ABSTRACT

Title of dissertation: DATA AND METHODS
FOR REFERENCE RESOLUTION
IN DIFFERENT MODALITIES

Anupam Guha, Doctor of Philosophy, 2017

Dissertation directed by: Professor Yiannis Aloimonos
Computer Science, University of Maryland
Professor Jordan Boyd-Graber
Computer Science, University of Colorado

One foundational goal of artificial intelligence is to build intelligent agents which interact with humans, and to do so, they must have the capacity to infer from human communication what concept is being referred to in a span of symbols. They should be able, like humans, to map these representations to perceptual inputs, visual or otherwise. In NLP, this problem of discovering which spans of text are referring to the same real-world entity is called Coreference Resolution. This dissertation expands this problem to go beyond text and maps concepts referred to by text spans to concepts represented in images. This dissertation also investigates the complex and hard nature of real world coreference resolution. Lastly, this dissertation expands upon the definition of references to include abstractions referred by non-contiguous text distributions.

A central theme throughout this thesis is the paucity of data in solving hard problems of reference, which it addresses by designing several datasets. To investigate hard text coreference this dissertation analyses a domain of coreference heavy text, namely questions present in the trivia game of quiz bowl and creates a novel dataset. Solving quiz bowl questions requires robust coreference resolution and world knowledge, something humans possess but current models do not. This work uses distributional semantics for world knowledge. Also, this work addresses the subproblems of coreference like mention detection. Next, to investigate complex visual representations of concepts, this dissertation uses the domain of paintings. Mapping spans of text in descriptions of paintings to regions of paintings being described by that text is a non-trivial problem because paintings are sufficiently harder than natural images. Distributional semantics are again used here. Finally, to discover prototypical concepts present in distributed rather than contiguous spans of text, this dissertation investigates a source which is rich in prototypical concepts, namely movie scripts. All movie narratives, character arcs, and character relationships, are distilled to sequences of interconnected prototypical concepts which are discovered using unsupervised deep learning models, also using distributional semantics. I conclude this dissertation by discussing potential future research in downstream tasks which can be aided by discovery of referring multi-modal concepts.

DATA AND METHODS FOR REFERENCE RESOLUTION IN DIFFERENT MODALITIES

by

Anupam Guha

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Yiannis Aloimonos, Advisor
Professor Jordan Boyd-Graber, Co-Advisor
Dr. Cornelia Fermüller
Professor Hal Daumé III
Professor Philip Resnik, Dean's Representative

Dedication

Dedicated to the fond memory of Mr. Deepak Guha,

My Late Father,

and to the fond memory of Sumegha Gulati,

Dearest Friend

Deepak Guha
(1954 - 2010)

Sumegha Gulati
(1990 - 2016)

"Attired with stars, we shall for ever sit,

Triumphing over Death, and Chance,

and thee, O Time!"

- John Milton

Acknowledgments

This dissertation would not have been possible without the advice, support, love, and concern of many people over the last six years of my academic life at UMD. I am most fortunate to have known these people and would like to thank them.

First, I thank Dr. Yiannis Aloimonos of the Computer Vision Lab and Dr. Jordan Boyd-Graber of the CLIP lab, my two advisors for their academic guidance and incredible patience with me. They both are excellent researchers with a deep grasp of their science, and also wise mentors with whom every conversation lead to me learning something. Yiannis with his grand experience and broad thinking inspired me to question conventional wisdom, and Jordan with his disciplined method taught me that one should not be satisfied with apparent results but analyse them thoroughly. I would like to think that over the years under their able tutelage has made me a much more optimistic and persevering researcher.

I would also like to thank Dr. Cornelia Fermüller of the vision group, and a part of the committee, with whom I have worked and without whose advice, academic rigour, and more importantly humour, life at UMD would have been quite dreary. I would also thank my two other committee members, Dr. Hal Daumé III and Dr. Philip Resnik without whose advice this dissertation wouldn't have existed. Hal was a part of many research groups I was in and his insights prompted a lot of ideas I have had. Here, I thank all other faculty members of the vision lab and CLIP. Outside of UMD I'm very grateful to have worked with Dr. Ferhan Ture of Comcast Research, a lot of that work is a part of this dissertation and has inspired

my future direction of research. Also, Ferhan is a wonderful colleague to work with.

My colleagues, former and current students at UMD, were an inspiring group of people and were responsible for shaping my research journey. I enjoyed working a lot with Dr. Yezhou Yang, who knows how to simplify complex problems and Dr. Mohit Iyyer, who redefines being at ease with himself, and was the life of the CLIP lab. I also worked with Michael Maynord, whose sincerity is unparalleled, and thanks to whose efforts we won a fellowship award! I would also like to mention Dr. Naho Orita, Hua He, Dr. Snigdha Chaturvedi, Dr. Anshul Sawant, Dr. Manish Purohit, Sudha Rao, Yogarshi Vyas, Aleksandrs Ecins, Dr. Alvin Grissom, and Varun Manjunatha. Thank you all for your support and camaraderie.

I thank the administrative staff at UMD who made this home of mine for six years a pleasant place to stay. Most importantly, Jennifer Story. She is indispensable to our work in the CS department and is an incredibly kind person. I also thank Jodie Gray and Sharron McElroy for being so helpful.

I will thank some dear friends at UMD who became important parts of my personal life and were vital for my mental well-being. First, my two best friends, Dr. He He and Yulu Wang. He taught, and continues to teach me to be a better version of myself, and she is easily the ablest peer, and the hardest working scholar I have known. Yulu is without doubt the kindest friend one could have. Her patience and empathy is a lesson to all us academics. Her taste in good food is also tremendous. I would also thank Apurv Mittal, Allyson Ettinger, and Sindhuja Devanapally, other friends I made at UMD. I would also like to thank my flatmates, Gregory Kramida and Prashanth Ravichandran, for being excellent company.

I thank my non-academic friends who stood by me, patiently bearing my caprices and absences, and thus did emotional labour for this dissertation. Shipra Jain, Sumit Arora, Puneet Gulati, Harish Alagappa, Harish Mandala, Nitesh Bhasin, Dinesh Kapur, Aditi Asthana, Anubhuti Arora, Emily Cheung, Chanpreet Kathuria, Surya Panicker, Varun Gupta, Latika Arora, and Karthikeya Ramesh, thanks for sticking around, and inspiring me to keep going on. In my family I would specifically like to mention Rudrani Banerjee and Ritika Kar. There are many more friends and family, both old and new, and I thank all of them for giving me the courage to make this dissertation possible.

Finally, I would like to thank the four most important people in my life. First, Sumegha Gulati, a young inspirational friend who believed in me, the bravest person I knew, and who left us so early after a gallant fight with cancer. Thank you for all the happiness you gave me and the meaning you gave my work. Second, my late father, Mr. Deepak Guha, who was a wonderful person with an unmatched wit and a great heart who gambled every hope and resource he had in me before leaving us so early. Thank you for helping me be the person I am and teaching me how to think. Third, my mother, Mrs. Bharati Rani Guha to whom I could never be the son she deserves. She has sacrificed so much while I was at the other end of the world, and I cannot thank her completely with words. I end by thanking Anupama Jha, my partner who stood by me for my entire academic journey and did not let me waver. She is my pole star, loyal opposition when need be, and most zealous champion. The history of my work is a history of our journey together, and I look forward to journeys ahead, thank you so much.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In the real world, humans have a strong ability to know from the context of a conversation, what they are talking about. While engaging in discourse (discourse being any text with more than one sentence) humans do not specify word-by-word what entity, event, or concept they are referring to, especially in conversations, and this does not prove to be a barrier to comprehension for the listener. In NLP, a reference is a symbolic relationship between a linguistic expression and a concrete object or a concept. This thesis tackles this problem of inferring what is being referred to.

For example, let us look at the sentence:

Marty assured Doc Brown that he ought to have faith in him

In this sentence, the first 'he' refers to Marty, while the second 'he' refers to Doc Brown. In NLP this problem of resolving which spans of text refer to the same real world entity is called *coreference resolution*. In the field of linguistics, this task is related to the theory of binding (Büring, 2005) which investigates what relations exist between these coreferent text spans. In this sentence, both the 'he's are called *anaphora* while 'Marty' and 'Doc Brown' are called *antecedents*. Thus, coreference resolution, alongside some other related problems, is also called anaphora resolution.

Just like anaphora, there can be text spans before their referent mentions in the text and these are called cataphora. Anaphora can be pronouns, nouns, or long phrases. This is an important phenomenon to be investigated because this is how discourse is constructed and understood.

However, coreference resolution is not the only kind of interesting problem in resolving references. References may not be to concrete objects but could be oblique associations. References may be to entities described through some other form of representation, like images. In this thesis I expand on the definition of references to include references to concepts which are abstractions like themes, archetypes, or topics, represented not by contiguous spans of text but rather by words dispersed over documents.

I present in this thesis an investigation of multiple reference problems in various kinds of data. I investigate coreference resolution in a domain rich with hard coreference problems of the kind humans can and do solve routinely but which current systems fail gracelessly at. I investigate why that happens and how datasets can reflect the kind of variety of referential problems humans solve in discourse. I demonstrate how distributional semantics can be used to help solve such problems. I also investigate the side problems of mention detection and bridging anaphora.

Aside from text reference, in this thesis I investigate the problem of spans of text referring to depictions in atypical images, i.e. I extend the problem to make it multimodal. Finally, I expand upon the conventional definition of reference and investigate how spans of text can refer to prototypical concepts (abstractions) rather than concrete entities, each concept defined by discontinuous text spans, which is

different from referring mentions which are contiguous spans. I present methods of discovering these concepts and matching them to text spans.

