Multimodal Character Representation for Visual Story Understanding

by

Mahmoud Azab

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2019

Doctoral Committee:

Professor Rada Mihalcea, Chair
Assistant Professor Jia Deng
Associate Professor Emily Mower Provost
Assistant Professor VG Vinod Vydiswaran

Mahmoud Azab

mazab@umich.edu

ORCID iD: 0000-0002-6013-8517

In memory of my father and my beloved grandmother

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Rada Mihalcea, for her priceless guidance and support. I also appreciate her willingness to give me the freedom to pursue my own ideas. On several occasions, Rada stayed up online until the 3 AM conference deadline to help me edit and polish my submissions. She also provided all the resources required to support my research. Although the PhD journey is full of ups and downs, Rada always managed to support and encourage me when I needed it the most.

I am deeply grateful to Jia Deng for his invaluable guidance and mentorship. Jia significantly helped with shaping and improving all the papers and work in this dissertation. His thoughtful comments and suggestions taught me how to identify research gaps and approach research problems in an organized way.

I also want to thank my thesis committee members, Emily Mower Provost and VG Vinod Vydiswaran, for their valuable feedback and questions. I also like to thank my supervisors at Carnegie Mellon Qatar, Kemal Oflazer and Behrang Mohit. They both taught me how to conduct academic research and encouraged me to pursue a PhD in the first place.

Special thanks to my main student collaborators over the years Mingzhe Wang and Noriyuki Kojima for the countless late-night meetings, and my friends Mohamed El-Banani and Zakaria Aldeneh for all the hours we spent discussing our research and their thoughtful feedback and suggestions. I want to thank all the current and former members of our LIT lab: Steven Wilson, Veronica Pérez-Rosas, Laura Burdick, Charlie Welch, Santiago Castro, Mohamed Abouelenien, Shibamouli Lahiri, Stephane Dadian, Aparna Garimella, MeiXing Dong, Jonathan Kummerfeld, Allie Lahnala, Laura Biester, Oana Ignat, Ashkan Kazemi, Paul Bara, Carol Zheng, and other students who were around. I learned a lot from our lab meetings and office discussions.

Special thanks go to my Ann Arbor friends for all the fun times and for making the city feels like home. You have helped me keep my sanity during the grind of graduate school.

Last but not least, I could not have completed this journey without the unconditional love and support of my wonderful wife, Dina Abdelmageed. Finally, words cannot express the gratitude I have for my family and friends. My loving mother, who keeps encouraging me to pursue my dreams, and my brothers and in-laws, who were always there for me.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Stories are one of the main tools that humans use to make sense of the world around them. This ability is conjectured to be uniquely human, and concepts of agency and interaction have been found to develop during childhood. However, state-of-the-art artificial intelligence models still find it very challenging to represent or understand such information about the world. Over the past few years, there has been a lot of research into building systems that can understand the contents of images, videos, and text. Despite several advances made, computers still struggle to understand high-level discourse structures or how visuals and language are organized to tell a coherent story.

Recently, several efforts have been made towards building story understanding benchmarks. As characters are the key component around which the story events unfold, character representations are crucial for deep story understanding such as their names, appearances, and relations to other characters. As a step towards endowing systems with a richer understanding of characters in a given narrative, this thesis develops new techniques that rely on the vision, audio and language channels to address three important challenges: i) speaker recognition and identification, ii) character representation and embedding, and iii) temporal modeling of character relations. We propose a multi-modal unsupervised model for speaker naming in movies, a novel way to represent movie character names in dialogues, and a multi-modal supervised character relation classification model. We also show that our approach improves systems ability to understand narratives, which is measured using several tasks such as their ability to answer questions about stories on several benchmarks.

CHAPTER 1

Introduction

Stories are ubiquitous and are one of the essential components of human communication tools. These stories can take multiple forms such as movies, books, comic strips, and so on. Understanding these stories usually engages multiple senses such as vision, listening, common sense, and sometimes even our imagination. Thus, story understanding is essential to understanding human intelligence, which is crucial to building artificial intelligence systems that can behave on par with humans [2]. In this thesis, we focus on visual stories, particularly movies and TV shows, and aim to develop models that improve their understanding. Movies and TV shows are interesting because understanding them requires processing many sources of information at once. When people watch movies, they integrate information from the dialogues between the movie characters as well as the visual channel representing the entities and the scenes to develop a good understanding of the plots.

1.1 Language & Vision

Over the past decade, there have been rapid advances in the fields of natural language processing (NLP) and computer vision (CV). With the development of large-scale benchmarks [3, 4, 5, 6], both worlds have seen a dramatic shift from using hand-crafted features such as Scale-Invariant Feature Transform (SIFT) or template patterns [7, 8] to trainable end-to-end models for image and text representation using deep learning models such as convolutional neural networks (CNN) [9, 10, 11, 12, 13]. These advances led to building

systems that can understand the contents of text, images, and videos. For instance, reading comprehension systems can answer questions about a given narrative paragraph by extracting the text span that answers a given question [14, 15]. An example of image and video understanding is a system that recognizes and localizes objects in images or detects actions in videos [16, 17, 18, 19, 20]. Building upon these advances led to an increasing interest in tasks that involve joint analysis of both language and vision such as image captioning [21, 22], phrase localization [16, 23], and description of short video clips [24, 25]. Visual question answering (VQA) [26, 27, 28] is another task that aims to answer questions about the content of a given image. The questions asked are typically based on the visual content of the image, like the attributes of a given object (e.g., color/size), counting a specific object (e.g., how many), yes/no questions (e.g., if objects exist or not), etc. Such questions are basically tied by the static nature of images. Thus, several visual story question answering benchmarks were proposed, such as MovieQA [29], TVQA [30], PoroQA [31].

Unlike still images or text, visual stories require humans to process many sources of information. For instance, movies contain visual information (in the form of shots and scenes), and language information in the form of dialogues between characters. Also, the temporal organization of the story as a sequence of frames and the development of events through interactions between characters requires humans to integrate information across scenes and make connective inferences to build a coherent story.

1.2 Characters in Stories

Stories often revolve around characters around which the events of a story's plot unfold. These characters do not necessarily have to be human. They can be almost anything: people, animals, objects, and so on. When we read a story or watch a movie, we easily identify the story characters, interpret their relations from the dialogue or actions. As the story evolves, we start to understand the dynamics of these characters relations, emotions, inten-

tions, and can sometimes predict their next course of actions and how these actions affect the rest of the storyline. Consider the plot of the movie "There's something about Mary":

> A shy 16-year-old high-schooler *Ted* lands a prom date with his dream girl *Mary*, only to have it cut short by an embarrassing accident. He subsequently loses touch with *Mary*.
>
> Thirteen years later, *Ted* is still in love with Mary. On the advice of his friend *Dom*, he hires private detective *Healy* to track her down. *Healy* finds that she is living in Miami but *Healy* falls in love with the irresistible Mary as well. *Healy* resorts to lying and cheating but is exposed by *Mary*'s friend, *Tucker* who turns out to be also in love with *Mary*.
>
> *Ted*, aided by *Dom*, drives down to Florida to reconnect with *Mary*. *Ted* seems to have won *Mary*'s love, until an anonymous letter exposes his being less than honest about his link to *Healy*. *Ted* confronts *Healy* and *Tucker*. *Mary* breaks up with *Ted*. Then *Ted* leaves tearfully. But *Mary* forgives *Ted* after that and chases him saying that she would be happiest with him. [1]

This example illustrates the importance of characters to stories. For instance, the plot starts with introducing the two main characters, "Ted" and "Mary." It then goes on to describe the story events, introduce new characters, and explain how the story progresses. In this example, every sentence in the summary talks about at least one character of the story. It also shows how their relations evolve, how their motivations change, and how their actions affect the events. In the last paragraph of the story, "Ted" and "Mary" loved each other. Then she broke up with him after thinking that he tricked her. She forgives him after that, and they become lovers again.

As this example illustrates, character representation is essential for deep story understanding, and this representation includes information such as their names, appearances,

[1] Plot summary from: https://www.imdb.com/title/tt0129387

and relations to other characters. This thesis takes a step towards endowing systems with a richer understanding of characters in a given story by leveraging information from the visual, acoustic, and textual modalities, with a focus on visual stories such as movies and TV shows.

1.3 Research Questions

The goal of the research described in this thesis is to explore and analyze various aspects of character representation. Specifically, the thesis seeks to find answers to the following research questions:

Q1: Can we augment the representation of each entity with speaker names automatically predicted from dialogue? (Chapter 4)

Humans identify and infer character names from dialogues and can identify the speaker of each dialogue turn. Thus, this work starts with building a large speaker naming dataset, which consists of eighteen movies and six episodes of a TV show. For each video, we have human annotators watch and label each dialogue segment with the name of the characters that uttered it. Then, we develop a new multimodal model that leverages in a unified framework of the visual, speech, and textual modalities that are naturally available while watching a movie.

Q2: Can this augmented character representations be used to improve the performance of the downstream task of question answering? (Chapter 4)

To address this question, we develop a question answering system based on memory networks. We compare the accuracy of this model to answer questions when it is introduced with movie subtitles and when it is introduced with subtitles of movies in addition to the speaker names. We use an existing question answering benchmark *MovieQA* [29], which consists of 15k questions created about the plots of 408 movies. The results suggest that using speaker names significantly improved the accuracy of the question answering models.

Q3: Can we create character representations that can more effectively identify the relations between characters? (Chapter 5)

Existing models represent name mentions of characters by looking up their embeddings from Word2Vec or Glove models. Unlike regular words, for which their distance from other words in an embedding space represents their semantic relatedness, we find that names' embeddings from such models do not reflect character relatedness in the same way. To approach this problem, human judges watch the 18 movies from the speaker naming dataset, and for each pair of characters, they give a score to how related these characters are to each other. We propose a change to the existing Word2Vec models to include the context word in addition to the previous and following speakers as a part of the Word2Vec objective function. The resulting embeddings are significantly better correlated with human judgment than a regular Word2Vec model. We also evaluate the resulting embeddings on different downstream tasks such as relation classification and question answering. We also hypothesize that other channels of information such as the visual or acoustic channels carry useful information that can help representing characters better than using language alone.

Q4: Can we use these character representations to model the relation between characters over time? (Chapter 6)

Many of the representations proposed so far are static and do not consider the changes that occur to each character and how their relation to other characters change over the events of the story. We hypothesize that using character-specific memory units that would temporally encode character-related information such as emotions and interactions would outperform the use of static representation on different tasks such as predicting the flow of emotions and interactions between characters.

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews previous work in the areas of our contributions, particularly: joint modeling of language and vision, speaker identification and naming, entity embedding, inter-character relationship modeling, and story understanding. In Chapter 3, we discuss our first attempt to use linguistic cues to improve the alignment between textual and visual modalities for the phrase localization task, thus laying the ground for more advanced language/vision tasks.

In the following three chapters, we discuss different aspects for character representation. Chapter 4 explores the usage of textual, visual, and acoustic information to identify and name speakers in movies. Chapter 5 discusses how to represent the characters based on the dialogue between them and how well does the resulting embedding space correlate with human judgment about how related these characters are to each other. In Chapter 6, we discuss our next step, which aims to build a graph of relations that models each character and its relations with other characters temporally.

Finally, Chapter 7 provides a summary of the thesis and highlights the contributions made by this thesis in the field of character representation and visual story understanding.

1.5 Funding Acknowledgment

The work in this dissertation is supported by a Samsung research grant and by a DARPA grant HR001117S0026-AIDA-FP-045. Any opinions, findings, or conclusions in this dissertation are those of the authors and do not necessarily reflect the views of these organizations.

CHAPTER 2

Related Work

Recent years have witnessed great interest in combining natural language processing and computer vision after the tremendous advances in both fields. Several datasets have been created to facilitate the development and evaluation of deep neural models on tasks like image/video captioning [32, 33, 34], grounding image descriptions [33, 35, 36], and visual question answering [26, 37, 28, 38].

In this chapter, we start with an overview of previous work that jointly models language and vision. We then survey the previous work most closely related to the problems of (i) speaker naming; (ii) character embedding; and (iii) character relationship modeling. These problems are the main focus of this thesis.

2.1 Language & Vision Modeling

Different language and vision tasks require different levels of reasoning about information from textual and visual modalities to solve. For example, caption generation aims to resonate about an image or video as a whole and generate textual descriptions for them. At a finer level of granularity, tasks like phrase localization and object retrieval using natural language aim to locate visual objects based on natural language descriptions or queries [39, 40, 41, 17, 16].

Grounding image descriptions. Alignment of textual descriptions and visual scenes has been extensively studied [42, 43, 44, 33, 45, 46]. Kong et al. [44] align mentions in textual

descriptions of RGB-D scenes with object cuboids using a Markov random field. Karpathy et al. [42] generate image fragments from object detection and sentence fragments from dependency parsing, and match them in a neural embedding framework. Karphathy & Fei-Fei [43] use a similar framework but replace dependency parsing with bidirectional recurrent neural networks to generate image descriptions. It is worth noting that Karpathy et al. [42] and Karpathy & Fei-Fei [43] only evaluate the performance using proxy tasks (image retrieval and captioning) and do not evaluate the quality of matching.

Plummer et al. [33] introduced the Flickr30K Entities dataset, making it possible to evaluate image-sentence alignments directly. Using this dataset, Wang et al. [45] learn neural embeddings of phrases and regions under a large-margin objective and localize each phrase by retrieving the closest region in the embedding space. Instead of producing explicit embeddings, Rohrbach et al. [46] train a neural network to predict the compatibility of a region and a phrase directly.

Object retrieval using natural language. The task of object retrieval using natural language is to locate visual objects based on natural language queries. Guadarrama et al. [39] generate textual descriptions for each region proposal and retrieval objects by matching a query with the descriptions. In contrast, Arandjelovic and Zisserman [40] convert a query into multiple query images using Google image search and retrieve objects by matching the query images with images in a database. Hu et al. [41] use a recurrent neural network to score each object proposal given an input text query and an input image, and retrieve the objects with the highest score. The difference between object retrieval using natural language and phrase localization is that the former aims to match an image region to a whole sentence whereas the latter aims to match an image region to only one part of a sentence.

Image captioning and retrieval. There has been a large body of prior work on image captioning and retrieval. Previously proposed methods for image captioning [42, 43, 45, 47, 48, 49, 50, 51] and video captioning [52, 53] model the task as a machine translation problem, in which the visual information gets translated into text. Typical approaches

include recurrent neural networks [54, 55, 51, 42, 43], Canonical Correlation Analysis [47], encoder-decoder architectures [50], and Discriminative Component Analysis [49]. The emphasis of learning and evaluation is placed on matching images and sentences as a whole rather than matching their individual components such as regions and phrases.

Visual Question Answering. A common strategy to assess natural language and visual understanding capabilities is to answer questions about them. Visual question answering [56, 26, 37, 28] has become a very active topic. Given an image or video and a natural language question, the task is to reason about the visual content of the input and infer the correct answer to the question. Various visual question answering datasets contain questions at different levels of complexity, which require different reasoning capabilities. For example, while datasets such as VQA [26, 57] and Visual7W [38] focus on reasoning about the visual contents of a given image, other datasets require spatio-temporal reasoning to answer questions such as TGIF-QA [37, 58]. A large number of attention-based deep neural networks have been proposed to solve these problems. Generally, these models can be categorized into top-down and bottom-up attention [59, 60, 61, 62]. Typically, visual features are extracted using one or more convolution neural network layer, and the textual features are provided by encoding the question using a bi-directional recurrent neural network. The attention mechanism aims to learn a weighting for each spatial location in the output of the visual features that would help with answering the question.

Another category of visual question answering dataset focus on story understanding such as MovieQA [29], TVQA [30], PororoQA [31]. These datasets focus on developing a high-level understanding of information from the dialog and the visual channels. It is hard to model such a temporally long sequence of information using a regular recurrent network. Therefore, most of the proposed approaches tackle these tasks using deep end-to-end multimodal memory networks [29, 63, 31, 64, 65]. The context with represented using the question and several attention mechanisms have been applied to select the memory slots that can answer a given question about the movie [63, 31, 66, 65].

2.2 Speaker Naming

The problem of speaker naming in movies has been explored by the computer vision and the speech communities. In the computer vision community, the speaker naming problem is usually considered as a face/person naming problem, in which names are assigned to their corresponding faces on the screen [67, 68, 69, 70, 71]. On the other hand, the speech community considered the problem as a speaker identification problem, which focuses on recognizing and clustering speakers rather than naming them [72, 73]. In this work, we aim to solve the problem of speaker naming in movies, in which we label each segment of the subtitles with its corresponding speaker name whether the speaker's face appeared on in the video or not.

Previous work can be furthered categorized according to the type of supervision used to build character recognition and speaker recognition models: supervised vs. weakly supervised models. In the movie and television domains, utilizing scripts in addition to subtitles to obtain time-stamped speaker information was also studied in [67, 71, 69, 74]. Moreover, they utilized this information to resolve the ambiguity introduced by co-occurring faces in the same frame. Features were extracted through the period of speaking (detected via lip motion on each face). Then they assigned the face based on candidate names from the time-stamped script. Thus, these studies used speaker recognition as an essential step to construct cast-specific face classifiers. [75] extended the face identification problem to include person tracking. They utilized available face recognition results to learn clothing models for characters to identify person tracks without faces.

In [68, 70], the authors proposed a weakly supervised model depending on subtitles and a character list. They extracted textual cues from the dialog: first, second, and third person references, such as "I'm Jack", "Hey, Jack!", and "Jack left". Using a character list from IMDB, they mapped these references onto true names using minimum edit distance, and then they ascribed the references to face tracks. Other work removed the dependency on a true character list by determining all names through coreference resolution. However,

this work also depended on the availability of scripts [76]. In our model, we removed the dependency on both the true cast list and the script, which makes it easier to apply our model to other movies and TV shows.

Recent work proposed a convolutional neural network (CNN) and Long Short-Term Memory (LSTM) based learning framework to automatically learn a function that combines both facial and acoustic features [77, 78]. Using these cues, they tried to learn matching face-audio pairs and non-matching face-audio pairs. They then trained an SVM classifier on the audio-video pairings to discriminate between the non-overlapping speakers. In order to train their models, they manually identified the leading characters in two TV shows, Friends and The Big Bang Theory (BBT), and collected their face tracks and corresponding audio segments using pre-annotated subtitles. Despite the very high performance reported in these studies, it is tough to generalize their approach since it requires a lot of training data.

On the other hand, talking faces have been used to improve speaker recognition and diarization in TV shows [79, 80, 81]. In the case of [82], they modeled the problem of speaker naming as facial recognition to identify speakers in news broadcasts. This work leveraged optical character recognition to read the broadcasters' names that were displayed on the screen, requiring the faces to be already annotated. Recently, Nagrani et al. [83, 84] introduced *VoxCeleb*, a large-scale speaker identification dataset, relying on obtaining videos from Youtube and performing active speaker identification method to train a speaker verification CNN model. They used their model to identify characters in TV shows [84]. However, their method relied on matching actors pictures and names on IMDB to their appearance in the videos.

2.3 Character Embedding

Learning distributional representation of words plays an increasingly important role in representing text in many tasks [85, 86]. The existence of massive datasets allowed learning high quality word embeddings in an unsupervised way by training a neural network on some fake objectives [87, 88, 89]. A major strength of these learned word embeddings is that they are able to capture useful semantic information that can be easily used in other tasks of interest such as semantic similarity and relatedness between pair of words [87, 13, 90] and dependency parsing [86, 91]. However, these models treat names and entities no more than the tokens used to mention them. As a result, these models are unable to well represent names in narrative understanding task because the word "John" in a given story can be very different from the word "John" in another narrative. In this work, we only focus on representing character names and not the whole embedding space [92].

Ji et al, [92] approached this problem by tracking and building a dynamic representation of each entity in a given text. While this method improves the performance of entity tracking and linking in text documents, it is very hard to apply in a dialogue setting due to the sparseness of the name mentions in dialogues such as in movies [93]. Despite this problem, most of the existing story understanding work feed the model the vector representations of names based on a global model, which hinders the ability of these models to understand the dialogue [29, 63, 93]. Unlike [92, 93], we focus on representing character names in dialogue settings and learning different embeddings for characters from different story dialogues in a way that reflect the relatedness between the story characters.

Identifying and analyzing character relations in literary texts is a very well studied problem [94, 95, 96, 97]. Most of these models depend on analyzing the co-occurrence of the characters and stylistic features used while characters address each other. These models are important to summarizing, understanding, and generation of stories [96]. In this work, we use the task of character relation classification as an extrinsic evaluation task to evaluate the impact of character embeddings on this task.

2.4 Character Relationship Modeling

Relation extraction between characters in literary texts has been studied in-depth and from multiple perspectives. We roughly categorize the related work into three topics: (1) literary character analysis in novels, (2) social network analysis, and (3) movie/narrative understanding and question answering.

Literary Character Analysis. Various approaches have been proposed to analyze fictional characters and/or their relationships in novels and movies. Bamman et al. [98, 99] proposed latent variable models to learn character personas in novels and movies plot summary on Wikipedia. In their model, a character persona is defined as a set of mixtures over latent lexical classes that capture their attributes (e.g., female, 28 years old) and the stereotypical actions (e.g., "strangle", "kill"). Instead of following a character-centric approach, other methods focused on modeling interpersonal relationships between characters. Supervised methods have been proposed to classify the inter-character relations from narratives. However, they make limiting assumption that relationships are of two types cooperative vs. non-cooperative [100, 101] and sometimes assume a fixed relationship between characters within a narrative [100]. Other unsupervised methods aim to learn the evolving relations between characters in novels [97] or plot summaries [97]. These models are similar to topic models: given a narrative text and a character-pair, their proposed approach learns a sequence of relationship descriptors (topics). However, these unsupervised methods are hard to evaluate given the subjectivity nature of the task and the coherence issues of the learned descriptors. To support the training and evaluation of automatic methods for relation type prediction, manually annotated relationships between characters in literary texts dataset have been proposed [102]. For a given pair of main characters, this dataset includes four-dimensional types of relation annotations: coarse-grained relation (professional, social, familial), fine-grained relation (lovers, husband/wife), affinity (positive, negative, neutral), and to whether this relation changes or not. While we aim to infer inter-character relationships in movies based on dialog, speech, and video, previous work focused on novels and

summarized movie plots in which text provide high-level semantic information explicitly grounded in text. Therefore, our problem setting is quite different.

Social Network Analysis. Apart from literary character analysis, several social network analysis based methods have been proposed to analyze movies [103, 104, 105, 106, 107, 108]. Generally, these approaches model movie characters as nodes and the edges between the nodes represent the relation between characters. However, these methods represent limited types of relationships such as the volume of interaction [96, 106], co-occurrence [104], participation in social events [107, 109], adversarial vs. non-adversarial communities [105]. Example of these models is RoleNet [104], which constructs a social network as the accumulation of the connected graphs of characters that appear in the same scene and uses the frequency of scenes with co-appearance as the weight to classify the character role as leading vs. supporting role. Another example, Character-Net [106], it uses the volume of dialog exchange between characters as the weight to the edges regardless of the content of the dialog and classifies the role of each character as a major, minor or extra according to the centrality of a character in the constructed social network. Similarly, [105] uses support vector regression model to estimate adverseness at the scene level using extracted visual and acoustic features. Then, based on character co-occurrence in adversarial and non-adversarial scenes, they cluster the characters into two communities adversarial communities and identify the leading character within each community. These approaches are different from ours because they do not necessarily model varied aspects of inter-character relationships, and they do not model the dynamic nature of their relations.

