Distribution Agreement

Yu-Hsin (Henry) Chen	April 7, 2017

Character Identification on Multi-party Dialogues

by

Yu-Hsin (Henry) Chen

Jinho D. Choi

Advisor

Department of Mathematics and Computer Science

Jinho D. Choi

Advisor

James J. Lu

Committee Member

Heather Julien

Committee Member

2017

Character Identification on Multi-party Dialogues

By

Yu-Hsin (Henry) Chen

Jinho D. Choi

Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2017

Abstract

Character Identification on Multi-party Dialogues

By Yu-Hsin (Henry) Chen

This thesis introduces a subtask of entity linking, called character identification, that maps mentions in multiparty conversation to their referent characters. Transcripts of TV shows are collected as the sources of our corpus and automatically annotated with mentions by linguistically-motivated rules. These mentions are manually linked to their referents and disambiguate with abstract referent labels through crowdsourcing. Our annotated corpus comprises 448 scenes from 2 seasons and 46 episodes of the TV show *Friends*, and shows the inter-annotator agreement of $\kappa = 79.96$. For statistical modeling, this task is reformulated as coreference resolution, and experimented with two state-of-the-art systems on our corpus. A novel mention-to-mention ranking model is proposed to provides better mention and mention-pair representations learned from feature groupings of dialogue-specific features After linking coreferent clusters to their referent entity with our proposed rule-based remapping algorithm, the best model gives a purity score of 57.27% on average, which is promising given the challenging nature of this task and our corpus.

Character Identification on Multi-party Dialogues

By

Yu-Hsin (Henry) Chen

Jinho D. Choi

Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2017

Acknowledgements

I would like to first acknowledge my parents in Taiwan. Tseng-Chou Chen, my father, and Sao-Fan Ma, my mother, have made possible my four years at Emory University. They have selflessly supported me both financially and mentally in order for me to success in college. I am extremely appreciative for their efforts. Thanks to them, I was able to study Computer Science and enjoy my academic experiences with no concern. Without them, I will not have the opportunity to study aboard in the States at Emory University.

I also want to acknowledge my advisor and mentor in college, Jinho D. Choi. I was fortunately enough to meet him and take his first class at Emory during my sophomore year. He welcomed me to his lab and granted me the privilege of doing research under him. He introduced me to the field of Natural Language Processing, of which I am now deeply passionate in. I am grateful to Jinho's guidances and resources that have made my research and this thesis possible.

Contents

List of Figures

List of Tables

Chapter 1

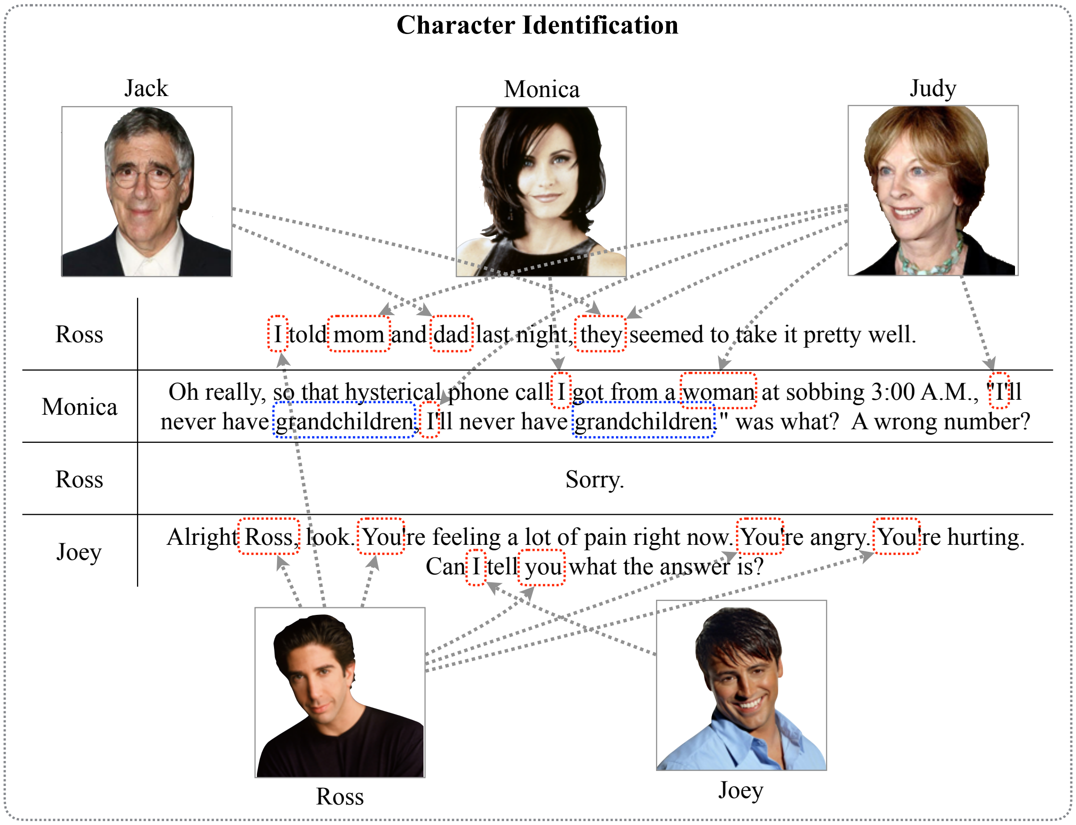
Introduction

1.1 Task definition

In this work, we introduce an new entity linking task, called Character Identification [5]. Character identification is a task of mapping each mention in context to one or more characters in a knowledge base. It is a subtask of entity linking; the main difference is that mentions in character identification can be any nominals indicating characters (e.g., *you*, *mom*, *Ross* in Figure 1.1), whereas they are mostly related to Wikipedia entries in entity linking [21]. Furthermore, character identification allows plural or collective nouns to be mentions such that a mention can be linked to more than one character. The characters can either be pre-determined, inferred, or dynamically introduced; however, a mention is usually linked to one pre-determined entity for entity linking.

The context can be drawn from any kind of documents where characters are present (e.g., dialogues, narratives, novels). Our work focuses on context extracted from multiparty conversation, especially from transcripts of TV shows. Entities, mainly the characters in the shows or the speakers in conversations, are predetermined due to the nature of the dialogue data.

Knowledge base regarding entities can be either pre-populated or populated from context. For the example in Figure 1.1, all the speakers can be introduced to the knowledge base without



Character Identification

Ross	I told mom and dad last night, they seemed to take it pretty well.
Monica	Oh really, so that hysterical phone call I got from a woman at sobbing 3:00 A.M., "I'll never have grandchildren, I'll never have grandchildren" was what? A wrong number?
Ross	Sorry.
Joey	Alright Ross, look. You're feeling a lot of pain right now. You're angry. You're hurting. Can I tell you what the answer is?

Figure 1.1: An example of character identification.

All three speakers are introduced as characters before the conversation (Ross, Monica, and Joey), and two more characters are introduced during the conversation (Jack and Judy). The goal of this task is to identify each mention as one or more of these characters.

reading the conversation. However, certain characters, mentioned during the conversation but not the speakers, should be dynamically added to the knowledge base (e.g., Ross' mom and dad). This is also true for many real-life scenarios where the participants are known prior to conversations, but characters outside of the participants are mentioned during the conversation.

The task of coreference resolution disregards the entity assignments of mentions. It targets at linking mentions to the correct antecedents [8, 47, 35]. A linked grouping could imply a connection between its mentions and some unknown or abstract entity. For example, given a sentence “I just bought a car, and I love it", a coreference resolution system should link “I" to “I" and “it" to “car". Character identification is distinguished from coreference resolution because mentions are linked to global entities in character identification whereas they are linked to others without considering any global entities in coreference resolution.

The task of entity linking, such as Wikification, primarily emphasizes on disambiguating the referred entity of mentions in discourses [22, 12]. For instances, given a sentence “Emory is located in Atlanta, Georgia", a system should identify “Emory" as Emory University rather than the American Methodist bishop, John Emory. Character identification is harder than typical entity linking duo to the rapid and frequent contexts switch of topics in dialogues.

In this work, mentions of plural or collective nouns are discarded, and knowledge base does not get populated from context dynamically. Adding these two aspects will greatly increase the complexity of this task, which we will explore in the future.

1.2 Motivation

The motivation for introducing and researching on the task of character identification is twofold. First, as one of the main targeted challenges in natural language processing [40, 16, 18], machine comprehension aims to provide syntactic- and semantic-rich information for the better understanding of natural language text. Though the latest approaches have shown great promises, most of them still face difficulties in understanding and synthesizing information scattered across different parts of documents. Reading comprehension in dialogues is particularly hard due to not only speakers switches but also context switches during conversations. Thus, it is necessary to learn the connections between mentions within and across utterances in order to derive meaningful inferences.