1.1 What are we talking about?

When humans talk about the world, meaningfully, they need to have a mental mechanism of what talking about something is. Thus, humans *refer* to objects (and concepts). In semantics, references are relations existing between various kinds of representations and concrete objects or abstractions. These representations/tokens do not need to be text based, they can be images, or even mental states. A lot of the work on reference in semantics is concerned with the *mechanism* of reference, i.e. why does a word attach to an object? Data driven methods in NLP leave out the question of why something is a reference but focus on which token is a reference to what. This work, maps references from both tokens in text as well as images, and when it addresses images it has to infer from tokens what might indicate a reference relation.

Furthermore, all attachments between tokens and objects are not references. References differ from denotations (Engelbretsen, 1980), which are simple expressions between concepts in language and entities in the world. References depend on context as well as who is speaking, making the task more complex. If a reference uniquely identifies a thing in the world, like a proper noun or a pronoun with a determiner, it is called a definite reference. But references could also be indefinite, i.e. not care about the identity of the entity. These often use indefinite articles with

the NP.

What of entities which are not concrete? While the field of reference does make allowances for fictional named objects, say, a dragon, with some philosophers claiming there is a reference here (Meinong, 1960), and others claiming if not a reference, at least meaning exists (Quine et al., 2013), this work expands on the definition of reference to include mappings from tokens to unstated abstractions, which are not necessarily named entities. It does this because when humans talk of the world they also obliquely refer to such abstractions and these unstated abstractions are a part of the shared knowledge. Thus, if the context of a section of text indicates that it is talking about "tragedy", without explicitly referring to a tragic entity, for the purposes of this work it is a reference from that text section to the concept of tragedy.

The way references are investigated as an NLP research problem currently leaves a lot to be desired if compared to the capabilities humans have. For example, humans can use implicit understandings, common sense, world knowledge, and semantic reasoning to infer on the fly what is being referred to (Ng, 2010). Here, world knowledge is defined as knowledge which has nothing to do with linguistic competence (which includes semantics) but rather shared information (Clark and Marshall, 1978) between about how the world (or even some shared fictional frame) works (Hobbs, 1987). For example, humans know that *dragons fly*, despite there being no real entity of dragon. This kind of knowledge is vital for resolving references.

Humans can detect much more than coreferrent spans of text in an "is-a"

relationship (also called an identity relationship), they can infer many kinds of semantic relationships and for much more than just entities, they can infer what topic, event, concept, or prototype is being referred to. One such kind of reference is called *Bridging Anaphora* (Clark, 1975).

For example, let us look at these two sentences:

I heard you went to the Thai restaurant this morning? Oh yes, the waiter is from our school!

In this small conversation between two people, it is obvious that the waiter being referred to in the second sentence is one who works in said Thai restaurant. Although this fact is not explicitly stated, the speaker of the second sentence assumes that since it is common knowledge that waiters work at restaurants, and since the sentence immediately follows the one where such a restaurant is mentioned, it need not be stated again. This kind of reference is different from the *Identity Anaphora* of the last example.

Moreover, in their daily interactions, humans use visual information alongside world knowledge. When two humans talk of some phenomenon which just happened, they do not need to mention in precise words the descriptions of what they observed. It is assumed that the listener will be able to infer a lot from shared knowledge. Humans use descriptions in natural language text of visual scenes in order to convey an understanding of the world to other humans, and are able to resolve successfully which parts of their descriptions match what parts of their vision and vice versa.

Similarly, over large spans of text, humans can infer themes, prototypes, and

other kinds of abstractions which are referred repeatedly if obliquely, even if these concepts are not directly mentioned by name. For example, a human can tell from a story that certain spans of text in it has some reference to the concept of "tragedy", indicated by the presence of certain words, even if there is no explicit sentence stating that it is tragic. In discourse humans can easily infer that certain prototypical concepts exist and where they are being referred. Hence, as mentioned before, this work expands the conventional usage of the term reference. Also, it is evident that to do the more conventional kind of reference resolution, a shared knowledge of context is needed which can be provided by this expanded definition.

For autonomous systems to interact successfully with humans, to engage in idiomatic discourse, they need to solve reference to entities and concepts robustly. This is necessary for tasks like question answering (Morton, 1999) and entailment (Mirkin et al., 2010) which depend on knowing what is being talked about. The importance of this problem is not limited to human-AI interaction either. Tasks like opinion analysis (Ding and Liu, 2010), if done with sophisticated questionnaires with unstructured data need to know who has what opinion on which subject, a reference heavy task.

Robust multi-modal reference resolution makes it easy to annotate large datasets, something that has become important for data hungry machine learning research. It is easier for human annotators to convey information about images or videos in natural language instead of complicated annotation tags and thus, it is possible to scale vision datasets if the annotations are in natural language. In order to use visual or linguistic *concepts* to regularise the training of any vision model in machine

learning, reference resolution across modalities becomes essential.

1.2 Current Methods

Current work in text based anaphora resolution in NLP focuses largely on entity resolution and not much on other kinds of related phenomena. This is primarily due to a limitation of datasets, an artefact of decisions made during the MUC-6 conference in '95 (MUC-6, 1995) which had very specific requirements, and thus research into many areas of reference is disincentivised. Even within entity coreference resolution, the current datasets of OntoNotes, ACE, and MUC described in MUC-7 (1997) and Pradhan et al. (2011), do not cover a wealth of problems which humans solve on a regular basis to complete downstream tasks. The history of coreference resolution research, especially in English, with respect to both policy and available data has had a tangible effect of restricting the problem to what this work calls "newswire" data, i.e. data obtained from journalistic and related sources. This has created a monoculture, and human discourse is richer than what is reflected in such data. This dissertation attempts to rectify this.

The current data driven methods (Björkelund and Kuhn, 2014; Durrett and Klein, 2013, 2014; Fernandes et al., 2012), as opposed to past rule-based methods (Lee et al., 2011; Pradhan et al., 2011), to solve entity coreference resolution depend on first finding *mentions*, i.e. spans of text which may refer to some real world entity, and then grouping these mentions into clusters. These current methods can be broadly divided into two groups, the first kind goes through the mentions

one at a time, each mention either starting its own cluster or being ranked against all the previous discovered mentions to assign one cluster which is similar enough to it as its cluster. The process continues until all the mentions have been evaluated and clustered. This method is called *mention ranking*. The second class of methods finds for all possible pairs of mentions a binary decision of whether or not they are coreferent. Then the binary linkages are clustered. This method is thus called *pair-wise classification*. Other methods for mention clustering do exist like antecedent trees but they can be considered an extension of these two methods.

Generally, the process of first finding mentions and then clustering them is pipelined, though joint models include Durrett and Klein (2014) and Peng et al. (2015). In current methods, mention detection is an ignored problem, generally a rule based method with very high precision and very low recall (Durrett and Klein, 2013) is used which creates many spurious mentions. The second step of the pipeline, namely clustering, is also used to weed out the spurious mentions by not assigning them to any clusters. This class of methods has a lot of problems, one of which is that it ignores the question of "what is a mention?" by marking every possible nominal or pronominal phrase as a candidate whereas, as I show in this work, what a mention is differs widely among datasets and domains and depends primarily on "what we are talking about", i.e. what referring entities are interesting to the downstream task. This has a lot to do with what the annotators agree on, and has consequences for domain adaptation problems, and incorporating generalised world knowledge into coreference resolution. For example, a dataset built around literature will have to decide whether entities mentioned in conversations between two characters count as

"real" entities. To further complicate this problem, the largest dataset for entity coreference, Ontonotes, is not annotated for mentions which are not coreferrent with any other mention, thus lacking a vital tool which could have been used to train data driven mention detectors, at least for that domain. While there have been efforts to incorporate semantic information into models optimised for this kind of data, the performance gains were small as there was a lack of datasets which needed world knowledge to solve the sort of coreference humans can solve. Consequently, there is a lack of research in models attempting to solve such problems. This dissertation addresses these issues.

As mentioned before, humans can resolve bridging anaphora with as much facility as identity anaphora. This is an extremely hard problem with current methods reaching nowhere near human performance. There are few datasets like ISNotes (Markert et al., 2012) and SciCorp (Rösiger, 2016) where bridging information is reliably annotated which makes the task of investigating new methods, especially deep learning methods which are data hungry, very difficult. Existing models split the task into two parts, the first being discovering which mentions are anaphors, and the second finding which antecedent matches the anaphor. Current methods generally ignore the first part of the process with the notable exception Hou et al. (2013a) which solves the first stage with a cascading collective classification model and the second stage with a joint inference model, reaching the performance of rule based systems (Hou et al., 2014). In this work I investigate both the stages of this pipeline.

The current methods in machine learning which employ both images and text

are generative for the text, they are about creating linguistic descriptions of images (Chen and Zitnick, 2014; Chen et al., 2015; Donahue et al., 2014; Karpathy and Fei-Fei, 2014; Mao et al., 2014; Vinyals et al., 2014; Xu et al., 2015) instead of using the complex relations in text to inform the learning of vision tasks. Thus, current methods tend to ignore the knowledge present in the conceptual scaffolding of language. Neither is the generation done in a *coreferent* manner, rather state-of-the-art systems are used to generate object classes and objecthood locations from within the image (this too is generally separate) and this data is used along with an bidirectional RNN to represent the object features and text descriptions in the same space, thus achieving alignment, and then using that to train an RNN to generate sentences. The text itself is treated as raw and semantic groupings within the text are completely ignored. This work on the other hand uses groupings of coreferent text to constrain the recognition task in images which aids in the identification of hard to recognise images, namely paintings.