Movie Understanding. While studying character roles and their relations is critical to computationally represent and interpret narratives and movies, recently, there has been growing interest to achieve this goal through question answering. Multiple narrative and movies question answering datasets have been created, such as MovieQA [29], TVQA [30], NarrativeQA [110], PororoQA [31]. NarrativeQA is a reading comprehension dataset in which questions are generated based on stories and their summaries and requires integrating

information and reasoning about events, entities, and their relations across a full document. Similarly, MovieQA questions are generated based on text summaries of movie plots; thus, they require a high-level understanding of the movie using the subtitles and video clips to answer. This task is very challenging and hard to solve given the types of questions. To overcome this problem, TVQA collected localized and compositional based on short video clips (around 1 minute on average) of TV shows and did not focus on character relations or plot. Rather than question answering, our work focuses on the character-centric approach and aim to infer character relations and how they evolve in movies based on reasoning about the dialog, speech, and video.

CHAPTER 3

Setting the Stage: Aligning Linguistic and Visual Modalities

3.1 Introduction

This chapter addresses the problem of phrase localization: given an image and a textual description, locate the image regions that correspond to the noun phrases in the description. [1] For example, an image may be described as "a man wearing a tan coat signs papers for another man wearing a blue coat". We wish to localize, in terms of bounding boxes, the image regions for the phrases "a man", "tan coat", "papers", "another man", and "blue coat". In other words, we wish to ground these noun phrases to image regions.

Phrase localization is an important task. Visual grounding of natural language is a critical cognitive capability necessary for communication, language learning, and the understanding of multimodal information. Specifically, understanding the correspondence between regions and phrases is important for natural language based image retrieval and visual question answering. Moreover, by aligning phrases and regions, phrase localization has the potential to improve weakly supervised learning of object recognition from massive amounts of paired images and texts.

Recent research has brought significant progress on the problem of phrase localization [33, 45, 46]. Plummer et al. introduced the Flickr30K Entities dataset, which includes

[1] The work described in this chapter was done in collaboration with Mingzhe Wang

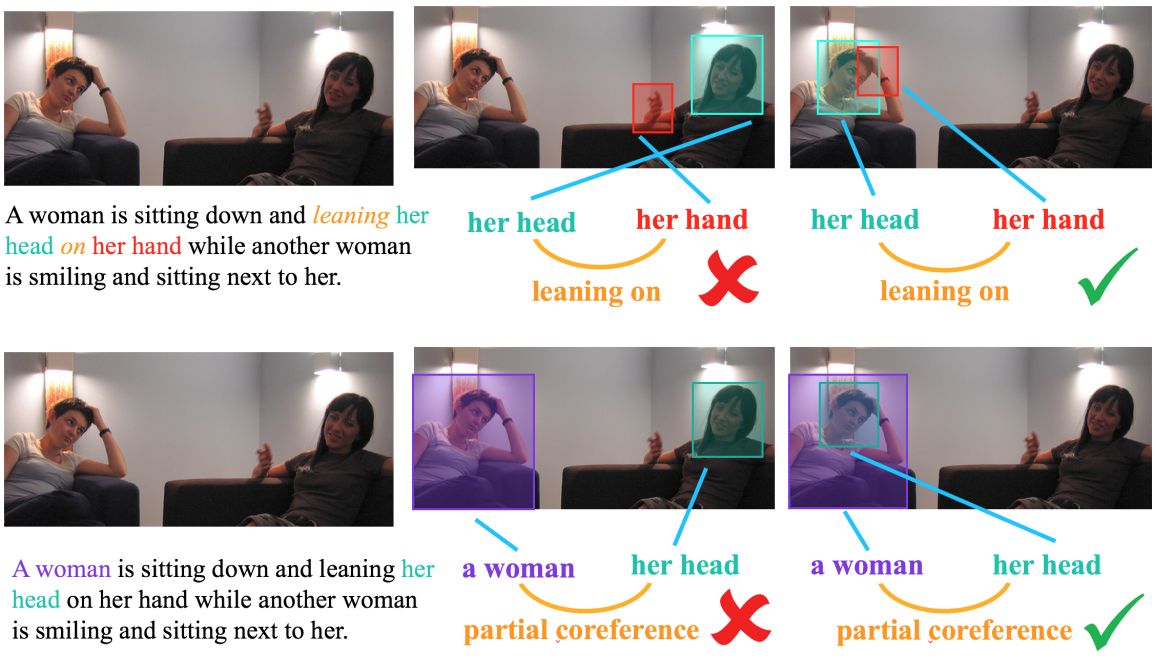


Figure 3.1: Structured matching is needed for phrase localization: it is not enough to just match phrases and regions individually; the relations between phrases also need to agree with the relations between regions.

images, captions, and ground-truth correspondences between regions and phrases [33]. To match regions and phrases, Plummer et al. embedded regions and phrases into a common vector space through Canonical Correlation Analysis (CCA) and pick a region for each phrase based on the similarity of the embeddings. Subsequent works by Wang et al. [45] and Rohrbach et al. [46] have since achieved significant improvements by embedding regions and phrases using deep neural networks.

But existing works share a common limitation: they largely localize each phrase independently, ignoring the semantic relations between phrases. The only constraint used in previous research was that different phrases should describe different regions, i.e., that each region should be matched to no more than one phrase [46]. But phrases have more complex semantic relations between each other, and phrase localization is often impossible without a deep understanding of those semantic relations. For example, in Fig. 3.1, an image from Flickr30K Entities is captioned as "a woman is sitting down and leaning her head on her hand while another woman is smiling and sitting next to her." Consider the localization of

"her head" and "her hand" from "leaning her head on her hand". There are two women, two heads, and two hands visible in the image, but only one head and one hand have a "leaning on" relation. So "her head" and "her hand" cannot be localized independently without verifying whether the head is actually leaning on the hand.

This brings forward the problem of *structured matching* of regions and phrases, that is, finding an optimal matching of regions and phrases such that not only does the visual content of each individual region agree with the meaning of its corresponding phrase (e.g. the regions must individually depict "head" and "hand"), but the visual relation between each pair of regions also agrees with the semantic relation between the corresponding pair of phrases (e.g. the pair of regions together must depict "leaning her head on her hand"). The problem of structured matching is closely related to the standard (maximum weighted) bipartite matching, although significantly harder: the nodes on the same side have relations between them, and thus not only the nodes but also the relations need to be matched with the other side.

In this chapter, we introduce a new approach to phrase localization based on the idea of structured matching. We formulate structured matching as a discrete optimization problem and relax it to a linear program. We use neural networks to embed regions and phrases into vectors, which then define the similarities (matching weights) between regions and phrases. We integrate structured matching with neural networks to enable end-to-end training. Experiments on Flickr30K Entities demonstrate the empirical effectiveness of our approach.

3.2 Approach

Figure 3.2 illustrates our approach. Given an image and a description, our goal is to localize the image region that corresponds to each phrase in the description. Following prior work [33], we assume that short noun phrases ("a man", "tan coat") have already been extracted from the description. We also assume that pronouns and non-visual phrases have

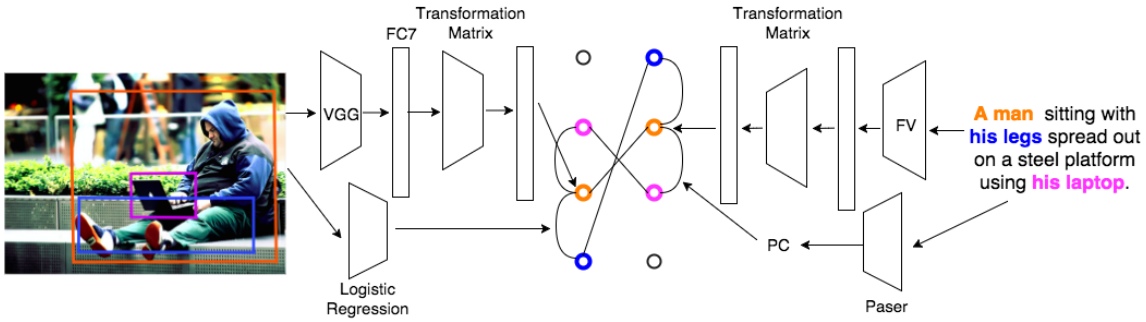


Figure 3.2: We embed regions and phrases into a common vector space and perform structured matching that encourages not only the individual agreement of regions with phrases but also the agreement of phrase-phrase relations with region-region relations. In particular, we consider "partial match coreference" (PC) relations—the relation between phrases such as "a man" and "his legs".

been removed. Also following prior work [33], we generate a set of region proposals in the form of bounding boxes.

Given these phrases and regions, the next step is to match them. To this end we adopt the same approach by Wang et al. [45]: we extract visual and phrasal features, embed them into a common vector space using neural networks, and measure region-phrase similarities in this common vector space. Using these similarities we then solve a structured matching problem: finding the optimal matching subject to two types of constraints: (1) a region can be matched to no more than one phrase, and (2) if two phrases have a certain semantic relation, their corresponding regions should have a visual relation that is consistent with the semantic relation.

If we have only the first type of constraints, we arrive at a standard maximum weighted bipartite matching problem, which can be solved exactly by linear programming. The second type of constraints, however, pose significant new difficulties because it appears intractable to obtain exact solutions. As a result, we propose a relaxation to a linear program that gives approximate solutions.

We learn end to end with a structured prediction loss. That is, the learnable parameters of our framework are jointly optimized with the objective that for each image-sentence pair in the training set the ground truth matching should have a higher score than all other

possible matchings. It is worth noting that although prior work on phrase localization has considered the constraint that a region should be matched to no more than one phrase [46], they have only used it as a post-processing heuristic, whereas we integrate this constraint into end-to-end training.

3.2.1 Representing Regions and Phrases

We generate regions proposals using Edgebox [111]. These regions serve as the candidates to be matched with phrases. To represent each region, we use Fast-RCNN [112] features, that is, features from a 16-layer VGG convolutional network that is pre-trained on the ImageNet [113] classification dataset and fine-tuned on the VOC2007 detection dataset [114]. In particular, we extract the fc7 layer activations to represent each region with a $4,096$ dimensional feature vector.

To represent phrases, we use Fisher vectors [47]. Following [47], we extract Fisher Vectors of 18,000 dimensions by first applying ICA on the 300-dimensional word2vec [115] word vectors and then constructing a codebook with 30 centers from a Hybrid Gaussian-Laplacian mixture model (HGLMM). Similar to [45], to save time during training, we apply PCA to reduce the dimensionality of the Fisher vectors from 18,000 to 6,000.

Next, we apply a linear transformation to the fc7 activations of the regions and another linear transformation to the Fisher Vectors of the phrases in order to embed them in the same vector space. That is, given for a phrase p and a region r and their feature vectors x_p and x_r, we compute the embedded features $\tilde{x}_p$ and $\tilde{x}_r$ as

$$\begin{aligned} \tilde{x}_p &= M_1 \mathbf{x}_p + b_1, \\ \tilde{x}_r &= M_2 \mathbf{x}_r + b_2, \end{aligned} \tag{3.1}$$

where M_1, M_2, b_1, b_2 are learnable parameters. We define the similarity w_{ij} between a

phrase p and a region r as

$$\cos(x_p, x_r) = \frac{< x_p, x_r >}{\|x_p\|\|x_r\|}, \tag{3.2}$$

i.e. the cosine similarity between the embedded vectors.

Given phrases $p_1, p_2 \ldots p_n$ and regions $r_1, r_2 \ldots r_m$, we obtain a similarity matrix $W_\theta = \{w_{ij}\}$, where

$$w_{ij} = \cos(x_{p_i}, x_{r_j}) \tag{3.3}$$

and θ represents the learnable parameters M_1, M_2, b_1, b_2.

3.2.2 Bipartite Matching

Given the similarities between regions and phrases, we are ready to solve the matching problem. We first consider bipartite matching, matching with only the constraint that each region should be matched to no more than one phrase. We refer to this constraint as the exclusivity constraint.

It is worth noting that in the most general case, this constraint is not always valid. Two phrases can refer to the same region: for example, “a man” and “he”, “a man” and “the man”. In these cases we need to perform coreference resolution and group the coreferences before phrase localization. Here we assume that this step has been done. This is a valid assumption for the Flickr30K Entities dataset we use in our experiments—coreferences such as “he” and “she” have been removed.

Given phrases $p_1, p_2, \ldots, p_n$, regions $r_1, r_2, \ldots, r_m$, and their similarity matrix W_θ, we have a standard bipartite matching problem if we consider only the exclusivity constraints. We first formulate this problem as an integer program.

We define a binary variable $y_{ij} \in \{0, 1\}$ for each potential region-phrase pair $\{p_i, r_j\}$ to indicate whether r_j is selected as a match for p_i in a matching y. Let $S(W, y)$ be a score that measures the goodness of a matching y, computed as the sum of similarities of the

matched pairs:

$$S(w, y) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} y_{ij}. \tag{3.4}$$

The best matching maximizes this score and can be found by a linear program that relaxes y_{ij} to continuous variables in $[0, 1]$.

$$\begin{aligned} \max_{y} & \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} y_{ij} \\ \text{s.t.} & \sum_{j=1}^{m} y_{ij} = 1, i = 1, 2, \ldots, n \\ & \sum_{i=1}^{n} y_{ij} \leq 1, j = 1, 2, \ldots, m \\ & 0 \leq y_{ij} \leq 1, i = 1, \ldots, n, j = 1, \ldots, m. \end{aligned} \tag{3.5}$$

Here, the first constraint guarantees that each phrase is matched with exactly one region, and the second constraint guarantees that each region is matched with no more than one phrase. This linear program is guaranteed to have an integer solution because all of its corner points are integers and this integer solution can be found by the simplex method.

To learn the embedding parameters, we optimize an objective that encourages the ground truth matching to have the best matching score. Let $y^{(l)}$ be the ground truth matching for the l^{th} image-sentence pair in a training set, and let $W_\theta^{(l)}$ be the region-phrase similarities. We define the training loss L as

$$L(\theta) = \sum_{l} \max(0, \max_{y'} S(W_\theta^{(l)}, y') - S(W_\theta^{(l)}, y^{(l)})), \tag{3.6}$$

where θ represents the learnable parameters. Note that although this loss involves a max operator that ranges over all possible matchings, computing the gradient of L with respect to θ only involves the best matching, which we can find by solving a linear program. It is also worth noting that this loss function is a simplified version of the structured SVM loss [116] with a margin of zero: although the ground truth matching needs to have the

highest score among all possible matchings, the score does not need to be higher by a fixed positive margin than that of the best non-ground-truth matching. With no margin requirement, this loss is not as stringent as the original structured SVM formulation but is significantly easier to implement.

3.2.3 Partial Match Coreference

We now address the matching of relations. In this work, we consider "partial match coreference" (PC) relations, a specific type of sematnic relations between phrases. Partial match coreference is composed of a noun phrase and another noun phrase including a possessive pronoun such as "his" or "her". Such relations indicate that the second phrase refers to an entity that belongs to the first phrase. For instance, the following are examples of partial match coreference:

1. **A woman** is dressed in Asian garb with a basket of goods on **her hip**.

2. **An instructor** is teaching **his students** how to escape a hold in a self-defense class.

Partial match coreference points to a strong connection between the two noun phrases, which places constraints on the visual relation between their corresponding regions. In particular, partial match coreference relations can give strong cues about the appearance and spatial arrangement of the corresponding regions. We thus use PC relations in the task of phrase localization and study if it can bring any improvements.

We extract partial match coreference relations using the Stanford CoreNLP library [117]. Since a partial match coreference is indicated by possessive pronouns such as "his" or "her", we first extract coreference relations between entity mentions in each image caption.

Among the extracted coreferences, some are "full matches", such as "he" as a coreference of "a man" where the entire phrase "he" and the entire phrase "a man" are mutual coreferences. The rest are "partial matches" such as "her hip" as a coreference of "a

woman" where only a part of the phrase "her hip", i.e. the possessive pronoun "her", is a coreference of "a woman". We discard all "full match" coreferences and keep only the "partial match" coreferences. Note that full match coreference is also useful because it indicates that two phrases should be matched to the same region. We discard them only because in Flickr30K Entities, the dataset we use for experiments, all pronouns that are full match coreferences are annotated as non-visual objects and excluded in evaluation.

3.2.4 Structured Matching with Relation Constraints

Given a certain type of semantic relations between phrases, we would like to enforce the constraint that the visual relations between regions should agree with these semantic relations. In the rest of this chapter we use the partial coreference relation as an example.

Formally, consider two arbitrary phrases p_i, p_s. Let r_j, r_t be two arbitrary regions, which are potential matches for the two phrases. Given a matching y, let $z_{ijst} \in \{0, 1\}$ be a binary variable indicating whether phrases p_i and p_s are simultaneously matched to regions r_j and r_t. In other words,

$$z_{ijst} = y_{ij} \wedge y_{st}. \tag{3.7}$$

Let $g(r_j, r_t)$ be a non-negative function that measures whether two regions r_j, r_t have a visual relation that agrees with the partial match coreference (PC) relation, that is, whether the the two regions have a "visual PC" relation.

We can now modify our matching objective to encourage the agreement between relations:

$$\max_y \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} y_{ij} + \lambda \sum_{(i,s) \in Q}^{n} \sum_{j,t}^{m} z_{ijst} g(r_j, r_t), \tag{3.8}$$

where Q is the set of all pairs of phrases with PC relations. The term $z_{ijst} g(r_j, r_t)$ makes a matching y more desirable if whenever a pair of phrases have a PC relation the corresponding pair of regions have a visual PC relation.

This new objective poses additional challenges for finding the best matching. It is an integer program that appears difficult to solve directly, and it is not obvious how to relax it to a linear program with the Boolean term $z_{ijst} = y_{ij} \wedge y_{st}$.

We propose a linear program relaxation by introducing a probabilistic interpretation. We relax the binary variables y and z into real values in $[0, 1]$. We imagine that the matching is generated through a probabilistic procedure where each phrase p_i chooses, not necessarily independently, a region from all regions according to a multinomial distribution parametrized by the relaxed, continuous variables y_{ij}. That is, we interpret y_{ij} as $\Pr(R(p_i) = r_j)$, where $R(p_i)$ represents the region chosen by phrase p_i. This interpretation naturally requires that

$$\sum_j y_{ij} = \sum_j \Pr(R(p_i) = r_j) = 1, \tag{3.9}$$

which is the same constraint used earlier in bipartite matching that requires a phrase to match with exactly one region. We also add the exclusivity constraint that each region is matched to no more than one phrase:

$$\sum_i y_{ij} = \sum_i \Pr(R(p_i) = r_j) \leq 1. \tag{3.10}$$

We treat z_{ijst} as the joint probability $\Pr(R(p_i) = r_j, R(p_s) = r_t)$, that is, the probability that we match p_i to r_j and p_s to r_t simultaneously. It follows from the rule of marginalization that

$$\begin{aligned} \sum_{t=1}^{m} z_{ijst} = &\quad \sum_t \Pr(R(p_i) = r_j, R(p_s) = r_t) = \Pr(R(p_i) = r_j) = y_{ij} \\ \sum_{j=1}^{m} z_{ijst} = &\quad \sum_j \Pr(R(p_i) = r_j, R(p_s) = r_t) = \Pr(R(p_s) = r_t) = y_{st}. \end{aligned} \tag{3.11}$$

Putting all the constraints together we have the following linear program for structured

matching:

$$
\begin{aligned}
&\max_{y \in \mathcal{Y}} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} y_{ij} + \lambda \sum_{(i,s) \in Q}^{n} \sum_{j,t}^{m} z_{ijst} g(r_j, r_t) \\
&\text{s.t.} \sum_{j=1}^{m} y_{ij} = 1, \text{for } i = 1, 2, \ldots, n \\
&\quad \sum_{i=1}^{n} y_{ij} \leq 1, \text{for } j = 1, 2, \ldots, m \\
&\quad \sum_{t=1}^{m} z_{ijst} = y_{ij} \text{ for any } i, j, s \\
&\quad \sum_{j=1}^{m} z_{ijst} = y_{st} \text{ for any } i, s, t \\
&\quad 0 \leq y_{ij} \leq 1, \text{for all } i, j \\
&\quad 0 \leq z_{ijst} \leq 1, \text{for all } i, j, s, t.
\end{aligned}
\tag{3.12}
$$

In this linear program, each pair of phrases with a partial match coreference relation p_i, p_s will lead to n^2 instances of z and g. This means that the linear program may have too many variables to be solved in a reasonable amount of time. To remedy this issue we adopt a heuristic that only applies the relation constraints to a subset of regions that are the most likely to be matched to phrases. Specifically, for a pair of phrases p_i and p_s, we only introduce z variables for the top 10 regions of p_i and the top 10 regions of p_s as measured by the cosine similarity. That is, the index j in z_{ijst} ranges over only the top 10 most similar regions of p_i and the index t ranges over only the top 10 most similar regions of p_s. This heuristic helps avoid a bloated linear program.

This linear program is easy to solve but is not guaranteed to produce an integer solution. A fractional solution indicates multiple possible regions for some phrases. In such cases we run a depth first search to enumerate all feasible solutions contained in this fractional solution and find the best matching. Since we limit the number of z variables, the search space is usually small and our approach remains efficient.

To learn or fine-tune parameters for structured matching we use the same loss function as defined in Equation 3.6, except that the matching score S is given by the solution value

of this new linear program.

We implement the "visual PC" scoring function $g(r_j, r_t)$ as a logistic regressor. For a pair of regions r_j, r_t, we concatenate their fc7 feature vectors and pass the longer feature into the logistic regressor. The parameters of this logistic regressor can be learned jointly with all other parameters of our method.

3.3 Experiments

Setup We evaluate our approach using the Flickr30k Entities dataset [33]. Flickr30k Entities is built on Flickr30K, which contains 31,783 images, each annotated with five sentences. In each sentence, the noun phrases are provided along with their corresponding bounding boxes in the image. These region-to-phrase correspondences enable the evaluation of phrase localization. There are more than 500k noun phrases (a total of 70k unique phrases) matched to 275k bounding boxes. Following [33], we divide these 31,783 image into three subsets, 1,000 images for validation, 1,000 for testing, and the rest for training. Also following [33], if a phrase is matched with multiple ground truth bounding boxes, we merge them into a new enclosing box. After this merging, every phrase has one and only one ground truth bounding box.

Following prior work [33, 45, 46], we generate 100 region proposals for each image using Edgebox [111] and localize each phrase by selecting from these regions. We select one region for each phrase and the selection is deemed correct if the region overlaps with the ground truth bounding box with an IoU (intersection over union) over 0.5. We evaluate the overall performance in terms accuracy, the percentage of phrases that are correctly matched to regions. Note that some prior works [33, 45] have reported performance in terms of recall@K: each phrase can select K regions; recall@K is 1 if one of them overlaps with the ground truth and 0 otherwise. Our definition of accuracy is the same as recall@1. We do not report recall@K with a K larger than 1 because it is unclear what it means to

select more than one region for each phrase when we jointly localize phrases subject to the exclusivity constraints and relation constraints.

