. These results showed the effectiveness of our system in clustering quotes into clusters with the same speaker entities.

To improve the runtime of our system, we tested an alternative second step where the distance metric was trained as a diagonal matrix. Results showed a drastic reduction in runtime, especially when using quote vectors with high dimensions, with a reduction of 9 hours to 2 minutes was observed on our computing environment.

Although the performance decreased slightly with longer vectors, the short runtime that was achieved suggests that our method is well-suited for applications involving the processing of a large amount of literary text.

Investigation of different cluster numbers for K-means clustering showed interesting trends, with a low number of clusters achieving the highest scores. We believe this requires further investigation.

Chapter 7

Future Work

In the future, we wish to test the proposed method on a large-scale dataset with a varied set of literary works, as well as working on other methods that are applicable across languages.

In this chapter, we describe potential directions of future research.

Dataset

Although the task of identifying speaker entities to quotes has been investigated by various studies in recent years, the amount of available annotated data is extremely limited, both in size and domain. As was shown in Table 3.1, the QuoteLi3 dataset, which was the largest and most comprehensive dataset available for this task, contains only partial excerpts from three novels, from the same time period, two of which have the same authors. Therefore, it is imperative that a larger dataset be built or otherwise procured.

Since there is the issue of copyright, a practical solution would be to annotate more of the same books, or find other books of the same era and in the same language to annotate. While the QuoteLi3 dataset used professionals to annotate the data, we believe that this could be accomplished through crowdsourcing. Although some

of the English used in 19th century novels may be antiquated, many books from that time period are popular today in unabridged versions, and significant conflicts between annotators is unlikely to occur.

Since the unsupervised task does not allow for training data, it will be useful to annotate novels in different genres or writing styles, to see how various methods are able to generalize across genres and styles. This is particularly important when evaluating against rule-based systems including the first step of our system, since some of these rules may only apply to the writing or the English of that era. A variation in the literary domains will provide a chance to see how well each system performs for literature in general.

To our knowledge, there is no quote-mention or quote-speaker annotated dataset available for languages other than English. Since most of the world's literature is not translated, it is important that datasets be made for other languages, so that we are able evaluate systems for cross-lingual and multilingual performance.

Improving the Proposed Method

While our method produced good results for our proposed task, there is much room for improvement. Since we only tested six different features when building the quote vector, there may be room for further additions that may improve the clustering performance. For example, we might design a feature that captures the position of the quote in relation to the entire text, since some characters might talk more in some scenes and less or none in others. This might be particularly useful for plays, since the number of actors in a certain scene is often static.

Alternatively, we may modify existing features to improve performance. As an example, the current feature $\boldsymbol{F}^{\text{near}}$ simply captures the existence of nearby entities. Similar to the distance baseline used in O'Keefe et al.[11] for mention linking, we can modify the feature to weight entities based on the distance from the quote.

Finally, many famous books are translated into different languages, or made into comics, TV shows, and movies. We may be able to leverage these sources to improve the clustering performance, for example, by aligning the quotes and jointly training a machine translation system or building new features from images or video.

Bibliography

[1] Micaleah Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69 2 Pt 2:026113, 2004.

[2] Hua He, Denilson Barbosa, and Grzegorz Kondrak. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1312–1320. Association for Computational Linguistics, 2013.

[3] Grace Muzny, Michael Fang, Angel Chang, and Dan 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*, pages 460–470. Association for Computational Linguistics, 2017.

[4] Jeff Rydberg-Cox. Social networks and the language of greek tragedy. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science*, 1, 01 2011.

[5] Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social network analysis of alice in wonderland. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 88–96. Association for Computational Linguistics, 2012.

[6] John Lee and Chak Yan Yeung. An annotated corpus of direct speech. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck,

Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[7] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003. Association for Computational Linguistics, 2016.

[8] Bruno Pouliquen, Ralf Steinberger, and Clive Best. Automatic detection of quotations in multilingual news. In *Proceedings of Recent Advances in Natural Language Processing*, pages 487–492, 2007.

[9] Kevin Glass and Shaun 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 (PRASA' 07)*, pages 1–6, 2007.

[10] David K. Elson and Kathleen R. McKeown. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 1013–1019. AAAI Press, 2010.

[11] Tim O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew 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*, EMNLP-CoNLL '12, pages 790–799, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[12] Mariana S. C. Almeida, Miguel B. Almeida, and André F. T. Martins. A joint model for quotation attribution and coreference resolution. In *Proceedings of*

the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 39–48. Association for Computational Linguistics, 2014.

[13] Dario Pavllo, Tiziano Piccardi, and Robert West. Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping. *CoRR*, abs/1804.02525, 2018.

[14] Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The actortopic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*, 2010.

[15] Chak Yan Yeung and John Lee. Identifying speakers and listeners of quoted speech in literary works. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 325–329. Asian Federation of Natural Language Processing, 2017.

[16] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics, 2014.

[17] Francis Heylighen and Jean-Marc Dewaele. Variation in the contextuality of language: An empirical measure. *Foundations of Science*, 7(3):293–340, Sep 2002.

[18] Scott Nowson, Jon Oberlander, and Alastair J Gill. Weblogs, genres and individual differences. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, volume 1666, page 1671. Stresa, 2005.

[19] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, 2003.

[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[21] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 45–52, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[22] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, 1998.

[23] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 25–32, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Publications

Publications related to the thesis

Domestic conferences

(Poster Presentation)

- 遠田哲史，吉永直樹．“文学作品における教師なし話者同定．” 言語処理学会年次大会，2019.

Other publications

International conferences

(Oral and Poster Presentation)

- Masato Neishi*, Jin Sakuma*, Satoshi Tohda*, Shonosuke Ishiwatari, Naoki Yoshinaga, and Masashi Toyoda. “A Bag of Useful Tricks for Practical Neural Machine Translation: Embedding Layer Initialization and Large Batch Size.” In *Proc. WAT*, AFNLP, 2017. (* Joint First Author)