Second, character identification will serve as a stepping stone to a bigger task called

Character Mining [5]. Character mining will be a extended task that utilizes the results of character identification. It focuses on extracting information and constructing knowledge base associated with particular characters or any personal entities in contexts. A knowledge base can be seemed as a collection of either structured or unstructured information that may or may not encompass multiple ideas [14], and character-centric knowledge base is the desired result from the task of character mining. The task can be subdivided into three sequential tasks, character identification, attribute extraction, and knowledge base construction. The knowledge base generated could be highly applicable and beneficial to provide entity-specific information and facilitate other systems, such as question answering and dialogue generation. Therefore, character identification is essential to the success of character mining. With linking information between mentions and characters identified, it would be possible to synthesize and infer information across contexts regarding specific characters.

Through our investigation in character identification on multiparty conversations, we hope to assess the feasibility of the task and tackle an unexplored yet crucial branch of machine comprehension. Furthermore, we aim to lay the ground work for our future task of character mining.

1.3 Objectives

Since character identification is a brand new task we proposed, there are five main objectives of our work in order to tackle the task in a systematically fashion. The objectives, listed below, consider corpus creation, task analysis, system design, and result evaluation.

- Creating a new corpus for character identification with thorough analysis.

- Assessing the feasibility of the task with coreference resolution systems.
- Reformulating character identification into a coreference resolution task.
- Implementing a coreference resolution system to solve the task.
- Evaluating our approach to character identification on our corpus.

To the best of out knowledge, this is the first time that character identification is formally proposed and experimented on a large and newly-created corpus.

Chapter 2
Background

2.1 Entity linking

Entity linking is a natural language processing task of determining entities and connecting related information in context to the entities [21]. There are multiple aspects of the task as well as its applications. One branch of entity linking, perhaps the more prominent one, is Wikification. It aims to associate concepts to their corresponding Wikipedia pages [30]. For example, given a statement below.

“The 44$^{\text{th}}$ president of the United States wishes to publish a memoir ..."

The task of Wikification would try to link “The 44$^{\text{th}}$ president of the United States" to the Wikipedia page of “Barrack Obama" even if the name is not mentioned in the statement. Another branch of entity linking that also take advantage of massive Wikipedia corpus is Entity Disambiguation. Different from entity linking that finds the distinct one-to-one or one-to-many relations between mentions to concepts, entity disambiguation aim to clarify the connections to concepts when the concepts are confusing due to their similar names or traits [39, 23]. An example is given in Section 1.1 where “Emory" refers to the university and not the person in the context.

Other than linking mentions to their Wikipedia concepts, works have also been done on domain-specific information using extracted local context [34]. Instead of using Wikipedia which serves as an universal knowledge base that contains factual information of various domains, entity linking systems can be trained on corpus of specific domains, such as law and medicine, to create in more versatile systems that helps with computations and automations in other fields. In the case of character identification, a system will be trained on genre-specific corpus, transcripts of multiparty conversations, rather than a domain-specific corpus. However, it can easily be trained on conversations that occur around a particular character or topic, and thus making it domain-specific to the character.

2.2 Speaker identification

Speaker identification is a task that has been proposed and worked on before. In their work, Kundu et al. [25] proposes different approaches to identify speakers at the turn levels for film dialogue scripts. For each line in the transcripts, the system makes prediction of a possible speaker groups, and there are three main categories of the speaker groups, “primary", “secondary", an “others". The work has helped to filter speaker candidates of individual utterances, however, the scope and the applications of the proposed systems are limited for character identification. Speaker identification does not concern the mentions within utterances and serves more of a documentation classification task. It does not identify the exact speakers of utterances but only the groupings of the speakers. The task is beneficial to ours, given the scenarios where the speakers of conversations are unknown, thus marking it a valuable task to perfect in the future.

2.3 Conversation Corpora

There exists several corpora of multi-party conversations. SwitchBoard is a large telephone speech corpus with focuses on speaker authentication and speech recognition [11]. It consists 2,500 audio recordings of phone conversations collected from 500 speakers in the US, and it is designed for training and testing different speech processing systems.

ICSI Meeting Corpus is a collection of meeting audios recording and their transcripts created for researches in speech recognition, dialog modeling, and etc [20]. It contains three years worth of data from natural meetings at the International Computer Science Institute (ICSI) in Berkeley, California. In addition to the audios and transcripts, metadata information of the participants, meetings, and hardwares is also included in the corpus. The Ubuntu Dialogue Corpus is a recently introduced dialogue corpus that provides task-domain specific conversations with multiple turns [27]. It includes about 1 million multi-turn dialogues with over 7 millions utterances and 100 millions words. Speaker information is also provided in the corpus.

All of the corpora provide immense amount of data, however, they lack the necessary information for our task, mentions and their referent information. The primary purpose of them aims to solve tasks like speech-related tasks in recognition and generation. Thus, they are not applicable to our task, and we will have to create our own corpus for character identification.

2.4 Neural network

Artificial neural network was first proposed in the 1940s as a learning model that simulates neuron activities in human brain [29]. It embraces the concept of Hebbian Learning [15], which stated “The connection between two neurons is strengthened if the neurons fire simultaneously." The model is made possible for learning tasks, like regression, classification, and prediction, with the later advancements in computer technology and the introduction of back-propagation [45].

Artificial neural network has shown successes in various applications, particularly in learning non-linear and complex features. Models tend to outperform traditional statistical models when trained on large datasets. Different architectures of the neural network have since been introduced as the concept gains its popularity.

2.4.1 Word2Vec and Word Embeddings

Word2vec is a shallow word embedding model that learns to map each word in the vocabulary into a continuous vector of low-dimension from its distributional property observed in raw text [31]. Word2Vec provides valuable distributional semantic information of words. When trained on large corpus, it learns the word embedding information that can be considered as the vector representations of words. Compared to the sparse vector that the previous bag-of-word [13] approach generates, the dense word vectors from Word2vec not only help reduce the dimension but also provide richer syntactic and semantic information.

The model has two architectures, continuous bag-of-words (CBOW) and skip-gram with negative sampling(SGNS). Both architectures use feed-forward neural networks [1] with one hidden layer as their backbone learning models. The CBOW architecture is trained to predict a target word t given the surrounding context words $c_0..c_n$ with the goal of maximizing $p(t|c_0..c_n)$. The SGNS architecture differs in which it tries to predict the context words $c_0..c_n$ given the target word t. The idea of negative sampling is introduced to allow skip-gram model to be more efficient in training [31]. Empirically, finer-grained vectors are produced from the skip-gram model trained on a large corpus as opposed to CBOW.

2.4.2 Convolutional Neural Network

One particular architecture of neural network, used in our proposed system in this work, is the Convolutional Neural Network (CNN). Unlike conventional neural network architecture, such as feed-forward neural network, that consider all input information at once, CNN selects the most salient information through convolution and pooling operations. [24]

Given a input image of dimension $m \times n$ and f as the number of filter used for CNN. A convolution filter of window size $h \times w$ convolutes through the input matrix, and each convolution operation generates f number of values that represent the input values of a certain window. As result of applying the operation on the input image, a convoluted feature image with dimension of $f \times (m - h + 1) \times (n - w + 1)$. A pooling layer of window size $f \times 1 \times 1$ is then applied on the feature image to collect the features that best describe the convoluted input image, resulting a image of size $(m - h + 1) \times (n - w + 1)$ that can be seemed as a smaller representation of the original image.

Due to its unique nature, CNN quickly gains popularity for computer vision tasks and is later adapted by natural language processing task. CNN allows the learning of different combinations and subsets of features and thus improves performance in desired tasks. Furthermore, CNN has the advantage of inexpensive computation compared to other neural network architectures.

2.5 Coreference resolution

Similar to entity linking, coreference resolution is a natural language processing task that connects mentions to their antecedents [37]. The task focuses on finding pair-wise connections between mentions and forming coreference chains of the pairs. Dialogues have previously been studied as a domain of coreference resolutions [41], however, there are no robust system proposed for multiparty conversational data due to its complex and context-switching nature. For the system proposed for dialogue data, most of them focus on narrations or conversations between two parties, such as a tutoring system [33]. Despite the similarity between coreference resolution and character identification, reformation of coreference resolution systems is still needed for character identification since resolved coreference chains do no directly refer to any character entities. There are three systems that we experiment in this work, one rule-based and two state-of-the-art neural systems developed by the Stanford and Harvard NLP groups.

2.5.1 Evaluation Metrics

In order to evaluate the performance of coreference resolution systems, there are three mainstream evaluation metrics used: MUC, B^3, and CEAF$_e$.

MUC [44] concerns the number of pairwise links needed to be inserted or removed to map system responses to gold keys. The number of links the system and gold shared and minimum numbers of links needed to describe coreference chains of the system and gold are computed. Precision is calculated by dividing the former with the latter that describes the system chains, and recall is calculated by dividing the former with the latter that describes the gold chains.