We use the same evaluation code released by [33]. It is also worth noting that since each phrase can only be localized to one of the region proposals, the quality of the region proposals establishes an upperbound of performance. Consistent with prior work, with EdgeBox the upperbound in our implementation is 76.91%.

Implementation We implement the following approaches:

1. *CCA+Fast-RCNN*: We produce CCA embeddings using the code from Klein et al. [47] except we replace the VGG features pretrained on ImageNet with the Fast-RCNN features (VGG features pretrained on ImageNet and fine-tuned on the VOC2007 detection dataset).

2. *Bipartite Matching*: using the same features as CCA+Fast-RCNN, we embed the features into a common vector space through (shallow) neural networks and perform bipartite matching with exclusivity constraints.

3. *Structure Matching*: Same as Bipartite Matching except we perform structured matching with relation constraints.

In training we modify the set of candidate regions: for each image we start with the 100 region proposals from EdgeBox; then we remove those with an IoU larger than 0.5 and add ground truth bounding boxes. The reason for this modification is that the EdgeBox region proposals may not contain the ground truth matches for all phrases. This modification ensures that all ground truth matches are included and each phrase has only one ground truth region.

For all training we use Stochastic Gradient Descent (SGD) with a learning rate of 1e-4. We decrease the learning rate slightly after each epoch. We use a momentum of 0.9 and a weight decay of 0.0005. The hyperparameter λ in the matching loss (Equation 3.6) is selected on the validation set. For bipartite matching, we initialize the embedding matrices

Methods	Accuracy (Recall@1)
CCA [33]	25.30
NonlinearSP [45]	26.70 (43.89)
SCRC [41]	27.80
GroundR [46]	29.02 (47.70)
MCB [119]	(48.69)
CCA [120]	(50.89)
Ours: CCA+Fast-RCNN	39.44
Ours: Matching	41.78
Ours: Structured Matching	42.08

Table 3.1: Accuracy (Recall@1) of our approach compared to other methods. Results in parentheses were released after the submission of this work for peer review and are concurrent with our work.

Methods	accuracy (Recall@1) on PC phrases only
Bipartite Matching	47.8
Structured Matching	49.3

Table 3.2: Performance of bipartite matching and structured matching on only phrases with partial match corerference (PC) relations.

M_1, M_2 with Canonical Correlation Analysis(CCA) [118] and fine-tune all parameters end to end for 3 epochs, optimizing the matching loss defined in Equation 3.6. Since CCA provides a good initialization, the matching loss converges quickly.

For structured matching, we pre-train the "visual PC" logistic regressor using the 10,325 pairs of regions in the training set that have a ground truth "visual PC" relation and an equal number of negative pairs of regions. This pre-trained logistic regressor has an accuracy of 78% on the validation set. Then we initialize all other parameters using the pre-trained bipartite matching model and fine-tune all parameters (including those of the logistic regressor) for 2 epochs optimizing the structured matching loss with relation constraints.

Results Table 3.1 summarizes our results and compares them with related work. Table 3.3 provides accuracy of phrase localization for different categories of phrases. Fig. 3.3 and Fig. 3.4 show qualitative results including success and failure cases.

Our results show that Fast-RCNN features leads to a large boost of performance, as can been seen by comparing the CCA result from [33] with our CCA+Fast-RCNN result. Similar results have also been reported in [120] and the latest version of [45].

Also we see that Bipartite Matching further improves CCA+Fast-RCNN, which demonstrates the effectiveness of end-to-end training with our matching based loss function. Structured Matching with relation constraints provides a small additional improvement over Bipartite Matching. It is worth noting that the improvement from Structured Matching appears small partly because in the test set only 694 phrases out of a total of 17519 are involved in partial match coreference relations, limiting the maximum possible improvement when averaged over all phrases. If we consider only these 694 phrases and their accuracy as shown in table 3.2, we see that Structured Matching outperforms Bipartite Matching.

Methods	person	clothing	body parts	animals	vehicles	instruments	scene	other
CCA[33]	29.58	24.20	10.52	33.40	34.75	35.80	20.20	20.75
GroundR[46]	44.24	9.93	1.91	45.17	46.00	20.99	30.20	16.12
	(53.80)	(34.04)	(7.27)	(49.23)	(58.75)	(22.84)	(52.07)	(24.13)
CCA[120]	(64.73)	(46.88)	(17.21)	(65.83)	(68.75)	(37.65)	(51.39)	(31.77)
Ours: CCA+FRCN	55.39	32.78	16.25	53.86	48.50	19.14	28.97	23.56
Ours: Bipartite	57.94	34.43	16.44	56.56	51.50	27.16	33.42	26.23
Ours: Structured	57.89	34.61	15.87	55.98	52.25	23.46	34.22	26.23
Upperbound	89.36	66.48	39.39	84.56	91.00	69.75	75.05	67.40

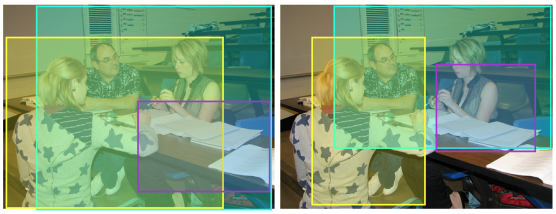
Table 3.3: Performance within categories. "Upperbound" is the maximum accuracy (recall@1) possible given the region proposals. Results in parentheses were released after the submission of our work.

3.4 Conclusion

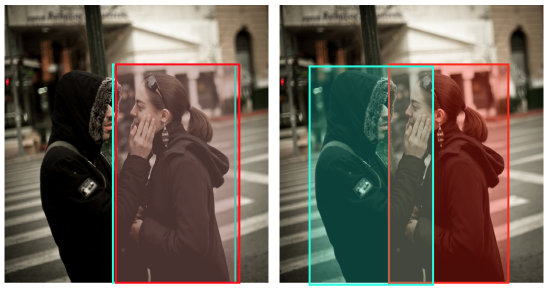
In this chapter, we have introduced a new approach to phrase localization. The key idea is a structured matching of phrases and regions that encourages the relations between phrases to agree with the relations between regions. We formulate structured matching as a discrete

optimization problem and relax it to a linear program. We integrate structured matching with neural networks to enable end-to-end training. Experiments on Flickr30K Entities have demonstrated the empirical effectiveness of our approach. Such language/vision alignments are essential for more advanced understanding tasks, such as building character representations, which we address for the remainder of this thesis.

(a) A man with a helmet is using an ATM.

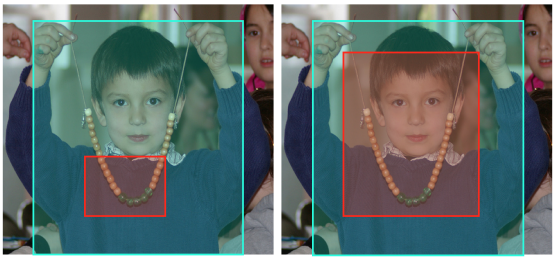


(b) Two women and a man discuss notes in a classroom.



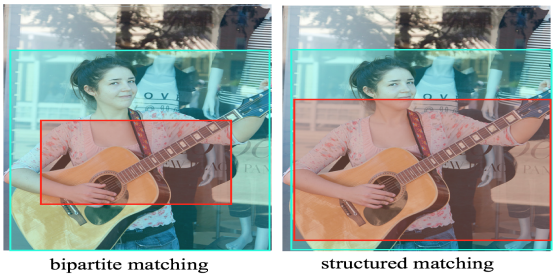
(c) A man wearing a black jacket with a woman wearing a black jacket are standing close to each other.

(d) A man and a boy holding microphones.

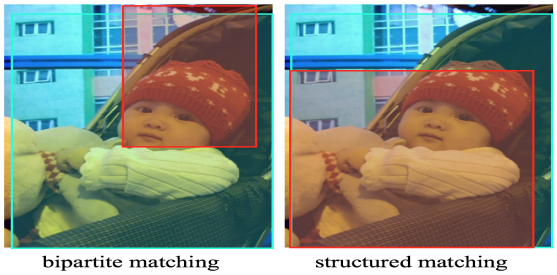


(e) A young boy shows his brown and green bead necklace.

(f) A man is working on his house by repairing the windows.

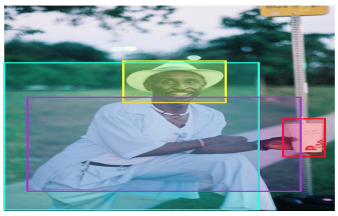


(g) This lady is wearing a pink shirt and tuning her guitar.



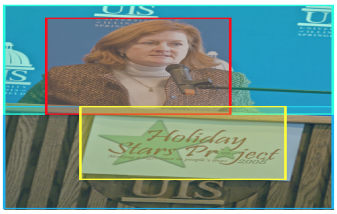
(h) A baby with red hat sit in his stroller.

Figure 3.3: Qualitative results. The first two rows compare CCA with bipartite matching. The rest compare bipartite matching with structured matching.

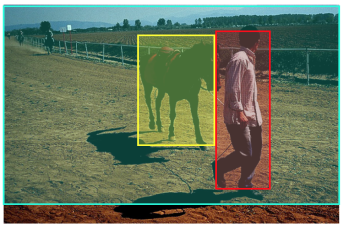


(a) A black man wearing a white suit and hat is holding a paper cup.

(b) A boy wearing an orange shirt is playing on a swing.

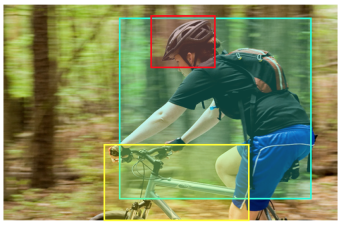


(c) A blond woman speaks at a podium labeled Holiday Stars Projectin front of a blue wall.

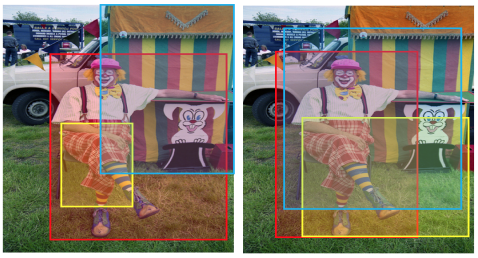


(d) A man is walking his horse on a racetrack.

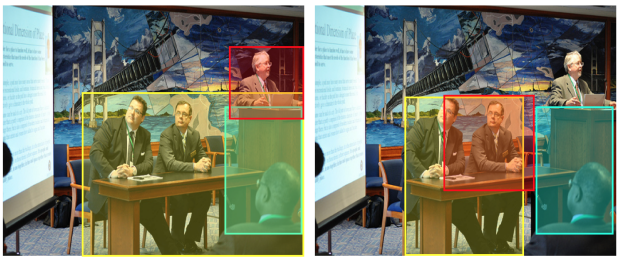
(e) A dark-haired bearded man wearing a turquoise shirt with a yellow peace sign on it.



(f) A womanwearing a black helmet riding on a bike.

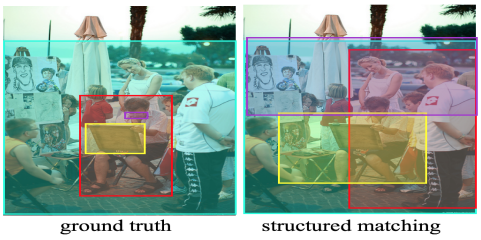


(g) A clown in red plaid overalls relaxes in the back of his tent.



(h) A man at a podium is speaking to a group of men at a conference.

(i) A man in a green shirt is jumping a ramp on his skateboard.



(j) Several people are standing on a street corner watching a cartoonist with glasses draw on his sketch pad.

Figure 3.4: Qualitative results. The first two rows compare CCA with bipartite matching. The rest compare bipartite matching with structured matching.

CHAPTER 4

Speaker Naming in Movies

4.1 Introduction

Identifying speakers and their names in movies and videos is a primary task for many video analysis problems, including automatic subtitle labeling [77], content-based video indexing and retrieval [121], video summarization [122], and video storyline understanding [122]. It is a very challenging task, as the visual appearance of the characters changes over the course of the movie due to several factors such as scale, clothing, illumination, and so forth [123, 67]. The annotation of movie data with speakers' names can be helpful in a number of applications, such as movie question answering [29], automatic identification of character relationships [121], or automatic movie captioning [77].

Most previous studies relied primarily on visual information [123, 67] and aimed for the slightly different task of face track labeling; speakers who did not appear in the video frame were not assigned any names, which is common in movies and TV shows. Other available sources of information such as scripts were only used to extract cues about the speakers' names to associate the faces in the videos with their corresponding character name [67, 71, 69, 74]; however, since scripts are not always available, the applicability of these methods is somehow limited.

Other studies focused on the problem of speaker recognition without naming, using the speech modality as a single source of information. While some of these studies attempted

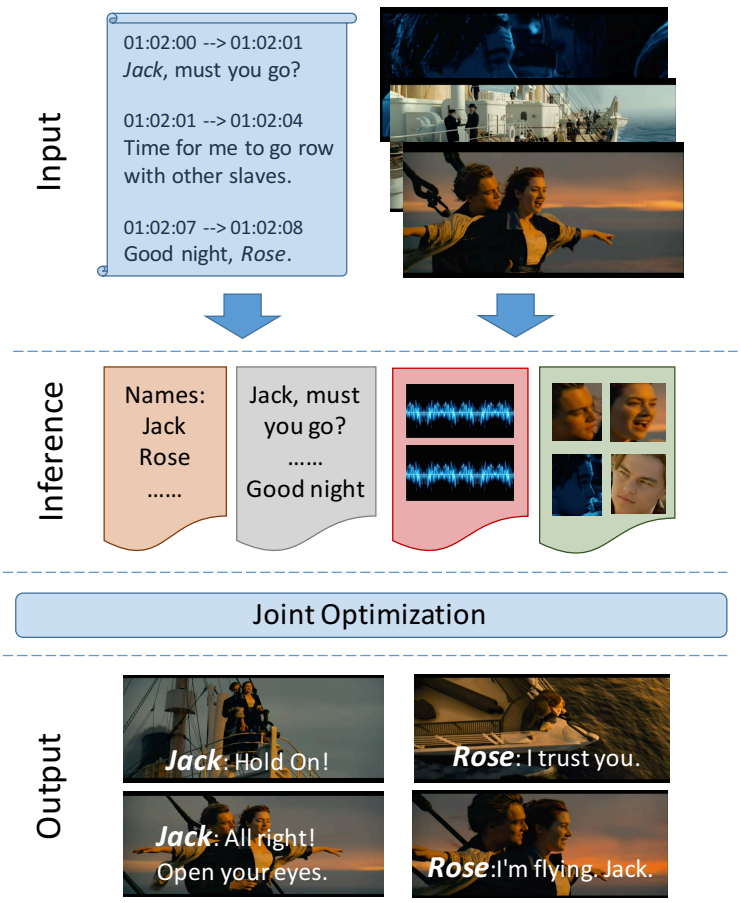


Figure 4.1: Overview of our approach for speaker naming.

to incorporate the visual modality, their goal was to cluster the speech segments rather than name the speakers [124, 80, 125, 79, 77, 78]. None of these studies used textual information (e.g., dialogue), which prevented them from identifying speaker names.

In this work, we address the task of speaker naming, and propose a new multimodal model that leverages in a unified framework of the visual, speech, and textual modalities that are naturally available while watching a movie. We do not assume the availability of a movie script or a cast list, which makes our model fully unsupervised and easily applicable to unseen movies.

In this chapter, we make two main contributions. First, we introduce a new unsupervised system for speaker naming for movies and TV shows that exclusively depends on

videos and subtitles and relies on a novel unified optimization framework that fuses visual, textual, and acoustic modalities for speaker naming. Second, we construct and make available a dataset consisting of 24 movies with 31,019 turns manually annotated with character names. Additionally, we also evaluate the role of speaker naming when embedded in an end-to-end memory network model, achieving state-of-the-art performance results on the subtitles task of the MovieQA 2017 Challenge.

4.2 Datasets

Our dataset consists of a mix of TV show episodes and full movies. For the TV show, we use six full episodes of season one of the BBT. The number of named characters in the BBT episodes varies between 5 to 8 characters per episode, and the background noise level is low. Additionally, we also acquired a set of eighteen full movies from different genres, to evaluate how our model works under different conditions. In this latter dataset, the number of named characters ranges between 6 and 37, and it has varied levels of background noise.

We manually annotated this dataset with the character name of each subtitle segment. To facilitate the annotation process, we built an interface that parses the movies subtitles files, collects the cast list from IMDB for each movie, and then shows one subtitle segment at a time along with the cast list so that the annotator can choose the correct character. Using this tool, human annotators watched the movies and assigned a speaker name to each subtitle segment. If a character name was not mentioned in the dialogue, the annotators labeled it as "unknown." To evaluate the quality of the annotations, we double annotated five movies in our dataset. The Cohen's Kappa inter-annotator agreement score for these five movies is 0.91, which shows a strong level of agreement.

To clean the data, we removed empty segments, as well as subtitle description parts written between brackets such as "[groaning]" and "[sniffing]". We also removed segments with two speakers at the same time. We intentionally avoided using any automatic means

to split these segments, to preserve the high-quality of our gold standard.

Table 4.1 shows the statistics of the collected data. Overall, the dataset consists of 24 videos with a total duration of 40.28 hours, a net dialogue duration of 21.99 hours, and a total of 31,019 turns spoken by 463 different speakers. Four of the movies in this dataset are used as a development set to develop supplementary systems and to fine-tune our model's parameters; the remaining movies are used for evaluation.

	Min	Max	Mean	σ
# characters/video	5	37	17.8	9.55
# Subtitle turns/video	488	2212	1302.4	563.06
# words/turn	1	28	8.02	4.157
subtitles duration (sec)	0.342	9.59	2.54	1.02

Table 4.1: Statistics on the annotated movie dataset.

4.3 Data Processing and Representations

We process the movies by extracting several textual, acoustic, and visual features.

4.3.1 Textual Features

We use the following representations for the textual content of the subtitles:

SkipThoughts uses a Recurrent Neural Network to capture the underlying semantic and syntactic properties, and map them to a vector representation [126]. We use their pre-trained model to compute a 4,800 dimensional sentence representation for each line in the subtitles.[1]

TF-IDF is a traditional weighting scheme in information retrieval. We represent each subtitle as a vector of tf-idf weights, where the length of the vector (i.e., vocabulary size) and the idf scores are obtained from the movie including the subtitle.

[1] https://github.com/ryankiros/skip-thoughts

4.3.2 Acoustic Features

For each movie in the dataset, we extract the audio from the center channel. The center channel is usually dedicated to the dialogue in movies, while the other audio channels carry the surrounding sounds from the environment and the musical background. Although doing this does not fully eliminate the noise in the audio signal, it still improves the speech-to-noise ratio of the signal. When a movie has stereo sound (left and right channels only), we down-mix both channels of the stereo stream into a mono channel.

In this work, we use the timestamps of the subtitles as an estimate of the boundaries that correspond to the uttered speech segments. Usually, each subtitle corresponds to a segment being said by a single speaker. We use the subtitle timestamps for segmentation so that we can avoid automatic speaker diarization errors and focus on the speaker naming problem.

To represent the relevant acoustic information from each spoken segment, we use iVectors, which is the state-of-the-art unsupervised approach in speaker verification [127]. While other deep learning-based speaker embeddings models also exist, we do not have access to enough supervised data to build such models. We train unsupervised iVectors for each movie in the dataset, using the iVector extractor used in [128]. We extract iVectors of size 40 using a Gaussian Mixture Model-Universal Background Model (GMM-UBM) with 512 components. Each iVector corresponds to a speech segment uttered by a single speaker. We fine-tune the size of the iVectors and the number of GMM-UBM components using the development dataset.

4.3.3 Visual Features

We detect faces in the movies every five frames using the recently proposed MTCNN [129] model, which is pre-trained for face detection and facial landmark alignment. Based on the results of face detection, we apply the forward and backward tracker with an implementation of the Dlib library [130, 131] to extract face tracks from each video clip. We represent a face track using its best face in terms of detection score, and use the activations of the fc7

layer of pre-trained VGG-Face [132] network as visual features.

We calculate the distance between the upper lip center and the lower lip center based on the 68-point facial landmark detection implemented in the Dlib library [130, 133]. This distance is normalized by the height of face bounding boxes and concatenated across frames to represent the amount of mouth opening. A human usually speaks with lips moving with a certain frequency (3.75 Hz to 7.5 Hz used in this work) [71]. We apply a band-pass filter to amplify the signal of true lip motion in these segments. The overall sum of lip motion is used as the score for the talking face.

4.4 Unified Optimization Framework

We tackle the problem of speaker naming as a transductive learning problem with constraints. In this approach, we want to use the sparse positive labels extracted from the dialogue and the underlying topological structure of the rest of the unlabeled data. We also incorporate multiple cues extracted from both textual and multimedia information. A unified learning framework is proposed to enable joint optimization over the automatically labeled and unlabeled data, along with multiple semantic cues.

4.4.1 Character Identification and Extraction

In this work, we do not consider the set of character names as given because we want to build a model that can be generalized to unseen movies. This strict setting adds to the problem's complexity. To extract the list of characters from the subtitles, we use the Named Entity Recognizer (NER) in the Stanford CoreNLP toolkit [134]. The output is a long list of person names that are mentioned in the dialogue. This list is prone to errors including, but not limited to, nouns that are misclassified by the NER as person's name such as "Dad" and "Aye", names that are irrelevant to the movie such as "Superman" or named animals, or uncaptured character names.

To clean the extracted names list of each movie, we cluster these names based on string minimum edit distance and their gender. From each cluster, we then pick a name to represent it based on its frequency in the dialogue. The result of this step consists of name clusters along with their distribution in the dialogue. The distribution of each cluster is the sum of all the counts of its members. To filter out irrelevant characters, we run a name reference classifier, which classifies each name into first, second, or third person references. If a name is only mentioned as a third person throughout the whole movie, we discard it from the list of characters. We remove any name cluster that has a total count of less than three, which takes care of the misclassified names' reference types.

4.4.2 Grammatical Cues

We use the subtitles to extract the name mentions in the dialogue. These mentions allow us to obtain cues about the speaker name and the absence or the presence of the mentioned character in the surrounding subtitles. Thus, they affect the probability that the mentioned character is the speaker or not. We follow the same name reference categories used in [68, 70]. We classify a name mention into: first (e.g., "I'm Sheldon"), second (e.g., "Oh, hi, Penny") or third person reference (e.g., "So how did it go with Leslie?"). The first person reference represents a positive constraint that allows us to label the corresponding iVector of the speaker and his face if it exists during the segment duration. The second person reference represents a multi-instance constraint that suggests that the mentioned name is one of the characters that are present in the scene, which increases the probability of this character to be one of the speakers of the surrounding segments. On the other hand, the third person reference represents a negative constraint, as it suggests that the speaker does not exist in the scene, which lowers the character probability of the character being one of the speakers of the next or the previous subtitle segments.

