University of Alberta

AUTOMATIC SPEAKER IDENTIFICATION IN NOVELS

by

Hua He

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2011
Edmonton, Alberta

Abstract

Speaker identification is the task of attributing utterances to characters in literary narratives. Although only some of the utterances are explicitly attributed in novels, humans readers are able to determine the speakers of the remaining utterances because of their understanding of the plot. This dissertation proposes a method to automatically identify the speakers using supervised machine learning methods that utilize various text clues and a speaker alternation pattern. In addition, the method incorporates an unsupervised actor-topic model that aims to distinguish speakers depending on the content of their statements. The experimental results show that the method substantially outperforms a baseline method, and is competitive and more general when compared to previous approaches to the problem.

Acknowledgements

I would like to express my utmost gratitude to my supervisors, Dr. Grzegorz Kondrak and Dr. Denilson Barbosa. I am deeply indebted to them, whose help and encouragement guided me throughout the whole research project. Writing this thesis has been one of my most enjoyable and rewarding experiences, and without the wisdom my supervisors attempted to pass on me, this document would not exist.

I would like to thank Dr. Asli Celikyilmaz who provided the impetus for this research, as well as many very helpful suggestions, the initial program and the unsupervised features.

I would also like to thank Dr. Susan Brown and her students for providing us the high quality annotated data which was extensively used for this research.

And, I would like to thank my parents.

Table of Contents

List of Figures

Chapter 1

Introduction

Novels are considered as important social communication documents, with which novelists are interested in recording ordinary people's lives, usually by creating interesting conversations between different characters. The interactions of characters, especially in the forms of direct and indirect speeches, can be highly helpful to guide readers through the whole novel and help them ponder themes. Based on the conversation and utterance analysis, readers can easily grasp an idea of how the story goes, how the relationship between two characters change over time, and possibly what will be the next big thing happening. For example in the famous novel *Pride and Prejudice* (P&P) by *Jane Austen*, after finishing reading the first few conversations, readers might be wondering, will *Miss Elizabeth* fall in love with *Mr. Darcy*?

The topics on the conversations and interactions of characters are always popular among researchers (Atkinson and Heritage 1984), and there are various literature and social network study related questions that ask, for example, what are the typical roles that a major character may play, what can be found on the behavior differences between major and minor characters, and how can a social group be formed in novels. In addition, the social network analysis based on conversations in novels also researches the patterns of the character connections with a focus on how the collection of characters is linked to one another. Previous work in this fields include network construction (Elson et al. 2010), link prediction (Miller et al. 2009), discourse analysis (Redeker and Egg 2006), etc.

A specialized expert can claim to have answers to the above questions by spending a long time studying a few novels, but computer-assisted literary analysis with huge amounts of novels is able to play bigger role, by providing more insights on numerous literature and social network topics. However, such analysis is complicated; we cannot hope to create a computer program that can fully understand the human natural languages and novel contents

with today's Natural Language Processing technology. Instead, we can take full advantages of the information that is relatively easier to obtain, such as the identification of speakers in novels.

For example, assume you are tasked to build a social network of characters given a novel, instead of understanding everything about the whole novel before focusing on the task, there is a simpler alternative that you only need those characters who are talking in conversations. Because if two characters are talking to each other, it is reasonable to use this fact as an evidence of their relationship. Previous work on spoken language processing for broadcast conversations and multi-party meetings (Salamin et al. 2010, Favre et al. 2009) has already shown that this speaker identification idea is useful by including each conversation participant to extract social networks. Creating effective methods of identifying the speaker given an utterance can also provide a new window into the understanding of social communities of novels for even deeper analysis (Elson and McKeown 2010).

Moreover, when it comes to industry applications, the identification of speakers can be used to create higher quality audio books. By attributing characters to the corresponding utterances, the different sounds and tones from different utterances can be changed, therefore it surely provides premium experiences for audio book readers.

This thesis addresses the problem of the automatical speaker identification in novels. The results of the work serve several promising directions.

1.1 Speaker Identification Problem

Speaker identification is the process of automatically attributing utterances to their corresponding characters in novels. Because different characters talk in countless ways and the novelists also have diverse writing styles, this task is challenging. Imagine now you are given an utterance randomly picked from Chapter 2 of the novel *P&P*, here is one example,

Example 1

"I am sick of Mr. Bingley," cried his wife.

You might be able to identify the speaker of the above utterance is "*his wife*" by using a simple syntactic rule like this:

Pattern *"UTTERANCE"* said/replied/cried/continued SPEAKER.

However, troubles still exist in determining who on earth is the mention *"his wife"*. And it is quite challenging to answer the question, even for humans who have never read the P&P

novel before, because no explicit helpful information can be found near this utterance.

In order to solve the speaker identification problem with satisfactory results, a total of three major challenges should be addressed:

Lack of Predictive Information As mentioned in the Example 1 above, it is quite difficult, even for humans, to make predictions for utterances which do not have any explicit speaker information nearby. In the P&P novel, however, there are only less than about 40% of utterances that contains helpful explicit information, while the rest 60%, which is a major part of the novel, has nothing useful for prediction.

Large Number of Candidates The total number of characters in a novel is usually large, so traditional multi-class classification or rule-based methods cannot be directly applied to obtain acceptable results. For example, in the novel *P&P*, there are at least 52 characters and to pick a speaker directly from such a large pool of candidates given an utterance is never easy.

Generalization The third key challenge is generalizing a good identification model for new books which are not included in the training set. The ultimate goal for the speaker identification system is to make the model trained on one novel and at the same time it should be used and generalized to other books in new domains, because the annotation data is always expensive to obtain. In order to deal with the lack of data issue, a cross-domain model is indeed preferred for real practice, for example, the model should be able to learn something from the book *Emma*, and can be used to make predictions on the popular novel *Harry Porter*.

1.2 Summary of Main Contributions

The key contribution of this thesis is to propose a direction on the utilization of the speaker alternation pattern for the speaker identification problem. Previous work focuses on the traditional multi-class classification which treats each prediction task independently, however, it is quite obvious that all prediction tasks on different utterances have relationship, and such dependencies should be taken into consideration during the classification. Thus by using a speaker alternation pattern, the problem should be treated as a structured prediction problem. The details of the speaker alternation pattern are described in Chapter 6. The utilization of speaker alternation pattern can be a key to treatment for the first challenge: the lack of predictive information, because this pattern can provide extra information for

prediction, and the experiments on oracle models have proven its effectiveness.

The contributions of my work are shown below:

- limit the number of potential speaker candidates given an utterance by using a dialogue model which can be considered as a solution for the second challenge mentioned above: large number of candidates (Chapter 4) and the model is also firstly used in our paper (Celikyilmaz et al. 2010).
- verify the effectiveness of the speaker alternation pattern by using an oracle model (Chapter 7).
- increase the final performance and be more generalizable with new domains by using several novel features, including the vocative features, the unsupervised actor-topic model features and the speaker alternation features (Chapter 5).
- annotate new novels with the self-developed annotation toolkits (Chapter 3).
- incorporate the pattern information by using several graphical models (Appendix A).
- construct social networks based on the speaker identification system (Appendix B).

1.3 Outline of Thesis

The thesis is divided into three major parts. The first part is about the text processing and background from Chapter 1 to Chapter 3. Chapter 2 summarizes related work with a focus on both rule-based and machine learning approaches. Chapter 3 describes the data sets throughout the research, and also introduces the annotation process and self-developed toolkit.

The second part starts from Chapter 4 describing the dialog model with dialog chains, which provides a limited number of characters as candidates. And Chapter 5 describes the details on feature engineering.

The third part is about the statistical models and experiments. In Chapter 6 the speaker alternation pattern is described. Chapter 7 describes the supervised local models and oracle models, also Appendix A lists various attempts on the graphical model side, Appendix B shows how to extract the social network based on the speaker identification results. Chapter 8 summarizes the thesis and discusses the future work.

Chapter 2

Related Work

This chapter introduces the previous work that is highly related to the approaches of this thesis. An overview of the research on the speaker identification problem is also provided. The chapter begins with a discussion of early computational efforts, including work in rule-based methods and later the machine learning approaches. The corresponding issues associated with each method will then be discussed.

2.1 Early Computational Efforts

2.1.1 Rule Based Method

The number of NLP applications of identifying the speakers given an utterance is small, although the theory and linguistic principles of the quoted speech, dialogues and conversations have long been researched over the past decades.

Glass and Bangay (2007) presents a rule-based scoring scheme method for the speaker identification of the quoted speech in novels. In general they focus on the verbs in each quoted speech, especially the ones that can be used for communication. Once they can identify the verb, a speech-verb-actor link will be extracted from the sentence, then they make use of this link pattern to exploit the potential information for the identification task. More specifically, their method consists of three steps:

1. The first step is to locate the speech verb, especially when there is more than one verb in the sentence. Their method tries to locate the right speech verb from such a candidate pool with a scoring scheme, which takes several factors into consideration and each factor can add score to those verb candidates. As a result the candidate with the highest score will be used as the speech verb. Their scoring factors include the distance of the verb to the quote, and the hypernym of the verb because of verb

similarity. They mainly use three categories in WordNet (Miller et al. 1990) to utilize the verb similarity: communicate, verbalize and breathe, because for most cases a typical speech verb in WordNet can have one of the three common ancestors.

2. Once the speech verb is obtained, the arguments of the identified verb can be located based on the dependency between its related elements. By using a part of speech tagger, a Connexor FDG parser and their scoring scheme, again several new factors will be examined, including the subject and the object relationship, the distance from verb, an abbreviation filter and so on; the actor will be chosen with the highest score from the system. Their key is the usage of the speech-verb-actor link.

3. This step is to match the real speaker name by using the identified actor in the previous step. Based on their scoring scheme, factors including direct reference, pronominal anaphora and nominal anaphora are responsible for choosing the right speaker name.

Here is one example of the speech-verb-actor link, provided in (Glass and Bangay 2007):

> **She shuddered** when she **heard** little Jammes speak of the ghost, **called** her a "silly little fool" and then, as she was the first to believe in ghosts in general, and the Opera ghost in particular, at once **asked** for details: *"Have you seen him?"*

The link for the above sentence can be shown below to identify the correct actor by using a Connexor FDG parser:

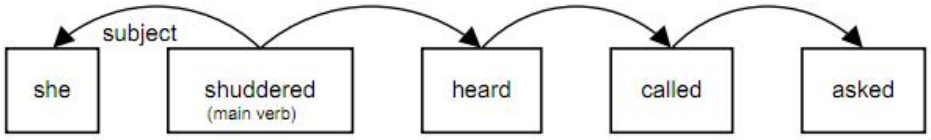

Figure 2.1: The Speech-verb-actor Link

Their work is predominately inspired by the work done in the anaphora resolution, which is to link different named entities together in the text. As described above, their work is purely based on hand-coded rules to implement the scoring scheme.

Another work also by the same authors (Glass and Bangay 2006) uses a rule generalization method instead of the scoring scheme. The general idea of their work is to use a seeding

rule set to generate new extra rules by adopting a merging scheme. The seeding rule sets are firstly constructed in tree forms, then based on a merging scheme the new hierarchical rule set can be generated.

Also the work in (Sarmento and Nunes 2009) recently presents a news feeds extraction system called *Verbatim*, which can extract speech quotes and index them from online texts in the news domain. They manually define 19 different variations that are similar to the typical speaker pattern, and then identify a total of 35 verbs as a candidate pool for the speech verb (while the work in (Glass and Bangay 2007) uses WordNet to score speech verbs). Thus based on syntactic rules, the above 19 variation patterns and the pool of the speech verbs, the speaker identification work can be done.

In terms of results, the precision obtained by the above rule-based methods is high, for example, the system from (Glass and Bangay 2007) gets nearly 90% precision on their own data set. Such high precision can be expected, because once the speaker pattern is identified and the rules are strictly matched within the text, the rule-based methods rarely make any mistakes.

The rule based methods can give us high precision, however, they also have their own problems, of which the biggest one is about the rule coverage gap, in other words, the hand made rules can not be generalized well. In terms of the F-score, the performances of all above rule based systems are highly limited.

For example, in the P&P data, as mentioned in the previous section only less than 40% of the data can somehow be matched with rules. The rest 60% has no hope to be solved with rules at all. Here is one example, which is a conversation extracted from Chapter 6 in P&P with three continuous utterances,

> ...
> *"Never, sir."*
> *"Do you not think it would be a proper compliment to the place?"*
> *"It is a compliment which I never pay to any place, if I can avoid it."*
> ...

Each of the utterances above appears by itself in a paragraph with no explicit speaker names and no other clues nearby. No rules can match the above cases and no speakers can be inferred if simply based on rules. Such cases are quite common both in the P&P data set and the Columbia corpus, which is introduced in Chapter 3

The papers in (Glass and Bangay 2006; 2007) also mention another related error source which is the misuse of the rules in the system, because natural language is always changing and no rule set can be big enough and subtle enough to capture every corners. For example, the list of possible speech verbs can be arbitrarily long, but it is not enough to manually define a fixed list of verbs or pick one verb from certain categories in the WordNet dictionary. In addition, it is always time consuming to create rules, especially when the data set grows larger.

In conclusion, the methods purely based on hand-coded rules are not feasible. Instead, the statistical methods can create rules implicitly and they are described in the following sections.