I have discussed the idea of referring to concepts rather than entities. Humans can infer if a block of text refers to some kind of concept, and more importantly, if it is a repeating theme over a body of text, for example a book. For discovery of such prototypical concepts, the only method available was adapting temporal topic models like the HTMM (Gruber et al., 2007) to infer bags of words as topics, a mixture of which would be distributed over the document. The temporal nature of the HTMM, unlike other topic models like the LDA (Blei et al., 2003) ensures that the inferred topics are sequential in nature, thus reflecting the assumption that concepts in certain kind of text, like narratives, follow one another in order. Recently,

an unsupervised deep neural network model has been developed called the Relationship Modelling Network (RMN) (Iyyer et al., 2016), which can, given suitable data extracted from narrative text (say, a novel), generate a list of prototypical concepts, with each pairwise relationship between two characters being a sequence of some of these concepts. The limitation of the RMN is that it assumes that these relationship prototypes occur in isolation in the text from other kinds of concepts. In real narratives, prototypical concepts are interconnected, for example, the relationship prototype of "death", would be influenced by the character prototype of "sadness", which in turn could be influenced by the global theme of "war", in a novel which refers to these concepts in its text. This work extends the RMN to a joint model which can deal with multiple kinds of prototypes.

1.3 Contribution and Roadmap

In this thesis I build multiple datasets which are used to investigate novel problems of reference discovery, especially the kind of hard and interesting reference problems which are not reflected in current datasets and not solved by current methods. I demonstrate the need for better methods to solve these problems on such data, and build them. I solve reference problems in three domains, namely, hard text coreference, analysing (atypical) images using referential texts, and discovering references to prototypical concepts.

In addition to these being hard, they are a unique take on existing problems and have research consequences in other areas, for example, distilling narratives

as chains of prototypes can be used to aid recommendation. Similarly, current deep learning methods with images and text are data hungry, and are brittle when faced with small datasets of sufficiently different images like paintings or cartoons. Current work with images and text is generative, focused on generating text captions to images, while in Chapter 4 I investigate the use of language for retrieving and ranking atypical images. These methods are organised into three chapters plus a background chapter to ease comprehension. I end by presenting some potential future research directions for these problems.

Chapter 2 presents the background information needed for this dissertation. It goes over in detail the problem on coreference resolution, and the work done on it both past and contemporary. It explains the parts of the pipeline used in current approaches to the problem. The chapter covers the methods for mention detection and all the methods for grouping those mentions into coreferent clusters. The the chapter delves into various machine learning concepts which have been referred to in the following chapters, namely distributional semantics, deep networks, multimodal architectures, sequence labelling, and topic modelling, as understanding these methods is needed for this dissertation. The next chapters discuss my contribution.

Chapter 3 focuses on the domain of Quiz Bowl, a factoid question answering game of trivia whose questions are rich with referent phrases, and thus humans need to do coreference resolution on them to answer these questions. I review the current datasets and evaluation metrics, followed by an explanation of why the current material in both data and models do not a cover a vast scope in

the interesting phenomena occurring in coreference resolution. Then, I build a new dataset from the domain of quiz bowl, and demonstrate the inadequacy of current methods to solve such coreferences and the need for investigating the problem of world knowledge. Then, I design a simple method to solve such coreference with the help of distributional semantics. Then the chapter tackles the problem of bridging anaphora, describing its existing datasets and methods used to solve it. I investigate the utility of embeddings for bridging resolution, analogous to the method employed for coreference. Barring the experiments on bridging which is unpublished, most of the material comes from Guha et al. (2015), a work in which I am the primary contributor.

Chapter 4 motivates the need for extending the problem of reference to images. In it I review current methods which use images with language. I design a method for retrieving and ranking complex images based on groupings of coreferent text found in image descriptions. I build a complex image dataset which requires real world knowledge to understand which is adapted to a problem of retrieval and ranking. I solve this problem by a method which infers visual properties from text using distributional semantics. I also describe another interesting dataset, which comprises of complex images and referential text and how it can be used to investigate the problem of hard multi-modal references. The material in this chapter comes from two published works, Guha et al. (2016) and Iyyer et al. (2017).

Chapter 5 deals with concepts which are not entities, and thus cannot be referred to by contiguous spans of text. Instead they are "prototypes", like themes

of a narrative, events in a character arc, or relationships between characters, etc. In this chapter I expand upon the conventional usage of *reference* from referencing concrete entities, to referencing abstractions or concepts which are also required to interpret discourse (and other references in it). Humans regularly figure out what part of a text (or spoken dialogue) refers to what prototypical concept. In this chapter I design a dataset from movie scripts, as movies are rich in such concepts. I describe a model which can be used to discover such prototypes from narrative text, then I design an adaptation of this model into a joint model which discovers multiple categories of inter-related prototypical concepts from the movie script data. Then, I perform experiments demonstrating the utility of discovering these prototypical references. The joint model is unpublished work, done in collaboration with Dr. Ferhan Ture of Comcast Research in which my contribution is primary, namely the model and the majority of the analysis. The RMN model on which the model is based comes from Iyyer et al. (2016).

In **Chapter 6** I conclude by a summary of this work and describe the potential future avenues of this research.

Chapter 2: Background Machine Learning Concepts for Reference Resolution

In this chapter I provide the background information of relevant past and present work and briefly explain some machine learning concepts which will be useful to understand the thesis. I first explain in detail what the problem of reference resolution is and what subprocesses it involves. Next, to do hard reference problems, systems need a mechanism to embed a feature which captures world knowledge, and thus, this chapter describes word embeddings, which give a sense of how "similar" two words are. In this thesis I also extend the problem of references to the domain of vision, and for that I describe the current work which involves both vision and text, and to understand that I present machine learning systems used for computer vision. All these concepts are described briefly in the following sections. These concepts are interconnected, for example joint vision and language systems use both convolutional neural networks and sequence labelling. Reference resolution systems also use sequence labelling. All three use vector representations. I begin with describing existing work in coreference resolution.

2.1 Coreference Resolution: Past and Present

As mentioned in the introduction, the general body of work in coreference resolution treats it as a pipeline with mention detection followed by various methods of clustering those mentions into coreferent groups. There is one major exception to this: Peng et al. (2015) which treats both the tasks as a joint model. This model splits the mention detection task into two, finding the "head" of the mention (the head is the parent node in the dependency parse of a phrase) and finding the whole mention. This model after finding all the possible mention head candidates feeds them into a model which jointly decides whether each head is a mention head and its similarity to other heads, and only after this coreference is done are the complete mentions determined. However, most coreference resolution models are not joint and the mention detection is a separate earlier step described next.

2.1.1 Current Approaches in Mention Detection

The earlier works in statistical coreference resolution assumed that the mentions are noun phrases, as in Cardie et al. (1999) and Ng and Cardie (2002). Later, various sequence labelling statistical methods (Florian et al., 2004, 2006) were used for mention detection with the ACE (Doddington et al., 2004) dataset. Mention detection is a hard task and the lack of singleton annotations and detailed mention type annotations in the currently used and larger Ontonotes dataset (Pradhan et al., 2011) unlike the past ACE dataset incentivises rule based methods like the mention detection methods in Lee et al. (2011), Lee et al. (2013), or Durrett and

Klein (2013). In general, aside from Peng et al. (2015) there is paucity of recent work in mention detection relative to work in mention clustering.

The recent rule based systems follow the method which the Berkeley Coreference system (Durrett and Klein, 2013) uses. It marks three kinds of spans, proper mentions, pronominals, and all maximal NP projections. Proper mentions are discovered by Named Entity Recognition (NER gold annotations are provided in the Ontonotes dataset) with all NER types being proper mentions aside from those labelled CARDINAL, QUANTITY, or PERCENT. Pronominals and maximal NP projections are obtained via POS tags and parse trees, also provided in gold annotations. This method is based on the Stanford rule based coreference system (Lee et al., 2011) aside from any mechanism to weed out the spurious mentions thus generated. Some systems (Björkelund and Farkas, 2012; Rahman and Ng, 2009) do try to learn spuriousness in a data driven fashion but this is not considered necessary by present systems as the next step in the pipeline is trained not to put any of these spurious candidates in clusters.

Thus, current methods concentrate on a high precision mention candidate generation with a low recall, followed by a high recall clustering procedure to address the noise. This class of methods have a flaw. As Peng et al. (2015) point out, current systems suffer significantly in evaluation metrics when running on system generated mentions. Recently, due to the advent of various deep learning models in NLP tasks there has been work done to rule out spurious mentions in a data driven manner (Wiseman et al., 2015). These methods take raw, unconjoined features as input and attempt to find if the mention is an anaphor. Of course the same method

can be used to rank the potential antecedent of an anaphoric mention as described in the next section.

2.1.2 Current Approaches in Mention Ranking vs. Pairwise Classification

Once the mentions are detected the task is to cluster them into coreferent groups. The methods to do this can be divided broadly into two categories as described in the introduction, namely mention ranking and pairwise clustering as shown in Figure 2.1. Another class of methods exists called antecedent trees which encodes all antecedent decisions for all anaphora but it can be considered an extension of mention ranking. While these approaches to coreference resolution vary a lot in their modelling of the problem, Martschat and Strube (2015) unify them by treating the problem as one of structured prediction with latent variables.