To identify first, second, and third person references, we train a linear support vector classifier. The first person, the second and third person classifier's training data are

extracted and labeled from our development dataset, and fine-tuned using 10-fold cross-validation. Table 4.2 shows the results of the classifier on the test data. The average numbers of first, second, and third-person references in each movie are 14.63, 117.21, and 95.71, respectively.

	Precision	Recall	F1-Score
First Person	0.625	0.448	0.522
Second Person	0.844	0.863	0.853
Third Person	0.806	0.806	0.806
Average / Total	0.819	0.822	0.820

Table 4.2: Performance metrics of the reference classifier on the test data.

4.4.3 Unified Optimization Framework

Given a set of data points that consist of l labeled[2] and u unlabeled instances, we apply an optimization framework to infer the best prediction of speaker names. Suppose we have $l+u$ instances $X = \{x_1, x_2, ..., x_l, x_{l+1}, ..., x_{l+u}\}$ and K possible character names. We also get the dialogue-based positive labels y_i for instances x_i, where y_i is a k-dimension one-hot vector and $y_i^j = 1$ if x_i belongs to the class j, for every $1 \leq i \leq l$ and $1 \leq j \leq K$. To name each instance x_i, we want to predict another one-hot vector of naming scores $f(x_i)$ for each x_i, such that $\text{argmax}_j f^j(x_i) = z_i$ where z_i is the ground truth number of class for instance x_i.

To combine the positive labels and unlabeled data, we define the objective function for predictions f as follows:

$$\begin{aligned} L_{initial}(f) = \frac{1}{l}\sum_{i=1}^{l} ||f(x_i) - y_i||^2 \\ + \frac{1}{l+u}\sum_{i=1}^{l+u}\sum_{j=1}^{l+u} w_{ij}||f(x_i) - f(x_j)||^2 \end{aligned} \tag{4.1}$$

[2]Note that in our setup, all the labeled instances are obtained automatically, as described above.

Here w_{ij} is the similarity between x_i and x_j, which is calculated as the weighted sum of textual, acoustic, and visual similarities. The inverse Euclidean distance is used as a similarity function for each modality. The weights for different modalities are selected as hyperparameters and tuned on the development set. This objective leads to a convex loss function which is easier to optimize over feasible predictions.

Besides the positive labels obtained from first person name references, we also introduce other semantic constraints and cues to enhance the power of our proposed approach. We implement the following four types of constraints:

Multiple Instance Constraint. Although the second person references cannot directly provide positive constraints, they imply that the mentioned characters have high probabilities of being in this conversation. Following previous work [68], we incorporate the second person references as multiple instances constraints into our optimization: if x_i has a second person reference j, we encourage j to be assigned to its neighbors, i.e., its adjacent subtitles with similar timestamps. For the implementation, we simply include multiple instances constraints as a variant of positive labels with decreasing weights s, where $s = 1/(l - i)$ for each neighbor x_l.

Negative Constraint. For the third person references, the mentioned characters may not occur in the conversation and movies. So we treat them as negative constraints, which means they imply that the mentioned characters should not be assigned to corresponding instances. This constraint is formulated as follows:

$$L_{neg}(f) = \sum_{(i,j)\in N} [f^j(x_i)]^2 \tag{4.2}$$

where N is the set of negative constraints x_i doesn't belong class j.

Gender Constraint. We train a voice-based gender classifier by using the subtitles segments from the four movies in our development dataset (5,543 segments of subtitles). We

use the segments in which we know the speaker's name and manually obtain the ground truth gender label from IMDB. We extract the signal energy, 20 Mel-frequency cepstral coefficients (MFCCs) along with their first and second derivatives, in addition to time- and frequency-based absolute fundamental frequency (f0) statistics as features to represent each segment in the subtitles. The f0 statistics have been found to improve the automatic gender detection performance for short speech segments [135], which fits our case since the median duration of the dialogue turns in our dataset is 2.6 seconds.

The MFCC features are extracted using a step size of 16 msec over a 64 msec window using the method from [136], while the f0 statistics are extracted using a step size of 25 msec over a 50 msec window as the default configuration in [137]. We then use these features to train a logistic regression classifier using the Scikit-learn library [138]. The average accuracy of the gender classifier on a 10-fold cross-validation is 0.8867.

Given the results for the gender classification of audio segments and character names, we define the gender loss to penalize inconsistency between the predicted gender and character names:

$$\begin{aligned} L_{gender}(f) = & \sum_{(i,j)\in Q_1} P_{ga}(x_i)(1 - P_{gn}(j))f^j(x_i) \\ & + \sum_{(i,j)\in Q_2} (1 - P_{ga}(x_i))P_{gn}(j)f^j(x_i) \end{aligned} \tag{4.3}$$

where $P_{ga(x_i)}$ is the probability for instance x_i to be a male, and $P_{gn(j)}$ is the probability for name j to be a male, and $Q_1 = \{(i,j)|P_{ga}(x_i) < 0.5, P_{gn}(j) > 0.5\}$, $Q_2 = \{(i,j)|P_{ga}(x_i) > 0.5, P_{gn}(j) < 0.5\}$.

Distribution Constraint. We automatically analyze the dialogue and extract the number of mentions of each character in the subtitles using Stanford CoreNLP and string matching to capture names that are missed by the named entity recognizer. We then filter the resulting counts by removing third person mention references of each name as we assume that this character does not appear in the surrounding frames. We use the results to estimate the

distribution of the speaking characters and their importance in the movies. The main goal of this step is to construct a prior probability distribution for the speakers in each movie.

To encourage our predictions to be consistent with the dialogue-based priors, we penalize the square error between the distributions of predictions and name mentions priors in the following equation:

$$L_{dis}(f) = \sum_{j=1}^{K} (\sum (f^j(x_i)) - d_j)^2 \tag{4.4}$$

where d_j is the ratio of name j mentions in all subtitles.

Final Framework. Combining the loss in Eqn. 4.1 and multiple losses with different constraints, we obtain our unified optimization problem:

$$\begin{aligned} f^* = \arg\min_f \lambda_1 L_{initial}(f) + \lambda_2 L_{MI}(f) \\ + \lambda_3 L_{neg}(f) + \lambda_4 L_{gender}(f) + \lambda_5 L_{dis}(f) \end{aligned} \tag{4.5}$$

All of the λs are hyper-parameters to be tuned on the development set. We also include the constraint that predictions for different character names must sum to 1. We solve this constrained optimization problem with projected gradient descent (PGD). Our optimization problem in Eqn. 4.5 is guaranteed to be a convex optimization problem and therefore projected gradient descent is guaranteed to stop with global optima. PGD usually converges after 800 iterations.

4.5 Evaluation

We model our task as a classification problem, and use the unified optimization framework described earlier to assign a character name to each subtitle.

Since our dataset is highly unbalanced, with a few main characters usually dominating the entire dataset, we adopt the weighted F-score as our evaluation metric, instead of using

an accuracy metric or a micro-average F-score. This allows us to take into account that most of the characters have only a few spoken subtitle segments, while at the same time placing emphasis on the main characters. This sometimes leads to an average weighted F-score that is not between the average precision and recall.

	Precision	Recall	F-score
B1: MFMC	0.0910	0.2749	0.1351
B2: DRA	0.2256	0.1819	0.1861
B3: Gender-based DRA	0.2876	0.2349	0.2317
Our Model (Skip-thoughts)*	0.3468	0.2869	0.2680
Our Model (TF-IDF)*	0.3579	0.2933	0.2805
Our Model (iVectors)	0.2151	0.2347	0.1786
Our Model (Visual)*	0.3348	0.2659	0.2555
Our Model (Visual+iVectors)*	0.3371	0.2720	0.2617
Our Model (TF-IDF+iVectors)*	0.3549	0.2835	0.2643
Our Model (TF-IDF+Visual)*	0.3385	0.2975	0.2821
Our Model (all)*	**0.3720**	**0.3108**	**0.2920**

Table 4.3: Comparison between the average of macro-weighted average of precision, recall and f-score of the baselines and our model. * means statistically significant (t-test p-value < 0.05) when compared to baseline B3.

One aspect that is important to note is that characters are often referred to using different names. For example, in the movie "The Devil's Advocate," the character Kevin Lomax is also referred to as Kevin or Kev. In more complicated situations, characters may even have multiple identities, such as the character Saul Bloom in the movie "Ocean's Eleven," who pretends to be another character named Lyman Zerga. Since our goal is to assign names to speakers, and not necessarily solve this coreference problem, we consider the assignment of the subtitle segments to any of the speaker's aliases to be correct. Thus, during the evaluation, we map all the characters' aliases from our model's output to the names in the ground truth annotations. Our mapping does not include other referent nouns such as "Dad," "Buddy," etc.; if a segment gets assigned to any such terms, it is considered a misprediction.

We compare our model against three baselines:

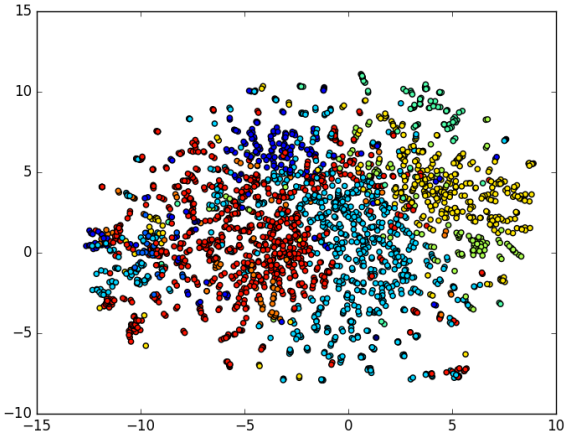
B1: Most-frequently mentioned character consists of selecting the most frequently mentioned character in the dialogue as the speaker for all the subtitles. Even though it is a simple baseline, it achieves an accuracy of 27.1%, since the leading characters tend to speak the most in the movies.

B2: Distribution-driven random assignment consists of randomly assigning character names according to a distribution that reflects their fraction of mentions in all the subtitles.

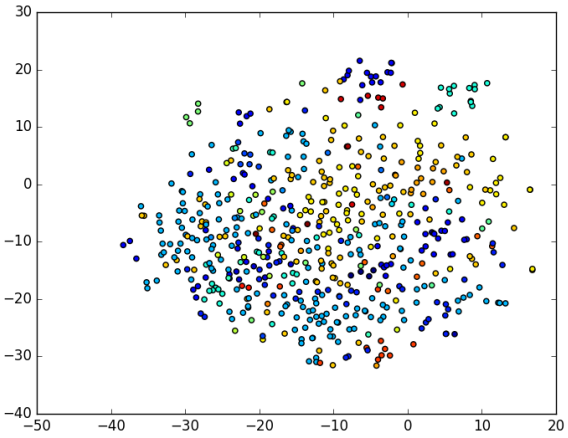
B3: Gender-based distribution-driven random assignment consists of selecting the speaker names based on the voice-based gender detection classifier. This baseline randomly selects the character name that matches the speaker's gender according to the distribution of mentions of the names in the matching gender category.

The results obtained with our proposed unified optimization framework, and the three baselines are shown in Table 4.3. We also report the performance of the optimization framework using different combinations of the three modalities. The model that uses all three modalities achieves the best results and outperforms the strongest baseline (B3) by more than 6% absolute in average weighted F-score. It also significantly outperforms the usage of the visual and acoustic features combined, which have been frequently used together in previous work, suggesting the importance of textual features in this setting.

The ineffectiveness of the iVectors might be a result of the background noise and music, which are difficult to remove from the speech signal. Figure 4.2 shows the t-Distributed Stochastic Neighbor Embedding (t-SNE) [1], which is a nonlinear dimensionality reduction technique that models points in such a way that similar vectors are modeled by nearby points and dissimilar objects are modeled by distant points, visualization of the iVectors over the whole BBT show and the movie "Titanic." In the BBT, there is almost no musical background or background noise, while, Titanic has a musical background in addition to the background noise such as the screams of the drowning people. From the graph, the difference between the quality of the iVectors clusters on different noise-levels is clear. Similarly, the short duration of the speech segment might have a negative impact on the



(a) The Big Bang Theory Season One

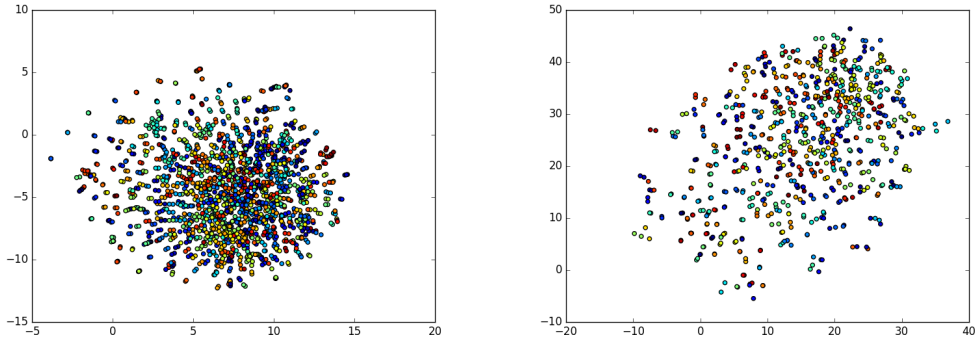


(b) Titanic

Figure 4.2: For each speech segment, we applied t-SNE [1] on their corresponding iVectors. The points with the same color represent instances with the same character name.

quality of the iVectors. The average utterance duration is 2.54 seconds. Figure 4.3 shows the t-SNE visualization of the entire BBT show iVectors of segments shorter than 3 seconds vs. segments longer than 3 seconds.

Table 4.4 shows the effect of adding components of our loss function to the initial loss L_{init} function. The performance of the model using only L_{init} without the other parts is very low due to the sparsity of first person references and errors that the person reference classifier introduces.



(a) Segments of duration less than 3 seconds (b) Segments of duration more than 3 seconds

Figure 4.3: For each speech segment in the BBT TV show, we applied t-SNE [1] on their corresponding iVectors. In each figure, the points with the same color represent instances with the same character name.

	Precision	Recall	F-score
$L_{initial}$	0.0631	0.1576	0.0775
$L_{initial}$ + L_{gender}	0.1160	0.1845	0.1210
$L_{initial}$ + $L_{negative}$	0.0825	0.0746	0.0361
$L_{initial}$ + $L_{distribution}$	0.1050	0.1570	0.0608
$L_{initial}$ + $L_{MultipleInstance}$	0.3058	0.2941	0.2189

Table 4.4: Analysis of the effect of adding each component of the loss function to the initial loss.

4.6 Additional Analyses

To gain further insights into the system, we perform several analyses.

First, we look at the performance of the system when considering the top two characters returned by our system. That is, if the correct character is listed in this top two, the system receives full credit. The average weighted precision, recall, and F-score grow to 0.5069, 0.4230, and 0.4177, which indicates there are many cases where the system finds the correct character as the second option.

Second, in order to analyze the effect of the errors that several of the used modules (e.g., gender and name reference classifiers) propagate into the system, we test our framework by replacing each one of the components with its ground truth information. As seen

in Table 4.5, the results obtained in this setting show significant improvement with the replacement of each component in our framework, which suggests that additional work on these components will have positive implications on the overall system.

	Precision	Recall	F-score
Our Model	0.3720	0.3108	0.2920
Voice Gender (VG)	0.4218	0.3449	0.3259
VG + Name Gender (NG)	0.4412	0.3790	0.3645
VG + NG + Name Ref	0.4403	0.3938	0.3748

Table 4.5: Comparison between our model while replacing different components with their ground truth information.

Third, to isolate the role played by each of the constraints used in the optimization model, we perform an evaluation where we add to the initial loss function one constraint at a time. Table 4.4 shows the results of these evaluations. As seen in this table, the biggest improvement in our framework is brought by the multiple instance constraint, followed by the gender constraint. That is expected because the first person reference instances are rare and the only constraint that provides weak supervision is the multiple instance constraint, which uses the second person references for the labels. Then the gender loss makes sure that the speaker is not assigned a name of an opposite gender.

Finally, we also analyze the effect of the number of characters on the performance of our framework. The number of characters in the movies in our dataset varies from 5 to 37 characters. Figure 4.4 shows the performance of our model on videos categorized according to the number of their named speakers. As expected, the performance is better when there is a limited number of characters, however, from the figure, we can see that the performance does not decrease by much after a certain number of characters. This could be explained by the fact that even movies with many characters are dominated by a few leading characters, and a large number of characters that only utter a few subtitle segments is not significantly affecting the performance of the system.

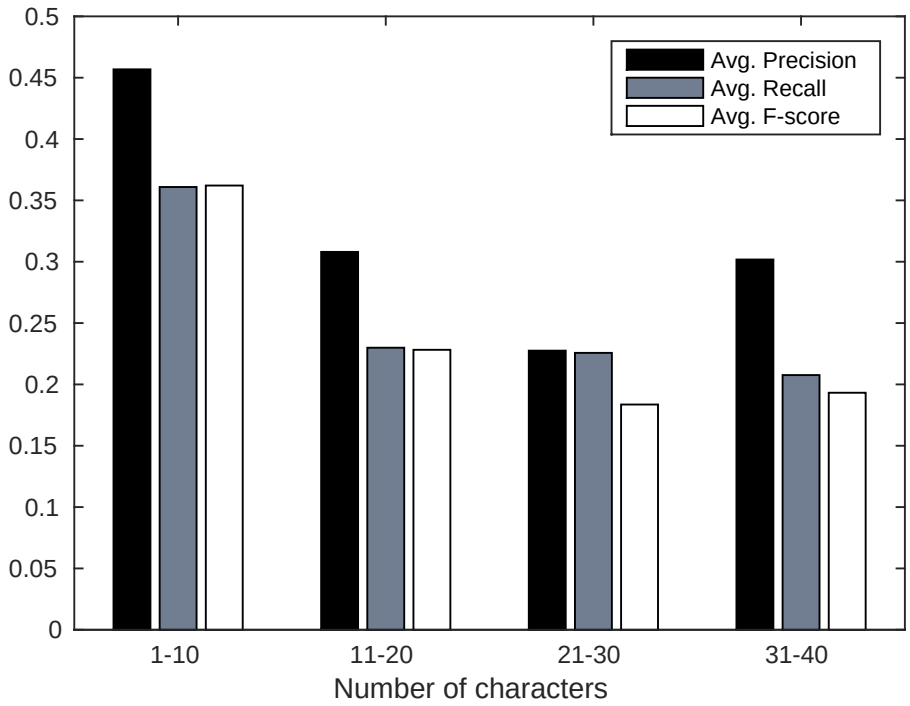


Figure 4.4: The average weighted precision, recall and f-score of our model across videos with different number of characters.

4.7 Speaker Naming for Movie Understanding

Identifying speakers is a critical task for understanding the dialogue and storyline in movies. MovieQA is a challenging dataset for movie understanding. The dataset consists of 14,944 multiple choice questions about 408 movies. Each question has five answers, and only one of them is correct. The dataset is divided into three splits: train, validation, and test according to the movie titles. Importantly, there are no overlapping movies between the splits. Table 4.6 shows examples of the question and answers in the MovieQA dataset.

The MovieQA 2017 Challenge[3] consists of six different tasks according to the source of information used to answer the questions. Given that for many of the movies in the dataset the videos are not entirely available, we develop our initial system so that it only relies on the subtitles; we thus participate in the challenge subtitles task, which includes

[3]http://movieqa.cs.toronto.edu/workshops/iccv2017/

Movie	Question	Answers	
Fargo	What did Mike's wife, as he says, die from?	A1: She was killed A3: Leukemia A5: Complications due to child birth	**A2: Breast cancer** A4: Heart disease
Titanic	What does Rose ask Jack to do in her room?	A1: Sketch her in her best dress A3: Take a picture of her nude A5: Take a picture of her in her best dress	A2: Sketch her nude **A4: Paint her nude**

Table 4.6: Example of questions and answers from the MQA benchmark. The answers in bold are the correct answers to their corresponding question.

the dialogue (without the speaker information) as the only source of information to answer questions.

To demonstrate the effectiveness of our speaker naming approach, we design a model based on an end-to-end memory network [139], namely Speaker-based Convolutional Memory Network (SC-MemN2N), which relies on the MovieQA dataset and integrates the speaker naming approach as a component in the network. Specifically, we use our speaker naming framework to infer the name of the speaker for each segment of the subtitles and prepend the predicted speaker name to each turn in the subtitles.[4] To represent the movie subtitles, we represent each turn in the subtitles as the mean-pooling of a 300-dimension pre-trained Word2Vec [12] representation of each word in the sentence. We similarly represent the input questions and their corresponding answers. Given a question, we use the SC-MemN2N memory to find an answer. For questions asking about specific characters, we keep the memory slots that have the characters in question as speakers or mentioned in and mask out the rest of the memory slots. Figure 4.5 shows the architecture of our model.

Table 4.7 includes the results of our system on the validation and test sets, along with the best systems introduced in previous work, showing that our SC-MemN2N achieves the best performance. Furthermore, to measure the effectiveness of adding the speaker names and masking, we test our model after removing the names from the network (C-MemN2N).

[4]We strictly follow the challenge rules, and only use text to infer the speaker names.

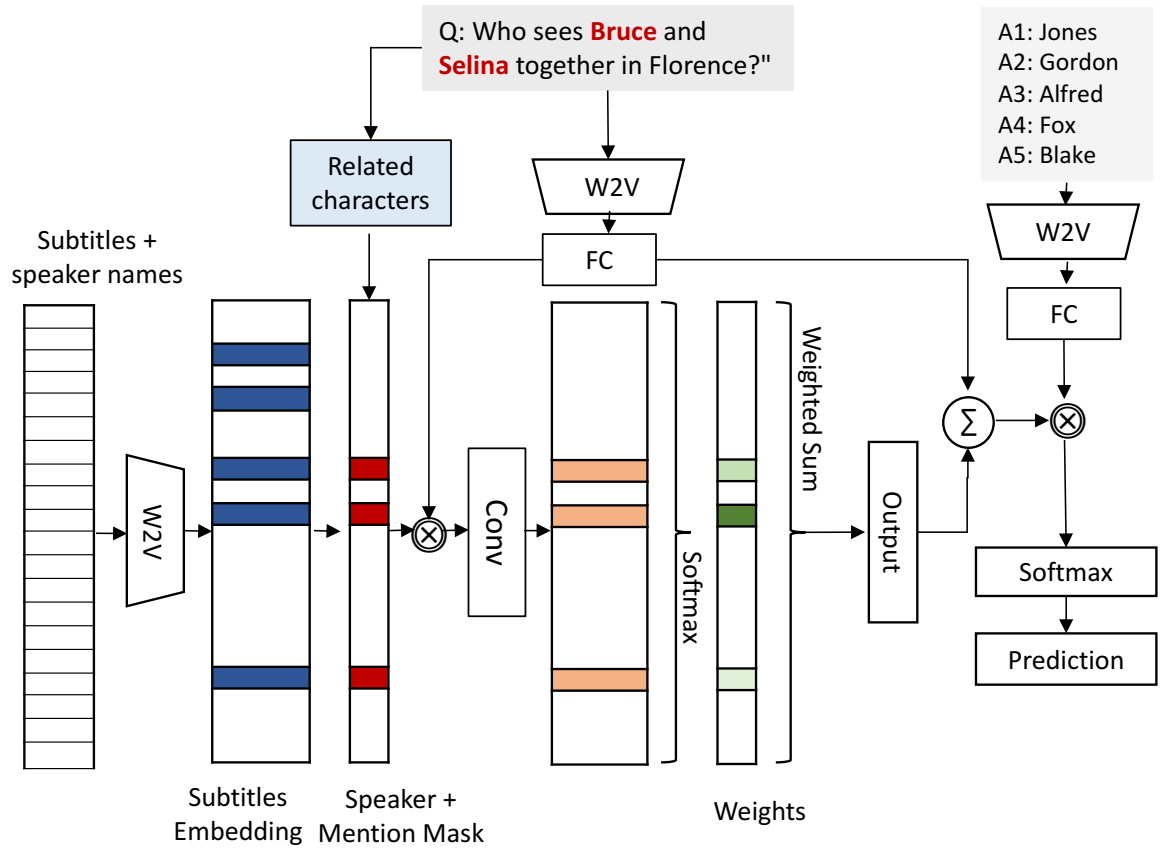


Figure 4.5: The diagram describing our Speaker-based Convolutional Memory Network (SC-MemN2N) model.