2.2 Machine Learning Approach

The rule-based methods suffer from the coverage and generalization problems. To address these concerns, it is time to turn to machine learning methods for help. Machine learning methods have been popular for the past few decades and can automatically learn implicit rules in the form of statistics without much human efforts. In real practice, machine learning methods are also capable of generalizing to new domains.

2.2.1 Supervised Learning for Multi-Class Classification

A very recent work by (Elson and McKeown 2010) provides a supervised statistical learning solution with the two-class logistics regression, J48 and JRip model. Given an utterance they extract one feature vector for each speaker candidate, then use machine learning models with all the extracted feature vectors to build the speaker identification system. Instead of using hand coded rules, the major advantage of statistical learning is that all necessary features extracted from texts can be incorporated automatically to make prediction decisions. Here is a list which includes their important features:

- Distance in words between the speaker candidate and the utterance
- Existence of punctuation between the candidate and the utterance
- Relative position to the utterance for a given speaker
- Number of words in the utterance
- Number of appearances of candidate speakers

- Number of speaker names and utterances in each paragraph
- Presence and absence of speaker name within the given utterance
- Proportion of recent utterances that were spoken by the candidate

Given an utterance, their statistical models, such as logistic regression, will give binary labels and probability scores to each speaker candidate. In the final stage their method uses a reconciliation scheme to combine all results together and choose the speaker with the highest probability score provided by the models. The score here is created differently from the score for the work in (Glass and Bangay 2007), because this is generated based on the training data instead of the hand-coded rules.

In general, their work can be illustrated in the following figure:

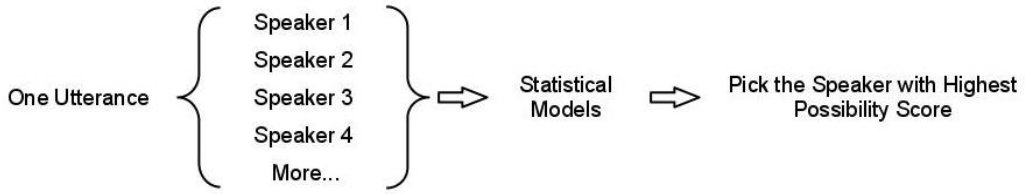

Figure 2.2: Multi-Class Classification in the Model by Elson and McKeown (2010)

Their method is a multi-class classification. However, one problem of their model exists because they treat each utterance independently, and the important relationship between utterances is never taken into consideration. In addition, they do not fully utilize the context information which can be obtained from the text surrounding the given utterance. But actually it is the relationship between utterances that can provide a promising way to improve the existing systems; this topic will be covered in later chapters.

The work by (Elson and McKeown 2010) also does not mention whether their method can be generalized to new domains, which should be important for the speaker identification, because the ultimate goal is to train the speaker identification system on annotated novels and can be applied to new unannotated novels for real usage.

Another work by the same author in (Elson et al. 2010) also utilizes the above system to extract social networks in novels to allow a systematic look and analysis at different novels.

2.2.2 Unsupervised Learning Classification

In the meanwhile, our recent paper (Celikyilmaz et al. 2010) also researches on an unsupervised method to identify the speakers given an utterance, and utilize the results to construct social networks in novels. Unsupervised learning does not require any labeled training data, and an author-topic generative model is used based on the relationship of utterances. An unsupervised system can be important because the data sparsity problem is always an issue in the supervised learning, so that the incorporation of unsupervised results as features can be highly necessary. To identify the relationship between utterances, this paper is also the first paper to use a dialogue-based model, which is described in Chapter 4 in details.

2.3 Conclusion

This chapter details several related works for the speaker identification problem, including the syntactically rule based approaches in the early stage and more recent machine learning approaches. The issues with those rule based approaches lead us to the machine learning method; however, the state-of-the-art machine learning system are also limited in several ways. The following chapters will try to deal with these issues.

Chapter 3

Data Sets

3.1 Introduction

In this chapter the self-annotated data from *Pride and Prejudice* and the publicly available Columbia corpus is described. The annotation process with our annotation toolkit is also introduced. In the last section, the comparison between two data sets is made.

3.2 Annotation Data on *Pride and Prejudice*

One contribution of this thesis is an annotation toolkit for the speaker identification problem. With the help of the toolkit, the annotation data set from the P&P novel provides a reliable source for any related research in future. One advantage of the P&P data is its high quality, while the Columbia corpus has much lower accuracy due to the nature of *Mechanic Turks* provided by *Amazon*.

Given a prepared list of all characters from the well-known nineteenth century English novel P&P, all utterances in the novel are labeled with correct characters. The P&P novel is chosen mainly because:

- Firstly, it is comparable to the public Columbia corpus which contains only nineteenth century English novels by influential authors between 1815 and 1899, such as *Jane Austen* and *Conan Doyle*.

- Secondly, the format of P&P is standard and traditional, and the book comes with a classical writing style where the paragraphs and quoted speeches are well organized. Meanwhile, since P&P has been studied thoroughly by literature researchers all over the world for decades, the available resources can enrich our knowledge on this book and thus help us further the research on speaker identification problem.

- Thirdly, although P&P is different from the well-established play or poem, it is still related to those older forms and consists of many independent dialogues between two or more individuals in a scene. The independent dialogues have a large proportion in the whole novel, and they are crucially important for this work.

All utterances in the novel are annotated. Here are some basic facts about the P&P data:

- Total number of annotated utterances: 1284.
- Total number of paragraphs after text preprocessing: 2012.
- Total number of characters in the data set: 52.

3.2.1 Annotation Procedure

The original text of the P&P novel is preprocessed with three steps as follows:

Step 1 Those paragraphs that contain quoted texts are extracted out. If the quoted speeches appear as the paragraph without interruption by narrations, they are kept and no changes are made. If the quoted speeches are separated by narrations, narration texts are replaced with a marker "[X]", then combine the separated quoted speeches which belong to the same paragraph, and keep them as new utterances. Please refer to Figure 3.1 below for an example.

Original Text		**Processed Text**	
	...		...
Para1	*"I am not likely to leave Kent for some time. Promise me, therefore, to come to Hunsford."*	UTT	*"I am not likely to leave Kent for some time. Promise me, therefore, to come to Hunsford."*
Para2	Elizabeth could not refuse, though she foresaw little pleasure in the visit.	NAR	Elizabeth could not refuse, though she foresaw little pleasure in the visit.
Para3	*"My father and Maria are to come to me in March,"* **added Charlotte**, *"and I hope you will consent to be of the party. Indeed, Eliza, you will be as welcome to me as either of them."*	UTT	*"My father and Maria are to come to me in March, **[X]** and I hope you will consent to be of the party. Indeed, Eliza, you will be as welcome to me as either of them."*
	...		...

Figure 3.1: An Example of Preprocessing, from Chapter 26 in P&P

Step 2 Once the preprocessing step on the original text is finished, a list of major characters of the novel is prepared. This work assumes that the character list is always provided beforehand. However, to prepare this list is not easy, and the way in this thesis to do it is, firstly use a name entity recognizer to identify all possible names, and then use existing knowledge to determine whether the output names should be included in the list or not. In the meanwhile, the gender of each character should also be provided.

One of the most important things about the preparation of the list is, both the formal name of a single speaker, and its corresponding aliases should be identified. For example, as shown in Figure 3.2 below, if *Lizzy* is found in the texts, it is important to know *Lizzy* also refers to *Elizabeth*. The third column of the figure displays the corresponding aliases for each character name. If there are multiple aliases, they are separated by semicolons.

SPEAKER NAME	**GENDER**	**ALIASES**
Elizabeth Bennet	F	Liz; Lizzy; Miss Lizzy; Miss Bennet; Miss Eliza; Eliza Bennet; Eliza; Elizabeth; Miss Elizabeth Bennet
Jane Bennet	F	Jane
Lydia Bennet	F	Lydia; Miss Lydia Bennet; Miss Lydia
Kitty Bennet	F	Catherine Bennet; Kitty
Mary Bennet	F	Mary
Mrs. Bennet	F	Bennet
Mr. Bingley	M	Bingley
Caroline Bingley	F	Caroline; Miss Bingley
Charlotte Lucas	F	Charlotte; Mrs. Collins; Miss Lucas
Mr. Collins	M	William Collins
Lady Catherine	F	Catherine
Mr. Darcy	M	Darcy;Mr. Fitzwilliam Darcy; Fitzwilliam Darcy
Lady Anne Darcy	F	Anne; Lady Anne
Georgiana Darcy	F	Georgiana
Colonel Fitzwilliam	M	Colonel F.Fitzwilliam
Mr. Gardiner	M	EDW. Gardiner; E. Gardiner
Maria Lucas	F	Maria; Miss Lucas
Louisa Hurst	F	Louisa; Mrs. Hurst
Sir William	M	Sir William Lucas
Anne de Bourgh	F	Bourgh; Miss de Bourgh
Mr. Wickham	M	Wickham; George Wickham; George
...		...

Figure 3.2: List of Major Characters that Appear in the Novel P&P

Step 3 Now it is time to move on to use the annotation toolkit for annotators to decide the

correct speakers for utterances. The toolkit can be ported to other novels as well.

3.2.2 Self-Developed Annotation Toolkit

The annotation toolkit is developed in C#. Here is the GUI interface:

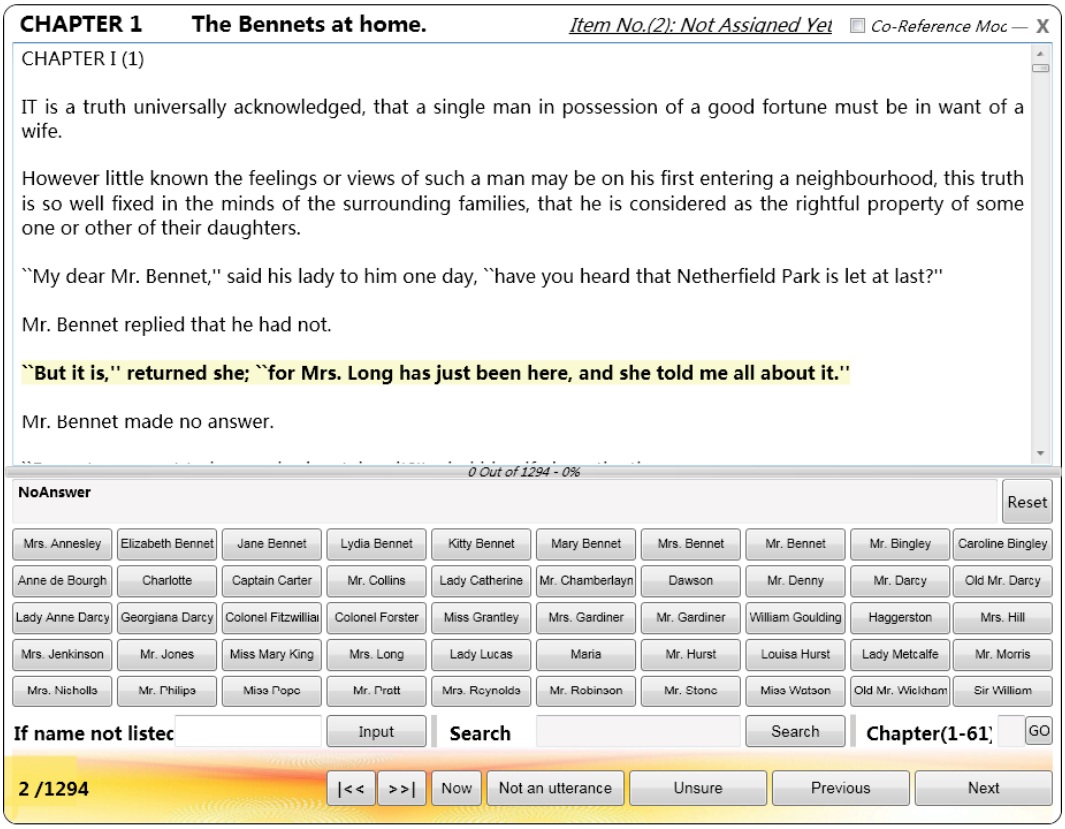

Figure 3.3: Annotation Toolkit GUI

This toolkit puts the whole text in front of the annotators, and highlights the utterance which needs annotation by automatically scrolling down the whole text. Once the list of characters is prepared, the buttons listed on the toolkit GUI can automatically associate characters and display their names. Then the annotators can decide who is the correct speaker; they can also read the texts around the utterance to avoid annotation mistakes. The annotation toolkit supports keyboard shortcut, and it is also possible to make changes on existing annotation if any mistakes are identified. The annotation toolkit is also designed to support large amount of annotation work with safety. Whenever a decision has been made, the tool ensures that all updated results are pushed into the file on disk, thus no matter what happened no data can be lost. Finally, the toolkit can run on any Windows platform with

.NET framework.

3.2.3 Data Format

The output file in the P&P novel is organized into the form of three columns. An example can be found in Figure 3.4 below.

Chapter	TYPE	CONTENT
1	NAR	IT is a truth universally acknowledged, that a single man in possession of a good fortune must be in want of a wife.
1	NAR	However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered as the rightful property of some one or other of their daughters.
1	Mrs. Bennet	*"My dear Mr. Bennet, [X] have you heard that Netherfield Park is let at last?"*
1	NAR	Mr. Bennet replied that he had not.
1	Mrs. Bennet	*"But it is, [X] for Mrs. Long has just been here, and she told me all about it."*
1	NAR	Mr. Bennet made no answer.
1	Mrs. Bennet	*"Do not you want to know who has taken it?"*
1	Mr. Bennet	*"You want to tell me, and I have no objection to hearing it."*
1	NAR	This was invitation enough.
...		...