B^3 [2] metric computes precision and recall on a mention level, instead of evaluating the coreference chains solely on their links. System performance is evaluated by the average of all mention scores. Given a set M that contains mentions denoted as m_i. Coreference chains S_{m_i} and G_{m_i} represent the chains containing mention m_i in system and gold responses. Precision(P) and recall(R) are calculated as below:

$$P(m_i) = \frac{|S_{m_i} \cap G_{m_i}|}{|S_{m_i}|}, \quad R(m_i) = \frac{|S_{m_i} \cap G_{m_i}|}{|G_{m_i}|}$$

CEAF$_e$ [28] metric further points out the drawback of B^3, in which entities can be used more than once during evaluation. For **B**^{3}, both coreference chains of the same entity and chains with mentions of multiple entities are not penalized. To cope with this problem, CEAF evaluates only on the best one-to-one mapping between the system's and gold's entities. Given a system entity S_i and gold entity G_j. An entity-based similarity metric $\phi(S_i, G_j)$ gives the count of common mentions that refer to both S_i and G_j. The alignment with the best total similarity is denoted as $\Phi(g^*)$. Thus precision(P) and recall(R) are measured as below.

$$P = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)}, \quad R = \frac{\Phi(g^*)}{\sum_i \phi(G_i, G_i)}$$

For simplicity, B^3 evaluation metric will be used to evaluated the coreferent result in this work since MUC has flaws in treating all error equally, and CEAF$_e$ is too strict for our task and too slow to compute.

2.5.2 Stanford Sieve System

The Stanford multi-pass sieve system [26] is used to provide insight of how a rule-based coreference resolution system would perform on our task. The system is composed of multiple sieves of linguistic rules that are in the orders of high-to-low precision and low-to-high recall. Information regarding mentions, such as plurality, gender, and parse tree, is extracted during mention detection and used as global and local features. Pairwise links between mentions are formed based on defined linguistic rules at each sieve in order to construct coreference chains and mention clusters. Although no machine learning is involved, the system offers efficiency in decoding while yielding reasonable results. More importantly, the system is not domain-specific since it is based on linguistic theories and properties of the English language. The rules are the same for both written and spoken discourses.

2.5.3 Stanford Neural System

The first state-of-the-art statistical coreference resolution system is the Stanford neural-network coreference resolution system [8, 9]. The system is consisted of two encoders and two ranking models. The two encoders are a mention-pair encoder and a cluster-pair encoders. The mention-pair is an auto-encoder [3] of three hidden layers with Rectifier(ReLu) [32] activation functions. Given a concatenated feature vector of any mention pair, the mention-pair encoder captures and generates the mention-pair representation from the vector. This helps to reduce the dimension of the input feature vectors while learning the important features [17]. (The complete table of feature used is included in the paper from Clark and Manning [8].)

Given $1 \times d$ mention-pair representations $\mathbf{r_m}(m_i, m_j)$ of any mention pairs $(m_i,\, m_j)$ generated from the mention-pair encoder where d is the encoded dimension, cluster representation can be constructed. Given two clusters $c_i = \{m^i_{1...m}\}$ and $c_j = \{m^j_{1...n}\}$, a $(m(n-1)/2) \times d$ mention-pair matrix $\mathbf{R_m}(c_i, c_j)$ has rows of mention-pair representations of $\mathbf{r_m}(c_i \times c_j)$. A cluster-pair encoder $\mathbf{r_c}(c_i, c_j)$ is defined as the following:

$$\mathbf{r_c}(c_i, c_j) = \begin{cases} max\{\mathbf{R_m}(c_i, c_j)_{k,\cdot}\} & \text{for } 0 \leq k < d \\ \\ avg\{\mathbf{R_m}(c_i, c_j)_{k-d,\cdot}\} & \text{for } d \leq k < 2d \end{cases}$$

The two ranking models are mention- and cluster-ranking models. The mention ranking model is a feed-forward neural network with one hidden layer of sigmoid activation function. The model takes any mention-pair representation as input and output $\mathbf{s_m}(m_i, m_j) \in [0, \infty)$ indicating the likelihood of a link between two mentions. The model is pre-trained for cluster-ranking models with the purposes of easy-first clustering and search space pruning.

The cluster-ranking model is the principal model for the Stanford system. In the initial state, all mentions are treated as clusters of one mention. A policy network [43] is applied to learn the decisions for *MERGE* or *PASS* given any cluster pairs. Since coreference resolution is a sequential prediction problem where prior prediction affects later decisions, the learning-to-search algorithm is applied to improve the quality of the final coreferent results [10, 7, 9]. As result of the combinations of encoders and ranking models, Stanford neural-network-based coreference resolution system is able to achieve 65.29% CoNLL F1 average.

2.5.4 Harvard Neural System

The Harvard neural coreference resolution system is another state-of-the-art system [46, 47]. Like the Stanford system, it also takes a deep learning approach to train their models. Instead of constructing a cluster-pair encoder and cluster-ranking model to capture global features of the clusters, the system takes advantage of the sequence information from individual cluster-based recurrent neural networks [19] learned from previous sequential predictions of mention pairs.

The Harvard neural system is primarily consisted of two parts, anaphoricity scoring and antecedent ranking models. Unlike the ranking models, the anaphoricity scoring takes in the local context of a mention and scores its anaphoricity. This provides additional feature for later linking, and it also helps to identify singleton mention early on to reduce the complexity of the ranking model. Given two feature maps ϕ_a and ϕ_p, one for mention features and one for pairwise features as described in the paper from Wiseman et al. [46], antecedent mention x, the truth y (if x is a singleton, $y = \epsilon$), a scoring function $f(x, y)$ consisting of two functions $u(x, y)$ and $v(x)$ can be defined as shown below, where the former scores anaphoricity of a mention pair, and the latter scores that of a mention if it is a singleton or has no antecedent.

$$f(x,y) = \begin{cases} \mathbf{u}^{\intercal} \begin{bmatrix} \mathbf{h_a}(x) \\ \mathbf{h_p}(x,y) \end{bmatrix} + \mathbf{u}_0 & \text{for } y \neq \epsilon \\ \mathbf{v}^{\intercal} \mathbf{h_a}(x) + \mathbf{v}_0 & \text{for } y = \epsilon \end{cases}$$

$$\mathbf{h_a}(x) \triangleq tanh(\mathbf{W_a}\ \phi_{\mathbf{a}}(x) + \mathbf{b_a})$$

$$\mathbf{h_p}(x,y) \triangleq tanh(\mathbf{W_p}\ \phi_{\mathbf{p}}(x,y) + \mathbf{b_p})$$

The antecedent ranking builds on top of the anaphoricity scoring model and makes pairwise decision for mention pairs with cluster information. A global scoring function g can be defined as below, where $\mathbf{h}_c(x_n)$ denotes the mention representation of mention x_n, $\mathbf{h}_{<n}^{(m)}$ is the cluster representation learned from the recurrent neural network of cluster m, and $\mathrm{NA}(x_n)$ denotes the score of mention x_n as a singleton.

$$\mathbf{h_c}(x) \triangleq tanh(\mathbf{W_c}\ \phi_\mathbf{a}(x) + \mathbf{b_c})$$

$$g(x_n, y_n, \mathbf{z}_{1:n-1}) \triangleq \begin{cases} \mathbf{h}_c(x_n)^\intercal\ \mathbf{h}_{<n}^{z_{y_n}} & \text{if } y \neq \epsilon \\ \mathrm{NA}(x_n) & \text{if } y = \epsilon \end{cases}$$

$$\mathbf{h_j^{(m)}} \leftarrow \mathbf{RNN}(\mathbf{h_c}(x)(X_j^{(m)}),\ \mathbf{h_{j-1}^{(m)}},\ \theta)$$

$$\mathrm{NA}(x_n) = \mathbf{q}^\intercal\ tanh(\mathbf{W}_s \begin{bmatrix} \phi_a(x) \\ \sum_{m=1}^{M} \mathbf{h}_{<n}^{(m)} \end{bmatrix} + \mathbf{b}_s$$

A more greedy searching algorithm compared to that of the Stanford neural system is applied on top of the models to optimize the resultant coreferent clusters. As result, the Harvard neural system yields 64.21% CoNLL F1 average.

Chapter 3

Corpus

3.1 Corpus creation

As discussed before, although there are existing corpora of multiparty conversations, they do not have sufficient mention and referent annotations that are specific to our task. Thus it is both novel and necessary for us to generate a corpus for our task. Our corpus is published and publicly available online[1]. This work also introduce a systematic framework for annotating referent information of mentions in order to create a large scale dataset for character identification.

3.1.1 Data collection

We choose TV shows as our sources of multiparty conversation data. They are selected because they represent everyday conversation well, nonetheless they can very well be domain-specific depending on the plots and settings. Their contents and exchanges between characters are written for ease of comprehension. Moreover, prior knowledge regarding characters is usually not required and can be learned as show proceeds. TV shows also cover a variety of topics and are carried on over a long period of time by focused groups of characters.