Method	Subtitles		Video+Subtitles	
	val	test	val	test
SSCB-W2V [29]	24.8	23.7	-	-
SSCB-TF-IDF [29]	27.6	26.5	-	-
SSCB Fusion [29]	27.7	-	-	21.9
MemN2N [29]	38.0	36.9	-	34.2
Understanding visual regions	-	37.4	-	34.34
RWMN [63]	40.4	38.5	38.67	36.25
Local Average Pooling Networks	-	-	-	38.16
C-MemN2N (w/o SN)	40.6	-	38.4	-
SC-MemN2N (Ours)	**42.7**	**39.4**	**40.8**	**38.16**

Table 4.7: Performance comparison for the subtitles task on the MovieQA 2017 Challenge on both validation and test sets. We compare our models with the existing models (from the challenge leaderboard). (-) means that we do not have the numbers for this on the dataset.

As seen from the results, the gain of SC-MemN2N is statistically significant[5] compared to a version of the system that does not include the speaker names (C-MemN2N). Figure 4.6 shows the performance of both C-MemN2N and SC-MemN2N models by question type. The results suggest that our speaker naming helps the model better distinguish between characters, and that prepending the speaker names to the subtitle segments improves the ability of the memory network to correctly identify the supporting facts from the story that answers a given question.

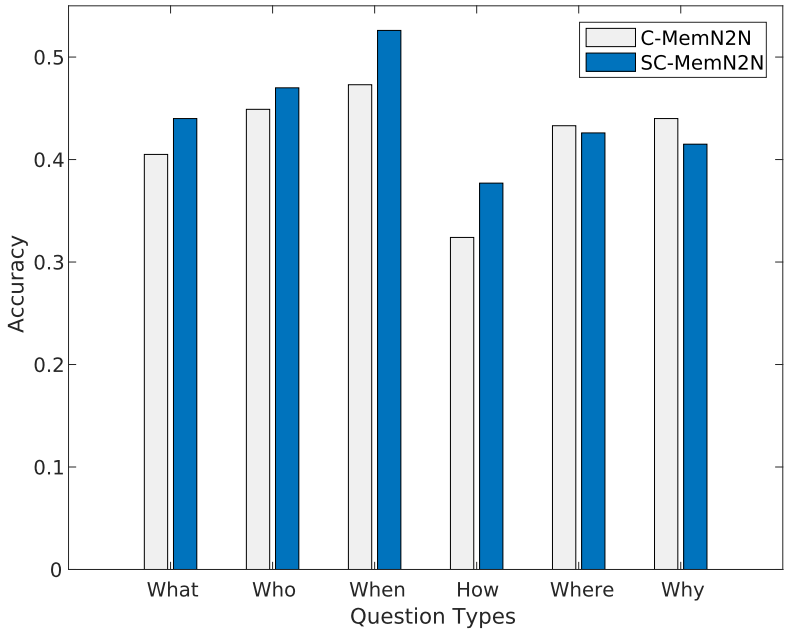


Figure 4.6: Accuracy comparison according to question type.

4.8 Conclusion

In this chapter, we proposed a unified optimization framework for the task of speaker naming in movies. We addressed this task under a difficult setup, without a cast-list, without supervision from a script, and dealing with the complicated conditions of real movies. Our

[5]Using a t-test p-value$<$0.05 with 1,000 folds each containing 20 samples.

model includes textual, visual, and acoustic modalities, and incorporates several grammatical and acoustic constraints. Empirical experiments on a movie dataset demonstrated the effectiveness of our proposed method with respect to several competitive baselines. We also showed that an SC-MemN2N model that leverages our speaker naming model can achieve state-of-the-art results on the subtitles task of the MovieQA 2017 Challenge. The dataset annotated with character names introduced in this chapter is publicly available from http://lit.eecs.umich.edu/downloads.html.

CHAPTER 5

Representing Movie Characters in Dialogues

5.1 Introduction

Understanding characters (or more broadly people) plays a critical role in the human-level interpretation of dialogues – be those in stories, movies, or day-to-day conversations. The verbal interaction between characters provides important information [97, 96]. In these contexts, the names of characters trigger reasoning at a much deeper level than other regular words, due to the character background, behaviors, social network, and so forth. Currently, the most commonly used word embedding models such as Word2Vec [87, 88] and Glove [13] represent characters using the embeddings corresponding to the tokens used to name them. Using these models in a dialogue setting to represent the characters poses three main issues. First, name mentions in dialogues are sparse [93], which makes it difficult for these models to learn a good quality representation for these names [140]. Second, in dialogues or narratives, names often do not refer to the same person, and yet these embeddings have a single vector representation for each word in the vocabulary. For example, "Danny" in the dialogue of the "American History X" movie is different from "Danny" in the "Ocean's Eleven" movie. Finally, the learned embeddings of these names reflect the co-occurrences of these name mentions and other words uttered by these characters, but do not model how related these characters are. Thus, the resulting embeddings cannot be effectively used to further reason about the characters and their relations.

Henry:	I did not know you could fly a plane.
Indiana:	Fly yes. Land no. Dad, you have to use the machine gun. Get it ready. Eleven o'clock!
Henry:	What happens at eleven o'clock?
Indiana:	Twelve, eleven, ten. Eleven o'clock, fire! Dad, are we hit?
Henry:	More or less. Son, I am sorry. They got us.
Indiana:	Hang on, dad. We are going in.

Table 5.1: A snippet of conversation between two characters from the "Indiana Jones and the Last Crusade" movie with each dialogue turn annotated with its corresponding speaker name. We aim to generate embedding representations for "Indiana" and "Henry" in a way that captures their relation.

The representation of characters in dialogues has been an important task for social network extraction [96], character relation modeling [101], and persona-based models [141, 98]. However, most of the previous work relies upon the extraction of linguistic features like explicit forms of address [95], the length of the utterance, or the frequency of exchanges between the characters [96].

In this work, we address the task of representing characters in dialogues, specifically focusing on movies and plays. Given a set of dialogue turns, annotated with the corresponding speaker names, our goal is to generate a vector representation for each of these characters that captures the relation with other characters. We propose a new approach to embed characters in dialogues based not only on what a character is saying, but also to whom. This model allows the information from the words in a dialogue turn to propagate to the representation of the previous and following speakers.

Despite its simplicity, our model yields strong empirical performance. By evaluating our model on two different tasks – namely character relatedness and character relation classification (fine-grained, coarse-grained, and sentiment) – we find that the model exceeds by a large margin several strong baselines, which indicates that our model effectively captures the various characteristics of characters. Additionally, in the process of evaluating the model, we build a new dataset consisting of 4,761 character relation pairs obtained from eighteen movies, manually annotated with relatedness scores and relations of various

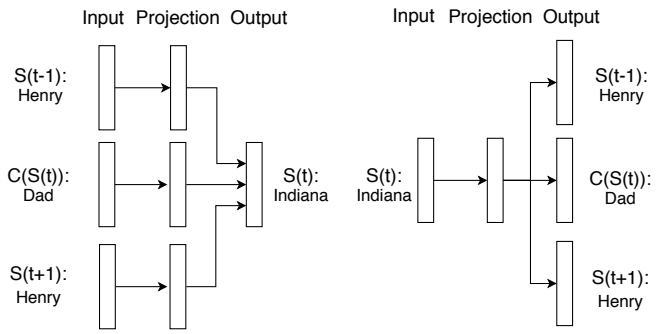


Figure 5.1: The conceptual figure describing input /output pairs of our character embedding model. The diagram describes when both the speaker window and the context window are size one. **Left**: Character Embedding(CBOW), **Right**: Character Embedding(SG).

granularities. We are making the dataset publicly available.

Characters play an important role in any dialogue, including movies or plays. Yet, work to date has rarely considered specialized character representations. We hypothesize that a representation that leverages both the language uttered by the characters as well as information on the other characters in the dialogue could result in richer encodings. The intuition behind our hypothesis is explained by the example in table 5.1. Here, the word "Dad" should be associated not only with "Indiana" but also propagate its information to "Henry", conditioned by "Indiana". Our proposed model is well conveying this intuition to encode characters.

5.1.1 Setup

Our architecture builds on a pretrained embedding model generated by standard Word2Vec models [87, 88] or pre-trained contextualized word representations from neural language models (ELMo) [142]. We start by collecting sets of (current speaker, previous speakers, next speakers, context words) as training examples. We split the four elements in the sets into target and context depending on our objectives. Figure 5.1 describes the input-output (target-context) pairs of our system. Additionally, our model works as an unsupervised post-training of existing embeddings, rather than starting the training from scratch. This is due to the fact that getting a good representation for characters is a separate task from get-

ting a general representation of tokens. A good pre-trained embedding space is an essential component to map characters so that they will be distributed in a semantically meaningful embedding space. While a good pre-trained embedding is important, our models focus on "moving" the character embeddings without affecting any other word representations.

5.1.2 Architecture

We propose two post-training schemes, which we refer to as Character Embedding (SG) and Character Embedding (CBOW). The differences stand in the objective of post-training, given sets of (current speaker, previous speakers, next speakers, context words) as training examples. Formally, given the sequence of speakers at each turn $S = s_1, s_2, s_3, , , s_{T-1}, s_T$, we define context words C for turn t as the set of words found by a sliding context window in the utterance. We propose our post-training objectives as following:

$$L = \frac{1}{N} \sum_{s_i \in S} \sum_{w_i \in C(s_i)} \sum_{-sw \leq j \leq sw} \log(p(w_i|s_{i+j}))) \tag{5.1}$$

$$L = \frac{1}{N} \sum_{s_i \in S} \sum_{w_i \in C(s_i)} (\log(p(s_i|w_i) + \sum_{-sw \leq j \leq sw, j \neq 0} \log(p(s_i|s_{i+j}))) \tag{5.2}$$

Our Character Embedding (SG) model maximizes the objective on Equation 5.1, while Character Embedding (CBOW) maximizes the objective on Equation 5.2, where N indicates the number of training examples and sw indicates the size of the speaker window (speaker window of size one means we consider speakers of one preceding turn and one succeeding turn). Our formulation defines probabilities $p(s_i|w_i)$, $p(s_i|s_{i+j})$ and $p(w_i|s_{i+j})$

using the softmax equation. We also define two transformations of our network – lookup table (LUT) initialized by embedding of pre-trained embedding model and Linear Projection Layer W.

To examine the generality of our post-training schemes, we also apply them to another pre-trained word embedding model. Given a dialogue turn, we encode it using ELMo's pre-trained Bi-LSTM model [142] to generate a sequence of contextualized vectors for words. We add a linear projection layer on top that takes the generated embedding, in addition to the previous and following speakers, and train it to predict the speaker of the current turn. We refer to this model as Character Embedding (ELMo).

Algorithm 1: Character Embedding(SG)

E: The embedding from pre-trained model
W: Linear Projection Layer
α: Learning Rate
maxepoch: maximum epoch to run
LUT $\leftarrow$ E, epoch $\leftarrow 1$;
while $epoch \leq maxepoch$ **do**
  for *t from 2 to* $T-1$ **do**
    $x_1 \leftarrow LUT[s_{t-1}]$;
    $x_2 \leftarrow LUT[s_t]$;
    $x_3 \leftarrow LUT[s_{t+1}]$;
    for w_0 *in* $C(s_t)$ **do**
      $target \leftarrow LUT[w_0]$;
      $logits = tanh(W^T(x_1 + x_2 + x_3))$;
      $prediction = softmax(logits)$;
      $loss = -target + \log(prediction)$;
      $W := W - \alpha * \frac{\delta loss}{\delta W}$;
      $LUT[s_{t-1}] := x_1 - \alpha * \frac{\delta loss}{\delta x_1}$;
      $LUT[s_t] := x_2 - \alpha * \frac{\delta loss}{\delta x_2}$;
      $LUT[s_{t+1}] := x_3 - \alpha * \frac{\delta loss}{\delta x_3}$;
    end
  end
  $epoch := epoch + 1$
end

5.1.3 Training

We represent our contexts and targets as a one hot vector of length equal to the vocabulary size. The purpose of our model is to update the embedding of characters in LUT by propagating the gradient from our objectives. We use cross-entropy to calculate the loss, and we use gradient descent to update the parameters. The description of our Character Embedding (SG) model with a speaker window size of one is showed in Algorithm 1.

5.2 Evaluation Tasks and Datasets

We evaluate the quality of our speaker embedding model across two different tasks. Our goal is to evaluate how well each embedding model captures simple and complex character representations and interactions.

5.2.1 Character Relatedness

Measures of semantic relatedness between words indicate the degree to which words are associated with any kind of semantic relationship such as synonymy, antonymy, and so on. Semantic relatedness is commonly used as an absolute intrinsic evaluation task to assess and compare the quality of different word embeddings [143, 144, 145] and phrase embeddings [90].

Similarly, we define character relatedness as the degree to which a pair of characters in a given story are related to each other based on the story plot and their level of interaction throughout the dialogue. Given a pair of characters, we would like the relatedness score between their embedding representations to have a high correlation with their corresponding human-based relatedness score. Thus, the distance of the embeddings between closely related characters should be smaller than the distance between less related ones.

To measure the relatedness between characters in movies, we construct a new annotated dataset based on a publicly available dataset [93]. That dataset includes 28K turns spoken

by 396 different speakers in eighteen movies covering different genres, with the subtitles of each movie labeled with the character name of their corresponding speakers. On average, each character uttered 452 words.

For each movie in that dataset, two human annotators watched the movies and annotated a dense relatedness matrix of characters on a 1-5 scale. Table 5.2 shows the meaning of each score. These scores reflect the level of interaction or how closely related the characters are over the course of the movie. For example, given two characters X and Y, a high score for X and Y is assigned if e.g., X is the father of Y, regardless of the amount of interaction between the two characters. We also give a high score for the cases where X and Y are closely interacted, even if they are unrelated in terms of kinship. Due to the sparseness of the number of closely related characters, we asked the annotators to select the higher score when hesitating between two scores.

For three movies, the Pearson correlation between the two annotators is 0.8394, which reflects a very good agreement. We then average the scores assigned by the annotators and use the result as the human relatedness ground-truth score for each pair of characters.

5	interacted frequently/closely related
4	interacted/related
3	moderately interacted/somewhat related
2	interacted few times/not related
1	did not interact/not related

Table 5.2: Relatedness annotation scores.

In this dataset, we have 4,761 unique character pairs annotated with a relatedness score. Figure 5.2 shows the statistics over the relatedness scores. As shown in the table, only a small number of character pairs are closely related, while the majority of the characters have either interacted very few times or did not interact at all. However, it is important to include these unrelated pairs while evaluating the quality of the character embeddings, as unrelated pairs might be closer than related ones especially for minor characters that do not speak much during the dialogue.

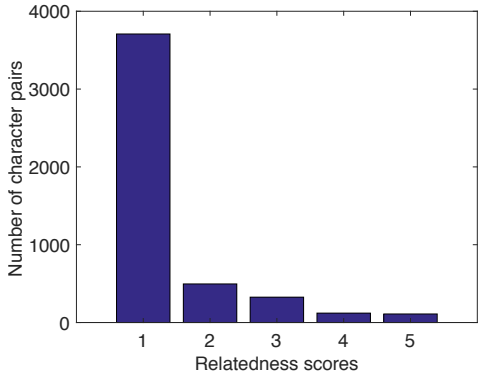


Figure 5.2: Statistics of the character relatedness dataset on movies of speaker naming dataset.

5.2.2 Character Relationships

Understanding the relationships between characters is a primary task in extracting and analyzing social relation networks from literary novels [96, 94]. It is also important for improving computational story summarization and generation methods [146, 147].

Character relationship is a more complex task than character relatedness. In this task, given a pair of character embeddings, we would like to classify the type of their relationship on multiple dimensions. Specifically, we consider: fine-grained relations, such as sister/father/friend/enemy; coarse-grained relations, such as familial/social/professional; and relation sentiment, i.e., positive, negative or neural. The goal of this task is to evaluate the quality of our character embeddings and how well it captures such complex information in an unsupervised fashion. It also serves as an extrinsic evaluation for the impact of our character representations on downstream tasks.

We use a subset of character relationships in a literary dataset [102]. This dataset includes annotations for 18 fine-grained relationship classes, 4 coarse-grained relationship classes, and 3 relation sentiment classes.[1] We use the 31 Shakespeare plays in this dataset and obtain their corresponding text from project Gutenberg. We use the Shakespeare plays

[1] Annotations on temporal change in the sentiment between each pair of characters is also included, but since our models do not have the ability to track such temporal information, we do not use these annotations.

because they have the dialogue turns annotated with speakers names, which is necessary for training our character embedding models. The plays include a total of 605 character pair relationship annotations.

5.3 Experiments

5.3.1 Baselines

For each task, we compare our character embedding models against five baselines:

Interaction Frequency. We count the number of exchanged dialogue turns between every pair of characters and normalize it by the total number of turns spoken by a given pair of characters.

TF-IDF. We treat all the utterances of a character as a document and calculate a tf-idf weight for each word. We then represent a character by its tf-idf vector of the words that they uttered.

Word2Vec (CBOW) model. We use the traditional Word2Vec architecture to train a word embedding space based on the continuous bag-of-words approach [87]. Given a sequence of words D, the context words that exist in a defined window size are considered as input to the network and the objective is to predict the target word by maximizing the average long probability:

$$L = \frac{1}{|D|} \sum_{w_i \in D} \log P(w_i | C(w_i)) \tag{5.3}$$

Word2Vec (SG) model. We use the skip-gram architecture of Word2Vec with negative sampling [88]. In this architecture, the objective is to learn a representation of the target word that would be good at predicting the words within a defined window by maximizing the average log probability:

$$L = \frac{1}{|D|} \sum_{w_i \in D} \sum_{w_0 \in C(w_i)} \log P(w_0|w_i) \tag{5.4}$$

Character BOW. We represent each character as the mean-pooling of a 300-dimension pretrained Word2Vec representation of all the words that this character has uttered through the entire dialogue.

Doc2Vec. We train a Doc2Vec model [148] as tagged documents using the character names as the document tags. We then represent each character as the Doc2Vec representation of all the words that this character has uttered through the entire dialogue.

ELMo (Mean-Pooling). We use pre-trained contextualized word representations from neural language models (ELMo) [142] to generate character names representations based on the sentences that include their names.[2] To generate these representations, we feed the pre-trained ELMo model with a Glove representation for the words and ELMo augments their representation with the hidden states of its two layers bi-directional LSTM to represent the words with respect to their context. For each character name, we average their contextualized representations through the entire dialogue.

5.3.2 Experimental Setting

To have these models trained on in-domain data, we use GenSim [149] to train the different architectures of Word2Vec on the almost 600K sentences / 4M words of subtitles and Shakespeare plays. For the target movies and plays, the speaker names are included in the training data so that we can have a vector representation for each character name. The names in our corpus have been manually normalized so that 'Joe' and 'Joseph' in a movie get the same representation, while 'Joseph' in a different movie gets a different representation. To achieve the first part of the name normalization, we utilize the name-clustering algorithm provided by Bamman [99] to extract and cluster name tokens from the text and

[2]We also tried training ELMo from scratch on our data but the pre-trained model produces better results.

annotate the true representation of names for each cluster. We achieve the second part of the name normalization by adding the text title to the name tokens (e.g., 'Michael' becomes 'Michael$_{Othello}$').

For Gensim [149], we set the learning rate to 0.1, the window size to 4 and the samples to 50 for negative sampling. We run 30 epochs to train our baselines. For post-training by our models, we use a gradient decent to update our parameters. For general experiments, we set the learning rate to 0.1 and the learning rate decays by the factor of 0.9 per 10 epochs. We run maximum 40 epochs for our post-training. For Character Embedding (CBOW), we use a context window of size two. We use a speaker window of size one for both the Character Embedding (CBOW) and the Character Embedding (SG).

Movie	Character	Methods	Closest	Second closest	Third Closest
The Devil's Advocate	Alice Lomax	Ground Truth	**Kevin Lomax**	**John Milton**	**Mary Lomax**
		Interaction Frequency	**Kevin Lomax**	Pam Garrety	John Milton
		TF-IDF	Mary Lomax	John Milton	Don King
		Character Average BOW	John Milton	Kevin Lomax	Barbara
		Word2Vec (CBOW)	Lloyd Gettys	Judge Poe	Alexander Cullen
		Word2Vec (SG)	Alfonse D'amato	Lloyd Gettys	Judge Poe
		ELMo (Mean-Pooling)	**Kevin Lomax**	Mary Lomax	Alexander Cullen
		Character Embedding (CBOW)	**Kevin Lomax**	Judge Poe	**Mary Lomax**
		Character Embedding (SG)	**Kevin Lomax**	**John Milton**	**Mary Lomax**
		Character Embedding (ELMo)	**Kevin Lomax**	Pam Garrety	**Mary Lomax**

Table 5.3: Example of character relatedness task. Given a character, we list the top three characters sorted in descending order from left to right according to their similarity scores.

5.3.3 Results

Character Relatedness. For each model, given a pair of characters we compute the cosine similarity score between the embeddings of these two characters, defined as:

$$similarity(\boldsymbol{C}1, \boldsymbol{C}2) = \frac{\boldsymbol{C}1 \cdot \boldsymbol{C}2}{||\boldsymbol{C}1|| \cdot ||\boldsymbol{C}2||} \tag{5.5}$$

and compute the similarity score between two characters in the embedding space similar to [150, 88]. The list of the nearest characters of a given character C are all the other characters from the same movie sorted in descending order by their similarity score with respect to C.

	Pearson Coeff
Interaction Frequency	0.3632
TF-IDF	0.3129
Doc2Vec	0.1771
Word2Vec (CBOW)	0.2081
Word2Vec (SG)	0.1989
Character BOW	0.2256
ELMo (Mean-Pooling)	0.3212
Character Embedding(CBOW)	**0.4644**
Character Embedding(SG)	**0.4933**
Character Embedding(ELMo)	**0.3475**

Table 5.4: Comparison between the average Pearson correlation coefficient scores of the different models against average human relatedness scores.