Figure 3.4: Output Format After Annotation from Chapter 1 in the P&P

Both narrations and utterances are in a sequential order which is the same as the original book. The contents are in the third column, and the annotation results of the corresponding third column can be found in the second column.

3.3 Columbia Corpus

The Columbia Corpus is firstly described in the paper by Elson and McKeown (2010) from the Columbia University. It is publicly available upon request.

The Columbia Corpus gathers novels by six authors who published in the nineteenth century. Such variety is to prevent overfitting to the style of any particular author: four authors wrote in English, one in Russian and one in French; two authors contribute short stories and the rest novels. Among them, excerpts are taken from *Emma*, *Madame Bovary* and *The Adventures of Tom Sawyer*. The full Columbia corpus consists of around 111,000

words with more than 5,000 utterances, about 3,176 instances of utterances have annotations.

To obtain gold standard annotations, Amazon's *Mechanical Turk* platform was used. For each quoted speech, 3 annotators were required to independently choose a speaker from the list of characters: up to 15 candidates were presented for each quote from up to 10 paragraphs preceding the paragraph with the quote.

There are three major differences between our data set on P&P and the Columbia Corpus:

Format Difference The way to organize utterances in novels between ours and theirs are different. For example, Figure 3.5 shows the difference in the format. Our dataset will by default combine the two parts of the whole utterance paragraph into one complete utterance if both parts are located in the same paragraph, while the Columbia Corpus divided them into 2 separate utterances.

ORIGINAL UTT	**OUR DATA**	**COLUMBIA CORPUS**
"How are you doing?" said John, "Long time no see!"	"How are you doing? [X] Long time no see!"	"How are you doing?" AND "Long time no see!"

Figure 3.5: Format Difference between the P&P Data and the Columbia Corpus

Annotation Difference Our data set contains all annotations from the novel, while in the Columbia corpus, not all utterances' annotation are provided by *Mechanic Turk*. For example, Figure 3.6 shows the non-annotated part of the Emma novel in the Columbia corpus, and the tag "UNK" in this figure means they are not annotated.

CHAP	**TAG**	**CONTENT**
...		...
1	UNK	*"Poor Miss Taylor!–I ... it is that Mr. Weston ever thought of her!"*
1	UNK	*"I cannot agree ... humours, when she might have a house of her own?"*
1	UNK	*"A house of her own!... And you have never any odd humours, my dear."*
1	UNK	*"How often we shall be ... we must go and pay wedding visit very soon."*
1	UNK	*"My dear, how am ... such a distance. I could not walk half so far."*

Figure 3.6: Non-annotated Part in Chapter 1 of Emma Novel from Columbia Corpus

Quality Difference As mentioned, the data quality in the Columbia Corpus is not high due to the nature of *Mechanic Turks*. For example, this utterance from Chapter 5 in

the novel *Emma* below should be spoken by one person, but in their annotation it is attributed to two different speakers, *Emma* and *Harriet* correspondingly,

> "*Part of my lace is gone*," said she, "*and I do not know how I am to contrive. I really am a most troublesome companion to you both, but I hope I am not often so ill-equipped. Mr. Elton, I must beg leave to stop at your house, and ask your housekeeper for a bit of ribband or string, or any thing just to keep my boot on.*"

A cursory inspection reveals that this type of mistake appears more than 60 times in the whole Columbia corpus, and this number is more than 3% of all utterances in the whole data, thus their corpus contains at least 3% mistakes.

Chapter 4

Generating Dialog Chains and Candidate Characters

4.1 Introduction

In this chapter, the dialogue model is introduced. With the dialogue model, each utterance is not independent, but is closely connected to other neighbor utterances and narrations.

A dialogue chain is a series of utterances together with narrations nearby. The purpose of adopting such a scheme is to model the conversations in novels. Also the reason of incorporating narrations near utterances is that they are usually the background context of conversations, and continuous utterances and narrations usually have a close relationship. As mentioned earlier, there are two advantages of adopting the dialogue model:

1. The dialogue model can construct the useful conversational pattern.

2. The pattern information can be utilized, thus unlike what the previous work does, the useful pattern information can be used to solve the speaker identification problem. The dialogue model and the corresponding dialogue chains are also the response to deal with the first challenge mentioned in Chapter 1: lack of predictive information.

One example of a dialogue in the novel P&P is shown in Figure 4.1 below, with the corresponding dialogue chain.

4.2 Literary Text Preprocessing

The text in novels needs to be preprocessed firstly, then dialogue chains can be generated based on the dialogue model.

Each paragraph is treated as a single unit and is processed separately within each chapter. For paragraphs that contain only narrations, they are simply kept. For the paragraphs

TYPE	SPKER	CONTENT
NAR		Her aunt assured her that she was; and Elizabeth having thanked her for...
UTT	Mrs. Bennet	*"wished they might be happy."*
UTT	Charlotte	*"I shall depend on hearing from you very often, Eliza."*
UTT	Elizabeth	*"That you certainly shall."*
UTT	Charlotte	*"And I have another favour to ask. Will you come and see me?"*
UTT	Elizabeth	*"We shall often meet, I hope, in Hertfordshire."*
UTT	Charlotte	*"I am not likely to leave Kent for some time. Promise me, therefore, to come to Hunsford."*
NAR		Elizabeth could not refuse, though she foresaw little pleasure in the visit.
UTT	Charlotte	*"My father and Maria are to come to me in March, [X] and I hope you will consent to be of the party. Indeed, Eliza, you will be as welcome to me as either of them."*
NAR		The wedding took place; the bride and bridegroom set off for Kent...
NAR		Jane had already written a few lines to her sister to announce their safe...
NAR		Her impatience for this second letter was as well rewarded as impatience...

Figure 4.1: One Example of a Dialogue, from Chapter 26 in the Novel P&P

that are wholly or partially surrounded by double quotations, there are two situations that they might need further processing:

Collapse Case Suppose there are two continuous paragraphs: the first paragraph ends with a comma or a colon instead of a period, and the other is an utterance within double quotations. If such cases can be identified in the text, then they will be combined together as a single paragraph.

For example, as shown in Figure 4.2 below, the two paragraphs in the second column will be combined to be a single paragraph in the third column.

To preprocess the text with collapse is beneficial, because unnecessary narrations in dialogue chains can be removed. Thus the collapse cases can help me "smooth" the resulting dialogue chain to better reveal the speaker alternation pattern.

Split Case Only for those collapsed paragraphs, if there are multiple narration sentences in the combined paragraph, then it is necessary to split this paragraph into several paragraphs. Suppose there are a total of $n = 3$ sentences in a collapse paragraph, the first $n - 1 = 2$ sentences are narrations. Then it is needed to split the first narrations

NO.	ORIGINAL TEXT	PROCESSED TEXT
P1	After listening one morning to their effusions on this subject, Mr. Bennet coolly observed,	After listening one morning to their effusions on this subject, Mr. Bennet coolly observed, *"From all that I can collect by your manner of talking, you must be two of the silliest girls in the country. I have suspected it some time, but I am now convinced."*
P2	*"From all that I can collect by your manner of talking, you must be two of the silliest girls in the country. I have suspected it some time, but I am now convinced."*	COMBINED
...	...	...

Figure 4.2: Collapse Case for Preprocessing, from Chapter 7 in the Novel P&P

into a single paragraph, and leave the rest sentences as the second paragraph. Here is one example in Figure 4.3 below,

NO.	ORIGINAL TEXT	STEP 1:COLLAPSE	STEP 2:SPLIT
P1	Miss Bingley immediately fixed her eyes on his face, and desired he would tell her what lady had the credit of inspiring such reflections. Mr. Darcy replied with great intrepidity,	Miss Bingley immediately fixed her eyes on his face, and desired he would tell her what lady had the credit of inspiring such reflections. Mr. Darcy replied with great intrepidity, "Miss Elizabeth Bennet."	Miss Bingley immediately fixed her eyes on his face, and desired he would tell her what lady had the credit of inspiring such reflections.
P2	"Miss Elizabeth Bennet."	COMBINED	Mr. Darcy replied with great intrepidity, "Miss Elizabeth Bennet."

Figure 4.3: Split Case for Preprocessing, from Chapter 6 in the Novel P&P

The purpose of the preprocessing step is to make sure the generated dialogue chains can be used realistically, and the split cases are used to avoid too many collapse cases. Thus the resulting texts can not only get rid of unnecessary narrations to maintain smooth dialogue chains, but also lower the information loss, finally to better reveal the speaker alternation pattern.

4.3 Tag Sequence Generation

This section describes the steps to generate the dialogue chains, which are the basis of the probabilistic model in this thesis.

In literary text, conversations take place between people in sequence with possible overlapping dialogues formed. A dialogue is defined as segmented structured text, or a sequence of sentences, which are either an utterance within a double quotation with an assignment of character, or a narration. This work wants to build such a dialogue model that can predict characters conversing in a dialogue.

After the text preprocessing, now it is needed to tag all paragraphs within each chapter. Several tags are prepared for different types of paragraphs. They are listed below:

1. Paragraphs that have only narrations, thus such paragraphs are tagged as narrations, or NAR in short.

2. Paragraphs that are wholly surrounded by double quotations, or are a mix of narrations and utterances. Such paragraphs are tagged as utterances, or UTT in short.

Once the available tags for all paragraphs in the novel are obtained, a sequence of tags can be generated, and it is shown in Figure 4.4 below as an example:

TAG SEQ	**CONTENT**
NAR	The invitation was accepted of course, and at a proper hour they joined the party...
NAR	Colonel Fitzwilliam seemed really glad to see them; any...
UTT	*"What is that you are saying, Fitzwilliam? What is it you are talking of? What are you telling Miss Bennet? Let me hear what it is..."*
UTT	*"We are speaking of music, Madam,"*
UTT	*"Of music! Then pray speak aloud. It is of all subjects my delight..."*
NAR	Mr. Darcy spoke with affectionate praise of his sister's proficiency....
UTT	*"I am very glad to hear such a good account of her..."*
UTT	*"I assure you, Madam, [X] that she does not need such advice. She practises very constantly."*

Figure 4.4: Tag Sequence Example, from Chapter 31 in the Novel P&P

4.4 Dialogue Identification

Once the previous step of preparing the tag sequences is done, dialogues which can capture the conversations will be extracted. Although it sounds like a natural idea to only consider

those continuous non-NAR paragraphs, this is not enough because the purpose of the dialogue identification is also to generate a list of candidate characters for each dialogue, and the information on those candidate characters can also be located in narration paragraphs.

This work extracts all dialogues separately from each chapter, and then constructs them as follows:

1. Firstly a parameter n is set to 3, this parameter is used in the following steps. And the processing is within each chapter.

2. Starting from the first UTT paragraph within each chapter, a new dialogue collects all UTT paragraphs sequentially and stops the collection only when there are more than n NAR paragraphs interrupting conversations after the last UTT paragraph in the current dialogue.

3. For the above dialogue with UTT paragraphs, this work also selects maximum of n NAR paragraphs preceding the first UTT paragraph and appends those NAR paragraphs to the head of the dialogue.

4. Maximum of n NAR paragraphs after the end of the last UTT paragraph are also appended to the foot of this dialogue.

5. If there are more UTT paragraphs, then this work creates new dialogues to capture all UTT paragraphs.

6. The chapter end is also an indication of a dialogue end.

And here is an example of how dialogues are constructed. Figure 4.5 below shows two dialogues together with the close connections between utterances, and also how the *context window* looks like. In Figure 4.5 the first column *SEQ* indicates the dialogue sequential number; the second column *TYPE* indicates whether the paragraph is a narration or an utterance, if the paragraph is an utterance then the third column *SPKER* shows the speaker of that utterance; the last column displays the content of the paragraph. In the above cases, both dialogues share the same narrations in the middle of the grey part.

SEQ	TYPE	SPKER	CONTENT
Dialog 1	...		...
	UTT	Elizabeth	*"Had you then persuaded yourself that I should?"*
	UTT	Mr. Darcy	*"Indeed I had. What will you think of my vanity..."*
	UTT	Elizabeth	*"My manners must have been in fault, but not..."*
	UTT	Mr. Darcy	*"Hate you! I was angry perhaps at first, but my..."*
	UTT	Elizabeth	*"I am almost afraid of asking what you thought of..."*
	UTT	Mr. Darcy	*"No indeed; I felt nothing but surprise."*
	UTT	Elizabeth	*"Your surprise could not be greater than mine..."*
	UTT	Mr. Darcy	*"My object then,"* replied Darcy,*"was to shew you..."*
	NAR		He then told her of Georgiana's delight in her...
	NAR		She expressed her gratitude again...
	NAR		After walking several miles in a leisurely manner....
Dialog 2	UTT	Mr. Darcy	*"What could become of Mr. Bingley and Jane!"*
	UTT	Elizabeth	*"I must ask whether you were surprised?"*
	UTT	Mr. Darcy	*"Not at all. When I went away, I felt that it..."*
	UTT	Elizabeth	*"That is to say, you had given your permission..."*
	...		...