Currently, mention ranking is achieved in two ways. In some works (Durrett and Klein, 2013; Wiseman et al., 2015) the mentions are compared against all the previous mentions individually, and the one with the highest rank tells which cluster the current mention belongs to. In other works (Björkelund and Kuhn, 2014; Wiseman et al., 2016) the mentions are not compared against individual mentions, but features obtained from previous clusters themselves to get the cluster with the highest rank and assigning the mention to it. The difference between these two approaches is illustrated in Figure 2.2. It has been long assumed that the "global" features of the cluster might give better ranking, than just ranking individually with

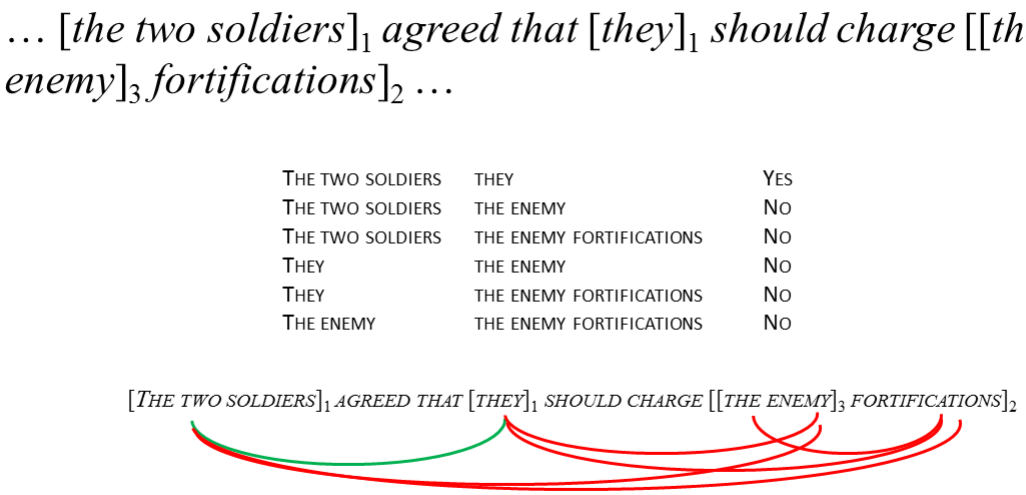

Figure 2.1: Pairwise classification vs. ranking of mentions among candidate antecedents. In the former there is a binary decision over all possible mention pairs but in the latter for a given mention the last mentions are considered in a sequence and only one is chosen.

local features per mention, but only in this recent work has the performance of the second type of ranking been better.

Generally, two kinds of raw feature sets are used in this task, the basic features which are obtained from the mention itself like type, number, gender, animacy, head word, etc. and the pairwise features which are obtained from the pair of mentions in consideration, like, are the two mention candidates in the same sentences, do their heads match, etc. Due to the manner in which pairwise features are computed it is more likely to do mention ranking with mentions rather than entire clusters. As pointed out in Wiseman et al. (2016) cluster level features are hard to define, because clusters are of different sizes. This was solved in this particular work by running an RNN over the vector representations (vector representations are defined in section 2.2) of mentions in a cluster, in sequence, and using the hidden state of

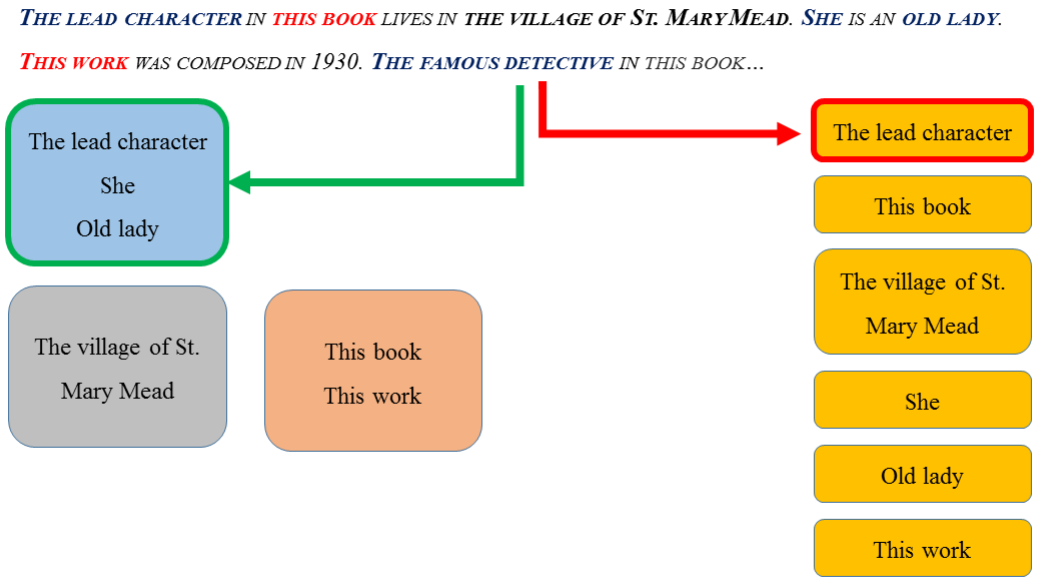

Figure 2.2: Mention ranking via comparing to individual mentions vs. comparing to entire clusters. In the former during the ranking process a mention is compared individually to previous mentions in a sequence using features inherent to the two mentions in question, but in the latter the mention to be considered is not compared to previous individual mention but to coreference chains already formed, and aside from features obtained from the mention, features at cluster level are used.

the RNN as the cluster feature. RNNs are described in subsection 2.5.2.

Another work which should be mentioned here is Durrett and Klein (2014) which also uses a joint model, namely doing Named Entity Recognition, entity linking, and mention clustering at the same time. Named Entity Recognition is essentially classifying mentions on semantic types. Entity linking is an allied problem to coreference resolution which involves linking a mention in text to a set of entities in a knowledge base. In the past entity linking was used for coreference, as well as coreference used for resolving ambiguity of semantic types or entity links, while this model does these three allied tasks at the same time.

Now let us discuss another way of clustering mentions. In pairwise classification as opposed to mention ranking the same feature sets are used but instead of giving scores to all the mentions against one mention, all possible pairs are given a binary decision (as is present in Chapter 3), and then clustering is done using that information. Clustering algorithms may vary for this task but usually some variant of best-first or closest-first is applied to the list of pairs. While it looks easier than mention ranking there are chances of more spurious linkages in the clustering. Mention pair models have another weakness, the sheer majority of negative examples for the pairwise instances in the training task make learning a robust classification model difficult, and it is easy to overfit to all negative decisions. To address this, these methods do resampling (Geibel and Wysotzki, 2003) and negative examples are taken only from a small neighbourhood of one mention, reducing the negative examples overall and improving performance. As this introduces cost sensitivity to learning, pairwise models do not need additional cost functions.

While this is one improvement on the mention ranking model, the pairwise model does not capture the comparison of scores between antecedents (or antecedent clusters) which the mention ranking model does and which reflects the online nature of natural language. A point of note here is that the same feature sets are used in both mention ranking and pairwise models. In both methods raw features have been seen to be not very useful for coreference resolution. However, conjoining the features with each other leads to massive improvements in all models, and thus state of the art systems rely on manual (Durrett and Klein, 2013) or automatic (Björkelund and Kuhn, 2014; Fernandes et al., 2012) conjunction schemes. Our method in Chapter 3

uses a manual scheme. An alternative to conjunction schemes is representation learning (Wiseman et al., 2015) but this has the disadvantage of being complex and requires pre-training for the feature representations.

2.2 Word Vector Representations

In all the following chapters there is a need for some measure of semantic similarity among phrases, a useful tool to do that is a kind of vector representation of words which which is a kind of shared world knowledge. Semantic similarity involves similarity in meaning, for example a car is more similar to a bus, than a carrot. A mechanism to obtain such representations is word embeddings (Mikolov et al., 2013; Pennington et al., 2014) (also called distributional semantics). These methods create vectors for words wherein the distance between two vectors captures the likelihood of them appearing in the same neighbourhood. While vector space models have existed since the nineties (Schütze, 1993), recent development in deep learning and existence of large text corpora make it possible to make improved embeddings reflecting similarity, compared to earlier methods like LSA (Landauer, 2006). Common methods used currently include the Continuous Bag of Words Model (CBOW) and the skip gram model. A skip gram model has for input, a one-hot vector representation of a word and as the output a series of one-hot vectors representing its neighbourhood. One-hot vectors are vocabulary length binary vectors with each dimension a separate word. As shown in Figure 2.3 the model predicts, given a word, its neighbourhood. The CBOW model is the opposite, given

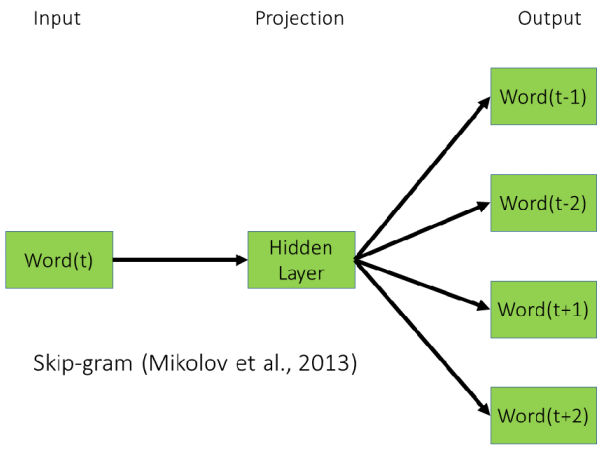

Figure 2.3: The skipgram model from Mikolov et al. (2013)

a vector representation of a neighbourhood, it attempts to predict the word.