[1] `http://github.com/emorynlp/character-mining`

Transcripts of the TV show, *Friends* is selected. The show serves as an ideal candidate due to its casual and day-to-day conversations among their characters. Seasons 1 and 2 of *Friends* (F1 and F2) are collected from a fan-generated site[2] . A total of 2 seasons, 47 episodes, and 620 scenes are collected (Table 3.1). The ratio relationships between the constituents of our corpus are shown in Table 3.2

	Epi	**Sce**	**Spk**	**UC**	**SC**	**WC**
F1	24	327	135	6,626	11,452	82,134
F2	23	293	141	6,048	9,638	82,211
Total	47	620	238	12,674	21,090	164,345

Table 3.1: Composition of our corpus.

Epi/Sce/Spk: # of episodes/scenes/speakers. UC/SC/WC: # of utterances/statements/words. Redundant speakers between F1 & F2 are counted only once thus result in a lower total count.

	Sce : Epi	**Spk : Sce**	**UC : Sce**	**SC : UC**	**WC : UC**	**WC : SC**
F1	13.63	0.41	4.84	1.73	12.40	7.17
F2	12.74	0.48	5.13	1.59	13.59	8.53
Total	13.19	0.38	4.98	1.66	12.97	7.79

Table 3.2: Constituent ratios of our corpus in percentages.

Epi/Sce/Spk: # of episodes/scenes/speakers. UC/SC/WC: # of utterances/statements/words.

Each season is divided into episodes, and each episode is divided into scenes based on the boundary information provided by the transcripts. Each scene is divided into utterances where each utterance belongs to a speaker (e.g., the scene in Figure 1.1 includes four utterances). Each utterance consists of one or more sentences that may or may not contain action notes enclosed by parentheses (e.g., *Ross stares at her in surprise*). A sentence with its action note(s) removed is defined as a statement.

[2] `http://friendstranscripts.tk`

3.1.2 Mention detection

Given the dataset in Section 3.1.1, mentions indicating humans are pseudo-annotated by our rule-based mention detector, which utilizes dependency relations, named entities, and personal noun dictionary information provided by the open-source toolkit, NLP4J.[3] Our rules are as follows: a word sequence is considered a mention if (1)it is a person named entity, (2)it is a pronoun or possessive pronoun excluding *it*, or (3)it is in the personal noun dictionary. The dictionary contains 603 common and singular personal nouns chosen from Freebase[4] and DBpedia.[5] Table 3.3 and Table 3.4 shows the break down of the mentions extracted with our rule-based mention detector.

	NE	**PRP**	**PNN(%)**	**All**
F1	946	7,549	811 (14.76)	9,306
F2	1,037	7,459	806 (14.30)	9,302
Total	1,983	15,008	1,617 (14.52)	18,608

Table 3.3: Count of the detected mentions.

NE: named entities, PRP: pronouns, PNN(%): singular personal nouns and its ratio to all nouns.

	NE : M	**PRP : M**	**PNN : M**
F1	10.17	81.12	8.71
F2	11.15	80.19	8.66
Total	10.66	80.65	8.69

Table 3.4: Composition of the detected mentions in percentages.

NE: named entities, PRP: pronouns, PNN: singular personal nouns, M: all mentions.

Plural (e.g., *we*, *them*, *boys*) and collective (e.g., *family*, *people*) nouns are discarded but will be included in the next version of the corpus.

[3] https://github.com/emorynlp/nlp4j

[4] http://www.freebase.com

[5] http://wiki.dbpedia.org

3.1.3 Corpus annotation

All mentions from Section 3.1.2 are first double annotated with their referent characters, then adjudicated if there are disagreements between annotators. Both annotation and adjudication tasks were conducted on Amazon Mechanical Turk[6]. Annotation and adjudication of 18,608 mentions took about 8 hours and costed about $450.

Each mention is annotated with either a main character, an extra character, or one of the followings: collective, unknown, or error. *Collective* indicates the plural use of *you*/*your*, which cannot be deterministically distinguished from the singular use of those by our mention detector. *Unknown* indicates an unknown character that is not listed as an option or a filler (e.g., *you know*). *Error* indicates an incorrectly identified mention that does not refer to any human character.

Our annotation scheme is designed to provide necessary contextual information and easiness for accurate annotation. The target scene for annotation includes highlighted mentions and selection boxes with options of main characters, extra characters, collective, unknown, and error. The previous and next two scenes from the target scene are also displayed to provide additional contextual information to annotators (Table 3.5). We found that including these four extra scenes substantially reduced annotation ambiguity. The annotation is done by two annotators, and only scenes with 8-50 mentions detected are used for the annotation; this allows annotators to focus while filtering out the scenes that have insufficient amounts of mentions for annotation.

[6] `https://www.mturk.com/mturk/`

	Friends: Season 1, Episode 1, Scene 1	
	. . .	
Ross:	I$_1$ told mom$_2$ and dad$_3$ last night, they seemed to take it pretty well.	1. ‘I$_1$’ refers to?
Monica:	Oh really, so that hysterical phone call I got from a woman$_4$ at sobbing 3:00 A.M.,	- . . .
	“I$_5$’ll never have grandchildren, I$_6$’ll never have grandchildren.” was what?	2. ‘mom$_2$’ refers to?
Ross:	Sorry.	- . . .
Joey:	Alright Ross$_7$, look. You$_8$’re feeling a lot of pain right now. You$_9$’re angry.	3. ‘dad$_3$’ refers to?
	You$_{10}$’re hurting. Can I$_{11}$ tell you$_{12}$ what the answer is?	- Main character$_{1..n}$
	. . .	- Extra character$_{1..m}$
	Friends: Season 1, Episode 1, Scene 2	- Collective
	. . .	- Unknown
	Friends: Season 1, Episode 1, Scene 3	- Error
	. . .	

Table 3.5: Example of our annotation task conducted.

Main character$_{1..n}$ displays the names of all main characters of the show.
Extra character$_{1..m}$ displays the names of high frequent, but not main, characters.

3.1.4 Corpus adjudication

Any scene containing at least one annotation disagreement is put into adjudication. The same template as that for the annotation task (Table 3.5) is used for the adjudication, except that options for the mentions are modified to display options selected by the previous two annotators. Nonetheless, adjudicators still have the flexibility of choosing any option from the complete list as mentioned in the annotation task. This task is done by three adjudicators. The resultant annotation is determined by the majority vote of the two annotators from the annotation task and the three adjudicators from this task.

Serval preliminary tasks were conducted on Amazon Mechanical Turk to improve the quality of our annotation using a subset of the *Friends* season 1 dataset. Though the result on annotating the subset gave reasonable agreement scores (F1$_p$ in Table 3.6), the percentage of mentions annotated as *unknown* was noticeably high. Such ambiguity was primarily attributed to the lack of contextual information since these tasks were conducted with a previous template design that does not provide additional scene information other than the target scene itself. The unknown rate decreased considerably in the later tasks (F1 and

F2) after the previous and the next two scenes were added for context. As a result, our annotation gave the absolute matching score of 82.83% and the Cohen's Kappa score of 79.96% for inter-annotator agreement, and the unknown rate of 11.87% across our corpus, which was a consistent trend across the seasons included in our corpus.

	Match	**Kappa**	**Col**	**Unk**	**Err**
$F1_p$	83.00	79.94	13.2	33.96	3.95
F1	84.55	80.75	11.2	21.42	3.71
F2	82.22	80.42	13.13	11.69	0.63
Avg.	**82.83**	**79.96**	**12.42**	**11.87**	**2.75**

Table 3.6: Inter-annotation agreement analysis.

Match and Kappa show the absolute matching and Cohen's Kappa scores between two annotators (in %). Col/Unk/Err shows the percentage of mentions annotated as collective, unknown, and error, respectively.

3.1.5 Corpus disambiguation

Despite our best effort to annotate and adjudicate our corpus, mentions labeled as *Unknown* still make up for more than 10% of our corpus. Additional instructions and procedures to our annotation and adjudication tasks for unknown mention disambiguation would not be ideal. Disambiguation requires extra and unknown number of referent options be introduced to our annotation scheme shown in Table 3.5. This would increase the complexity of the tasks and thus compromises the inter-annotation agreement of the annotated results.

Instead of augmenting our previous tasks, we put our annotated and adjudicated corpus up for another disambiguation task that targets at unknown mentions. The task utilizes the scheme shown in Table 3.5 with some minor tweaks. First, only mentions that are labels *Unknown* are highlighted for annotation. Second, additional scenes before and after the target one are provided for more contextual information. Third, arbitrary and generic referent options , as shown below, are provided to annotators in additional to known character names.

- *General*: Mention used in reference to a general case rather than an specific entity.

  eg. People are generically nice to others.