Figure 4.5: Several Dialogues from Chapter 58 in P&P

4.5 List of Candidate Characters Extraction

The character names and aliases found in each dialogue will be maintained as potential candidates for utterances within that dialogue.

Only the names appearing in this dialogue are treated as candidates and are kept in a list for the utterances of that dialogue. To identify the list of characters, the whole dialogue for characters names and aliases is scanned. The aliases are necessary because a character in a novel is usually referred to several different ways, e.g. *Caroline Bingley*, a character in the novel P&P, can be also referred as *Caroline*, or *Miss Bingley*.

Here is an example of a dialogue from the novel P&P and how the system extracts its candidate characters.

TYPE	CONTENT	CANDIDATES
NAR	Her aunt assured her that she was; and **Elizabeth** having thanked her for ... being given on such a point without being resented.	Elizabeth
UTT	*"wished they might be happy."*	Jane
UTT	*"I shall depend on hearing from you very often, **Eliza**."*	Maria
UTT	*"That you certainly shall."*	Mr. Collins
UTT	*"And I have another favour to ask. Will you come and see me?"*	Charlotte
UTT	*"We shall often meet, I hope, in Hertfordshire."*	Lady Catherine
UTT	*"I am not likely to leave Kent for some time. Promise me, therefore, to come to Hunsford."*	
NAR	**Elizabeth** could not refuse, though she foresaw little pleasure in the visit.	
UTT	*"My father and **Maria** are to come to me in March, [X] and I hope you will consent to be of the party. Indeed, **Eliza**, you will be as welcome to me as either of them."*	
NAR	The wedding took place; the bride and bridegroom set off...as usual. **Elizabeth** soon heard from her friend; and their correspondence was as regular and frequent as it had ever been; that it should be equally unreserved was impossible. **Elizabeth** could never ... of what had been, rather than what was. **Charlotte**'s first letters were received ... but be curiosity to know how she would speak of her new home, how she would like **Lady Catherine**, and how happy she would dare pronounce herself to be; though, when the letters were read, **Elizabeth** felt that **Charlotte** expressed ...and Lady Catherine's behaviour was most friendly and obliging. It was **Mr. Collins**'s picture...	
NAR	**Jane** had already written a few lines to her sister to announce their safe arrival in London; and when she wrote again, **Elizabeth** hoped it would be in her power to say something of the Bingleys.	

Figure 4.6: A Dialogue and Its list of Candidate Characters

In the above Figure 4.6, names that appear in the prepared list of characters are focused. As shown it is needed to extract all the names and aliases in bold, and keep those names in the third column as candidates for this dialogue. In the above example 6 names are found, thus the number of the potential candidates are only 6 instead of the total speaker number 52 in the novel P&P. Because of this limitation the number of the candidates is largely reduced, so a higher accuracy can be possible for the later speaker identification task.

It is worth mentioning that certain speaker names in one dialogue might not appear in that dialogue, but the evaluation on the P&P novel shows that, lists of candidate speakers

for each dialogue have a very low missing ratio, which is less than 1% in total.

4.6 Conclusion

This chapter describes how to identify the dialogues in the raw novel text, and how to extract the list of possible characters in each dialogue. The dialogues are extracted by paragraphs based on their types (NAR and UTT), and the purpose of constructing a list of possible characters for each dialogue is to reduce the number of candidates for multi-class classification.

The dialogues not only include the utterance paragraphs (UTT), but also contain narration paragraphs (NAR). Once dialogues are obtained, the next chapter describes how to extract the features to assist the classification.

Chapter 5

Feature Engineering

5.1 Introduction

The speaker identification method is statistical based on features by training a supervised classifier and testing on an independent data set. This chapter introduces all features that are used in the system. The features can represent various different aspects of the data, including the syntactic structures of the utterances, the surrounding narrations, the lexical clues in the novel, the semantics of the utterances, the whole dialogue discourse information and so on.

Generally speaking, the speaker identification system adopts two types of features, one belongs to the type of syntax, and the other is the type of semantics. Both types of features are closely related because of the linking theory from the work (Levin and Hovav 1995), which is about the relationship between syntactic surface cues and semantics content. The linking theory states that the syntactic patterns and realizations can be somehow predicted from semantics contents and vice versa. Thus due to the high relatedness of both feature types, it is obvious to adopt both syntactic and semantic features in order to present the whole view of the data set as much as possible. In the meanwhile, the speaker name related features are important, because they are the most direct clues of the correct speaker given an utterance.

The current utterance means an utterance that is to be assigned a speaker by the supervised mode, and the decision is being made.

5.2 Feature List

All of the important feature types are listed below in Figure 5.1 with comments. Here is an overview, and in the following sections of this chapter, all of them will be discussed.

FEATURE NAME	COMMENTS
Vocative Speaker Name	Binary, for each candidate speaker, possible to have more than one vocative feature given an utterance.
Dialogue Candidate Matching	Binary, and all utterances in the same dialogue have the same features.
Distance Feature	Not binary, this feature represents the top 2 closest speakers near the current utterance.
Gender Matching	Binary, whether the gender of the speaker is matched if the gender information is available.
Unsupervised Actor-Topic Model	Generated by unsupervised ATM model, the top four speaker identifiers as the feature values.
Speaker Name in Utterance	Not binary, speaker names appearing in utterance given an utterance, binary for each speaker.
Speaker Name in Narration	Not binary, speaker names within narrations of the UTT paragraph.
Speaker Appearance Count	Not binary, the count of the speaker names in the whole novel text, non-binary value
Neighbor Feature	Not binary, only used for oracle model, the feature value is the unique speaker identifier.

Figure 5.1: Feature List

5.3 Vocative Feature

The vocative feature is a novel feature for the speaker identification provided in this thesis. It is a binary feature for each corresponding speaker given an utterance. Intuitively, it represents a speaker name to be called by another speaker in a previous utterance during a conversation. When speaker A is talking to speaker B, it is highly possible for speaker A to mention the name of speaker B in his/her own utterance. For example, here is one conversation from Chapter 1 in the P&P novel,

> ...
> "*My dear **Mr. Bennet**,*" replied his wife, "*how can you be so tiresome! You must know that I am thinking of his marrying one of them.*"
> "Is that his design in settling here?"
> ...

In the above example, the first utterance is spoken by *Mrs. Bennet*, the second is by *Mr. Bennet*. *Mrs. Bennet* mentioned "My dear **Mr. Bennet**" in her utterance because she is talking directly to *Mr. Bennet*, this name is a good indicator of the speaker, and the statistical

model can intuitively guess that the next utterance should be spoken by *Mr. Bennet* with very high confidence.

To use this feature, a few requirements should be satisfied as shown below,

- Based on the dialogue chain (see Chapter 4), the current utterance should have a continuous previous utterance without any narration in between.
- It has to be a speaker name, or an alias from the prepared people list (see Chapter 3.2.1).
- The speaker name should only appear within the utterance part of the previous UTT paragraph, not from narration part of that UTT paragraph.
- This speaker name should be identified by the machine learning classifier to be vocative. The classifier will be described in the next section.

If all of the above requirements are satisfied, then the vocative feature of the corresponding speaker can be set to one. The vocative feature is not only limited for the one utterance, actually it can be used for three continuous utterances given a current utterance: based on the speaker alternation pattern the vocative speaker name features within the previous utterance and the second previous utterance can be added if they are identified by the classifier. One thing worth mentioning is, although a vocative name has very high possibility to be the speaker of the next continuous utterance, it is not always true, because when speaker A is talking to speaker B, speaker A can also mention speaker C's name with the form of vocative address.

This feature is found to be useful by the vocative prediction experiments - the system increased the accuracy by about 2% when this feature is plugged in. However, there exists a difficult question: among all speaker names appearing in the utterances, how can the system successfully choose the right speaker name to be vocative? As a short answer, a supervised classifier called Logistic Regression model is used. In the following sections, the experiment on vocative speaker name prediction is introduced.

5.3.1 Vocative Name Classification and Its Features

A Logistic Regression classifier (LR) (Agresti 2006) is built to predict whether a speaker name within an utterance can be vocative or not. Two things need to be done before the application of the machine learning model, shown as below:

Creating new training Data This work uses the training corpus in P&P to create a new vocative training data for the vocative name prediction, and all speaker names which satisfy the above requirements are annotated with either vocative tag or non-vocative tag manually. A total of about 900 names are tagged within utterances in the P&P novel; 25% of the tagged names are set to be vocative.

Gathering features In order to build the supervised classifier, various related features are extracted and collected after careful observations. Here is the list of all the features used to predict vocative names:

FEATURE NAME	**Example**
Beginning of the utterance	*"Mr. Bennet you are ..."*
Start with emotional words like "dear"/"dearest" etc.	*"O Dear Mr. Collins ..."*
Separated by comma	*"..., Elizabeth, ..."*
Names end with "!"	*".. Mrs. Bennet! ..."*
Names are located at end of the utterance with periods	*"... Mr. Darcy. "*
Names end with question mark	*"... Lizzy?..."*
Separated by both a comma and a period	*" , Kitty."*
With emotional words like "oh!" in front of names	*"Oh! Emma..."*
The utterance contains "you" in the sentence	*"My Eliza, you are ..."*

Figure 5.2: Feature List of the Vocative Name Prediction

The general idea of extracting features is to capture the emotional words and punctuation, such as question marks, exclamation marks, "oh", "my dear/dearest" and so on, because it is obvious that such emotional features with speaker names nearby can be good indicators, and experiment details are shown in the next section.

5.3.2 Experiment

This work uses the Logistic Regression classifier to extract the vocative features. Logistic Regression is a generalized linear model used for predictions of certain events' probability by fitting the training and testing data to a logistic curve, and it utilizes several contributing numerical and categorical features. In this experiment, with a 10-fold cross validation about 93% accuracy for correctly classified instances can be achieved, and the average F-score, recall and precision are all above 93%. Details are shown in the following figure.

Model NAME	Precision	Recall	F-Score
Logistic Regression	0.935	0.936	0.936

Figure 5.3: Experiment Results for the Vocative Name Prediction

After training, the system obtains the resulting weights by the LR model, and then applies the learned weights for future usage in the test corpus of the P&P data, and also the Emma data from the Columbia corpus.

5.4 Gender Matching Feature

The gender match feature is a binary feature, used to represent whether a potential speaker candidate can be matched by gender. Here is an example from Chapter 20 in the P&P,

> "*But depend upon it, Mr. Collins*," **she** added, "*that Lizzy shall be brought to reason. I will speak to her about it myself directly. She is a very headstrong foolish girl, and does not know her own interest; but I will make her know it*."

It is obvious to know this utterance is spoken by a female speaker, thus all female candidate speakers's gender matching feature value can be set to one, and all male speaker matching features can be set to zero. This feature is highly useful; however, the question is how to extract the gender information by rules. Next section presents an answer by using rule-based method. The rule-based method can provide high precision results which are exactly what this work needs. Although the recall (coverage) is lower, the coverage problem can be solved later with the overall statistical model for the speaker identification.

For the paragraphs that are wholly surrounded by double quotes, the gender information is usually not identifiable, thus the gender information extraction in this thesis focuses on the UTT paragraphs that contain parts of narrations. In order to successfully get the information, there is a big issue on the extraction of speaker aliases, who are associated with the communicative verb. The recognition of speaker aliases is important, because the gender extraction will be solely based on whether it is good to get the speaker mention of the current utterance correctly; once the speaker mention is identified, then its corresponding gender information can be extracted. Note that the gender information is not always available; such cases are treated differently depending on the models in this work. For example, in the SVM ranking model (Chapter 7) the missing gender cases are treated as "not matched" so that this gender matching feature is still binary, while in the CRF model, one more option called "not available" is added.

Inspired by the related work in Chapter 2, the system uses a deterministic method based on syntactic rules, and processes the narrations in the following steps:

Step 1 The speaker names and aliases appearing in the narrations should have already been replaced with the unique speaker IDs. In terms of the speaker name extraction, once the people list is in good status, the key word matching is good enough to capture all related names in the text. The example is shown below,

ORIGINAL TEXT	**SPEAKER IDENTIFIED TEXT**
"Lady Catherine is a very respectable, sensible woman indeed," added Charlotte, *"and a most attentive neighbour."*	*"Lady Catherine is a very respectable, sensible woman indeed,"* added **CHARACTERID10**, *"and a most attentive neighbour."*

Figure 5.4: One Example from Chapter 8 for the Speaker Identified Utterance

Step 2 Given narration text extracted from UTT paragraph, the system will first determine which type of the narration belongs to: Whether this piece of the narration text is located between two utterances, or before the utterance, or after. The three types of the narrations are listed below:

TYPE	**UTTERANCE EXAMPLE**
MIDDLE	"*I like her appearance,*" said CHARACTERID2, struck with other ideas. "*She looks sickly and cross. Yes, she will do for him very well. She will make him a very proper wife.*"
END	"*That is capital*," added her sister, and they both laughed heartily.
BEFORE	CHARACTERID19 thought the same, and added, "*She has nothing, in short, to recommend her, but being an excellent walker. I shall never forget her appearance this morning. She really looked almost wild.*"

Figure 5.5: Examples for the Different Utterance Types

One thing worth mentioning is that, there are many MIDDLE type of the utterances, and a lot fewer END and BEFORE cases. There also exists very few mixed utterances; For those mixed cases, the narration text for the MIDDLE type utterance is always with higher priority.