Once trained, the hidden layer in this neural network captures word co-occurrence or the likelihood that the word it formed from will occur in a particular neighbourhood. This hidden layer is then used as the vector representation for this word instead of its one-hot vector fed to the model. This allows these vectors to take a real value as opposed to a binary one, and more importantly, words which are "similar" tend to cluster in this low dimensional space. This happens because if there is large enough data with billions of word, taken from real world sources like the internet, words which occur in each other's neighbourhood often will capture some kind of real world similarity. These word vectors can be used in many NLP tasks requiring neural networks which need some kind of vector representation of words. Where vector representation of collection of words, or phrases, are needed, something as simple as averaging the vectors for the individual words might work, although there are more sophisticated techniques.

2.3 Convolutional Neural Networks

Another neural network which will useful to know about to understand systems which use language and vision together, which I will describe in the next section, is the CNN (Krizhevsky et al., 2012). CNNs are used generally in vision architectures. CNNs are quite similar to ordinary neural networks used for classification tasks, i.e. each neuron is a matrix which gets an input as a vector, performs a dot product, and put a non-linear function over it. In a neural network a layer is made of a set of neurons and the network itself comprises of a set of layers where each neuron is connected to all the neurons in the previous layer but are independent to other neurons of the same layer. The last layer is the output layer. For training, the input layer is fed a vector, while the edge weights of all the connections are altered till the output layer matches the instance label. This training ends when there isn't a need to alter the weights any more. However, regular neural networks do not work well with images as the vector sizes are so large, that to have every layer fully connected to the previous ones means an intractable amount of parameters and a tendency for overfitting. This is called the curse of dimensionality.

A CNN is a variant of this which takes advantage of the localised nature of pixels in images, i.e. the fact that the visual field of one neural need not cover the entire image, to solve the dimensionality problem. Every neuron is influenced only by a limited region of the image and these regions overlap. This is accomplished via a special layer called the convolution layer in which a neuron is only connected to a small region of the layer before it instead of all the neurons. This is done by

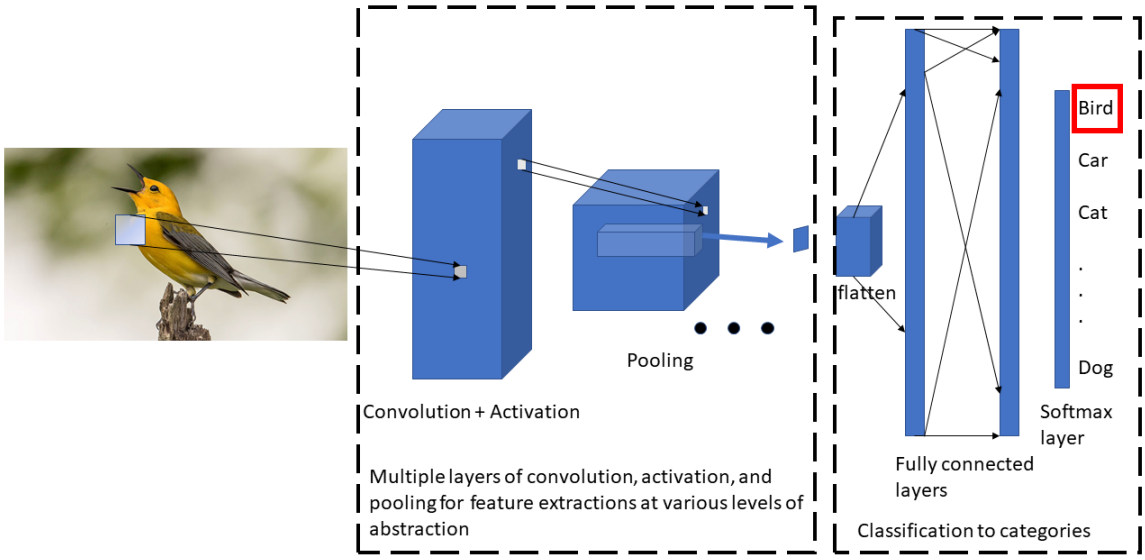

Figure 2.4: An image of a bird is filtered with multiple layers of convolution, activation, and pooling. The layers are represented as 3d instead of 2d because the depth represents the set of features each layer can learn. After all the abstractions have been learnt the layers are flattened and fed into fully connected layers like a standard neural network, hereupon they go through the softmax layer and yield decisions on the categories. The diagram is a simplified representation of common CNNs present in works like Krizhevsky et al. (2012)

convolving a small learnable filter as a sliding window across the entire previous layer computing dot products of a neuron with those of the previous layer under that filter. There are a number of these filters whose comprise the Feature Map. The more filters there are the more features are being extracted from the last layer, and thus the model is better. The number of filters in this feature map is called its depth, the size of the window is called its stride, and there is a third parameter which decides the padding of these filters. All the weights of all the filters in the feature map are learnt at training. Figure 2.4 demonstrates the basic architecture of the CNN.

In CNNs aside from these *convolutional layers*, there are a few fully connected layers, and there are other special purpose layers like layers with element-wise activation functions and layers for downsampling. Thus a CNN has four kinds of layers which work in conjunction. The CNN takes an image, each convolutional layer produces higher and higher abstractions of it, then the signals pass through a few fully connected layers, and the final layer gives the category, in case of a classification operation, or the results of the penultimate layer is used as the features for some other vision or multi-modal operation as described in the next section. Because of the convolutional layers the number of parameters are smaller and the CNNs become computationally tractable. Aside from being used to classify images, the penultimate layer of CNN architectures can be used as a feature vector for downstream tasks. These tasks include ones which use language and vision together.

2.4 Current Approaches in Vision with Text

The majority of past work in computer vision has been about labelling pixels with fixed visual categories. Thus, previous work has included segmentation, semantic segmentation, object detection, and scene understanding. With the advent of deep networks like CNN described in section 2.3, and GPU computational strength these tasks have been done with high degree of accuracy. Despite the rapid advance of these deep networks and independent excellence in each of the above tasks, they still lack an ability to have a robust understanding of semantic and other relationships of the artefacts in a visual scene to each other and to the scene, which

involves not just recognition and localisation of scene components like objects but also parsing their interrelations, as well as the non object parts of the scene.

This is a known problem, and one method of generating sophisticated representations of images is by combining the label complexity of generating sequences for an image (captioning) with the label density, or discovering multiple labels per image. Thus all the components in the scene can be described by complex text captions. This is known as the "dense captioning task" (Johnson et al., 2015) by some groups, though it can be argued that a complete semantic understanding is greater than learning dense enough captions. Generating dense captions of images has become extremely popular recently (Chen and Zitnick, 2014; Karpathy and Fei-Fei, 2014; Mao et al., 2014; Vinyals et al., 2014; Xu et al., 2015) after the advent of deep networks and resources such as the imagenet (Deng et al., 2009), MSCOCO (Chen et al., 2015), flickr dataset (Rashtchian et al., 2010; Young et al., 2014) etc. The way this is done is often like this, first recognition models trained on the imagenet give object recognitions from an image from a set of fixed object classes as described in the previous section. In certain models the object classes are not required but the matrix of the penultimate layer which serves as a feature vector for that particular image. In train time, an RNN, a model described in section 2.5.2 which essentially predicts sequences of tokens, is fed captions and the image feature vector till it learns what caption occurs with what kind of image. In test time this multi-modal RNN is fed only the image feature vector and it creates a new sequence off tokens, or a caption, for that image. While there are slight variances in these caption generation frameworks, a multi-modal RNN architecture, or a neural network analogous to

that, is the commonality between all these models. Since they use text as a purely generative task, and do not look at text to guide the output of the vision task, this way of generating captions does not actually use the full potential of human language to do vision tasks better.

There are very few models which do reference between spans of text and visual entities (Kong et al., 2014) or anything which attempts to utilise complex natural descriptions and semantics to aid vision tasks. Text has been used to weakly supervise vision tasks like learning correspondence between words and video clips (Yu and Siskind, 2013), words and action models (Ramanathan et al., 2013), or jointly learning language and perception models (Matuszek et al., 2012). Very few works try to relate semantics with vision retrieval (Lin et al., 2014) and even less has been done to use semantic understanding of text to improve anything more complex than tag generation or object classification, like jointly doing semantic segmentation, object detection, and scene understanding etc (Yao et al., 2012). This work attempts to address this by doing retrieval a hard vision task with word embeddings.

2.5 Sequence Labelling

One other concept which this thesis uses often, both as a component of joint language vision systems as in the last section, and also as in systems which only use text, is sequence labelling. In machine learning sequence labelling is one of the many pattern recognition tasks that is used to find out given a sequence of inputs what label should be assigned to which input. Sequence labelling techniques are

used in many NLP problems, and in the context of this thesis this is used in the mention detection task in the next chapter. Sequence labelling is done via many techniques like linear CRFs and RNNs and their analogues.

2.5.1 Conditional Random Fields

In machine learning, generative models are models that generate the observed data from random using a statistical model, also they model the joint probability between inputs and outputs. Discriminative models on the other hand model the posterior probability, that is given such and such input what is the expected output.

The Conditional Random Field (CRF) (Lafferty et al., 2001) is a discriminative model. It is a variant of a Markov network and is a conditional probability distribution on an undirected graph. While a classifier can predict one class variable, a probabilistic graphical model (PGM) can model many variables which are interdependent. The CRF specifically, is a *conditional* variant of undirected graphical models (Markov Random Fields), which are *generative*, i.e. the CRF graphs make the independence assumption for the output variables, but not the input. The reason a discriminative model has an advantage over a generative one for certain tasks is because it is hard to represent multiple interacting features or dependencies of the observations if they are long range.