- *Other*: Mention that refers to irrelevant singleton entity.

  eg. Some one licked my neck on the subway.

- *Man/Woman/Person #*: Mention that refers to a generic entity whose identity is not defined. Entities are distinguished by their group names and numberings.

  eg. That waitress is really cute. I think I am going to ask her out.

The numberings for distinct groups of the *Man/Woman/Person #* are limited to 5 per group for simplicity. The scenes that have unknown mentions that would require more than 5 entities of a certain group, like *Man 6*, are recursively annotated until all unknown mentions can be disambiguated. This task is proposed and supervised by this work, and the principle work in done by an undergraduate student, Ethan Zhou, at Emory University. The result of the disambiguation is shown in Table 3.7 with detailed break down of the counts of mentions in each group.

	Primary	**Secondary**	**Generic**	**Collective**	**General**	**Other**	**Total**
F1	5,101	2,610	152	1,150	109	184	9,306
F2	5,312	2,432	111	1,238	42	167	9,302
Total	10,413	5,042	263	2,388	151	351	18,608

Table 3.7: Count break down of mentions in our corpus after disambiguation.

Primary: main characters, Secondary: supporting characters with names identified, Generic: *Man/Woman/Person* entities, Collectives: collective use of *you/your/yourself*, General: reference to general cases, Other: irrelevant and singleton entities.

3.2 Corpus analysis

3.2.1 Annotation results

As mentioned in the later section Section 4.1, our annotated corpus can be formatted with two delimiters. Scene-delim documents treat every scene as a document, and episode-delim documents treat every episode as a document. Additional scenes that does not have annotation are included in the episode-delim documents to provide additional context information, however, for the scenes that do not have annotations, they are not included as scene-delim documents. The final result of the annotations in our corpus are shown in Table 3.8 for scene-delim documents and in Table 3.9 for episode-delim documents.

	D	**M**	**C**	**S**	**S%**	**Avg(|C|)**	**M : D**	**C : D**	**S : D**
F1	229	9,306	1,646	424	25.76	5.7	40.64	7.19	1.85
F2	219	9,302	1,462	349	23.87	6.4	42.47	6.68	1.59
Total	448	18,608	3,108	773	24.87	6.0	41.54	6.94	1.73

Table 3.8: Annotation statistics and constituent ratios for scene-delim documents.
D/M/C: Counts of documents/mentions/coreferent clusters. S/S%: Count of singletons and its composition ratio to all clusters. Avg(|C|): average count of mentions in a chain.

	D	**M**	**C**	**S**	**S%**	**Avg(|C|)**	**M : D**	**C : D**	**S : D**
F1	24	9,306	599	149	24.87	15.5	387.75	24.96	6.21
F2	22	9,302	518	122	23.55	18.0	422.82	23.55	5.55
Total	46	18,608	1,117	271	24.26	16.7	404.52	24.28	5.89

Table 3.9: Annotation statistics and constituent ratios for episode-delim documents.
D/M/C: Counts of documents/mentions/coreferent clusters. S/S%: Count of singletons and its composition ratio to all clusters. Avg(|C|): average count of mentions in a chain.

3.2.2 Mention Detection Error

Unlike most mention detector used in coreference systems, our rule-based mention detector focuses on finding personal mentions that are specific in the dialogue context. For quality assurance, 8.5% of the corpus is sampled and manually evaluated. A total of 1,584 mentions from the first episode of each season in each show are extracted. If a mention is not identified by the detector, it is considered a “miss”. If a detected mention does not refer human character(s), it is considered an “error”. Our evaluation shows an F1 score of 95.93, which is satisfactory (Table 3.10).

	Miss	**Error**	**Total**	**P**	**R**	**F**
F1	17	19	615	96.82	94.15	94.47
F2	15	3	448	99.31	95.98	97.62
B1	19	14	475	96.93	93.05	94.95
Total	51	36	1,538	97.58	94.34	**95.93**

Table 3.10: Evaluation of our mention detection.
P: precision, R: recall, F: F1 score (in %).

A further investigation on the causes is conducted on the misses and errors of our mention detection. Table 3.11 shows the proportion of each cause. The majority of them are caused by either negligence of personal common nouns or inclusion of interjection use of pronouns, which are mostly coming from the limitation of our lexicon.

1. Interjection use of pronouns (e.g., *Oh mine*).
2. Personal common nouns not included in the personal noun dictionary.
3. Non-nominals tagged as nouns.
4. Proper nouns not tagged by either part-of-speech tagger or name entity recognizer.
5. Misspelled pronouns (e.g., *I’m* → *Im*).
6. Analogous phrases referring to characters (e.g, *Mr. I-know-everything*).

Causes of Error and Miss	**%**
Interjection use of pronouns	27
Common noun misses	27
Proper noun misses	18
Non-nominals	14
Misspelled pronouns	10
Analogous phrases	4

Table 3.11: Proportions of the misses and errors of our mention detection.

3.2.3 Annotation Disagreement

In addition to the ambiguity occurred with the mentions labeled *Unknown*, the ambiguity of speakers of which the mentions *you*/*your*/*yourself* might refer to leads to a common disagreement in our annotations. Such confusion often occurs during a multiparty conversation when one party attempts to give a general example using personal mentions that refer to no one in specific. For the following example, annotators label the *you*'s as *Rachel* although they should be labeled as *unknown* since *you* indicates a general human being.

Monica: *(to Rachel)* You$_1$ do this, and you$_2$ do that. You$_3$ still end up with nothing.

The case of *you* also results in another ambiguity when it is used as a filler:

Ross: *(to Chandler and Joey)* You$_1$ know, life is hard.

The referent of *you* here is subjective and can be interpreted differently among individuals. It can refers to Chandler and Joey collectively. It can also be unknown if it refers to a general scenario. Furthermore, it potentially can refers to either Chandler or Joey indivisually based on the context. Such use case of *you* is often unclear to human annotators; thus, for the purposes of simplicity and consistency, this work treats them as *unknown* and considers that they do not refer to any speaker.

Another common disagreement among our annotators happens with the referent labels *Unknown* and *Error* due to insufficient context information provided by our annotation scheme. Take the line shown below as an example, the mentions *Marcel* refer to the pet monkey of Ross's. According to our annotation guidelines, annotators are supposed to label the mentions as *Error* since they refer to a non-personal entity.

Ross: Marcel$_1$, Marcel$_2$, come! come over here!

The two proceeding and succeeding scenes provided in our scheme do not have sufficient information about Marcel. In fact, the role of the Marcel as a monkey is introduced episodes ago. For some of our annotators who have prior knowledge about the show *Friends* are able to identify Marcel as a non-personal entity while others who are not familiar with the show cannot. This case of disagreement would create inconsistency in our annotations since a desired system would have learned the information in the previous episodes and ignore Marcel as a mention to any personal entity, but our annotation does not reflect such property.

Both cases of disagreement are adjudicated during our disambiguation task described in Section 3.1.5, and all annotations in our corpus provide clear and necessary referent information for character identification.

Chapter 4

Approaches

4.1 Data Formulation

4.1.1 Data Split

In order to train the models that will be discussed in the later sections, we split our corpus of both seasons 1and 2 of the TV show *Friends* into the training(∼75%), development(∼10%), and evaluation(∼15%) sets. As mentioned before, documents are formulated into two ways, one treating each episode as a document and the other treating each scene as a document, which allows us to conduct experiments with or without the contextual information provided in the previous and next scenes. The break down for the data splits are shown in Table 4.1 for scene-delim documents and in Table 4.2 for episode-delim documents.

	Sce	**M**	**C**	**S**	**S%**	**Avg(|C|)**
TRN	362	15,254	2,531	620	24.50	6.0
DEV	28	1,194	199	52	26.13	6.0
TST	58	2,160	378	101	26.72	5.7

Table 4.1: Data splits for scene-delim documents.

TRN/DEV/TST: training, development, and evaluation sets. Sce/M/C: Counts of scenes/mentions/coreferent chains. S/S%: Count of singletons and its ratio to all chains. Avg(|C|): average count of mentions in a chain.

	Epi	M	C	S	S%	Avg(|C|)
TRN	38	15,254	924	223	24.13	16.5
DEV	3	1,194	66	16	24.24	18.1
TST	5	2,160	127	32	25.20	17.0

Table 4.2: Data splits for episode-delim documents.
`TRN`/`DEV`/`TST`: training, development, and evaluation sets. Sce/M/C: Counts of scenes/mentions/coreferent chains. S/S%: Count of singletons and its ratio to all chains. Avg(|C|): average count of mentions in a chain.

4.1.2 Data formats

For training existing coreference resolution models, Our corpus is reformatted to adapt the convention of CoNLL'12 shared task on coreference resolution [37]. Each statement is parsed into a constituent tree using the Berkeley Parser [36] and labeled with part-of-speech and named entity tags using the NLP4J tagger [6]. The CoNLL data format [7] allows speaker information for each statement, which is used by both systems we experiment with. The converted format preserves all necessary annotation for our task.