Step 3 Prepare a list of communicative verb, which is manually collected, they are:
talked/continued/added/said/cried/replied/spoke/thought
If a sentence contains more than one communicative verb, the way to handle them is discussed in step 5.

Step 4 The system will parse all the narrations from the UTT paragraph, and focus on the communicative verbs. In order to locate the speakers who are associated with those verbs, several proper parsers including a dependency parser, a part-of-speech tagger, and a semantic labeling parser are needed.

For example, with a sentence below:

> "*We have not quite determined how far it shall carry us,*" said Mrs. Gardiner; "*but perhaps, to the Lakes*."

This sentence is quite straightforward to get the speaker mention since there is only one communicative verb and one speaker name. With a couple of simple rules it can be determined that *Mrs. Gardiner* is the speaker mention of this utterance. There are about 30% of such MIDDLE utterance type cases in the whole P&P data set: the narration part is simple and straightforward, and such case has one strong communicative verb.

Here is another example:

> "*We are speaking of music, Madam,*" said he, when no longer able to avoid a reply.

In the above example, several syntactic rules are used to determine *he* is a speaker mention. By using a dependency parser by Stanford, the following dependencies can be obtained from the parser:

RELATION	**DEPENDENCIES**
...	...
prep	prep(speaking-4, of-5)
pobj	pobj(of-5, music-6)
appos	appos(music-6, Madam-8)
nsubj	nsubj(said-11, he-12)
advmod	advmod(avoid-19, when-14)
advmod	advmod(able-17, no-15)
advmod	advmod(no-15, longer-16)
dep	dep(when-14, able-17)
aux	aux(avoid-19, to-18)
dep	dep(said-11, avoid-19)

Figure 5.6: Dependencies for the Above Utterance from a Dependency Parser

The system focuses on two relations from the Stanford dependency parser, one is *nsubj*, the other is *dobj*. The *nsubj* relation represents a noun phrase is the syntactic

subject of a clause, and the *dobj* relation means that the direct object of a VP is the noun phrase which is the object of the verb. Once those two relations are identified, the system will start checking whether the relation contains a communicative verb. In the above example, the relation *nsubj(said-11, he-12)* is important and the speaker mention *he* can be got. In theory the talker should always be a subject thus *nsubj* relation should be the only correct relation the system should focus on, however, the dependency parser is also possible to make mistakes when the *dobj* relation for certain utterances' narration can be seen. The reason is not the topic of the thesis - the system simply adopts both relations here to get the talker out.

Step 5 If there is more than one communicative verb, then the system needs to distinguish the locations where the narration texts are extracted. If the narration text is extracted at the beginning part of the utterance paragraph (BEGIN), then the system will only focus on the communicative verb that appears at the end of the narration text; if the narration text is extracted from the end of the utterance paragraph (END), as shown in the above example, then the focus is on the first communicative verb. The results show that the speaker mentions found in the MIDDLE type of utterances are the most accurate ones. At the final step, the cleaning of some obvious wrong output is also carried out.

5.5 Dialogue Candidate Matching Feature

The dialogue candidate matching feature is a binary feature, it represents whether a speaker candidate belongs to the current dialogue which includes the current utterance. As mentioned in Chapter 4, a list of speakers will be provided as potential candidates for utterances within that dialogue. If a speaker name or its alias appears in any position within the dialogue windows, this speaker name should be in the dialogue candidate list. This list is important because the number of candidates in each dialogue can be limited.

When a decision of the speaker identification needs to be made for an utterance, the dialogue is checked and the corresponding list of speakers is used to set the dialogue candidate matching feature. If a speaker belongs to this list, the dialogue feature of this corresponding speaker is set to one, and vice versa.

5.6 Distance Feature

In the novel context, one important observation is that an utterance tends to be close to its speaker name. A distance feature is then used to represent whether a speaker is close to the utterance. This is also similar to the one used in David Elson's work (Elson and McKeown 2010).

Instead of adopting a real value to represent the distance between a speaker and an utterance, a relative position type is put into work instead: only the top 2 closest speakers near the current utterance are included, because by experiments the top 2 closest names can achieve the maximum accuracy without much confusing the supervised model. Also because of the speaker alternation pattern, this feature is included for the second previous continuous utterance and the second next continuous utterance.

5.7 Unsupervised Actor-Topic Model Feature

This feature is generated by the unsupervised actor-topic model (ACTM) in the work (Celikyilmaz et al. 2010). The ACTM is essentially an unsupervised model to predict the most possible speaker of a given utterance in the novel context; it also utilizes the dialogue model to construct conversational chains, and then based on Author-Topic model (Rosen-Zvi et al. 2010) to make good use of the rich contextual information to identify the most possible speakers for a given utterance. The details on ACTM can be found in Chapter 2.

For a given utterance, a list of top 4 speakers is generated by ACTM, and this list is directly used in the system as features.

5.8 Other Speaker Name Related Features

Features in this category are related to the speaker names directly, and they include:

Names within utterance This feature focuses on the speaker names within each utterance's double quotation. The intuition behind this feature is that a speaker A usually will not call A's name in his/her own utterances, thus it can be used to exclude certain speaker candidates during the prediction. For example,

> "*Of Mr. Collins and Lizzy. Lizzy declares she will not have Mr. Collins, and Mr. Collins begins to say that he will not have Lizzy*"

In the above example from Chapter 20 in the P&P, although people have no idea of who is speaker of this utterance if only based on its content, but at least people can guess *Mr. Collins/Lizzy* should not be the speaker of this utterance because both names appear in the utterance content.

Names in the narrations A speaker name is located in the narration part of the utterance can be used to solve the coverage problem mentioned in Chapter 5. If such information can be extracted, they are treated as features.

Name appearance counts This feature is used to provide an overview of major and minor speakers for the system based on the counts of speakers' appearances in the whole novel text. For each speaker, by scanning through the whole text, the system extracts the name and its alias, then counts this speaker's appearing frequency. Intuitively, if the number of a certain speaker is small, then such a speaker is a minor speaker and has lower chance to be picked given an utterance, and vice versa. In other words, the appearances counts of a certain speaker name can indicate whether this speaker in the novel is a major speaker or not.

5.9 Neighbor Feature for Oracle Model

This set of features are only for the oracle model which is discussed in Chapter 7. An oracle model assumes the speakers of all neighbor utterances are known already, and the system needs to decide who is the correct speaker of the current utterance. The neighbor utterances are the previous utterance, the second previous utterance, the next utterance, the second next utterance. A total of 8 speaker features from neighbor utterances is added.

5.10 Conclusion

This chapter describes important features for the speaker identification system. Most of those features are broadly applicable and can also be easily extracted for other new novel texts. Some novel features include the vocative features and the unsupervised actor-topic model features. In the following chapters, the supervised models are described.

Chapter 6

Speaker Alternation Pattern

6.1 Introduction

A conversational pattern of the speaker alternation exists commonly in any conversational corpus, it can represent a typical conversation between two speakers, who talk in turn both in a regular manner. This pattern contains the important information of the relationship of neighbor utterances in each dialogue and it is the most straightforward conversational pattern in the data, the pattern is also the basis for the oracle model (Chapter 7) and the local graphical models (Appendix A).

The discovery and the utilization of the speaker alternation pattern is the response to the first challenge: the lack of predictive information, which is mentioned in Chapter 1. Also this is the key contribution of this work by using the speaker conversational pattern to assist the multi-class classification, in other words, the speaker identification problem now becomes a structured prediction problem with the help of the alternation pattern, and is no longer a single traditional multi-class classification problem. All the following chapters are based on this combination of linguistics intuition and statistical machine learning method.

6.2 How It Looks Like

Figure 6.1 shows one dialogue extracted from the beginning of Chapter 4 in the P&P novel.

SPEAKER	**CONTENT**
NARRATION	When Jane and Elizabeth were alone, the former, who had been...
Jane	*"He is just what a young man ought to be," said she, "sensible ..."*
Elizabeth	*"He is also handsome," replied Elizabeth, "which a young man..."*
Jane	*"I was very much flattered by his asking me to dance a second time..."*
...	...

Figure 6.1: An Example of One Conversation Extracted from Chapter 4 in P&P

In the above example, the speaker of the second utterance in the middle is *Elizabeth*, based on intuition the next utterance's speaker who is talking to Elizabeth cannot be Elizabeth herself, in this example the next speaker is *Jane*. Again the same also applies to the utterance that is right before the utterance uttered by *Elizabeth*, the speaker of the previous utterance (the first utterance in the figure) cannot be *Elizabeth* either and this is again true in the above example. Intuitively it can be concluded with high confidence that neighbor speakers should not be the same, although this statement sounds quite strong, this is true for almost all cases in the P&P and Emma data set. And this judgment can be very useful, for example, as features (or as a graphical structure) for statistical models.

For another example, Figure 6.2 below shows utterances with two speakers *Elizabeth* and *Lady Cathering*, and they are talking to each other. This dialogue is extracted from Chapter 29 in P&P.

NO	SPEAKER	CONTENT
	NARRATION	Elizabeth could hardly help smiling as she assured her...
1	Lady Catherine	*"Then, who taught you? who attended to you? Without a... "*
2	Elizabeth	*"Compared with some families, I believe we were; but such..."*
3	Lady Catherine	*"Aye, no doubt; but that is what a governess will prevent...."*
4	Elizabeth	*"Yes, Ma'am, all."*
5	Lady Catherine	*"All! – What, all five out at once? Very odd! And you only..."*
6	Elizabeth	*"Yes, my youngest is not sixteen. Perhaps she is full young..."*
7	Lady Catherine	*"Upon my word," said her ladyship, "you give your opinion..."*
8	Elizabeth	*"With three younger sisters grown up," replied Elizabeth...*
	NARRATION	Lady Catherine seemed quite astonished at not receiving a...

Figure 6.2: An Example of Speaker Alternation Dialogue from Chapter 29 in P&P

In the above dialogue, two ladies are talking emotionally with anger. It is quite obvious that they are talking in turn, and this is a typical speaker alternation pattern, which means speakers within the same dialogue talk one by one and one after the other. Hence when it comes to the speakers of No. 1 utterance and No. 3 utterance, the alternation pattern assumes that they should have highly chance that both speakers are the same. The same applies to No. 2 and No. 4, and for all utterances are not neighbors the speakers are the same. Again the statement seems quite strong, but such alternation chains are very common in the data, including the novel P&P and Emma.

6.3 Two Rules

Based on what has been discussed, here is a typical speaker alternation pattern, shown below in Figure 6.3 as a summary,

SPEAKER
...
SPEAKER A
SPEAKER B
SPEAKER A
SPEAKER B
SPEAKER A
SPEAKER B
...

Figure 6.3: A Typical Speaker Alternation Pattern

To conclude the above figure in words, here are the two important rules,

1. The speakers of neighbor utterances can not be the same.
2. The speakers of the current utterance and the second previous utterance can have high chance to be the same.

The above two rules are commonly observed in the data, but both rules are not used directly as strict rules, and both rules are learned automatically by using machine learning models. One thing worth mentioning is, it is of course possible that a third speaker will join the dialogue, make a talk suddenly and go away. This behavior interrupts the alternation pattern and provides a noise source for the statistical models. This issue is the only challenge during the utilization of the alternation pattern information, and it is discussed in Appendix A in details, which can serve as another contribution of this work.

6.4 Conclusion

When it is needed to predict the speaker for a given utterance, the neighbor utterances can be of great value, but such implicit information on the relationship between utterances is never used before. This chapter describes the commonly seen speaker alternation pattern with examples and provides two important rules that live on the pattern. This conversational pattern actually exists in any conversational corpus, and to leverage such information is the key to solve the speaker identification problem. In the next chapter, oracle model tries to verify its value by experiments.

Chapter 7

Oracle Model via Ranking

7.1 Introduction

In order to verify the identified speaker alternation pattern is of great use, this work builds an oracle model via SVM ranking. The oracle model assumes that all the neighbor utterances' labels are known so that the system can utilize the neighbor information to make the prediction of the current utterance's speaker. In this chapter the oracle model experiments show that the speaker alternation pattern can significantly improve the performance of the system and also provide a promising direction on how to construct the graphical models. In the following sections, the SVM ranking model is firstly introduced, then the oracle model experiments and the comparison of experiment results with and without the oracle features are described. Finally, the model generalization test on the Emma corpus is discussed and the models are compared.

7.2 SVM Ranking Model

A SVM ranking model is a discriminative model to rank candidates based on confidence scores. SVM Ranking is firstly used in the work (Joachims 2002) for optimizing the search engine with query logs, and later it produces numerous applications in NLP and Information Retrieval fields.