For our purposes, I describe the linear version of the the CRF. This is just like an HMM (Eddy, 1996) aside from the fact that it is discriminative and supports arbitrary features. In an HMM the observation at a time depends on only that par-

ticular state, this is called the Markov property (also used in subsection 2.6.1). This is a non reflective of real world sequences where an observation can be dependent on features from previous states. Unlike the HMM a linear CRF has access to the entire observation sequence till the given time, so unlike the HMM it can model overlapping dependent features while modelling a sequence.

Thus, given a sequence of textual words it can infer what their transition probabilities are, and what their label probabilities are, which is what is needed for the task of finding labels of a sequence of word. Of course instead of only one feature, the token itself, other sequential features like the POS tag of the token, or its dependency relation can also be used. The only drawback of this method is, as the words themselves are purely symbols with no meaning in themselves, the graph cannot model things like inter-word similarity while modelling the sequence. For that a neural network based method is needed where the words are not tokens but vectors. One such model is the RNN.

2.5.2 Recurrent Neural Networks

The RNN is a neural network, that analogous to a linear CRF, considers the order of the input signals and is thus suitable for sequence labelling. In an RNN a sequence of text, represented as vectors, yields a vector as the hidden state representing the sequence, and a softmax layer over this hidden state predicts the output layer. Every hidden state for every token depends on the vector representation of the current input token and the vector which is previous hidden state, thus the

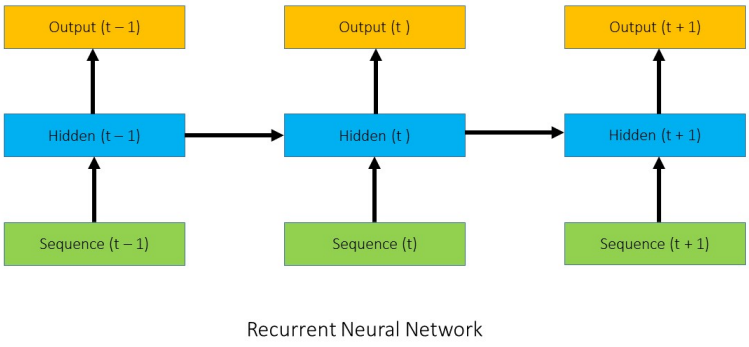

Figure 2.5: Sequence labelling via recurrent neural network

RNN reads a sentence left to right, as shown in the Figure 2.5. The hidden state is computed as a non-linear function (like *tanh*) over the vector composed of the hidden state of the previous token and the vector representation of the current input. Then, the entire model is trained.

Once trained the model can eat any sequence of tokens and find the probability of what token comes after and also the label at each. However, basic RNNs suffer from the vanishing gradient problem, i.e. due to the sequence of layers at every token, the signal starts becoming weak. To overcome this problem which occurs with long text sequences, variants of RNN, like the Long Short Term Memory (LSTM), are used which have mechanisms to prevent backpropagated errors from vanishing, even over long sequences.

Aside from being used in joint vision-language models as described in the last section, RNNs can be used to convert sequences to vectors. This will be used in Chapter 5 in a neural network model which finds references to concepts from documents. An analogous task is done by a generative model in the next section.

2.6 Topic Models

Topic models are a class of statistical methods in machine learning which assume that there are a bunch of hidden or latent topics in a set of documents, and all words in them belong with a certain probability to each of these topics. This is useful as using these models one can discover these unnamed topics which can be used in various downstream applications. In the context of this thesis topics are analogous to abstractions which we want to discover.

In topic models, each topic can be described by a list of probabilities over the entire vocabulary of the document set, and each word in the documents can be given one topic of which it has the maximum probability. Having discovered these hidden topics, each document thus can also be described as a combination of topic probabilities. For example in the Figure 2.6 a topic model using a commonly used algorithm Latent Dirichlet Allocation or LDA (Blei et al., 2003) is run on the text of the AP corpus, it discovers a a few topics and assigns each word with one. Topic models can be used to discover concepts distributed over text.

The way topic models work, they consider each document a "bag of words", i.e. the order of words in them is not important, they explain these collections of words by discovering the unobserved groupings that may generate these collections. In NLP this is an example of a generative model, i.e. a statistical model which randomly generates observed data values given some latent (hidden) variables. This is useful for this thesis because it is not only interested in discovering spans of text referring to real world entities, it is also interested in expanding the definition of

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Figure 2.6: An article from the Associated Press corpus run with a topic model using the LDA algorithm (Credit: David M. Blei) demonstrating some topics discovered and how tokens in the article are assigned those topics.

reference to include what "concept" the spans might be referring to. The concepts are analogous to topics. However, with the kind of text this work deals with, and the kind of concepts spans in such text refer to, the order, or sequence of words is important because we might want to make assumptions about how those concepts influence the existence of succeeding concepts. There are variants of topic models which account for this. One variant uses the Markov property, also used by HMMs described in 2.5.1.

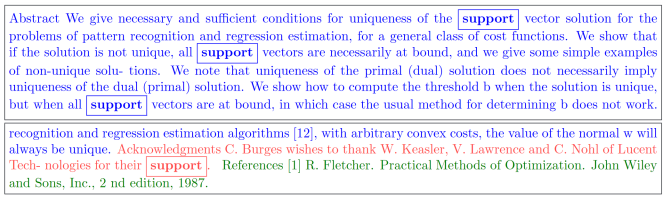

Figure 2.7: Two portions of text from the beginning and end of a NIPS paper being assigned topics via an HTMM topic model. Because the model assumes that topics of contiguous text must be similar, it is a good model to perform text segmentation. In this example, all of the front part and some of the last part is assigned a topic which has mathematical terms, whereas the part of the text with acknowledgements and the one with references are assigned two different topics (Credit: Amit Gruber)

2.6.1 Markov Topic Models

The assumption of topic models that the order of the words is irrelevant causes problems for the task of reference detection of concepts because the presence of a concept in a sentence in some structured text is not independent of what text came before or after, and also the model needs to consider the sequential property of these sentences. For example, in narrative text, a "sad" sentence must have had text before it which caused its words to have that theme, and the sentences following will have themes whose existence is influenced by this reference. One class of topic models which can address this are Markov topic models, like the HTMM (Gruber et al., 2007) which assume that the topic assignments form a Markov chain, and thus the existence of one topic or concept is influenced by existence of other topics before it.

As shown in the Figure 2.7 if the assumption is that sentences refer to the same topics and adjacent sentences are likely to have same topics, then this model is useful for neatly breaking the text into sequential topics.

2.7 Summary

In this chapter I have discussed the various background machine learning concepts needed to understand the following chapters of this thesis. This included a description of current work in coreference, the methods used therein for the various stages of the coreference pipeline, vector representations of words and how they are generated using neural networks, deep learning vision architectures and how they intersect with models which work on text, statistical methods for sequence labelling, using both neural networks and probabilistic graphical models, and topic models and their temporal variants. The next chapter describes my work in discovering hard coreference chains requiring world knowledge from a new coreference data source.

Chapter 3: Discovering Referring Entities and Related Problems

In **Chapter 2** I describe the current body of work in both mention detection and resolution. In this chapter, I first talk about the issues with the existing sources of data used sources for coreference resolution.[1] Then I build a better dataset which captures the kind of problems this work wishes to investigate. After that I present our own methods for end-to-end resolution which uses distributional semantics. Finally, I present experiments for using a similar method to investigate the harder phenomena of bridging references.

3.1 Quiz Bowl Data

As mentioned briefly in the introduction, many current limitations of coreference resolution research in text, namely, restricting the problem to easy entity centric coreference, is due to the both the lack of diverse datasets and the absence of datasets which have hard coreference problems. There are three major English

[1] Aside from the section at the end about Bridging Anaphora which is work in preparation, most of this chapter comes from the published work called "Removing the Training Wheels: A Coreference Dataset that Entertains Humans and Challenges Computers" published in NAACL, 2015. The authors of this work are Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. In this work my contribution is primary, namely the annotation of the dataset, the active learning method for selecting instances to annotate, experiments with both the mention detection via a sequence labelling method and clustering, and using similarity as a means of incorporating world knowledge

datasets for this task, namely MUC (MUC-6, 1995; MUC-7, 1997), ACE (Doddington et al., 2004), and Ontonotes (Pradhan et al., 2011), and their data sources and annotation conventions have had consequences in shaping the problem.

3.1.1 Issues with Current Datasets and Approaches

Newswire text, defined as text obtained from journalistic and allied sources, like broadcast conversations, is widely used as training data for coreference resolution systems. The standard English datasets used in the MUC (MUC-6, 1995; MUC-7, 1997) and CoNLL shared tasks (Pradhan et al., 2011) contain such text while the ACE (Doddington et al., 2004) dataset, though it has non newswire text as well, focuses on specific entity types instead of general coreference. Even in the new Ontonotes dataset version, which has a pivot corpus from non newswire sources, and more telephonic conversations, there is a severe lack of *hard coreference* examples, i.e. coreferences which need more than lexical and semantic features, like shared world knowledge which appears in all kinds of human discourse.