For distribution purpose, our corpus is published in JSON format. Each season is released as a JSON file that contains a list of episodes. Each episode contains a list of scenes. Each scene contains a list of utterance. Each utterances contains speaker information and lists of script lines and statements. The scripts lines and statements are tokenized with the NLP4J tokenizer. In alignment to the tokenization, the token-associated referent annotations are included in BILOU [38] convention.

4.2 Coreference resolution

Character identification is tackled as a coreference resolution task here, which takes advantage of existing state-of-the-art systems although it may not yield the best results for our task,

[7] `http://conll.cemantix.org/2012/data.html`

since our task is more similar to entity linking. Most of the current entity linking systems are accustomed to find entities in Wikipedia [30, 39], which are not intuitive to adapt to our task. As mentioned in the previous section, our corpus is first reformed into the CoNLL'12 shared task format, then experimented with three open-source systems and our proposed system. The resultant coreference chains from these systems are linked to a specific character using our cluster remapping algorithm. In addition to showing the result of our proposed system, this work also includes the results of the Stanford sieve, Stanford neural, and Harvard neural systems, as described in Section 2.5.2, 2.5.3, and 2.5.4.

4.2.1 CNN mention-to-mention model

In this work, we propose a mention-mention ranking model that takes advantage of convolutional neural network. Instead of treating all mentions and pairwise features at once, features sharing similar properties are divided into feature groups and learned separately. The model is able to generate more robust mention and mention-pair representations compared to those generated from the Stanford and Harvard neural systems.

Feature selection

There are three main categories of features, mention embedding features, discrete mention features, and pairwise mention features. The mention embedding features are further subcategorized into four different groups based on the properties of the embeddings. The word embeddings of dimension 50 are trained with the Word2vec SGNS model [31] discussed in Section 2.4.1 on raw text corpora including *New York Times*, *Wikipedia*, and *Amazon reviews*. Gender and plurality data used is provided from the work from Bergsma and Lin

[4]. Speaker embeddings of dimension 5 are uniformly randomized. Sentence and utterance vectors are the average word embedding of all the words in a sentence and an utterance. The completed features used in our model are described in Table 4.3.

Feat. group	**Feature**
Group 1	Word emb. of 1st ... 4th words in mention
Group 2	Word embs. of 3 proceeding words before mention Avg. word emb. of all words in mention Word embs. of 3 succeeding words after mention
Group 3	Sent. vecs. of 2 proceeding sent. before mention Sent. vecs. of the current sentence Sent. vecs. of 2 succeeding sent. before mention
Group 4	Utter. vecs. of 2 proceeding utter. before mention Utter. vecs. of the current utterance Utter. vecs. of 2 succeeding utter. before mention

(a) List of grouped mention embedding features

Feat. type	**Feature**
Discrete Feat.	Avg. gender info. of all words in mention Avg. plurality info. of all words in mention Speaker emb. of the current utterance Speaker emb. of the previous utterance
Pairwise Feat.	Exact string match of words between mentions Normalized *lcs* of words between mentions Matching speaker info. of mention pair Distance and position info. between mentions

(b) List of discrete and pairwise mention features

Table 4.3: Completed mention features used in our model.

emb/vec: embedding/vector. sent/utter: sentence/utterance. *lcs*: longest common subsequence.

System architecture

Features extracted as described in the previous section are fed into our model at different stages. Let d_w be the dimension of our mention embedding feature, and f be the number of filters used in each layer. Convolution layers with window sizes of $1 \times d_w$, $2 \times d_w$, and $3 \times d_w$ are applied to each mention embedding feature groups, and the result of each layer is max-pooled along the columns resulting a vector of $1 \times f$. A dropout layer [42] with 0.8 dropout rate is then applied to each of these vectors to regularize and introduce randomness to our model. The stacked result of the convolution and pooling is a matrix of dimension $12 \times f$. Another convolution and max-pooling layers of windows sizes $1 \times f$ and 12×1 are applied to create a $1 \times f$ vector that captures the significance of the grouped mention embedding features. The concatenated and flatten vector of the convoluted grouped mention embedding features and discrete mention features is defined to be a mention representation that encapsulates all necessary local context of a given mention. A pictorial illustration of our mention representation model is shown in Figure 4.1.

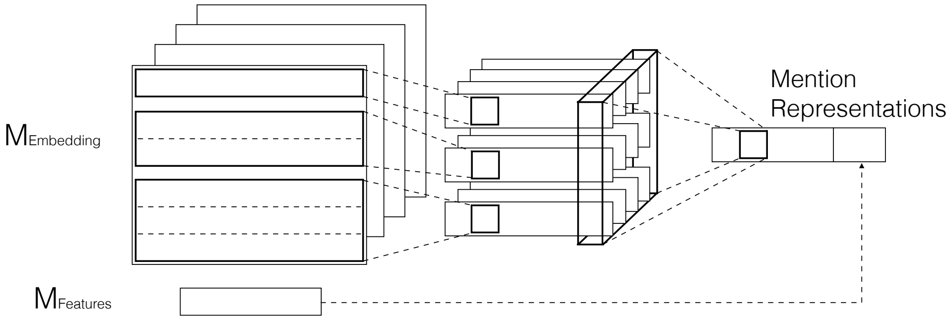


Figure 4.1: Architecture of our mention representation model.

Let $\mathbf{r}_m(m_i)$ be the mention representation of mention m_i of dimension d_{r_m}. Our mention-pair representation model stacks two mention representations $\mathbf{r}_m(m_i)$ and $\mathbf{r}_m(m_j)$ into a $2 \times d_{r_m}$ matrix. A single convolution (with the same number of filters f) and a max-pooling layers with windows sizes of $1 \times d_{r_m}$ and 2×1 are applied to the matrix, resulting a mention-pair vector of dimension f. The vector is concatenated with pairwise mention features extracted from mentions m_i and m_j. The concatenated vector is defined to be a mention-pair representation $\mathbf{r}_{mm}(m_i, m_j)$ of mentions m_i and m_j. The structure of our mention-pair representation model is shown in Figure 4.2 below.

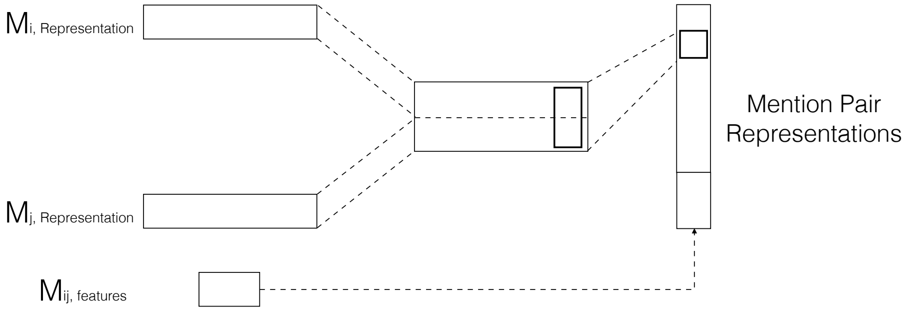


Figure 4.2: Architecture of our mention-pair representation model.

Mention-to-mention ranking model

Given a mention representation $\mathbf{r}_{mm}(m_i, m_j)$ between mention m_i and m_j, a scoring function determining the likelihood of a link between the two mention is defined as the following,

$$\mathbf{s}_{mm}(m_i, m_j) = \sigma(\mathbf{W}_{mm}\ \mathbf{r}_{mm}(m_i, m_j) + \mathbf{b}_{mm})$$

where $\mathbf{W}_{mm}$ and $\mathbf{b}_{mm}$ are the weights and bias of the scoring function, and σ is the sigmoid function. The function is implemented as a single hidden layer of dimension f that takes a mention-pair representation as input and outputs a score $p \in (0, 1)$.

In order to train our model, the scoring function is trained like a regression model with a mean squared error loss function. Let $\mathcal{A}(m_i)$ to be the set of antecedents of mention m_i, $\mathcal{C}(m_j)$ to be the cluster containing mention m_j. The gold linking score $p(a, m)$ given mention $a \in \mathcal{A}(m)$ and m of the training instances is defined as the following:

$$p(a, m) = \begin{cases} 1 & \text{if } m \in \mathcal{C}(a) \\ 0 & \text{otherwise} \end{cases}$$

For each mention, we extract training instances up to the closest antecedent that has the linking score of 1 to reduce complexity. However, all antecedents are included during decoding to maximize the chance of making a correct link. Given a decision threshold $d_{thres} \in (0, 1)$, the predicted cluster $\hat{c}$ for mention m_i is defined as shown below. If $\hat{c} = \epsilon$, then a new cluster $\{m_i\}$ will be introduced to the set of found clusters.

$$\hat{c} = \begin{cases} \mathcal{C}(\underset{m_j \in \mathcal{A}(m_i)}{\arg\max}\ \mathbf{s}_{mm}(m_i, m_j)\ ) & \text{if } p = \max\limits_{m_j \in \mathcal{A}(m_i)} \mathbf{s}_{mm}(m_i, m_j) \geq d_{thres} \\ \epsilon & \text{otherwise} \end{cases}$$

Through back-propagation of the loss function on our scoring function, mention and mention-pair representations are learned and optimized for the task of mention-pair ranking, and the representations can be used to train and introduce as additional features to more complex models in the future.