The SVM ranking is similar to the traditional SVM classification, but at least it has with different training data input and a different output from a specially designed ordering function:

- In SVM classification each utterance in the training data is associated with a unique label, and should contains various feature values, but in SVM ranking each utterance in the training set is a set of ordering numbers for multiple candidates. For example,

the following figure exemplifies the difference of the training data input between SVM ranking and SVM classification:

TRADITIONAL SVM	**SVM RANKING**
	3 qid:0 1:1 2:1 3:0 4:0 5:0 6:0 7:0 8:0 9:0 10:0
0 1:1 2:1 3:0 4:1 5:0 6:0 7:0 8:0 9:0 10:0 11:1 12:1 13:0 14:0 15:2 16:3 17:0 18:1	**2** qid:0 1:1 2:1 3:0 4:0 5:0 6:0 7:0 8:1 9:0 10:1
	1 qid:0 1:0 2:1 3:0 4:0 5:0 6:0 7:0 8:0 9:0 10:0

Figure 7.1: The Difference between the Traditional SVM and SVM Ranking

In the above example, the input for SVM ranking given an utterance is actually a set of three different speakers, which have their own feature vectors. The number in bold indicates the speaker id, which is not a part of the feature vector. The output of the ranking model is a set of scores for each character. On the other hand, the traditional SVM has a single feature vector with a single speaker id. Its output is the predicted speaker id.

- In SVM classification the output is a distinct class label for an utterance, while in SVM ranking the output is a set of scores for candidates of that utterance. By using the scores the ranking model can construct a global ordering. Thus if the output score for candidate A is bigger than the score for candidate B, then A has a higher chance to be the speaker of the current utterance than B. One thing needs to be mentioned is that the ordering is always strict, thus for all pairs x and y of the candidates, either $x > y$ or $x < y$.

With the above basic knowledge of the input and output and general behavior of SVM ranking, it is important to get the score values, which are calculated with a global ranking function learned by the SVM ranking model. Suppose the x is a feature vector for a certain candidate,

$$\exists x_1, x_2, F(x_1) > F(x_2) \Leftrightarrow wx_1 > wx_2 \tag{7.1}$$

The goal is to learn this global ranking function $F(x)$, or the weight vector w, to accord with most data pairs in the golden data. The figure below shows how two different weight vectors can rank different candidates for each utterance from (Joachims 2002),

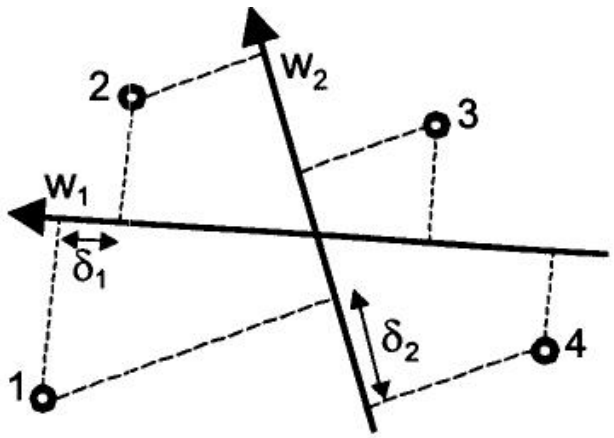

Figure 7.2: An Illustration on How SVM Ranking Ranks Four Points

The points in this figure represent different features, while the weight vector's direction decides the ranking order of the points. Thus just by dot product function, the order can be decided. Then the problem becomes finding the proper weight vector that can fulfill the maximum number of rankings in the training data. However, although the literature concludes that this problem is NP-hard (Höffgen et al. 1995), it is still possible to approximate the solution by using a non-negative slack variables $\xi_{i,j,k}$ and minimizing the upper bound $\sum \xi_{i,j,k}$ as follows,

$$\begin{aligned}
\text{minimize} \quad & V(w,\xi) = \frac{1}{2} ww + C \sum \xi_{i,j,k} \\
\text{subject to} \quad & \forall (d_i, d_j) \in r_1^* : w\Phi(q_1, d_i) \geq w\Phi(q_1, d_j) + 1 - \xi_{i,j,1} \\
& ... \\
& \forall (d_i, d_j) \in r_n^* : w\Phi(q_n, d_i) \geq w\Phi(q_n, d_j) + 1 - \xi_{i,j,n} \\
& \forall i \forall j \forall k : \xi_{i,j,k} \geq 0
\end{aligned} \tag{7.2}$$

In the meanwhile, by rearranging the above constraints as:

$$w(\Phi(q_1, d_i) - w\Phi(q_1, d_j)) \geq 1 - \xi_{i,j,k} \tag{7.3}$$

Then it becomes obvious that this problem is ultimately similar as a SVM classification problem based on pairwise difference vectors $x_1 - x_2$. That is why an existing traditional SVM model can be utilized to solve the problem. The pairwise difference vectors also look interesting because for each candidate data line in the training set, the SVM ranking model needs to do the "minus" operation to obtain the difference between features, that resulting features vector differences can make great sense mathematically and ultimately lead us to a ranking order. The SVM ranking model can surely be extended to higher dimensional

space with kernel tricks, however, the NLP applications usually favor the SVM with the linear kernel.

7.3 Oracle Model

An oracle model is constructed with the SVM ranking model, and the only difference between the oracle model and non-oracle model is whether the system includes the neighbor information during prediction. The oracle model adds 4 or 8 extra feature slots to represent such extra information in each candidate's feature vector of the SVM ranking input.

A graph can be used to represent a simple non-oracle model used in the experiments. Figure 7.3 below shows a simple multi-class classification model without the usage of the extra neighbor information, where the node of the lower position is the feature vector of the current utterance, and the node of the upper position is the speaker label that needs to be predicted for this utterance. One simple local model does not have the information from its neighbor labels, and essentially makes a single multi-class classification and pick a speaker from a candidate pool, which is the full list of speakers in the novel:

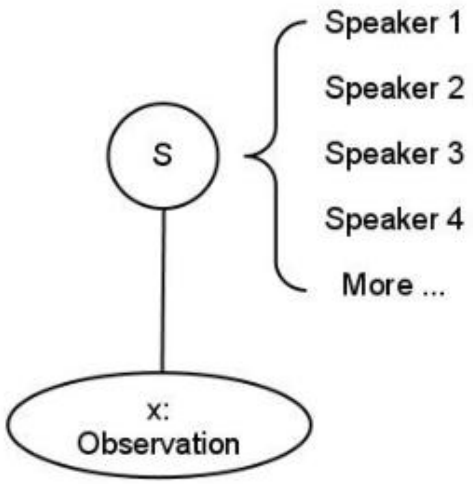

Figure 7.3: Single Multi-class Classification and Pick a Speaker from the Pool

In terms of how the model predicts a speaker label, SVM ranking can be used, but it is optional and other machine learning methods can also be applied here.

However, as a comparison Figure 7.4 below shows an oracle model with the information of neighbor utterances. In this graph there are a total of five continuous utterances in a dialogue: each of them contains a speaker label node (in the upper position) and a feature vector node (in the lower position). The oracle model makes a prediction based on not only

the current utterance's features, but also neighbor utterances' speaker labels (with label link connections shown below),

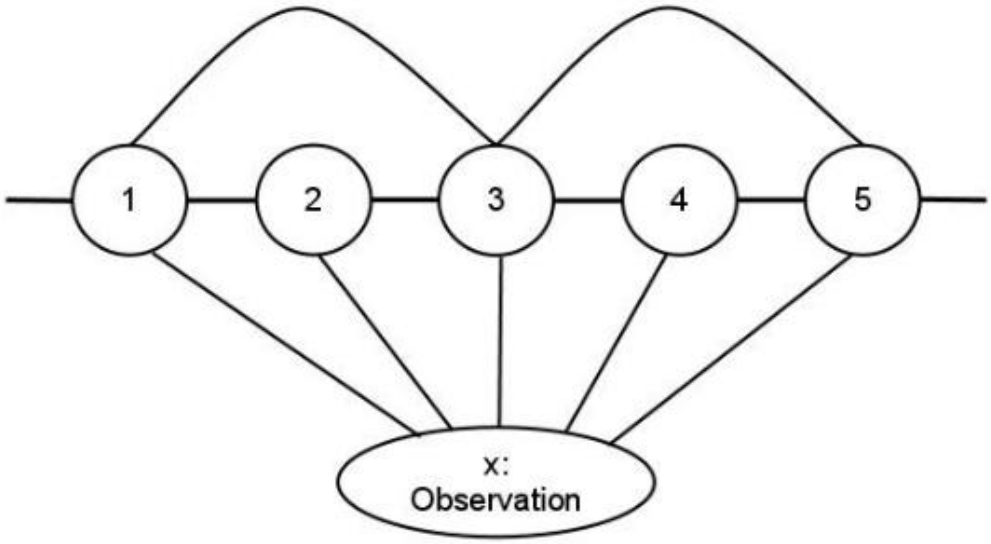

Figure 7.4: A Multi-class Model that Considers its Neighbor Information

In the above figure, the speaker of the utterance 3 in the middle is to be predicted, the oracle model assumes that it already knows the labels of its neighbor utterances, 1, 2, 4 and 5 because of the four link connections as shown above. Thus when it comes to the feature representation, 4 extra slots in the feature vector will be added and the value of each slot is the speaker label. This work also tries the 8 slots version, which means the oracle model knows up to 4 neighbor speakers' labels beforehand near the current utterance.

It is obvious if the extra few slots are included, the resulting extra neighbor information can be encoded in the model based on the speaker alternation pattern. Once an oracle model is built, it is important to verify whether or not the neighbor speakers' information is helpful for the prediction of the speakers. In the next section experiments are described with results shown.

7.4 Experiments

In order to verify the performance changes, a total of three sets of experiments are done. And each set comes with the result comparison between normal model (without neighbor information) and the oracle model. Both models are tested 1) on the P&P's randomly picked set; 2) with the 10-fold cross validation scheme on the whole P&P data; and 3) on the Emma novel data as the generalization test, and the Emma data is extracted from the

Columbia corpus which is described in Chapter 3. In those experiments, only the SVM ranking model is used, because the ranking model can provide the best results. One more thing worth mentioning is, although the local normal model cannot directly use the neighbor speakers' label, but in order to increase the accuracy, three neighbor utterances' feature vectors are combined together with the current utterance' features as an indirect usage of the neighbor information (the previous continuous utterance, the second continuous previous utterance and the next continuous utterance). The combination scheme can increase the overall performance of the system.

Here are some facts about the P&P and Emma data:

CATEGORY	NUMBER OF UTTERANCES
P&P Test Set: Chapter 19-26	126
Emma Whole Data	397

Figure 7.5: P&P and Emma Data Facts

7.4.1 On the Test Set of P&P

The experiments follow the standard procedure, first of all based on the training set, the development set is used as testing data to obtain the best possible parameters, during this step, the test set is independently collected and has no relationship with other data sets in order to avoid training overfitting. The second step is to use the obtained parameters to test on the test set. The test set used for this experiment is from the P&P, in which there are a total of 61 chapters. This work randomly divides the book into three parts. The test set has about 10% of the total number of utterances extracted only from Chapter 19 to Chapter 26, the development set has another 10% from Chapter 27 to Chapter 33, and finally the training set includes all the remaining chapters which are about 80% of the total utterances.

Here are the final results on the test set from Chapter 19 to Chapter 26, with normal local model and the oracle model.

MODEL	ACCURACY
Normal model	82.7%
Oracle model with 8 neighbors	86.6%

Figure 7.6: Experiment Results on the Test Set of the P&P

The results clearly show that a significant increase can be obtained with oracle model.

7.4.2 Cross Validation on the P&P Data

This thesis also conducts a 10-fold cross validation experiment on the whole P&P corpus. 10-fold CV can help me identify the overall performance on the data set. The whole P&P data is divided into 10 different smaller parts, the models are trained on 90% of the whole corpus, and are tested on another 10% of the data.

Here are the results for both oracle model and normal local model:

MODEL	**ACCURACY**
Normal 10-fold CV	80.2%
Oracle 10-fold CV	91.1%

Figure 7.7: Experiment Results with 10-fold CV on the Whole P&P Data

The accuracy is calculated by averaging all folds' results. The final results again clearly show that a significant increase can be obtained with oracle model.

7.4.3 Generalization Test on Emma Data

The generalization test on Emma data is different than the two previous sets of experiments, mainly because the training and testing are in different domains. This set of experiments uses the whole P&P data set as the training corpus, and use the Emma data from the Columbia corpus as the testing set. Training and testing on cross domain data can be usually challenging because the fact of cross domain model application can lead to data sparsity and poor coverage problems in features, thus the system is not able to perform as good as the one with the same domain data.

Here are the results for both oracle model and local model on the Emma data from the Columbia corpus,

MODEL	**ACCURACY**
Normal Generalization	74.4%
Oracle Generalization	80.5%

Figure 7.8: Experiment Results with the Generalization Test on the Emma Data

The accuracy drop can be observed by roughly 6% when the model is applied to the cross domain data. Despite this, once again the results show a significant increase with the oracle model, compared with the local model.

7.5 Model Comparison

Because the generalization test data on the *Emma* book is obtained from the Columbia corpus, it is necessary to compare the results from our ranking model with the model from (Elson and McKeown 2010) (EM model, see Chapter 2).

However, there are several difficulties to directly make such comparison, mainly because the data format is different (see Chapter 3.3), and also the *Emma* data is only one part of the whole Columbia corpus. Fortunately the work in the (Elson and McKeown 2010) divides all utterances in the Columbia corpus into a total of 7 different categories, such as quote-said-person category, quote-alone category. Among all categories, the most difficult one is the quote-alone category, thus it is reasonable to compare both models based on the results obtained on this category. Here is the comparison below,

MODEL	**ACCURACY**
Ranking Model on Quote-Alone Category	62.9%
EM Model on Quote-Alone Category	63%
Ranking Model Generalization Test on Emma Data	74.4%
Ranking Oracle Model Generalization Test on Emma Data	80.5%
EM on Columbia Corpus	83%

Figure 7.9: Model Comparison

From the above results both models have similar accuracy level on the quote-alone category, but given that our ranking model is with the generalization test, the ranking model is superior than the EM model.