	Fine-grained Relation			Coarse-grained Relation			Sentiment		
	P	R	F	P	R	F	P	R	F
Interaction Frequency	0.04	0.16	0.06	0.30	0.44	0.33	0.33	0.58	0.42
TF-IDF	0.11	0.12	0.10	0.39	0.42	0.40	0.43	0.53	0.40
Character Average BOW	0.08	0.16	0.05	0.33	0.43	0.28	0.28	0.53	0.37
Word2Vec (CBOW)	0.11	0.13	0.12	0.37	0.38	0.38	0.39	0.40	0.39
Word2Vec (SG)	0.09	0.12	0.10	0.37	0.37	0.37	0.41	0.43	0.42
Doc2Vec	0.12	0.12	0.12	0.40	0.40	0.40	0.42	0.42	0.42
ELMo (Mean-Pooling)	0.14	0.18	0.14	0.39	0.41	0.40	0.44	0.50	0.46
Character Embedding(CBOW)	0.11	0.14	0.12	0.43	0.44	0.43	0.44	0.47	0.44
Character Embedding(SG)	0.11	0.17	0.12	0.43	0.46	0.42	0.40	0.51	0.42
Character Embedding (ELMo)	**0.18**	**0.19**	**0.19**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**

Table 5.5: Comparison between the average of the precision, recall and macro-weighted f-score of the baselines and our character embedding model on both fine-grained, coarse-grained character relation and sentiment classification.

Table 5.4 shows the Pearson correlation coefficients of the resulting similarity scores of each model against the average human annotation scores. These results suggest that having the context window over the utterance and adding the previous and next speakers to the input layer greatly improves the ability of the character embeddings to capture the relatedness between the different characters in a given story dialogue.

Table 5.3 shows an example of characters that are most related to "Alice Lomax" from the movie "The Devil's Advocate" as calculated based on each model sorted in descending order according to their cosine similarity scores. It is worth noting that Kevin Lomax is Alice's son, John Milton is Kevin's father and Mary Ann Lomax is Kevin's wife. On the

Play	Character 1	Character 2	Methods	Fine-grained	Coarse-grained	Sentiment
			Ground Truth	**lovers**	**social**	**positive**
			Interaction Frequency	**lovers**	**social**	**positive**
The Two			TF-IDF	servant	**social**	negative
Gentlemen	Julia	Proteus	Character Average BOW	friend	**social**	**positive**
of Verona			Word2Vec (CBOW)	servant	familial	negative
			Word2Vec (SG)	servant	familial	**positive**
			ELMo (Mean-Pooling)	friend	**social**	**positive**
			Character Embedding (CBOW)	**lovers**	**social**	negative
			Character Embedding (SG)	**lovers**	**social**	**positive**
			Character Embedding (ELMo)	**lovers**	**social**	**positive**

Table 5.6: Example of classification task on Shakespeare's play, using different baselines and our character representation methods. The classification output consists of the relations of character 2 from character 1's perspective. A bold face indicates a correct relation classification.

other hand the characters suggested by both Word2Vec CBOW and SG models did not interact with Alice through the whole movie.

To further analyze the quality of the produced character embeddings, we evaluate the embeddings across different characters according to the their frequency of appearance in the movies. Figure 5.3 shows a comparison between the performance of the different models over minor and major characters based on the number of dialogue turns that each character uttered. These results show that our character embedding model consistently outperforms the traditional Word2Vec baseline models and reflect the robustness of our model in generating better character embeddings.

Character Relationship. We have three classification tasks for characters relationship: 1) fine-grained relationship classification; 2) coarse-grained relationship classification; 3) relation sentiment classification. For each of these tasks, we train a logistic regression classifier using the Scikit-learn library [138]. These classifiers take a pair of character embeddings as a concatenation of their vectors and predict their relationship. We use a leave-one-play-out cross-validation in which character pairs from each play are used as a test set and character pairs from the other plays are used to train the models. Table 5.5 shows the classification average precision, recall and weighted F-score obtained by training the logistic regression classifiers using the character embeddings produced by the

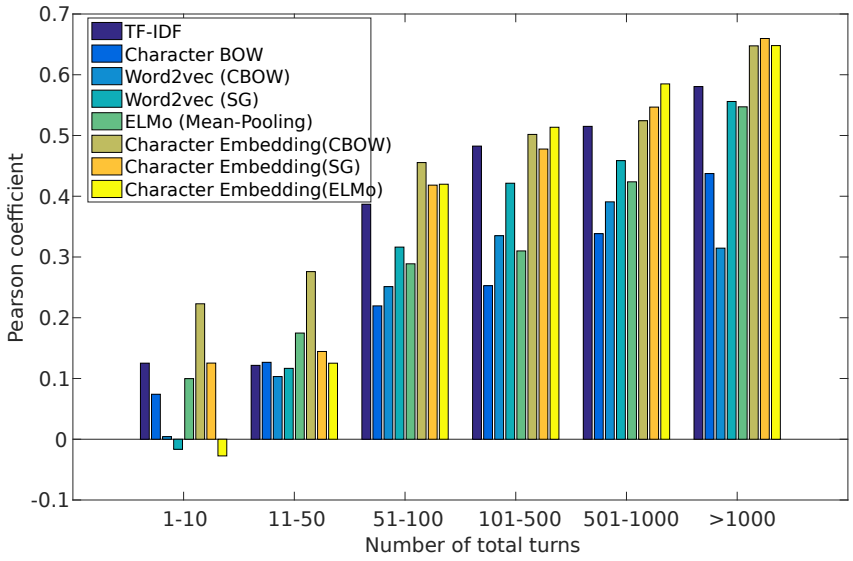


Figure 5.3: Comparison of the average Pearson correlation coefficient over characters who had different number of turns.

different models. Training classifiers using our character embedding models consistently outperforms the classifiers trained using the other models, which reflects the quality of the semantic information captured by our character embeddings when compared to other models. Table 5.6 shows examples of the three character relation classification tasks as classified by our character embedding models and the baselines.

	Accuracy	
	Q+S	Q+S+V
MS (Glove) [30]	0.6515	0.6770
MS (Glove w/o names)	0.6177	0.6467
MS (CharEmbedding(CBOW))	**0.6590**	**0.6852**
MS (CharEmbedding(SG))	**0.6554**	**0.6884**

Table 5.7: Comparison on the TVQA validation dataset using the MS method with Glove and Glove fine-tuned using our proposed character embedding method.

Question Answering. As a final evaluation, we test the impact of our character embedding on dialogue understanding. TVQA [30] is a challenging dataset that includes 152.5K multiple choice question answers about 21.8K video clips from 6 TV shows such as the *Big Bang Theory*, *House*, and so on. These questions were created in a way that requires

understanding of both the dialogue and the visual content of a given video. Each video clip includes the video frames and subtitles with speaker names aligned automatically with their corresponding show scripts (around 69% of the subtitle segments include speakers names). We follow the same dataset splits for training, validation, and test sets.

To evaluate our embedding, we use the baseline implementation proposed with the TVQA dataset, namely Multi-Stream (MS). This model relies on bidirectional attention between context (represented by subtitles and/or visual content) and question answer pairs as queries to predict the correct answer [30]. Visual features are included as textual labels of detected visual concepts in the frames of the video clip. To measure the effect of the person names on the model, we apply a named entity recognizer and replace the names with a fixed randomly generated embedding. Table 5.7 shows the results from the MS method using Glove, Glove with removing names from subtitles, and using a fine-tuned Glove using our character embedding model. The use of our character embeddings bring improvements over the pre-trained Glove embeddings, which demonstrates the usefulness of these character representations.

5.4 Conclusion

In this paper, we presented a novel unsupervised embedding model to represent characters and their interaction in a dialogue. Our embedding model produces character representations that reflect the language used by the characters as well as information about their relations with other characters. To evaluate the performance of our character embeddings, we experimented with two tasks on two datasets: (1) character relatedness, using a dataset we introduced consisting of a dense character interaction matrix for 4,761 unique character pairs over 22 hours of dialogue extracted from eighteen movies; and (2) character relation classification, for fine- and coarse-grained relations, as well as relation sentiment. Our experiments show that our model significantly outperforms the traditional Word2Vec

continuous bag-of-words and skip-gram models, thus demonstrating the effectiveness of the character embeddings we introduced. We further showed how the character embeddings can be used in conjunction with a visual question answering system to improve over previous results.

CHAPTER 6

Character Relation Classification

6.1 Introduction

Story understanding requires the ability to interpret characters' roles, relationships, and how they evolve as the story progresses. This interpretation is vital to improve methods for story summarization and generation [146, 147], as well as to analyze story characters' social network [96, 107]. It also provides basic facts that assist with reasoning about and justifying character actions in a story [95, 151].

Several character-centric story understanding attempts focus on modeling character persona [98] or their relationships [100, 101, 151] from textual narratives and movie plot summaries. Moreover, previously proposed methods often made several limiting assumptions such as assuming that relationships are static within a narrative, or that relationships can be modeled using a coarse binary polarity (e.g., cooperative vs. non-cooperative or adversarial vs. non-adversarial) [105, 100]. Recent unsupervised learning approaches have been proposed to overcome these limitations by modeling evolving character relationships using latent variables to capture their dynamics within narratives [97, 151]. The evaluation of these models is tricky and subjective since they result in relation descriptor words similar to topic models. However, most of the existing methods for analyzing character relations focus on textual narratives and movie plot summaries. These summaries include a high-level understanding of the story as interpreted by the summary writer grounded in

the text.

In this work, we address the problem of character relation classification in movies and the evolution of these relations as the movie story progresses. Unlike textual story summaries, movies require processing information from several modalities such as the dialog between characters, their interactions, and expressions reflected in the visual and acoustic channels. Figure 6.1 shows an example of how relations between characters in the movie "Single White Female" are established and evolve. As seen in the figure, the movie starts with "Allie" and "Sam" discussing their love life and their engagement plans. Allie then discovers that "Sam" had a relation with his ex-wife. She gets angry and decides to break up with him, but Sam still loves her. After several attempts to apologize to Allie, he succeeds, and they get back together. Recently, movie understanding from both text and video gained attention with several movie understanding datasets developed to facilitate movie story comprehension through question answering such as MovieQA [29] and TVQA [30]. They are, however, not enough to develop character-centric models for movie story understanding.

In this work, we make two main contributions. First, we construct a new inter-character relationship dataset for movies. This dataset captures the introduction and changes in relations throughout movie events. The collected relations also capture different facets of these relations. Second, we present a supervised multimodal framework for modeling inter-character relationships in movies. The input to our model is the subtitles, video, speech signal, and a pair of characters appearing in the movie. We model the relationships by leveraging all movie segments that each character speaks in as well as their common interaction segments. Finally, our model predicts their relationship at any given time. We evaluate our model on our new character relationship dataset and show that our multimodal model outperforms using single modality and several non-neural baselines as well.

This chapter is structured as follows: in Section 6.2 we describe our dataset; in Section 6.3, we explain our data processing and feature extraction process from each modality; in

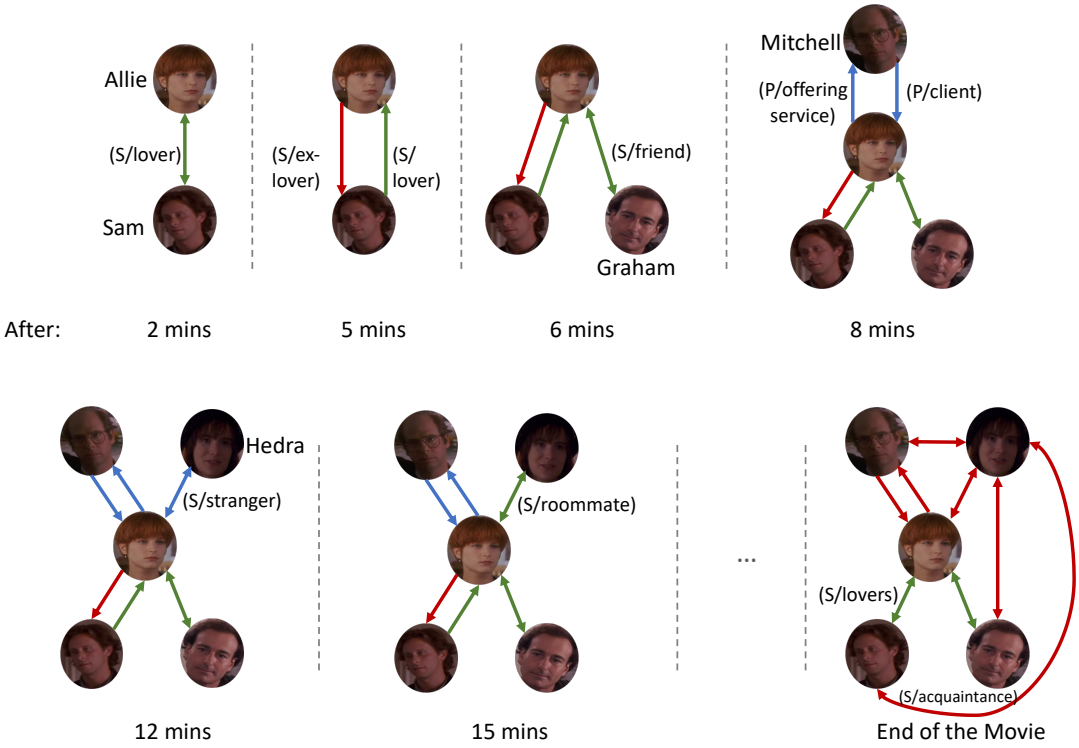


Figure 6.1: Example of "Single White Female" movie character relations evolve over the storyline. The head of an arrow represents the direction of the relation between the two characters, and its color represents the sentiment between the characters green, blue, red for positive, neutral, and negative sentiments, respectively. The labels on the arrow represent the coarse-grained relation **S**ocial/**P**rofessional/**F**amilial followed by the fine-grained relation such as lover or friend.

Section 6.4, we introduce our character relation classification model; Section Section 6.5 discusses our experiments and results; and finally, we conclude in Section 6.6

6.2 Datasets

6.2.1 Temporal Character Relations

We introduce our new dataset, Temporal Character Relations (TCR), of manually annotated inter-character relationships between characters in seventeen movies covering different genres. Our dataset is built using the movies in the speaker naming dataset [93]. Therefore, for each subtitle segment, we have the ground-truth speaker name. For every

pair of characters in each movie, we annotate their relationship when it is established, as well as changes throughout the movie. Inspired by [102], we collect annotations on three dimensions of interest: coarse-grained relation (professional, social, familial), fine-grained relation (lover, friend, husband/wife), sentiment (positive, negative, neutral). Unlike [102], we want our annotations to capture how the relations between characters evolve as the movie storyline progresses.

For each movie, a human annotator watches it and labels the introduction/change of relationships between every pair of characters, the direction of the relation, and the timestamp of the change. The annotated relation is directional, i.e., given two characters, c_1 and c_1, the relation between c_1 and c_2 is not necessarily the same relation between c_2 and c_1. For example, in the movie "Just Friends" at the beginning of the movie there are two characters "Chris Brander" and "Jamie Palamino": "Chris" loves "Jamie" while "Jamie" considers "Chris" only her friend. Thus, our dataset captures the temporal and dynamic aspects of the relations between characters over the course of the movie plot. Overall, the total number of fine-, coarse-grain, and sentiment labels in our dataset are 34, 4, and 3, respectively.

Overall, our dataset consists of seventeen full movies with a total duration of 35.6 hours. We treat each relation dimension independently; therefore, the number of relation types might differ because one relation type might not change while others do not. For example, when two friends fight, the sentiment relation between them changes, while the coarse and fine relations remain unchanged. Similarly, two friends can become lovers, thus their fine-relation changes while the coarse and sentiment relations stay the same.

We split our dataset into train, validation, and test sets. We use the validation set to fine-tune our model as well as the baselines parameters. The movies in each split are mutually exclusive. Therefore, we evaluate the ability of our models to generalize to completely unseen movies. Table 6.1 shows the statistics of our collected data. For training and evaluation purposes, we label character pairs that do not exist in our annotations with a "No

Relation" label. Figure 6.2 shows the label distribution of each relation type in our dataset.

	Train	Val	Test	Total
# movies	9	3	5	17
# characters	185	53	109	347
# fine-grained relations	1,367	311	576	2,251
# coarse-grained relations	1,314	253	539	2,106
# sentiment relations	1,139	225	548	1,912

Table 6.1: Statistics of our temporal character relations dataset across train, validation, and test splits. We show the number of movies, characters, fine-, coarse-grain, and sentiment relations.

6.3 Data Processing and Representation

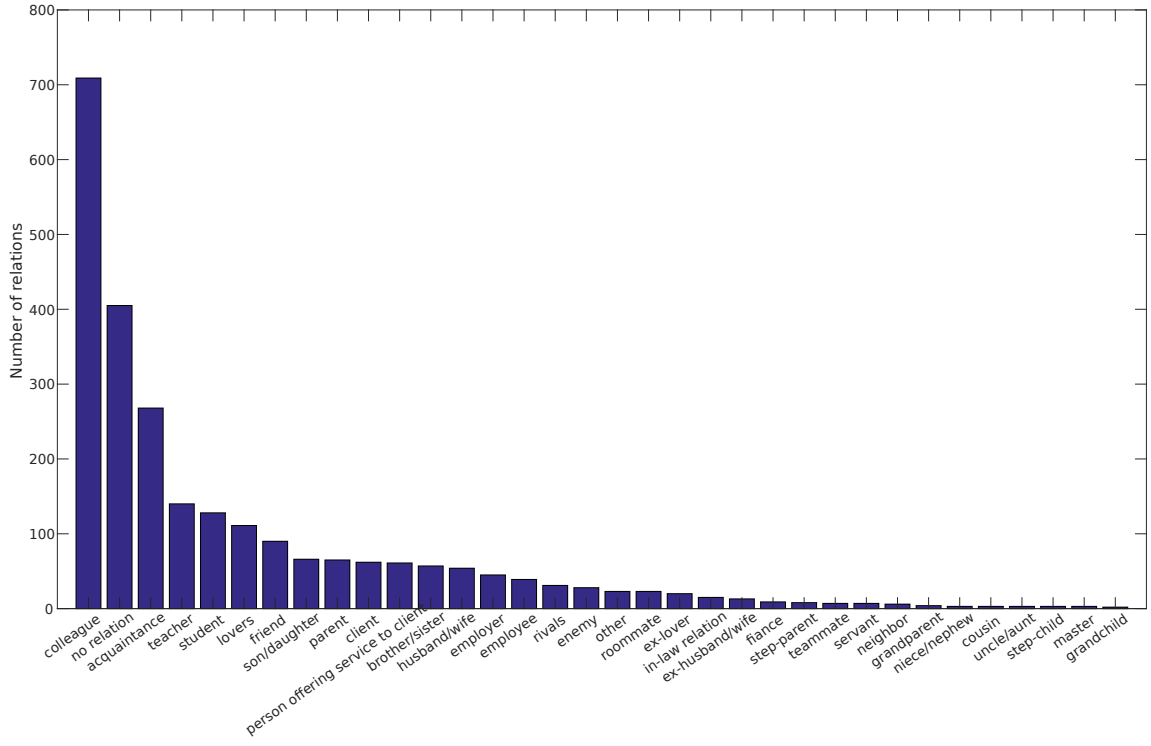
Given a movie, we split it into segments. We extract features from each segment by processing their corresponding textual, acoustic, and visual information. The textual information comes primarily from the subtitles, which represent the dialog between the characters. The visual and acoustic features are extracted from the video content of the movie. In this section, we discuss the process of obtaining the sequence of feature representations for each individual modality: text, video, and audio. We use these representations as input to our model for our character relation prediction task. Figure 6.3 shows an overview of our representation of each modality.

6.3.1 Textual Features

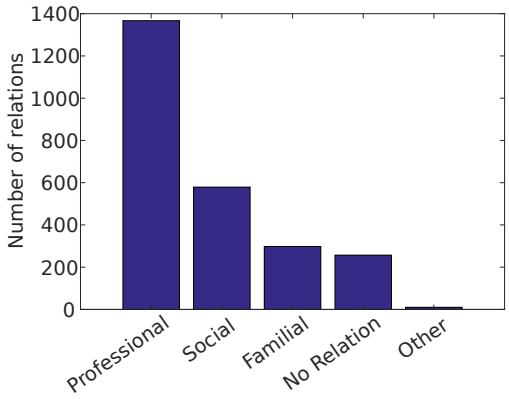
We evaluate the following representations to represent the subtitle segments:

Glove Word Embedding. For each subtitle segment s_i, we first tokenize the sentence and represent each token using its corresponding GloVe embedding [13]. We use the 300-dimensional version of GloVe vectors pre-trained on text from Wikipedia. We represent each sentence using the average representation of its individual tokens.

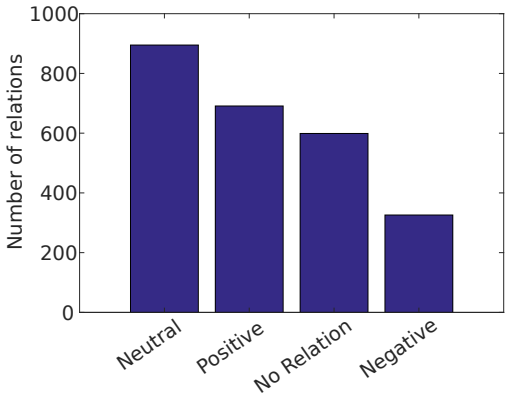
TF-IDF is a traditional weighting scheme in information retrieval. We represent each subtitle segment as a vector of tf-idf weights, where the length of the vector (i.e., vocabulary



(a)



(b)



(c)

Figure 6.2: Label distribution for each relation type in our temporal character relations dataset. (a) shows the distribution of fine-grained relations; (b) shows the distribution of coarse-grained relations; (c) shows the distribution of sentiment relations.

size) and the idf scores are obtained from the movie subtitles.