4.3 Character identification

4.3.1 Rule-based entity linker

Theoretically, the larger a document is the large its coreferent chains are, and in the context of our task, the clusters from running coreference resolution on entire TV show should each represent an entity. However, it is nearly impossible to have such a perfect system. The resultant chains are often segmented and noisy, therefore they are less obvious in the specific characters they are referring to. Since the predicted coreferent chains do not directly point to specific entities, a mapping mechanism is needed for linking those chains to certain characters.

We propose a rule-based entity linker that evaluates and pools the mention within found coreferent chains. The resultant chains from systems are mapped to either a character, collective, or unknown entity. Each coreference chain is reassigned through voting on the majority entity assignment of mentions. The referent of each mention is determined by the below high-precision rules:

1. If the mention is a proper noun or a named entity that refers to a known character, it is referent to the character.
2. If the mention is a first-person pronoun or possessive pronoun, it is referent to the speaker character of the utterance containing the mention.
3. If the mention is a collective pronoun or collective possessive pronoun, it is referent to the *collective* group.

If none of these rules apply to any of the mentions in a coreference chain, the chain is mapped to the *unknown* group.

Chapter 5

Experiments

5.1 Task analysis

As one of the objectives of this work, we wish to investigate the feasibility of solving the task of character identification using trained statistical models. We experiment with the Stanford rule-based sieve model, Stanford pre-trained neural model, and our CNN mention-to-mention ranking model. The former two are tested on the F1+F2 dataset to evaluate the performance of existing rule-based and statistical coreference resolutions systems. The CNN model is trained and tested on different combinations of datasets of F1, F2, and F1+F2 in order to provide holistic views of how feasible it is to tackle character identification with a coreference resolution system. As discussed before in Section 2.5.1, the results are evaluated with the B^3 metric and shown in Table 5.1. The results demonstrate a few interesting trends.

5.1.1 Task feasibility

Based on the results of the trained models, we can see a clear trend of increasing performance as sizes of the training datasets increase. In average, models trained on F1+F2 outperform those on only F1 or F2 by by around 2%. This shows that coreference resolution for dialogue is a trainable task and can be learned by statistical models. Moreover, this implies that with

TRN	TST	**Epi-delim**	**Sce-delim**	**Avg.**
Stanford sieve system	F1+F2	49.87	65.25	**57.56**
Stanford neural system	F1+F2	47.67	64.14	55.91
F1	F1	58.80	75.23	**67.02**
	F2	57.28	72.07	64.68
	F1+F2	58.16	73.88	66.02
F2	F1	60.06	75.43	67.75
	F2	59.82	76.82	**68.32**
	F1+F2	59.98	76.04	68.01
F1+F2	F1	63.64	75.29	69.47
	F2	62.31	77.90	70.11
	F1+F2	63.10	76.41	**69.76**

Table 5.1: Task analysis using existing coreference systems with B^3 evaluation metric.
Epi-/Sce-delim: episode-/scene-delim documents. All results are evaluated using the B^3 metric. The Stanford neural system are pre-trained on ConLL'12 data.

more data, systems have the potential to perform and generate better coreferent results and thus provide more meaningful features for tackling the task of character identification.

5.1.2 Rule-based vs. statistical model

Two observations can be made when comparing rule-based and statistical models. First, it might be unexpected to see a rule-based system outperforms a statistical one. The pre-trained Stanford neural model, which gives state-or-the-art results for the CoNLL corpus, is optimized for data primarily consisted of newswire, articles, and broadcast conversations, thus making it domain-specific for written text. Therefore, the model is not trained for casual and day-to-day conversations that are included in our corpus, whereas a rule-based system can be seemed as a domain-free model that has no bias for the types of discourses. It explains why a rule-based model would top a statistical one in this case.

Second, statistical models are necessary for coreference resolution of multi-party dialogue. When trained on our corpus, statistical model significantly outperforms the rule-based system. This is attributed to the nature of multi-party conversations. Traits and features of mentions spoken by different characters have to be learned. Rules are either too broad or too strict when making coreferent decisions, and it is not intuitive to create rules or define structures that are universal to human conversations. Statistical system with domain-specific model trained on dialogue data is crucial to the success of coreference resolution for character identification on multi-party conversations.

5.1.3 Episode-delim vs. scene-delim documents

We originally foresee the models trained on the episode-delim documents would outperform the ones trained on the scene-delim ones because the latter ones would not provide enough contextual information. However such speculation is not reflected in our evaluation; the results achieved by the scene-level models consistently yield higher accuracy, which is probably because the scene-delim documents are much smaller than the episode-delim ones so that fewer characters appear within each document. The smaller expected cluster sizes and fewer number of characters present help reduce the complexity of the task, but episode-level clusters are still preferred as discussed in Section 5.3.2.

5.1.4 Learning past vs. future conversations

The last intriguing finding from this experiment is the ability and disability of models in predicting past and future conversations. Focusing on only models trained on either F1 or F2, though both model performs the best when they are trained and tested on the same season, there is a difference in their behavior for past and future conversations. There is a greater drop in performance when resolving chains in F2 with the model trained on F1. In average, models are evaluated to be around 2% worse when resolving future conversations.

This is attributed to the persistence of information from previous conversations. Future conversations often include topics revolved around past events. Thus even if a model is not trained directly on past conversations, it is still able to capture pieces of information that resembles them. The model would have more advantage in making the correct prediction on past conversations than on future conversations.

5.2 Coreference resolution

All systems discussed in Section 2.5.3, 2.5.4, and 4.2.1 are experimented on the F1+F2 dataset. Models for each systems are trained and tested on scene- and episode-delim documents. The results are evaluated with the B^3 metrics. Table 5.2 below shows the average performances of three trials for all systems.

System	**Epi-delim**	**Sce-delim**	**Avg.**
Stanford neural	**69.12**	76.79	**72.96**
Harvard neural	57.66	**78.08**	67.87
This work	63.10	76.41	69.76

Table 5.2: Performance of coreference resolution systems on our corpus.
Epi-/Sce-delim: episode-/scene-delim documents. All results are evaluated using the B^3 metric.

The model proposed in this work is able to achieve 63.10% and 76.41% in B^3 F1 scores for episode- and scene-delim documents. The macro average system performance for the two document configurations is 69.76%. From the results of the systems, we are able to deduce the pros and cons for each system.

5.2.1 Stanford neural system

In average, the Stanford neural system [9] performs the best among all systems. In particular, it excels in resolving coreferent links for larger episode-delim documents. The combination of clustering policy network and cluster-pair ranking models are able to capture global cluster features without performance compromises as the size of documents increases. Shown in the evaluation results in Table 5.2, the system is able to achieve 69.12% B^3 F1 score for episode-delim documents, and this is in average 8% better than the other two systems.

5.2.2 Harvard neural system

The Harvard neural system [47] performs relatively well on scene-delim documents (78.08%). This is primarily attributed to its recurrent neural network(RNN) architecture. Since RNN is known for learning sequence information, the RNN learned for each individual cluster provides valuable global features regarding the cluster. However, as the sequence length, or the cluster size, increases, more noise and complexity are introduced to the network. RNN then becomes a drawback for the system on episode-delim documents. This explains the significant decrease in system performance for episode-delim documents.

5.2.3 CNN mention-to-mention ranking model

Compared to the other state-of-the-art coreference resolution systems, our CNN mention-to-mention ranking model is able to achieve acceptable results. The model is able to outperform the Harvard system for episode-delim documents and, in average it yields 63.10% and 76.41% B^3 F1 scores for scene- and episode-delim documents. It is notable that our system is consisted of only one model that does not take global cluster information into account, while both Stanford and Harvard have two pre-trained and one main models. This implies our model is able to learn comparatively better mention and mention-pair representations with the CNN and feature grouping architectures.

5.3 Character identification

Based on the rules we proposed in Section 4.3, our rule-based entity linker is used to solve the task of character identification. The system-generated coreferent chains are remapped to entities if possible based on voting. The linking experiment is done only on the test datasets. The percentages of mentions that each rule is applicable to are shown in Table 5.3, and the quality of the linked results are evaluated by the average purity score of individual clusters. The linking results are shown in Table 5.4.