It must be mentioned here that some large annotated coreference datasets in other languages are not newswire centric. `ANCOR_centre` (Muzerelle et al., 2014), the French dataset is based on conversational text while the Polish PCC (Ogrodniczuk et al., 2013) dataset has 14 different text genres. However the Japanese Kyoto corpus (Kawahara et al., 2002), the Spanish and Catalan ANCORA (Taulé et al., 2008), the Japanese NAIST dataset (Iida et al., 2007), the Czech PDT (Hajič et al., 2017), the German TüBa-D/Z (Telljohann et al., 2004), and the German

DIRNDL (Björkelund et al., 2014) are all all based on news sources indicating a monoculture of newswire in NLP research across languages.

Current systems are not able to solve hard coreference in the manner humans can because they are evaluated on data which isn't representative of the range in human ability to solve hard coreference resolution. This skews the research towards building models with lexical and at best semantic features, ignoring the need for knowledge.

Newswire like text is sparse in anaphoric mentions, and those that it has are mainly identity coreferences and appositives. In the CoNLL 2011 shared task (Pradhan et al., 2007) based on OntoNotes 4.0 (Hovy et al., 2006),[2] there are 2.1 mentions per sentence; in the next section I present a dataset with 3.7 mentions per sentence.[3] In newswire text, most nominal entities (not including pronouns) are singletons; in other words, they do not corefer to anything. OntoNotes 4.0 development data contains 25.4K singleton nominal entities (Durrett and Klein, 2013), compared to only 7.6K entities which corefer to something (anaphora). On the other hand, most pronominals are anaphoric, which makes them easy to resolve as pronouns are single token entities. While it is easy to obtain a lot of newswire data, the amount of coreferent-heavy mention clusters in such text is not correspondingly high, and text where coreference is among text spans which are not pronouns is even lower. In human coreference by contrast an entity can be referred to in various

[2]As our representative for "newswire" data, the English portion of the Ontonotes 4.0 contains professionally-delivered weblogs and newsgroups (15%), newswire (46%), broadcast news (15%), and broadcast conversation (15%).

[3]Neither of these figures include singleton mentions, as OntoNotes does not have gold tagged singletons. Our dataset has an even higher density when singletons are included.

NW	Later, [they]$_1$ all met with [President Jacques Chirac]$_2$. [Mr. Chirac]$_2$ said an important first step had been taken to calm tensions.
NW	Around the time of the [Macau]$_1$ handover, questions that were hot in [the Western media]$_2$ were "what is Macaense"? And what is native [Macau]$_1$ culture?
NW	[MCA]$_1$ said that [it]$_1$ expects [the proposed transaction]$_2$ to be completed no later than November 10th.
QB	As a child, [this character]$_1$ reads [[his]$_1$ uncle]$_2$ [the column]$_3$ [*That Body of Yours*]$_3$ every Sunday.
QB	At one point, [these characters]$_1$ climb into barrels aboard a ship bound for England. Later, [one of [these characters]$_1$]$_2$ stabs [the Player]$_3$ with a fake knife.
QB	[One poet from [this country]$_2$]$_1$ invented the haiku, while [another]$_3$ wrote the [*Tale of Genji*]$_4$. Identify [this homeland]$_2$ of [Basho]$_1$ and [Lady Murasaki]$_3$.

Table 3.1: Three newswire sentences and three quiz bowl sentences with annotated coreferences and singleton mentions. These examples show that quiz bowl sentences contain more complicated types of coreferences that may even require world knowledge to resolve.

ways, not necessarily using pronouns.

Current work in coreference resolution have downplayed the need world knowledge, due to not as much improvement on attempting to use semantic and world knowledge features (Durrett and Klein, 2013) compared to improvements obtained from better syntactic features. Before that, works like Daumé III and Marcu (2005) tried word clustering for world knowledge which led to methods like Web n-gram features (Bansal and Klein, 2012). In this work I show how using simple distributional semantics (word embeddings which reflect world knowledge by leveraging neighbourhood of words in large corpora) massively increases performance, in part because because of a change in datasets. Systems trained on news media data for a related problem—entity extraction—falter on non-journalistic texts (Poibeau and Kosseim, 2001). This discrepancy in performance can be attributed to the stylistic

conventions of journalism also known as Style Guides. Journalists are instructed in these style guides, for example, to limit the number of entities mentioned in a sentence to aid comprehension of their audience, and there are strict rules for referring to individuals (Boyd et al., 2008). Furthermore, writers cannot assume that their readers are familiar with all participants in the story, which requires that each entity is explicitly introduced in the text (Goldstein and Press, 2004). These constraints make for easy reading by design and, as a side effect, easy coreference resolution. Unlike this simplified "journalistic" coreference, everyday coreference relies heavily on inferring the identities of people and entities in language, which requires substantial world knowledge.

3.1.2 Quiz Bowl data and Annotation

One example of such data comes from a game called *quiz bowl*. Quiz bowl is a trivia game where questions are structured as a series of sentences, all of which indirectly refer to the answer. Each question has multiple clusters of mutually-coreferent terms, and one of those clusters is coreferent with the answer. Figure 3.1 shows an example of a quiz bowl question where all answer coreferences are marked.

A player's job is to determine the entity referenced by the question.[4] Each sentence contains progressively more informative references and more well-known clues. For example, a question on Sherlock Holmes might refer to him as "he", "this character", "this housemate of Dr. Watson", and finally "this detective and

[4]In actual competition, it is a race to see which team can identify the coreference faster, but we ignore that aspect here.

[The Canadian rock band by [this name]] has released such albums as Take A Deep Breath, Young Wild and Free, and Love Machine and had a 1986 Top Ten single with Can't Wait For the Night. [The song by [this name]] is [the first track on Queen's Sheer Heart Attack]. [The novel by [this name]] concerns Fred Hale, who returns to town to hand out cards for a newspaper competition and is murdered by the teenage gang member Pinkie Brown, who abuses [the title substance]. [The novel] was adapted into [a 1947 film starring Richard Attenborough]; [this] was released in the US as Young Scarface. FTP, identify [the shared name of, most notably, [a novel by Graham Greene]].

Figure 3.1: An example quiz bowl question about the novel *Brighton Rock*. Every mention referring to the answer of the question has been marked; note the variety of mentions that refer to the same entity and the variety of ways in which an entity can be referred to.

resident of 221B Baker Street". While quiz bowl has been viewed as a classification task (Iyyer et al., 2014), previous work has ignored the fundamental task of coreference. Nevertheless, quiz bowl data are dense and diverse in coreference examples. For example, references within references, called nested mentions, which are difficult for both humans and machines, are very rare in the newswire text of OntoNotes—0.25 mentions per sentence—while quiz bowl contains 1.16 mentions per sentence (Figure 3.2). Examples of nested mentions can be seen in in Table 3.1. Since quiz bowl is a game, it makes the task of solving coreference interesting and *challenging* for an annotator. I create a new dataset based from this domain thus altering the nature of the coreference resolution task.

Each document is a single quiz bowl question containing an average of 5.2 sentences. While quiz bowl covers all areas of academic knowledge, this work focuses on questions about literature from (Boyd-Graber et al., 2012), as annotation standards are more straightforward.

This webapp (Figure 3.3) allows users to annotate a question by highlighting a phrase using their mouse and then pressing a number corresponding to the coref-

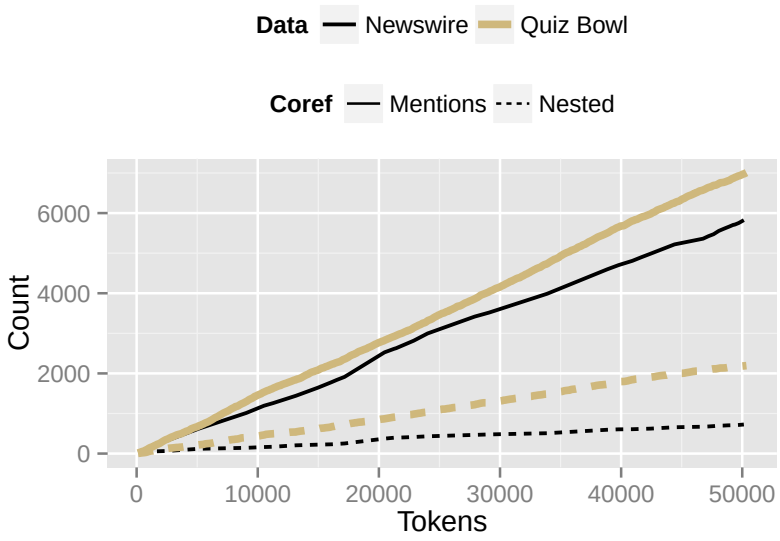

Figure 3.2: Density of quiz bowl vs. CoNLL coreference both for raw and nested mentions.

erence group to which it belongs. Each group is highlighted with a single color in the interface distinguishing the coreference chains. The webapp displays a single question at a time, and for some questions, users can compare their answers against gold annotations by the authors. The annotators are provided with the ability to see if their tags match the gold labels for a few documents as a mechanism was needed to help them learn the annotation guidelines as the annotators are crowdsourced volunteers. This improves inter-annotator agreement.

The webapp was advertised to quiz bowl players before a national tournament and attracted passionate, competent annotators preparing for the tournament. A leaderboard based on mentions tagged was implemented to encourage competitiveness, and prizes were given to the top five annotators.

Users are instructed to annotate all authors, characters, works, and the answer

to the question (even if the answer is not one of the previously specified types of entities). This work considers a coreference to be the maximal span that can be replaced by a pronoun.[5]

As an example, in the phrase *this folk sermon by James Weldon Johnson*, the entire phrase is marked, not just *sermon* or *this folk sermon*. Users are asked to consider appositives as separate coreferences to the same entity. Thus, *The Japanese poet Basho* has two phrases to be marked, *The Japanese poet* and *Basho*, which both refer to the same group.[6] Users annotated prepositional phrases attached to a noun to capture entire noun phrases.