7.6 Conclusion

This chapter mainly focuses on a few important things: the SVM ranking model, the oracle model and the experiments. Overall, the value of the speaker identification pattern is verified by experiment results. In the final section the model comparison is also conducted. Based on the promising direction obtained from the oracle model, the local graphical models can be constructed with new experiments, which are discussed in Appendix A.

Chapter 8

Conclusion

In this thesis, the issues of the automatic speaker identification in novels are examined. In particular, this thesis examined the SVMranking model for the oracle model by making use of an important speaker alternation pattern in the data. This work provides a promising direction on how to utilize the implicit relationship information in the speaker alternation pattern between utterances.

This work is the first to utilize the speaker alternation pattern for the speaker identification problem. While the previous methods only focus on a simple multi-class model, the approaches of this work treat the speaker identification problem as a structured prediction problem, thus the rich context information can be utilized within the dialogue model frame. Finally, a novel vocative feature is used, and it can be extracted by training a Logistic Regression model.

The positive results from the oracle model verify the value of the speaker alternation pattern with an accuracy increase of at least 4-10%. Inspired by the oracle model, the high order feature functions in local graphical models are designed and with the break link scheme a significant improvement is observed over a simple linear chain model and baseline method (see Appendix A). Both statistical models, together with the dialogue model, provide acceptable solutions to the three challenges discussed in Chapter 1.

8.1 Future Work

The ultimate goal of the speaker identification problem, as mentioned in Chapter 1, is to train the identification system on a few annotated novels and then to be applied to new unannotated novels for real usage, because the annotated data resource is always expensive to obtain. Although the ranking model can somehow deal with this challenge, current graphical models still cannot completely solve the generalization problem because of the

speaker tags, because the CRF models are only able to deal with the fixed tag scheme. One possible method to deal with this issue is to combine both models together, and to take advantage of the generalization aspect of the SVMranking model, and also the relationship utilization aspect of the CRF models.

Another major error source for the speaker identification system is the lack of the predictive information, a few attempts can be tried to utilize the web knowledge to include new knowledge source. For example, based on the Wikipedia corpus, utterances can be measured by the content similarity, given that same speaker tends to talk about similar topics.

Appendix A

Local Models

A.1 Introduction

This chapter describes our various attempts to utilize the relationship information to solve the speaker identification problem with graphical models. Based on the fact that utterances have strong relationship between each other in the novel data, the speaker identification problem can be treated as a structured prediction problem. Such relationship information is critical to help deal with the lack of predictive information challenge mentioned in Chapter 1. This chapter starts with an important concept of graphical models, the factor graph (Clifford 1990, Kschischang et al. 2001, Koller and Friedman 2009), and then the experiments are described.

A.2 Factor Graph and Feature Function

Factor graph (Bishop 2006, Klinger and Tomanek 2007) is not the same as the general graphical structure of the graphical models, and is a convenient way to represent different feature functions. Based on the conditional independence assumption (Dawid 1979), the factor graph for a CRF model can be shown below,

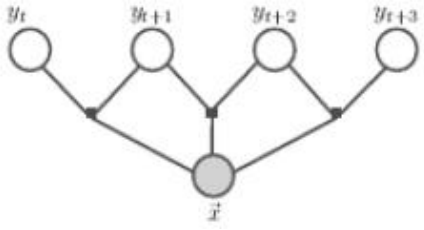

Figure A.1: Linear-chain CRF in Factor Graph Representation

In the above figure, the $\vec{x}$ is the observation, y_t, y_{t+1}, y_{t+2} and y_{t+3} are the labels and

are conditionally independent. The idea of decomposing complicated general graph into a set of independent factors makes the model much less complicated. The equation for the above factor graph is shown below,

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{c \subseteq C} Factor_c(\vec{x}_c, \vec{y}_c) \tag{A.1}$$

The independent factors are multiplied together, and the coverage of each factor node is now only limited to its connected variables. Within each factor node, there are a bunch of feature functions manually defined, the feature functions are crucial on the effectiveness of the system, and that is the reason factors are important. The equation is shown below,

$$Factor_c(\vec{x}_c, \vec{y}_c) = \exp(\sum_{i=1}^{m} w_i f_i(y_{i-1}, y_i, \vec{x}, i)) \tag{A.2}$$

From the above two equations, the final equation for the linear-chain CRF is,

$$p(\vec{y}|\vec{x}) = \frac{\prod_{i=1}^{n} \exp(\sum_{i=1}^{m} w_i f_i(y_{i-1}, y_i, \vec{x}, i))}{\sum_{\vec{y} \subseteq Y} \prod_{c \subseteq C} Factor_c(\vec{x}_c, \vec{y})} \tag{A.3}$$

During training, the above CRF function tries to obtain the best weight vector $\vec{w}$. The details on how the training works can be found in (Lafferty et al. 2001, Wallach 2004, Sutton and Mccallum 2006).

As indicated by previous work in (Keerthi 2007), the key of obtaining the best performance of CRF models lies in the selection of proper feature functions (Qi and Chen 2010, Sutton and Mccallum 2007). The feature function can be changed whenever there is a good reason for doing that. Suppose there is a training data item $\vec{x}_i = x_1...x_m$, where m is the number of features, and the corresponding label y_i, at each index i a list of possible feature functions can be created manually, including:

- $f(y_i)$, the simplest form of feature function, this means the label frequency should be used as one of the contributing factors for CRF models to pick labels.
- $f(y_i, \vec{x}_i)$, this function requires the current label is directly related to its own features.
- $f(y_{i-1}, y_i)$, the current label and the previous label are related. This simple function tries to utilize the implicit relationship between neighbor labels.
- $f(y_{i-2}, y_i)$, this function indicates that the model will use the relationship between the second previous label and the current label to make predictions.

- $f(y_{i-1}, y_i, \vec{x}_{i-1})$, a tri-token relationship, the current label, the previous label and all the previous features are related.

- $f(y_{i-2}, y_i, \vec{x}_i)$, 2nd previous label, current label and current features are related.

The above list provides examples of possible feature functions. The combination of choosing different feature functions can be crucial on the performance of the speaker identification system. For language problems, the relationship information can usually be obtained from linguistics theories, syntactic rules and dialogue discourse principles. One typical example in NLP research is the application on semantic role labeling systems (Haghighi 2005, Cohn and Blunsom 2005), for which the successfulness of feature function development depends on whether the implicit relationship in the language data can be identified.

A.3 Experiments

This section focuses on the experiments to solve the speaker identification problem with graphical models, which are mainly inspired by the two rules mentioned in Chapter 6.3 from the speaker alternation pattern. Two rules are, as a review,

1. The speakers of neighbor utterances cannot be the same.

2. The speakers of the current utterance and the second previous utterance, and the second next utterance, can have high chance to be the same.

In terms of the experimental settings, all experiments are carried out on the data set of the novel P&P, both the training set and testing set are the same as the section 7.4.1.

A.3.1 Linear-Chain CRF

In the hope of implementing the rule No.1, this thesis builds a linear-chain CRF, which is able to output a label sequence where the two neighbor utterances are not the same for most of the cases. The feature functions used for this model include,

1. $f(y_i, \vec{x}_i)$

2. $f(y_{i-1}, y_i)$

3. $f(y_i)$

4. $f(y_{i-1}, y_i, \vec{x}_{i-1})$

The corresponding factor graph for those feature functions can be viewed in Figure A.2. In order to make comparisons, a log linear model is also constructed without the implementation of the No.1 rule, which means this naive model does not have the second and fourth feature function listed above. In the meanwhile, the thesis also reports a baseline method result, which is similar to the work (Elson and McKeown 2010) and calculates the frequency when the characters, who are closest to utterances in terms of word distance, happen to be the speakers of those utterances.

The experiments run the model with *PocketCRF* (Xian 2010), a toolkit for building standard linear-chain CRF model, and also *Factorie* (McCallum et al. 2009, Wick et al. 2010; 2009), a toolkit to build arbitrary graphical model. Both models are different in terms of the inference methods they used. Results are shown below,

MODEL	**ACCURACY**
Linear-Chain CRF with *PocketCRF*	62.7%
Linear-Chain CRF with *Factorie*	62.7%
Naive model without links for neighbor labels	59.5%
Rule-based baseline	42.0%

Figure A.2: Experiment Results on the Test Set of the P&P

From the above comparison, the No.1 rule can increase the accuracy by least 3%.

A.3.2 Higher Order CRFs

The rule No.1 can be implemented when the neighbor labels in the model are connected together, it is then a natural idea to implement the rule No.2 with a higher order linear-chain CRF, which looks similar to the one shown below,

The above graph structure can capture the dependencies that label 1 and label 3, label 2 and label 4 might be identical. Thus this work constructs a few higher order CRF models with different combinations of feature functions to compare their performances,

CRF A Is 2 order, with links between neighbor labels, between current label and the second next label, between current label and the second previous label, and between current label and features. Any models that are more complicated than this model cannot be run with *PocketCRF* or other popular CRF toolkits, because the searching space grows exponentially with more links.

CRF A+ Contains extra links between current label and the third next label, and the third previous label, and also all links in the CRF A+ model.

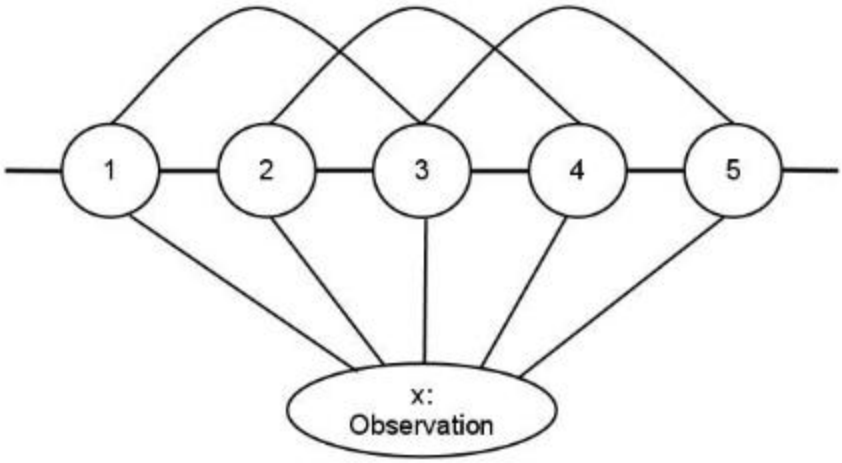

Figure A.3: Higher Order CRF in Graph Representation

CRF A++ Contains extra links between current label and the fourth next label, and the forth previous label, and also all links in the CRF A++ model.

The level of the model complexity rises from CRF A to CRF A+, then to CRF A++. In order to implement the higher order CRFs, a bunch of feature functions are used,

1. $f(y_i)$
2. $f(y_i, \vec{x}_i)$
3. $f(y_{i-1}, y_i)$
4. $f(y_{i-2}, y_i)$
5. $f(y_{i-2}, y_i, \vec{x}_{i-2})$
6. $f(y_{i-3}, y_i)$
7. $f(y_{i-4}, y_i)$
8. $f(y_{i-1}, y_i, \vec{x}_{i-1})$

It can be observed that based on the same feature sets, the accuracies increase with new links. Thus long dependencies between labels can be useful, and the relational information can also be utilized by higher order CRF models.

MODEL	ACCURACY
Higher order CRF A with *PocketCRF*	63.5%
Higher order CRF A with *Factorie*	65.8%
Higher order CRF A+	67.4%
Higher order CRF A++	68.2%
Linear-Chain CRF	62.7%
Naive model without links for neighbor labels	59.5%

Figure A.4: Experiment Results with Higher Order CRFs

A.3.3 Arbitrary Structure CRFs

Although No.2 rule is helpful and the long range dependencies can increase the performance of the system, there is still one problem: No.2 rule could be noisy.

For example, suppose there are speaker A and speaker B in one dialogue, they are talking one by one with the typical speaker alternation pattern, suddenly there is a third speaker C who joins this dialogue and starts talking, at this moment such interruption changes the conversing pattern and provides a noise source for the CRF model. In this case higher order links in the high order CRFs can have negative impact on the performance. In order to deal with such noises, an arbitrary structure CRF model is built,

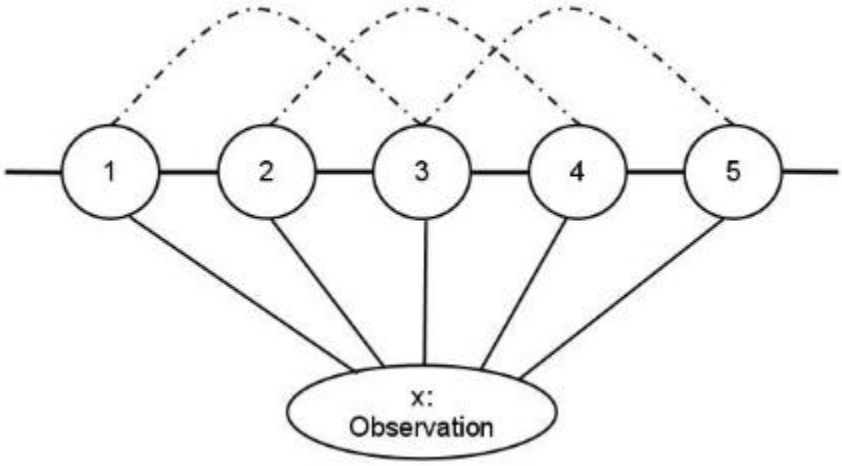

Figure A.5: Arbitrary CRF Representation

The difference between this model and the previous high order CRF models is on the long range links between current label and the second/fourth neighbor labels. In the previous high order CRFs, the links in the model are always connected, but in the arbitrary structure CRF model, the links between label 1 and label 3, or label 2 and label 4 can be broken, in other words, whether they are connected or not depends on situations. Thus the

following *conditional* feature functions can be used,

- $f(y_{i-2}, y_i)$, only if the gender of y_i and y_{i-2} is not different, otherwise it will be removed from the factor.
- $f(y_{i-4}, y_i)$, only if the gender of y_i and y_{i-2} is not different, otherwise it will be removed from the factor.