Linking rule	%
NNP matching	10.37
1st person PRP	37.69
Collective noun	12.73
Unknown	51.94

Table 5.3: Applicability of our mention-to-entity linking rules.
NNP/PRP: Proper noun/pronoun. The top three rules refer to rules 1 to 3 proposed in Section 4.3.1.

5.3.1 Linking rules

The linking rules we defined are able to identify the referent entities for 48.06% of the total mentions. More than half of the identified are 1st person pronouns {“I", “my", “me", “mine", “myself"}, and they are directly linked to the speakers of the utterances containing them. The rules for proper noun and collective noun matchings are used to identify around 10-12% of the mentions. Our rules are highly accurate, but in the cases of narratives, the 1st person pronoun cannot be applied. For example, the line

Joey: She looked at me and said, "I am leaving."

shows a case where the 1st person pronouns “I" does not refer to the speaker Joey since it is his impersonation of another person.

5.3.2 Cluster remapping

	Epi-delim			**Sce-delim**		
System	**UC%**	**Purity**	**Avg(|C|)**	**UC%**	**Purity**	**Avg(|C|)**
Stanford neural	47.36	**57.27**	15.19	27.76	50.76	8.13
Harvard neural	46.22	51.24	14.86	26.69	48.94	6.20
This work	43.84	56.14	15.44	28.55	**51.86**	9.83

Table 5.4: Performance of our rule-based entity linker of auto-generated coreferent chains. Epi-/Sce-delim: episode-/scene-delim documents. UC%: percentage of unknown clusters. Avg(|C|): average count of mentions in a cluster.

As discussed in Section 5.1.3, the scene-level models consistently outperform the episode-level models for coreference resolution. However, an opposite trend is found for character identification when the coreference chains are mapped to their referent characters. The purity scores of the overall character-mention clusters can be viewed as an effective accuracy score for character identification. The purity scores, or the percentages of recoverable character-mentions clusters, of the remapped clusters for the scene-level models are generally lower than the ones for the episode-level models. As shown in Table 5.4, systems that are more inclined to generate larger coreferent cluster have higher purity scores for their remapped clusters.

The percentages of unknown clusters and unknown mentions are considerably higher for the episode-level models. When looking at the output chains for episode-delim documents, most of the unknown clusters are small or singleton clusters. They consist of about one third of all the clusters, however, they only include around 15% of all mentions. We find these results more reasonable and realistic to the nature of our corpus, since the average percentage of mentions that are annotated as *unknown* are 11.87% for the entire corpus and 14.01% for the evaluation set. The primary cause of lower performance for the scene-level models is the lack of contextual information across scenes. The following example is excerpted from the first utterance in the opening scene of F1:

Monica: There's nothing to tell!
He$_1$'s just some guy$_2$ I$_3$ work with!

As the conversation proceeds, there is no clear indication of who He_1 and guy_2 refer to until later scenes which introduce the character. As a result, the coreference chains in the scene-level documents are noticeably shorter than those in the episode-level documents. When trying to determine the referent characters, there are fewer mentions in coreference chains produced by the scene-level models, therefore there is a higher chance for those chains to be mapped to the wrong characters. Thus, the episode-level models are recommended for better performance on character identification.

The purity scores of the remapped clusters using the coreferent chains our CNN mention-to-mention ranking model generated are comparable to the Stanford system for episode-delim documents and outperform the other systems for scene-delim documents. This shows that our model is desirable for the task of character identification, and the feature groupings and selections work better for multi-party conversations.

Chapter 6

Conclusion

This work introduces a new task, called character identification, that is a subtask of entity linking. A new corpus is created for the evaluation of this task, which is comprised of multiparty conversations from TV show transcripts. Our annotation scheme allows the creation of large dataset with the personal mentions and their referent characters annotated. We further disambiguate our corpus and introduce generic groupings of mentions with abstract referent entities. The nature of this corpus is analyzed with potential challenges and ambiguities identified for future investigation. Hence, this work provides baseline approaches and results using existing coreference resolution systems. We also propose a CNN mention-to-mention ranking model that provides better mention and mention-pair representations learned from feature groupings of dialogue-specific features. Experiments are run on combinations of our corpus in various formats to analyze the applicability of the current systems as well as the model trainability for our task. A rule-based entity linker is then proposed to connect the coreference chains to their referent characters. The appropriateness of the rules used is also analyzed, and we are able to identify desirable properties for both coreference resolution and character identification tasks.

Future work

Character identification is the first step to a machine comprehension task we defined as character mining. We are going to extend this task to handle plural and collective nouns. Additional models for coreference resolution will be added on top of our current mention-to-mention ranking model to improve performance. A statistical entity linking system customized for this task will be implemented for better cluster remapping results. With our own coreference resolution system and entity linker, we wish to release a end-to-end system focusing on character identification for multi-party conversations. Furthermore, we will explore an automatic way of building a knowledge base containing information about characters that can be used for tasks such as question answering.

Bibliography

[1] Peter Auer, Harald Burgsteiner, and Wolfgang Maass. A learning rule for very simple universal approximators consisting of a single layer of perceptrons. *Neural networks*, 21 (5):786–795, 2008.

[2] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer, 1998.

[3] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] Shane Bergsma and Dekang Lin. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Sydney, Australia, July 2006. Association for Computational Linguistics.

[5] Yu-Hsin Chen and Jinho D. Choi. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles,

September 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-3612.

[6] Jinho D. Choi. Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL'16, 2016.

[7] Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1136.

[8] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1061.

[9] Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1245.

[10] Hal Daumé III and Daniel Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, pages 97–104. Association for Computational Linguistics, 2005.

[11] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. SWITCHBOARD: Telephone Speech Corpus for Research and Development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'92, pages 517–520, 1992.

[12] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130 – 150, 2013. URL `http://www.sciencedirect.com/science/article/pii/S0004370212000446`.

[13] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[14] F. Hayes-Roth, D. Waterman, and D. Lenat. *Building expert systems*. Addison-Wesley,Reading,MA, Jan 1984.

[15] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[16] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching Machines to Read and Comprehend. In *Annual Conference on Neural Information Processing Systems*, NIPS'15, pages 1693–1701, 2015.

[17] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[18] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. Learning Knowledge Graphs for

Question Answering through Conversational Dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL'15, pages 851–861, 2015.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. The ICSI Meeting Corpus. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP'03, pages 364–367, 2003.

[21] Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *Proceedings of Text Analysis Conference*, TAC'15, 2015.

[22] Mahboob Alam Khalid, Valentin Jijkoun, and Maarten De Rijke. The impact of named entity normalization on information retrieval for question answering. In *European Conference on Information Retrieval*, pages 705–710. Springer, 2008.

[23] Seokhwan Kim, Rafael E. Banchs, and Haizhou Li. Towards Improving Dialogue Topic Tracking Performances with Wikification of Concept Mentions. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL'15, pages 124–128, 2015.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with

deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Amitava Kundu, Dipankar Das, and Sivaji Bandyopadhyay. Speaker identification from film dialogues. In *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*, pages 1–4. IEEE, 2012.

[26] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic Coreference Resolution Based on Entity-centric, Precision-ranked Rules. *Computational Linguistics*, 39(4):885–916, 2013.

[27] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL'15, pages 285–294, 2015.

[28] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics, 2005.

[29] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[30] Rada Mihalcea and Andras Csomai. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM'07, pages 233–242, 2007.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[32] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[33] Nobal B. Niraula, Vasile Rus, Rajendra Banjade, Dan Stefanescu, William Baggett, and Brent Morgan. The DARE Corpus: A Resource for Anaphora Resolution in Dialogue Based Intelligent Tutoring Systems. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, pages 3199–3203, 2014.

[34] Alex Olieman, Jaap Kamps, Maarten Marx, and Arjan Nusselder. A Hybrid Approach to Domain-Specific Entity Linking. In *Proceedings of 11th International Conference on Semantic Systems*, SEMANTiCS'15, 2015.

[35] Haoruo Peng, Kai-Wei Chang, and Dan Roth. A Joint Framework for Coreference Resolution and Mention Head Detection. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, CoNLL'15, pages 12–21, 2015.

[36] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (COLING-ACL'06)*, pages 433–440, 2006.

[37] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen

Zhang. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL'12, pages 1–40, 2012.

[38] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[39] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL'11, pages 1375–1384, 2011.

[40] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP'13, pages 193–203, 2013.

[41] Marco Rocha. Coreference Resolution in Dialogues in English and Portuguese. In *Proceedings of the Workshop on Coreference and Its Applications*, CorefApp'99, pages 53–60, 1999.

[42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

[43] Richard Stuart Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 1984. AAI8410337.

[44] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics, 1995.

[45] Paul Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1974.

[46] Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1416–1426, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1137`.

[47] Sam Wiseman, Alexander M Rush, and Stuart M Shieber. Learning global features for coreference resolution. In *Proceedings of NAACL-HLT*, pages 994–1004, 2016.