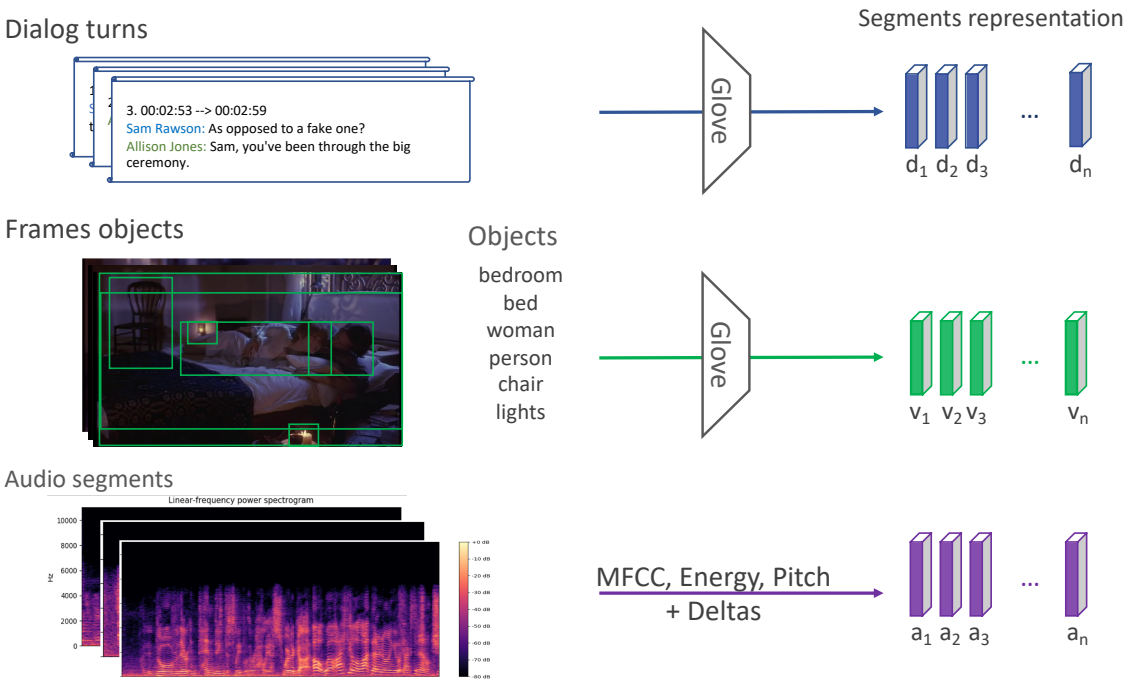


Figure 6.3: Overview of our representation of each movie modality. We represent the entire movie as a sequence of segments. Each movie segment s_i is represented using $< d_i, v_i, a_i >$ representing the feature vectors extracted from dialog, video frames, and speech signal, respectively.

6.3.2 Acoustic Features

For each movie in the dataset, we extract the audio from the center channel. The center channel is usually dedicated to the dialogue in movies, while the other audio channels carry the surrounding sounds from the environment and the musical background. For each dialog turn, we use the subtitles' timestamps to estimate its boundaries.

We utilized the open source openSMILE [137] for feature extraction. In particular, we extract the Interspeech 2009 emotion challenge features described in [152] to represent the acoustic information of each segment. In detail, the extracted features are: root mean square frame energy, normalized pitch frequency, mel-frequency cepstral coefficients (MFCC), harmonics-to-noise ration, and zero-crossing-rate (ZCR). To each of these features, the delta coefficients are computed. In addition to 12 other features including the mean, standard deviation, etc. This results in a 384-dimensional vector for each segment.

6.3.3 Visual Features

For each movie segment, we subsample one frame every ten frames. We process each frame using a Faster R-CNN [153] pre-trained on the Visual Genome dataset [154] to detect objects and their predicted labels. [1] Across the segment frames, we follow [62] to select the top-K objects, where K is a hyper-parameter that we fine-tune during training. Recent work found that using detected object labels as input to an image captioning system and question answering gave comparable performance to using regions visual features directly [155, 30].

Inspired by this previous work, we use the detected objects labels as visual inputs. Figure 6.3 shows an example of the detected objects and their predicted labels in a given segment. The model can detect rich visual concepts such as objects and locations, e.g., "bed" and "bedroom," which might suggest a familial or social relationship. For every segment, we represent it using the average Glove embedding of the predicted objects labels. The resulting segment representation is a 300-dimensional feature vector.

As a result of our feature extraction from the multiple modalities, each segment s_i in the input sequences consists of features $< d_i, v_i, a_i >$ where d_i represents the average word embedding for a dialog turn, v_i represents the embeddings of the objects extracted from the video frames, and a_i represents the acoustic features extracted from speech signal of segment i.

6.4 Model

6.4.1 Problem Definition

Given a movie and two characters, C_1 and C_2 appearing in it, our goal is to predict their relationship at any given time T in the story. For our relation classification task, we have

[1] We used the implementation available here: https://github.com/violetteshev/bottom-up-features

three sources of inputs (C_1 memory, C_2 memory, $C_{1,2}$ interaction memory). We use the word "memory" to refer to the movie segments that the characters uttered. We define:

- C_1 memory as the sequence of segments $S_{c_1} =< s_1, s_2, ..s_T >$ in which character C_1 is the speaker.
- C_2 memory as the sequence of segments $S_{c_2} =< s_1, s_2, ..s_T >$ in which character C_2 is the speaker.
- $C_{1,2}$ interaction memory as the sequence of segments $S_{c_1,c_2} =< s_1, s_2, ..s_T >$ in which both characters C_1 and C_2 are talking to each other. In order words, S_{c_1,c_2} include segments in which both C_1 and C_2 are present.

We represent each segment, as described in section 4.3. These sequences are usually chronologically ordered according to the movie events unless there is a flashback as a part of the movie plot.

Each input to our model is (C_1, C_2) or ($C_{1,2}$), in addition to the target relation label. Our model includes three encoders: i) dialog encoder; ii) visual encoder; iii) audio encoder. We use a Gated Recurrent Unit (GRU) [156] to encode each modality sequence into a vector.

$$
\begin{aligned}
z_t &= \sigma\big(x_t U^z + h_{t-1} W^z\big) \\
r_t &= \sigma\big(x_t U^r + h_{t-1} W^r\big) \\
\tilde{h}_t &= \tanh\big(x_t U^h + (r_t * h_{t-1}) W^h\big) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
$$

Here r is a reset gate, and z is an update gate. The reset gate determines how to combine the information from the new segment with the previous memory, and the update gate defines how much of the previous memory to keep.

$$\overrightarrow{h}_{di} = GRU(d_i), i \in [1, T]$$
$$\overrightarrow{h}_{vi} = GRU(v_i), i \in [1, T]$$
$$\overrightarrow{h}_{ai} = GRU(a_i), i \in [1, T]$$

We use the last hidden state to represent the encoding of the input sequence. We use the same encoders for the different input sequences C_1, C_2, $C_{1,2}$. We then represent the sequences of segments from all modalities as

$$S = [\overrightarrow{h}^{C_1}_{dT}; \overrightarrow{h}^{C_2}_{dT}; \overrightarrow{h}^{C_1}_{vT}; \overrightarrow{h}^{C_2}_{vT}; \overrightarrow{h}^{C_1}_{aT}; \overrightarrow{h}^{C_2}_{aT}]$$

in case of using C_1, C_2 as the input. In case of using $C_{1,2}$, the sequence is represented as,

$$S = [\overrightarrow{h}^{C_{1,2}}_{dT}; \overrightarrow{h}^{C_{1,2}}_{vT}; \overrightarrow{h}^{C_{1,2}}_{aT}]$$

where ; represent tensor concatenation. We finally pass it to a two fully connected feed-forward layers for classification. We use cross entropy loss function to train our model.

6.5 Experiments

To study the role of using information from multiple modalities, we conduct several experiments to evaluate the performance of models trained separately using each modality and combinations of them on each relation classification task. Additionally, we investigate a number of baselines for character relation classification.

6.5.1 Baselines

We also use naive baselines to study the inherited biases in the data, such as the majority class and random guessing classifier. For each relation classification task, we compare our multi-modal character relation classification model against several non-neural baselines.

These baselines have been frequently used to relation classification between literary novel characters [96, 95]. In particular, we train logistic regression, SVM, decision tree, and random forest models using the Scikit-learn library [138]. For each model, given a character pair, we train the model using TF-IDF representation of the dialog between the characters.

Majority Class. We select the majority class label in the training data as our prediction during test.

Random Guess. We randomly select labels according to class distribution from the training data as our prediction during test.

Interaction Frequency. Similar to [96, 106], we use the volume of interaction to predict the relationship between two characters. We use the number of turns between characters, in addition to the the number of turns of C_1 and C_2 up to the target segment.

Logistic Regression. We train a multi-class logistic regression classifier using one-vs-all strategy for each task.

SVM. Support Vector Regression has been used previously to estimate a weighting score for relations between characters [105]. In our work, we train a linear SVM classifier for relation classification. We experimented with several kernels, but the linear one achieved the best results on the validation set.

Random Forest. It fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve its predictive power and control over-fitting. These trees aim to find several important tokens in the dialog between characters to classify their relation (e.g., a word like "dad" is more probable to occur in "son/daughter"/ "familial" relation between characters.

End-to-End Memory Network. Memory networks have been widely adapted for movie understanding and question answering tasks [29, 63]. We examine the ability of a memory network to predict the relationship between two given characters.

	Fine Relation			Coarse Relation			Sentiment		
	P	R	F	P	R	F	P	R	F
Majority	0.2227	0.2764	0.2467	0.3961	0.4601	0.4257	0.3632	0.4434	0.3993
Random	0.0933	0.1002	0.0966	0.3447	0.3524	0.3485	0.3436	0.2363	0.2800
Interaction Frequency	0.2227	0.2806	0.2483	0.4634	0.5382	0.4980	0.4364	0.3759	0.4039
Logistic Regression	0.2186	0.2750	0.2435	0.3961	0.4601	0.4257	0.3898	0.4453	0.4157
SVM	0.2007	0.2279	0.2134	0.4042	0.4688	0.4341	0.3717	0.2828	0.3212
Random Forest	0.2227	0.2806	0.2483	0.3961	0.4601	0.4257	0.3632	0.4434	0.3993
MemoryN2N									
{dialog}	0.2181	0.2096	0.2138	0.3961	0.4601	0.4257	0.3256	0.3066	0.3158
{audio}	0.2227	0.2764	0.2467	0.3946	0.4583	0.4241	0.3750	0.4215	0.3969
{video}	0.2124	0.2597	0.2337	0.3901	0.4531	0.4193	0.385	0.4124	0.3982
{dialog, audio}	0.2248	0.2782	0.2487	0.3961	0.4601	0.4257	0.3800	0.3668	0.3733
{dialog, video}	0.2012	0.1929	0.197	0.4529	0.5260	0.4867	0.4453	0.4015	0.4223
{audio, video}	0.2231	0.2764	0.2469	0.3946	0.4583	0.4241	0.3639	0.4416	0.3990
{dialog, audio, video}	0.2453	0.1948	0.2172	0.3976	0.4618	0.4273	0.3647	0.4453	0.4010
Our Model									
{dialog}	0.2212	0.2746	0.2450	0.4200	0.4878	0.4514	0.3752	0.4252	0.3986
{audio}	0.2446	0.2542	0.2493	0.3961	0.4601	0.4257	0.3563	0.3303	0.3428
{video}	0.2335	0.2560	0.2442	0.4709	0.5469	0.5060	0.4003	0.4398	0.4191
{dialog, audio}	0.2227	0.2764	0.2467	0.3961	0.4601	0.4257	0.4366	0.4526	0.4444
{dialog, video)	0.3281	0.1929	0.2430	**0.4918**	**0.5712**	**0.5285**	**0.4466**	**0.5036**	**0.4734**
{audio, video}	0.2256	0.2746	0.2477	0.4469	0.5191	0.4803	0.3925	0.4197	0.4056
{dialog, audio, video}	**0.2505**	**0.2505**	**0.2505**	0.4686	0.4670	0.4678	0.4024	0.4325	0.4169

Table 6.2: Comparison between the average of the precision, recall and micro-averaged f-score of the baselines and our multi-modal character relation classifier model on fine-grained, coarse-grained and sentiment relation classification using C_1 and C_2.

6.5.2 Results

We model our task as a classification problem. Since the distributions in our dataset are unbalanced, we use micro-averaged precision, recall, and f-score as our evaluation metric. For all of our experiments, we use the validation set to fine-tune models' parameters. We then select the best model to evaluate on the test set. From our experiments, TF-IDF representation of the text gives better results than Glove when used with the non-neural baselines. Thus, for these models the textual modality in the results table refer to TF-IDF, while for the neural baselines text is represented using pre-trained Glove embeddings.

Table 6.2 and Table 6.3 show the results of our model in comparison to the baselines on each relation task when using C_1, C_2 and $C_{1,2}$ only, respectively. In both scenarios, our model outperforms the other baselines. Using C_1, C_2 memory for prediction, our model has the best results when relying on information from both dialog and video. It also out-

	Fine Relation			Coarse Relation			Sentiment		
	P	R	F	P	R	F	P	R	F
Majority	0.2227	0.2764	0.2467	0.3961	0.4601	0.4257	0.3632	0.4434	0.3993
Random	0.0933	0.1002	0.0966	0.3447	0.3524	0.3485	0.3436	0.2363	0.2800
Interaction Frequency	0.2227	0.2806	0.2483	0.4634	0.5382	0.4980	0.4364	0.3759	0.4039
Logistic Regression	0.2196	0.2486	0.2332	0.3931	0.4566	0.4225	0.3710	0.4434	0.4040
SVM	0.2003	0.2467	0.2211	0.4048	0.4688	0.4344	0.3843	0.4635	0.4202
Random Forest	0.2227	0.2806	0.2483	0.3961	0.4601	0.4257	0.3643	0.4434	0.4000
MemoryN2N									
{dialog}	0.2227	0.2764	0.2467	0.4245	0.4931	0.4562	0.4357	0.4453	0.4404
{audio}	0.2821	0.2077	0.2393	0.4081	0.4740	0.4386	0.3454	0.3120	0.3279
{video}	0.1880	0.1855	0.1867	0.3901	0.4531	0.4193	0.4223	0.4416	0.4317
{dialog, audio}	0.2227	0.2764	0.2467	0.4081	0.474	0.4386	0.4357	0.4453	0.4404
{dialog, video}	0.2196	0.1911	0.2044	0.3984	0.4601	0.4270	0.3831	0.4452	0.4118
{audio, video}	0.2272	0.1892	0.2065	0.4484	0.5208	0.4819	0.4357	0.4453	0.4404
{dialog, audio, video}	0.2302	0.1614	0.1897	0.4230	0.4913	0.4546	0.4072	0.4361	0.4211
Our Model									
{dialog}	0.2244	0.2764	0.2477	0.3961	0.4601	0.4257	0.3756	0.3996	0.3873
{audio}	0.2226	0.2560	0.2381	0.4081	0.4740	0.4386	0.4252	0.3321	0.3730
{video}	0.2222	0.2635	0.2411	0.435	0.5052	0.4675	0.4282	0.3431	0.3810
{dialog, audio}	0.2248	0.2560	0.2394	0.3896	0.4410	0.4137	0.4079	0.4124	0.4102
{dialog, video}	**0.2710**	**0.2504**	**0.2603**	**0.4854**	**0.5486**	**0.5151**	**0.4699**	**0.4562**	**0.4630**
{audio, video}	0.2227	0.2764	0.2467	0.3991	0.4635	0.4289	0.4207	0.3047	0.3534
{dialog, audio, video}	0.2301	0.2635	0.2457	0.4051	0.4705	0.4353	0.4057	0.4161	0.4108

Table 6.3: Comparison between the average of the precision, recall and micro-averaged f-score of the baselines and our multi-modal character relation classifier model on fine-grained, coarse-grained and sentiment relation classification using $C_{1,2}$.

performs our model trained on $C_{1,2}$ segments. However, these models struggle to classify the fine-grained relationships between the characters due to the unbalanced distribution of the fine-grained relationship classes.

Table 6.4 shows an example of our model predictions for the two characters, "Holden McNeil" and "Alyssa Jones" from the movie "Chasing Amy" over different segments of the movie. From the example, it can be seen that the network was able to identify their relationship correctly at the right time in many cases. In other cases, the network managed to identify these relations after time segments than the ground-truth annotations. This particular example is tricky as well. The two characters did not know each other initially and they got introduced in a business event through a friend. Both characters are writers and spent their first two meetings chatting about publishing and work topics. Such cases might impact the classification performance of the models. The current version of our dataset

assigns a single label for each relation type, which is the most common scenario in movies.

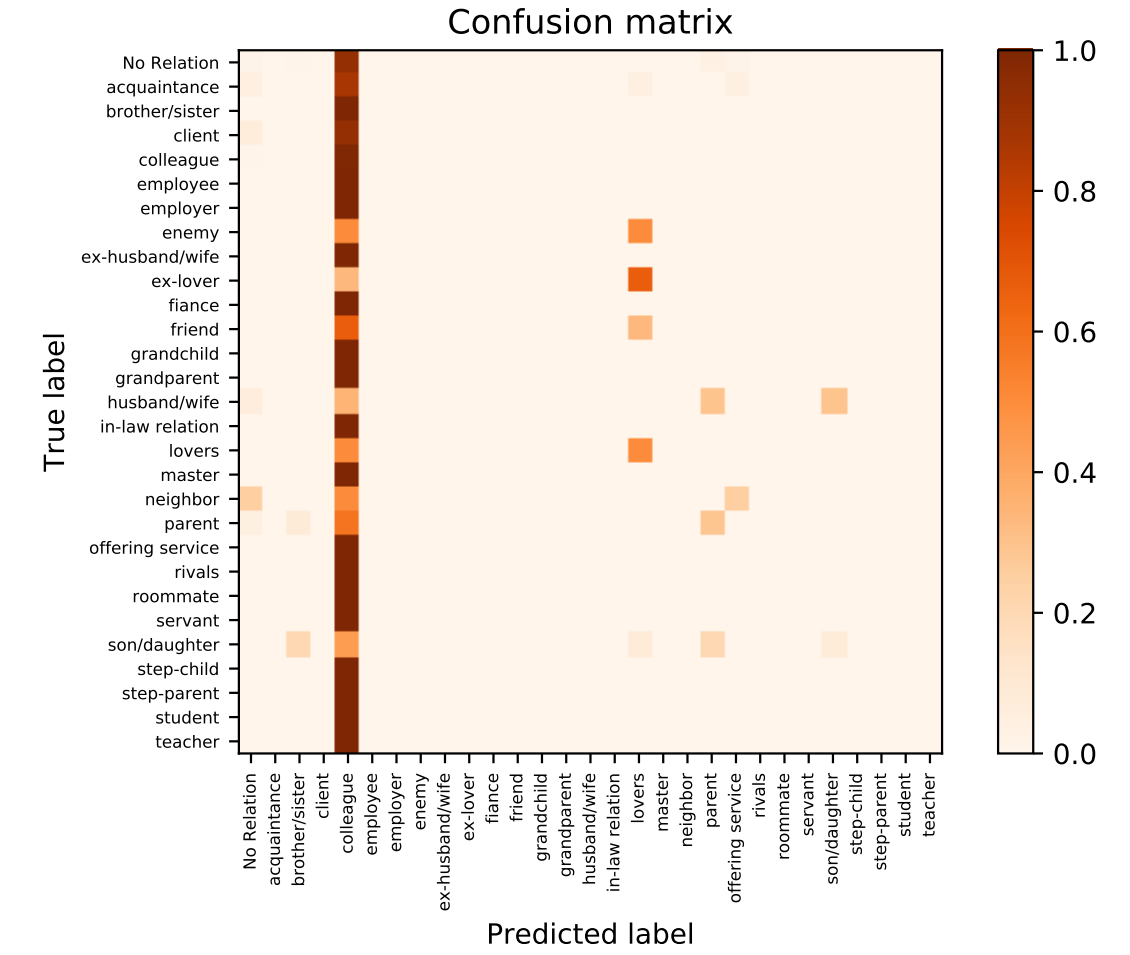
Figure 6.4 shows the classification confusion matrix for each relation classification task using our best model. The fine-grained relationships are sparse. Hence most of the predictions

Segment #	Character 1	Character 2	Ground Truth	Our Model
135	Holden McNeil	Alyssa Jones	(NR, NR, Positive)	(P, colleague, NR)
136	Alyssa Jones	Holden McNeil	(NR, NR, Positive)	(P, **NR**, Neutral)
220	Holden McNeil	Alyssa Jones	(S, acquaintance, Positive)	(P, colleague, **Positive**)
220	Alyssa Jones	Holden McNeil	(S, acquaintance, Positive)	(P colleague, **Positive**)
603	Alyssa Jones	Holden McNeil	(S, friend, Positive)	(**S**, colleague, **Positive**)
604	Holden McNeil	Alyssa Jones	(S, friend, Positive)	(**S**, colleague, **Positive**)
813	Holden McNeil	Alyssa Jones	(S, lovers, Positive)	(**S**, colleague, **Positive**)
889	Holden McNeil	Alyssa Jones	(S, lovers, Positive)	(**S**, friend, Negative)
939	Alyssa Jones	Holden McNeil	(S, lovers, Positive)	(**S**, **lovers**, Negative)
1224	Holden McNeil	Alyssa Jones	(S, lovers, Negative)	(**S**, **lovers**, **Negative**)
1257	Alyssa Jones	Holden McNeil	(S, ex-lover, Negative)	(**S**, lovers, **Negative**)

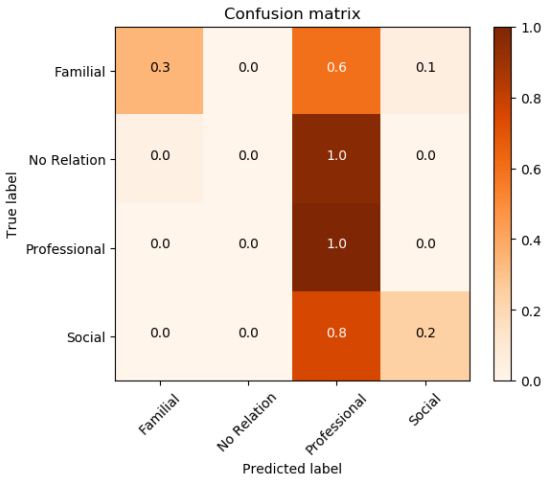
Table 6.4: Example of classification task on the movie "Chasing Amy", using our character relation classification model. The classification output consists of the relations of character 2 from character 1's perspective. A bold face indicates a correct relation classification. The relations are arranged as coarse-grain, fine-grain, and sentiment, respectively.

6.6 Conclusion

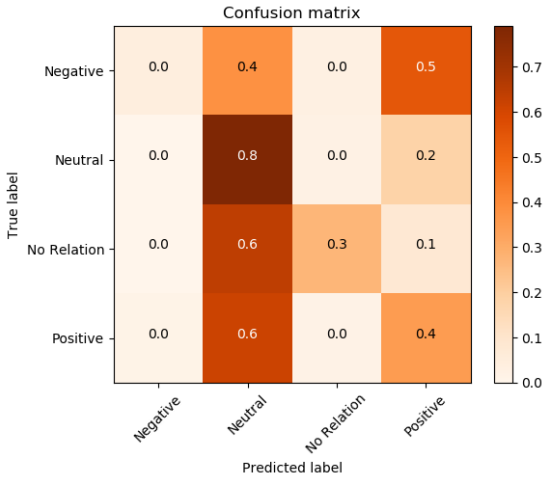
In this chapter, we explore modeling character relations and their changes in movies. To facilitate the development and evaluation of our character relations model, we constructed a new inter-character relationship dataset for movies. This dataset captures the introduction and changes in relations throughout movie events. The collected relations also capture different facets of these relations, in particular, sentiment, fine-, and coarse-grained relations. We then developed a supervised multimodal framework for inter-character relationship prediction between pairs of characters at any given point in the story. We modeled the relationships by leveraging all movie segments that each character speaks in as well as their common interaction segments. Our multimodal model outperformed methods that use a single modality and several non-neural baselines as well.