This work builds a set of CRF models with the break link scheme. In order to make direct comparisons, the break link scheme is added directly to the existing CRF A, CRF A+ and CRF A++ models. The results are shown below,

MODEL	**ACCURACY**
CRF A	65.8%
CRF A w. Breaklink	69.8%
CRF A+	67.4%
CRF A+ w. Breaklink	68.2%
CRF A++	68.2%
CRF A++ w. Breaklink	73.8%
Linear-Chain CRF	62.7%

Figure A.6: Experiment Results with Arbitrary Structure CRFs

The results show that the break link scheme can increase the accuracy significantly. By breaking unnecessary links (useless dependencies) between speaker labels, the speaker alternation pattern noises can be reduced, given that there is about 40% of the utterances in the data of which the gender information can be identified. Another reason is about the data sparsity, the removal of useless links reduce the data sparsity for training, thus CRF models cannot be confused by the overwhelming features and can get the most important information without much efforts.

So far the high order links in CRFs are always beneficial, no matter what the data is and what the model is the system can always increase the accuracy with new links because of the strong pattern in the data. The break link scheme is helpful for most of the times, and its effectiveness also more or less depends on other feature functions' combination.

A.4 Other Attempts

This section will describe a few of our attempts, although they are not really working but the ideas and learning opportunities can be extremely helpful for me.

- Research shows that the log-linear models (CRFs) and the maximum margin based models (structSVMs) are successful to solve the structured prediction problems (Nguyen and Guo 2007). A model called Maximum Margin Markov Model (M3N) in the work (Taskar et al. 2003) combines the advantages of both sides. However, the downside of this model is its efficiency, this model can be extremely slow, thus this limits the further attempts. Based on the experiments, the accuracy level of M3N is about the same as the linear-chain CRF model.

- The second attempt is to break down the whole task into smaller ones. Inspired by the work in semantic role labeling, a pipeline system can usually obtain better results than an integrated system. A model is built to predict the gender information first, once this is done then such extra gender information is utilized for the construction of the arbitrary structure CRFs. However, even though the accuracy on the gender prediction is high, the incorporation of such predicted gender information is not helpful for the overall system, because experiments found that the wrong gender prediction of the utterance is usually the wrong speaker prediction utterance, that is why a wrong gender information is not going to help predict the correct speaker of that utterance.

- The third one is about multi-pass scheme with SVMranking. The idea is from the oracle model, if the neighbor labels can be obtained correctly, then simply by using an oracle-like model the accuracy can be much higher because of the speaker alternation pattern. Thus a ranking model is built by firstly generating a sequence of labels, then those labels are inserted as oracle neighbor features. Then the multi-pass running is carried out, with the hope that the results obtained in later rounds can be better than the first round. However, the final results are not improved, the reason is similar to the previous attempt on the gender extraction. Because the mistakes can propagate, such erroneous information cannot be removed completely with the multi-pass scheme.

A.5 Conclusion

This chapter discusses the possibility of incorporating speaker alternation pattern with graphical models. The experiments and results provide promising directions on the utilization of the alternation pattern. Based on the experiments, the high order feature functions and the break link scheme with CRF models can significantly reduce the data noise and improve the performance of the system.

Appendix B

Relationship Extraction

As mentioned in Chapter 1, one of the important application for the speaker identification system is to extract the social network from novels. By using our system, a part of the relationship information between different characters can be obtained. In general, the construction of the social network follows two steps as below:

Step 1 The initial relationship information is obtained from the speaker identification results. For example, here is one conversation from Chapter 1 in the P&P novel,

> ...
> "*How so? how can it affect them?*"
> "*My dear Mr. Bennet,*" **replied his wife**, "*how can you be so tiresome! You must know that I am thinking of his marrying one of them*."

With the fact that the first utterance is uttered by *Mrs. Bennet* and the second is uttered by *Mr. Bennet*, because there is a clue *replied his wife* in the second utterance, it can then be concluded with high confidence that *Mrs. Bennet* and *Mr. Bennet* could be a couple, they have wife-husband relationship. In total, a set of 28 relationship are identified automatically with step 1.

Step 2 In previous step, a list of relationship can be extracted, but the list is not complete. By inserting a set of rules into the SQL database more relationship between characters can be generated. A total of 30 rules are used in the extraction systems, part of them are shown below,

1. Father(A,B) + Father(A,C) + Couple(A,D) = Mother(D,B) + Mother(D,C)
2. Mother(A,B) + Mother(A,C) + Couple(A,D) = Father(D,B) + Father(D,C)
3. Mother(A,B) + Mother(A,C) + Female(B) = Sister(B,C)

4. Mother(A,B) + Mother(A,C) + Male(B) = Brother(B,C)
5. Father(A,B) + Father(A,C) + Female(B) = Sister(B,C)
6. Mother(A,B) + Mother(A,C) + Male(B) = Brother(B,C)
7. Wife(A,B) = Husband(B,A)
8. Husband(A,B) = Wife(B,A)
9. Niece(A,B) + Male(B) = Uncle(B,A)

Here is an example for a part of the social network extracted from the P&P novel by our system.

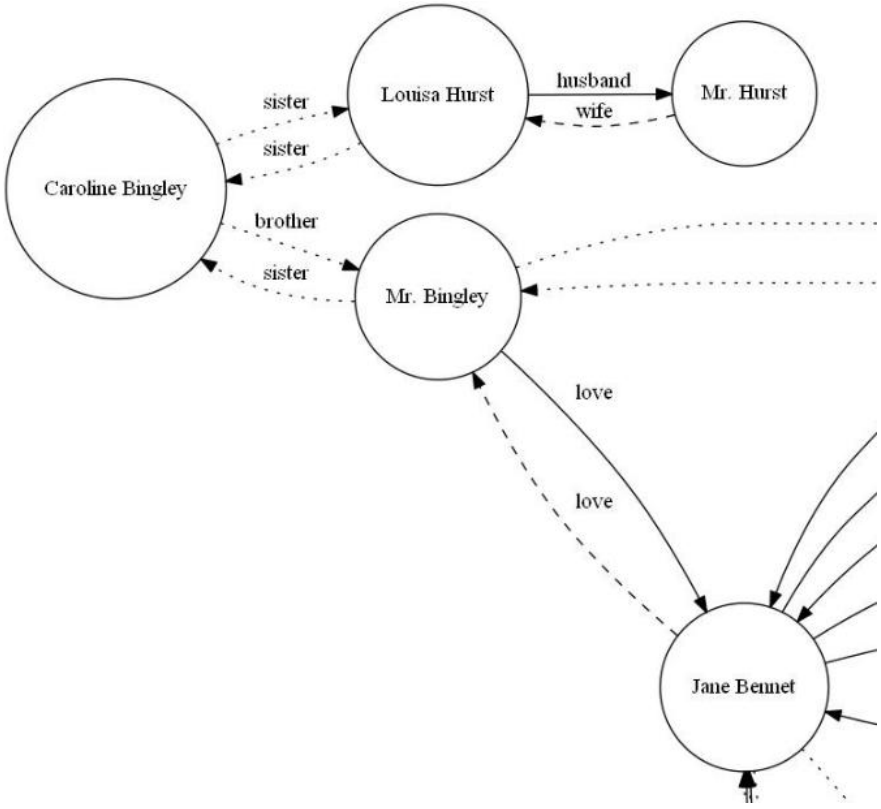

Figure B.1: Social Network Construction

There are a total of 113 relationship extracted, among them 28 (dotted edges) are directly from the speaker identification results and 27 (dashed edges) are from annotation results. The rest 58 (solid edges) relationship are populated automatically by using the rules.

The dotted edges are used to represent the relations directly from the speaker identification results. The dashed edges are directly from the annotation results and the relations

automatically generated by rules are represented in solid edges. The whole social network of the P&P novel is shown below,

Figure B.2: Social Network of the P&P Novel

Bibliography

Alan Agresti. *Building and Applying Logistic Regression Models*, pages 137–172. John Wiley & Sons, Inc., 2006. ISBN 9780470114759.

J. Maxwell Atkinson and John Heritage. *Structures of Social Action: Studies in Conversation Analysis*. Cambridge University Press, 1984.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 0387310738.

Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The actor-topic model for extracting social networks in literary narrative. In *Proceedings of the NIPS 2010 Workshop - Machine Learning for Social Computing*, page 7 pp, 2010.

P. Clifford. Markov random fields in statistics. 1990.

Trevor Cohn and Philip Blunsom. Semantic role labelling with tree conditional random fields. In *In Proceedings of CoNLL-2005*, pages 169–172, 2005.

A. P. Dawid. Conditional Independence in Statistical Theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(1):1–31, 1979. ISSN 00359246.

David K. Elson and Kathleen McKeown. Automatic attribution of quoted speech in literary narrative. In *AAAI*, 2010.

David K. Elson, Nicholas Dames, and Kathleen McKeown. Extracting social networks from literary fiction. In *ACL*, pages 138–147, 2010.

Sarah Favre, Alfred Dielmann, and Alessandro Vinciarelli. Automatic role recognition in multiparty recordings using social networks and probabilistic sequential models. In *ACM Multimedia*, pages 585–588, 2009.

Kevin Glass and Shaun Bangay. Hierarchical rule generalisation for speaker identification in fiction books. SAICSIT '06, pages 31–40, Republic of South Africa, 2006. South African Institute for Computer Scientists and Information Technologists.

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*, 2007.

Aria Haghighi. A joint model for semantic role labeling. In *Proceedings of CoNLL 2005 shared task*, 2005.

Klaus-U. Höffgen, Hans-U. Simon, and Kevin S. van Horn. Robust trainability of single neurons. *J. Comput. Syst. Sci.*, 50:114–125, February 1995. ISSN 0022-0000.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X.

Sathiya Keerthi. CRF versus SVM-Struct for Sequence Labeling. Technical report, Yahoo Research, 2007.

Roman Klinger and Katrin Tomanek. Classical probabilistic models and conditional random fields. 2007.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009. ISBN 9780262013192.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, February 2001. ISSN 00189448.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

B. Levin and M.R. Hovav. *Unaccusativity: at the syntax-lexical semantics interface*. Linguistic inquiry monographs. MIT Press, 1995. ISBN 9780262620949.

Andrew McCallum, Karl Schultz, and Sameer Singh. Factorie: Probabilistic programming via imperatively defined factor graphs. pages 1249–1257, 2009.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3: 235–244, 1990.

Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. Nonparametric latent feature models for link prediction. In *Proceedings of the NIPS*, 2009.

Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 681–688, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

Luole Qi and Li Chen. A linear-chain crf-based learning approach for web opinion mining. In *Proceedings of the 11th international conference on Web information systems engineering*, WISE'10, pages 128–141, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-17615-1, 978-3-642-17615-9.

G. Redeker and M. Egg. Says who? on the treatment of speech attributions in discourse structure. In *Proceedings of Constraints in Discourse.*, page 7 pp, 2006.

Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas L. Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Trans. Inf. Syst.*, 28(1), 2010.

Hugues Salamin, Alessandro Vinciarelli, Khiet Truong, and Gelareh Mohammadi. Automatic role recognition based on conversational and prosodic behaviour. In *ACM Multimedia*, pages 847–850, 2010.

Luis Sarmento and Sergio Nunes. *Automatic extraction of quotes and topics from news feeds*. 4th Doctoral Symposium on Informatics Engineering, 2009.

C. Sutton and A. Mccallum. An Introduction to Conditional Random Fields for Relational Learning. In *Introduction to Statistical Relational Learning*. MIT Press, 2006.

Charles Sutton and Andrew Mccallum. *An Introduction to Conditional Random Fields for Relational Learning*. 2007.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *NIPS*, 2003.

Hanna M. Wallach. Conditional random fields: An introduction. 2004.

Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew Mccallum. Samplerank: Learning preference from atomic gradients. In *the NIPS workshop on Advances in Ranking*, 2009.

Michael L. Wick, Andrew McCallum, and Gerome Miklau. Scalable probabilistic databases with factor graphs and mcmc. *CoRR*, abs/1005.1934, 2010.

Qian Xian. A CRF toolkit for sequence labeling tasks in natural language processing. Technical report, 2010. URL `http://code.google.com/p/pocketcrf/`.