(a)



(b)



(c)

Figure 6.4: Confusion matrix for each relation type in our temporal character relations dataset. (a) shows the confusion matrix of fine-grained relations; (b) shows the confusion matrix of coarse-grained relations; (c) shows the confusion matrix of sentiment relations.

CHAPTER 7

Conclusions

Throughout this dissertation, we explored various aspects of character representations that leverage multiple modalities such as the visual, textual, and acoustic channels. Our main goal was to use such representations to improve visual story understanding. Our approach used different representations to solve three challenging problems: (i) speaker identification and naming, (ii) character embedding, and (iii) character relationship representation.

7.1 Research Questions Revisited

The previous chapters have presented the details of our proposed approaches to represent movie characters and provided several experiments analyses to answer the fundamental research questions posed in Chapter 1. We will now recap these questions and discuss our findings.

Q1: Can we augment the representation of each entity with speaker names automatically predicted from dialogue?

Humans identify and infer character names from dialogues and can identify the speaker of each utterance when they see or hear them again. Unlike previous work that relied on scripts to identify speakers and focused on using vision and speech for their prediction, we solely used the information that is naturally available while watching a movie. In Chapter 4, our experiments have shown that although it is reasonably possible to develop speaker

naming system by identifying their names from the dialog, the predictive power is not sufficient when relying on a single modality. Out of all the single modal models, the textual features outperformed the visual and acoustic features, especially when applied to a movie with a noisy background. Utilizing information from multiple modalities, in particular, the dialog, video, and speech led to the best results. The speaker naming task remains difficult as a result of using several methods to infer and cluster names from the dialog. Further, our ablation studies showed that we could achieve more than 8% F-score improvement using perfect auxiliary systems for voice gender and name mentions classifiers.

Q2: Can this augmented character representations be used to improve the performance of the downstream task of question answering?

Empirical experiments on a movie question answering dataset demonstrated the effectiveness of our proposed method in comparison to several competitive baselines, as shown in Chapter 4. We have also demonstrated that an SC-MemN2N model that leverages our speaker naming model can achieve state-of-the-art results on the subtitles task of the MovieQA [29] 2017 Challenge. This dataset is very challenging because answering the questions requires a high-level understanding of the movie plot. Given that our model with no supervision inferred the character names of 408 movies in the dataset, it shows the value of identifying speakers to provide support for movie understanding, even if its predictive power is not extremely strong.

Q3: Can we create character representations that can more effectively identify the relations between characters?

Existing models represent name mentions of characters by looking up their embeddings from a Word2Vec or Glove models. From our analysis, we found that character name embeddings do not reflect useful information about them in a given movie or story. In Chapter 5, we proposed using speaker prediction as an auxiliary task to embed story character names in dialogues. To evaluate the quality of the character embeddings, we collected human judgments on how related each pair of characters are to each other in a given movie.

We then compared our character embedding models to other commonly used word and sentence embedding models. Our experiments have shown several positive properties of the character name embeddings. The distances between the embeddings representing the characters were better correlated with human judgments than a regular Word2Vec model. We also found these embeddings to predict relationships between characters better than several word embedding baselines. As our main goal in this thesis is story understanding, we experimented using our character embeddings for both character relation classification [102] and TV question answering [30] tasks. On both tasks, our character representations improved the classification and question answering tasks.

Q4: Can we use these character representations to model relations between characters over time?

Changes to characters and their relations are crucial elements for stories and their progression. Using static embeddings to represent a character through a story causes the temporal information in the story to be lost. In Chapter 6, we explored modeling character relations and their changes in movies. To facilitate the development and evaluation of our character relations model, we constructed a new inter-character relationship dataset for movies. This dataset captures the introduction and changes in relations throughout movie events. The collected relations also capture different facets of these relations, in particular, sentiment, fine-, and coarse-grain relations. We then developed a supervised multimodal framework for inter-character relationship prediction between pairs of characters at any given point in the story. We modeled the relationships by leveraging all movie segments that each character speaks in as well as their common interaction segments. Our multimodal model outperformed methods that use a single modality and several non-neural baselines as well.

7.2 Final Remarks

Throughout this thesis, we have taken several approaches to represent characters in movies and TV shows. We have shown that multimodal systems generally outperform systems that rely on one modality at a time. However, there is still a long way ahead for building machines that can "watch" a movie and fully understand it as humans do. Understanding movies is a very challenging task. Beside processing information from several sources at once, it requires a wealth of world knowledge and commonsense reasoning. An important question is how to build this knowledge and integrate it within a model during training.

Finally, we hope that the resources, tools, and methodologies presented in this thesis encourage further research on building character-centric approaches for movie understanding.

BIBLIOGRAPHY

[1] Van Der Maaten, L., "Accelerating t-SNE using tree-based algorithms." *Journal of machine learning research*, 2014.

[2] Winston, P. H., "The Strong Story Hypothesis and the Directed Perception Hypothesis," 2011.

[3] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A., "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, 2015.

[4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, 2015.

[5] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P., "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[6] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S., "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[7] Lowe, D. G., "Object recognition from local scale-invariant features," *The proceedings of the seventh IEEE international conference on Computer vision*, 1999.

[8] Nadeau, D. and Sekine, S., "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, 2007.

[9] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[10] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L., "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014.

[11] Kim, Y., "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[13] Pennington, J., Socher, R., and Manning, C., "Glove: Global vectors for word representation," *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[14] Wang, W., Yan, M., and Wu, C., "Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[15] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] Wang, M., Azab, M., Kojima, N., Mihalcea, R., and Deng, J., "Structured matching for phrase localization," *European Conference on Computer Vision*, 2016.

[17] Wang, L., Qiao, Y., and Tang, X., "Action recognition with trajectory-pooled deep-convolutional descriptors," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[18] Feichtenhofer, C., Pinz, A., and Zisserman, A., "Detect to Track and Track to Detect," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] Diba, A., Fayyaz, M., Sharma, V., Karami, A. H., Arzani, M. M., Yousefzadeh, R., and Van Gool, L., "Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification," *arXiv preprint arXiv:1711.08200*, 2017.

[20] Paul, S., Roy, S., and Roy-Chowdhury, A. K., "W-TALC: Weakly-supervised Temporal Activity Localization and Classification," *arXiv preprint arXiv:1807.10418*, 2018.

[21] Karpathy, A. and Fei-Fei, L., "Deep visual-semantic alignments for generating image descriptions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[22] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D., "Show and tell: A neural image caption generator," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[23] Plummer, B. A., Mallya, A., Cervantes, C. M., Hockenmaier, J., and Lazebnik, S., "Phrase localization and visual relationship detection with comprehensive image-language cues," *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[24] Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B., "A dataset for movie description," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[25] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T., "Long-term recurrent convolutional networks for visual recognition and description," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[26] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D., "VQA: Visual Question Answering," *International Conference on Computer Vision (ICCV)*, 2015.

[27] Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., and Batra, D., "Visual Dialog," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] Yu, L., Park, E., Berg, A. C., and Berg, T. L., "Visual madlibs: Fill in the blank description generation and question answering," *Proceedings of the ieee international conference on computer vision*, 2015.

[29] Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S., "MovieQA: Understanding Stories in Movies through Question-Answering," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] Lei, J., Yu, L., Bansal, M., and Berg, T. L., "TVQA: Localized, Compositional Video Question Answering," *EMNLP*, 2018.

[31] Kim, K.-M., Heo, M.-O., Choi, S.-H., and Zhang, B.-T., "Deepstory: Video story qa by deep embedded memory networks," *arXiv preprint arXiv:1707.00836*, 2017.

[32] Hodosh, M., Young, P., and Hockenmaier, J., "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, 2013.

[33] Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S., "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[34] Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L., "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[35] Zhou, L., Kalantidis, Y., Chen, X., Corso, J. J., and Rohrbach, M., "Grounded Video Description," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] Li, Y., Song, Y., Cao, L., Tetreault, J., Goldberg, L., Jaimes, A., and Luo, J., "TGIF: A New Dataset and Benchmark on Animated GIF Description," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[37] Jang, Y., Song, Y., Yu, Y., Kim, Y., and Kim, G., "Tgif-qa: Toward spatio-temporal reasoning in visual question answering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[38] Zhu, Y., Groth, O., Bernstein, M., and Fei-Fei, L., "Visual7w: Grounded question answering in images," *arXiv preprint arXiv:1511.03416*, 2015.

[39] Guadarrama, S., Rodner, E., Saenko, K., Zhang, N., Farrell, R., Donahue, J., and Darrell, T., "Open-vocabulary object retrieval," *Robotics: Science and Systems*, 2014.

[40] Arandjelovic, R. and Zisserman, A., "Multiple queries for large scale specific object retrieval." *BMVC*, 2012.

[41] Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., and Darrell, T., "Natural language object retrieval," *arXiv preprint arXiv:1511.04164*, 2015.

[42] Karpathy, A., Joulin, A., and Li, F. F. F., "Deep fragment embeddings for bidirectional image sentence mapping," *Advances in neural information processing systems*, 2014.

[43] Karpathy, A. and Fei-Fei, L., "Deep visual-semantic alignments for generating image descriptions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[44] Kong, C., Lin, D., Bansal, M., Urtasun, R., and Fidler, S., "What are you talking about? text-to-image coreference," *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014.

[45] Wang, L., Li, Y., and Lazebnik, S., "Learning Deep Structure-Preserving Image-Text Embeddings," *arXiv preprint arXiv:1511.06078*, 2015.

[46] Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., and Schiele, B., "Grounding of Textual Phrases in Images by Reconstruction," *arXiv preprint arXiv:1511.03745*, 2015.

[47] Klein, B., Lev, G., Sadeh, G., and Wolf, L., "Fisher Vectors Derived from Hybrid Gaussian-Laplacian Mixture Models for Image Annotation," *arXiv preprint arXiv:1411.7399*, 2014.

[48] Gong, Y., Wang, L., Hodosh, M., Hockenmaier, J., and Lazebnik, S., "Improving image-sentence embeddings using large weakly annotated photo collections," *Computer Vision–ECCV 2014*, Springer, 2014.

[49] Hoi, S. C., Liu, W., Lyu, M. R., and Ma, W.-Y., "Learning distance metrics with contextual constraints for image retrieval," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[50] Kiros, R., Salakhutdinov, R., and Zemel, R. S., "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[51] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D., "Show and tell: A neural image caption generator," *arXiv preprint arXiv:1411.4555*, 2014.

[52] Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., and Saenko, K., "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," *Proceedings of the IEEE international conference on computer vision*, 2013.

[53] Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., and Schiele, B., "Translating video content to natural language descriptions," *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[54] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T., "Long-term recurrent convolutional networks for visual recognition and description," *arXiv preprint arXiv:1411.4389*, 2014.

[55] Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A., "Deep captioning with multimodal recurrent neural networks (m-rnn)," *arXiv preprint arXiv:1412.6632*, 2014.

[56] Malinowski, M. and Fritz, M., "A multi-world approach to question answering about real-world scenes based on uncertain input," *Advances in neural information processing systems*, 2014, pp. 1682–1690.

[57] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D., "Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] Lei, J., Yu, L., Berg, T. L., and Bansal, M., "TVQA+: Spatio-Temporal Grounding for Video Question Answering," *Tech Report, arXiv*, 2019.

[59] Lu, J., Yang, J., Batra, D., and Parikh, D., "Hierarchical question-image co-attention for visual question answering," *Advances In Neural Information Processing Systems*, 2016.

[60] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M., "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[61] Yang, Z., He, X., Gao, J., Deng, L., and Smola, A., "Stacked attention networks for image question answering," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[62] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L., "Bottom-up and top-down attention for image captioning and visual question answering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[63] Na, S., Lee, S., Kim, J., and Kim, G., "A Read-Write Memory Network for Movie Story Understanding," *International Conference on Computer Vision (ICCV)*, 2017.

[64] Wang, B., Xu, Y., Han, Y., and Hong, R., "Movie question answering: remembering the textual cues for layered visual contents," *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[65] Kim, J., Ma, M., Kim, K., Kim, S., and Yoo, C. D., "Progressive Attention Memory Network for Movie Story Question Answering," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[66] Kim, K.-M., Choi, S.-H., Kim, J.-H., and Zhang, B.-T., "Multimodal dual attention memory for video story question answering," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 673–688.

[67] Everingham, M., Sivic, J., and Zisserman, A., ""Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video." *BMVC*, 2006.

[68] Cour, T., Sapp, B., Nagle, A., and Taskar, B., "Talking pictures: Temporal grouping and dialog-supervised person recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[69] Bäuml, M., Tapaswi, M., and Stiefelhagen, R., "Semi-supervised Learning with Constraints for Person Identification in Multimedia Data," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[70] Haurilet, M.-L., Tapaswi, M., Al-Halah, Z., and Stiefelhagen, R., "Naming TV Characters by Watching and Analyzing Dialogs," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[71] Tapaswi, M., Bäuml, M., and Stiefelhagen, R., "Improved weak labels using contextual cues for person identification in videos," *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015.

[72] Reynolds, D. A., "An overview of automatic speaker recognition technology," *Acoustics, speech, and signal processing (ICASSP)*, 2002.

[73] Campbell, J. P., "Speaker recognition: A tutorial," *Proceedings of the IEEE*, 1997.

[74] Sivic, J., Everingham, M., and Zisserman, A., ""Who are you?" – Learning person specific classifiers from video," *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2009.

[75] Tapaswi, M., Bäuml, M., and Stiefelhagen, R., ""Knock! Knock! Who is it?" probabilistic person identification in TV-series," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[76] Ramanathan, V., Joulin, A., Liang, P., and Fei-Fei, L., "Linking people with "their" names using coreference resolution," *IEEE Conference on European Conference on Computer Vision (ECCV)*, 2014.

[77] Hu, Y., Ren, J. S., Dai, J., Yuan, C., Xu, L., and Wang, W., "Deep Multimodal Speaker Naming," *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, 2015.

[78] Ren, J., Hu, Y., Tai, Y.-W., Wang, C., Xu, L., Sun, W., and Yan, Q., "Look, Listen and LearnA Multimodal LSTM for Speaker Identification," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[79] Bredin, H. and Gelly, G., "Improving speaker diarization of TV series using talking-face detection and clustering," *Proceedings of the 24th Annual ACM Conference on Multimedia*, 2016.

[80] Bost, X. and Linares, G., "Constrained speaker diarization of TV series based on visual patterns," *IEEE Spoken Language Technology Workshop (SLT)*, 2014.

[81] Li, Y., Narayanan, S. S., and Kuo, C.-C. J., "Adaptive speaker identification with audiovisual cues for movie content analysis," *Pattern Recognition Letters*, 2004.

[82] Liu, C., Jiang, S., and Huang, Q., "Naming faces in broadcast news video by image google," *Proceedings of the 16th ACM international conference on Multimedia*, 2008.

[83] Nagrani, A., Chung, J. S., and Zisserman, A., "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[84] Nagrani, A. and Zisserman, A., "From benedict cumberbatch to sherlock holmes: Character identification in TV series without a script," *arXiv preprint arXiv:1801.10442*, 2018.

[85] Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[86] Chen, D. and Manning, C., "A fast and accurate dependency parser using neural networks," *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[87] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[88] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, 2013.

[89] Turney, P. D. and Pantel, P., "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, 2010.

[90] Wilson, S. and Mihalcea, R., "Measuring Semantic Relations between Human Activities," *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017.

[91] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A., "Transition-based dependency parsing with stack long short-term memory," *arXiv preprint arXiv:1505.08075*, 2015.

[92] Ji, Y., Tan, C., Martschat, S., Choi, Y., and Smith, N. A., "Dynamic entity representations in neural language models," *arXiv preprint arXiv:1708.00781*, 2017.

[93] Azab, M., Wang, M., Smith, M., Kojima, N., Deng, J., and Mihalcea, R., "Speaker Naming in Movies," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[94] Agarwal, A., Kotalwar, A., and Rambow, O., "Automatic extraction of social networks from literary text: A case study on alice in wonderland," *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013.

[95] Makazhanov, A., Barbosa, D., and Kondrak, G., "Extracting family relationship networks from novels," *arXiv preprint arXiv:1405.0603*, 2014.

[96] Elson, D. K., Dames, N., and McKeown, K. R., "Extracting social networks from literary fiction," *Proceedings of the 48th annual meeting of the association for computational linguistics*, Association for Computational Linguistics, 2010.

[97] Iyyer, M., Guha, A., Chaturvedi, S., Boyd-Graber, J., and Daumé III, H., "Feuding families and former friends: Unsupervised learning for dynamic fictional relationships," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[98] Bamman, D., OConnor, B., and Smith, N. A., "Learning latent personas of film characters," *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[99] Bamman, D., "book-nlp: Natural language processing pipeline that scales to book-length documents," https://github.com/dbamman/book-nlp, 2014.

[100] Srivastava, S., Chaturvedi, S., and Mitchell, T., "Inferring interpersonal relations in narrative summaries," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[101] Chaturvedi, S., Srivastava, S., Daumé III, H., and Dyer, C., "Modeling Evolving Relationships Between Characters in Literary Novels." *AAAI*, 2016.

[102] Massey, P., Xia, P., Bamman, D., and Smith, N. A., "Annotating character relationships in literary texts," *arXiv preprint arXiv:1512.00728*, 2015.

[103] Weng, C.-Y., Chu, W.-T., and Wu, J.-L., "Movie analysis based on roles' social network," *2007 IEEE International Conference on Multimedia and Expo*, 2007.

[104] Weng, C.-Y., Chu, W.-T., and Wu, J.-L., "Rolenet: Movie analysis from the perspective of social networks," *IEEE Transactions on Multimedia*, 2009.

[105] Ding, L. and Yilmaz, A., "Learning Relations among Movie Characters: A Social Network Perspective," *ECCV*, 2010.

[106] Park, S.-B., Oh, K.-J., and Jo, G. S., "Social network analysis in a movie using character-net," *Multimedia Tools and Applications - MTA*, 2011.

[107] Agarwal, A., Kotalwar, A., and Rambow, O., "Automatic extraction of social networks from literary text: A case study on alice in wonderland," *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013.

[108] Fathi, A., Hodgins, J. K., and Rehg, J. M., "Social interactions: A first-person perspective," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012.

[109] Agarwal, A., Kotalwar, A., Zheng, J., and Rambow, O., "SINNET: Social Interaction Network Extractor from Text," *The Companion Volume of the Proceedings of IJCNLP: System Demonstrations*, 2013.

[110] Kočiskỳ, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E., "The narrativeqa reading comprehension challenge," *Transactions of the Association for Computational Linguistics*, 2018.

[111] Zitnick, C. L. and Dollár, P., "Edge boxes: Locating object proposals from edges," *European Conference on Computer Vision*, Springer, 2014.

[112] Girshick, R., "Fast r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[113] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "Imagenet: A large-scale hierarchical image database," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009.

[114] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A., "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[115] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, 2013.

[116] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y., "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, 2005.

[117] Clark, K. and Manning, C. D., "Entity-Centric Coreference Resolution with Model Stacking," *Association for Computational Linguistics (ACL)*, 2015.

[118] Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J., "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, 2004.

[119] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M., "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[120] Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S., "Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models," *arXiv preprint arXiv:1505.04870v3*, 2015.

[121] Zhang, Y.-F., Xu, C., Lu, H., and Huang, Y.-M., "Character Identification in Feature-Length Films Using Global Face-Name Matching," *IEEE Transactions on Multimedia*, 2009.

[122] Tapaswi, M., Bäuml, M., and Stiefelhagen, R., "StoryGraphs: Visualizing Character Interactions as a Timeline," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[123] Arandjelovic, O. and Zisserman, A., "Automatic face recognition for film character retrieval in feature-length films," *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[124] Erzin, E., Yemez, Y., and Tekalp, A. M., "Multimodal speaker identification using an adaptive classifier cascade based on modality reliability," *IEEE Transactions on Multimedia*, 2005.

[125] Kapsouras, I., Tefas, A., Nikolaidis, N., and Pitas, I., "Multimodal Speaker Diarization Utilizing Face Clustering Information," *International Conference on Image and Graphics*, 2015.

[126] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S., "Skip-thought vectors," *Advances in neural information processing systems*, 2015.

[127] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P., "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[128] Khorram, S., Gideon, J., McInnis, M., and Provost, E. M., "Recognition of Depression in Bipolar Disorder: Leveraging Cohort and Person-Specific Knowledge," *Interspeech*, 2016.

[129] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y., "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, 2016.

[130] King, D. E., "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, 2009.

[131] Danelljan, M., Häger, G., Khan, F., and Felsberg, M., "Accurate scale estimation for robust visual tracking," *British Machine Vision Conference (BMVC)*, 2014.

[132] Parkhi, O. M., Vedaldi, A., and Zisserman, A., "Deep Face Recognition," *British Machine Vision Conference (BMVC)*, 2015.

[133] Kazemi, V. and Sullivan, J., "One millisecond face alignment with an ensemble of regression trees," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[134] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D., "The Stanford CoreNLP Natural Language Processing Toolkit," *Association for Computational Linguistics (ACL) System Demonstrations*, 2014.

[135] Levitan, S. I., Mishra, T., and Bangalore, S., "Automatic Identification of Gender from Speech," *Speech Prosody*, 2016.

[136] Mathieu, B., Essid, S., Fillon, T., Prado, J., and Richard, G., "YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software," *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010.

[137] Eyben, F., Weninger, F., Gross, F., and Schuller, B., "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor," *Proceedings of the 21st ACM International Conference on Multimedia*, 2013.

[138] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.

[139] Sukhbaatar, S., Weston, J., Fergus, R., et al., "End-to-end memory networks," *Advances in neural information processing systems (NIPS)*, 2015.

[140] Barteld, F., "Detecting spelling variants in non-standard texts," *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[141] Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B., "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.

[142] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L., "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[143] Schnabel, T., Labutov, I., Mimno, D., and Joachims, T., "Evaluation methods for unsupervised word embeddings," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[144] Yih, W.-t. and Qazvinian, V., "Measuring Word Relatedness Using Heterogeneous Vector Space Models," *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012.

[145] Upadhyay, S., Faruqui, M., Dyer, C., and Roth, D., "Cross-lingual models of word embeddings: An empirical comparison," *arXiv preprint arXiv:1604.00425*, 2016.

[146] Elsner, M., "Character-based Kernels for Novelistic Plot Structure," *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2012.

[147] Gorinski, P. J. and Lapata, M., "Movie script summarization as graph-based scene extraction," *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

[148] Le, Q. and Mikolov, T., "Distributed representations of sentences and documents," *International conference on machine learning*, 2014.

[149] Řehůřek, R. and Sojka, P., "Software Framework for Topic Modelling with Large Corpora," *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta.

[150] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, Vol. 12, No. Aug, 2011.

[151] Chaturvedi, S., Iyyer, M., and Daume III, H., "Unsupervised learning of evolving relationships between literary characters," *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[152] Schuller, B., Steidl, S., and Batliner, A., "The interspeech 2009 emotion challenge," *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[153] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015.

[154] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, 2017.

[155] Yin, X. and Ordonez, V., "Obj2text: Generating visually descriptive language from object layouts," *arXiv preprint*, 2017.

[156] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.