[5]The instruction was phrased in this way to allow the educated but linguistically unsavvy annotators to approximate a noun phrase.

[6]The datasets, full annotation guide, and code can be found at http://www.cs.umd.edu/~aguha/qbcoreference.

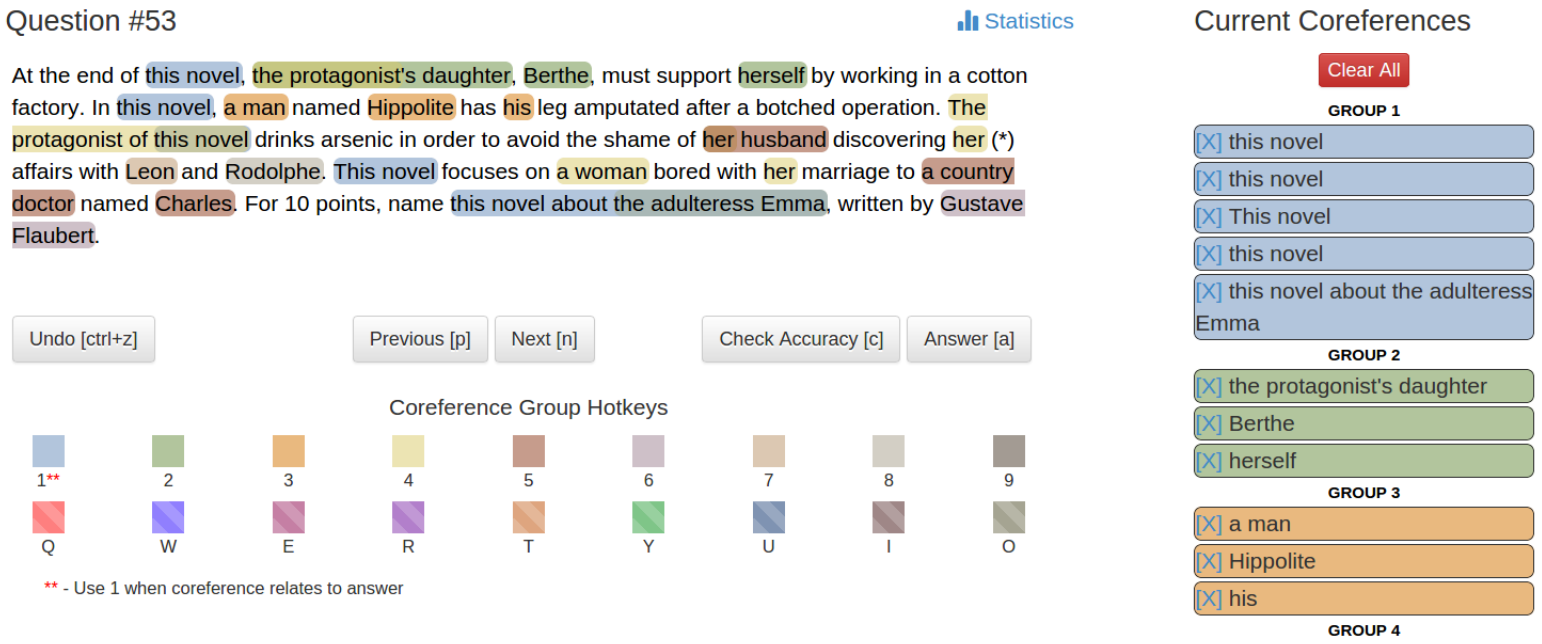

Figure 3.3: The webapp to collect annotations. The user highlights a phrase and then assigns it to a group (by number). Showing a summary list of coreferences on the right significantly speeds up user annotations.

Titular mentions are mentions that refer to entities with similar names or the same name as a title, e.g., "The titular doctor" refers to the person "Dr. Zhivago" while talking about the book with the same name. For our purposes, all titular mentions refer to the same coreference group. While the annotators do mark all anaphora, cataphora, and coreferring noun phrases, they are not instructed to mark split antecedents as it was considered too complex for the scope of this work, though split antecedents are present in such data. Split antecedents are non contiguous mentions that referred to by a single anaphor; for example, in the sentence *Romeo met Juliet at a fancy ball, and they get married the next day*, the word *they* refers to both *Romeo* and *Juliet*, and these are the antecedents which are not-contiguous, therefore split. Currently, this webapp cannot tag these cases. The annotation does not distinguish between coreference and bound variables. A bound variable is an antecedent, which unlike an anaphor is indefinite. For example in the phrase "every office got its desk", "its" doesn't refer to a particular office and is thus a bound variable and not an anaphor. Bound variables are practically not present in quiz bowl due to the nature of the text so this wasn't a problem.

To illustrate how popular the webapp proved to be among the quiz bowl community, 615 documents were tagged by seventy-six users within a month. The top five annotators, who between them tagged 342 documents out of 651, have an inter-annotator agreement rate of 87% with a set of twenty author-annotated questions used to measure tagging accuracy. Agreement here looks at complete mention span matches instead of only head matches, but does not do chance correction as this is not a classification task.

Number of ...	Quiz bowl	OntoNotes
documents[7]	400	1,667
sentences	1,890	44,687
tokens	50,347	955,317
mentions	9,471	94,155
singletons[8]	2,461	0
anaphora	7,010	94,155
nested mention	2,194	11,454

Table 3.2: Statistics of both our quiz bowl dataset and the OntoNotes training data from the CoNLL 2011 shared task.

This work only considers documents that have either been tagged by four or more users with a predetermined degree of similarity and verified by one or more author (150 documents), or documents tagged by the authors in committee (250 documents). Thus, the gold dataset has 400 documents.

Both the quiz bowl dataset and the OntoNotes dataset are summarized in Table 3.2. If coreference resolution is done by pairwise classification, this dataset has a total of 116,125 possible mention pairs. On average it takes about fifteen minutes to tag a document because often the annotator will not know which mentions co-refer to what group without using external knowledge, solving the coreference chains on this data is hard for humans as well. OntoNotes is 18.97 times larger than our dataset in terms of tokens but only 13.4 times larger in terms of mentions.[9] Next, I describe a technique that allows this webapp to choose which documents to display for annotation.

[7] This number is for the OntoNotes training split only.

[8] OntoNotes is not annotated for singletons.

[9] These numbers do not include singletons as OntoNotes does not have them tagged, while ours does.

3.1.2.1 Active Learning

Active learning is a technique that alternates between training and annotation by selecting instances or documents that are maximally useful for a classifier to learn (Settles, 2010). Because of the large sample space and amount of diversity present in the data, active learning helps us build our coreference dataset. To be more concrete, the original corpus contains over 7,000 literature questions, and it would be prudent to tag only the useful ones. Since it can take a quarter hour to tag a single document and the authors wanted at least four annotators to agree on every document included in the final dataset, annotating all 7,000 questions is infeasible.

I follow the work of Miller et al. (2012), who use active learning for document-level coreference rather than at the mention level. Starting from a seed set of a hundred documents and an evaluation set of fifty documents I sample 250 more documents from our set of 7,000 quiz bowl questions.[10] I use the Berkeley coreference system (described in the next section) for the training phase. In Figure 3.4 the effectiveness of our iteration procedure can be seen. Unlike the result shown by Miller et al. (2012), it is observed that for this dataset voting sampling beats random sampling, which supports the findings of Laws et al. (2012).

Voting sampling works by dividing the seed set into multiple parts and using each to train a model. Then, from the rest of the dataset this method selects the document that has the most variance in results after predicting using all of the models. Once that document gets tagged, it is added to the seed set, retrain, and

[10]These were documents tagged by the quiz bowl community, so I didn't have to make them wait for the active learning process to retrain candidate models.

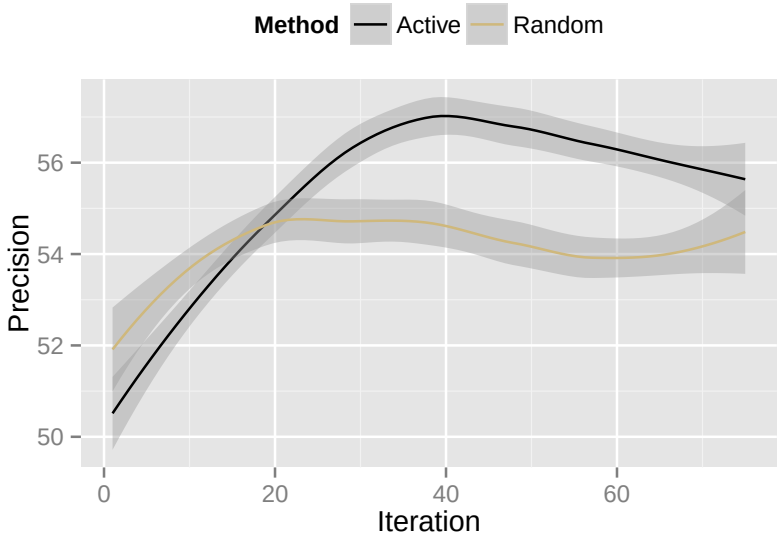

Figure 3.4: Voting sampling active learning works better than randomly sampling for annotation.

repeat the procedure. This process is impractical with instance-level active learning methods, as there are 116,125 mention pairs (instances) for just 400 documents. Even with document-level sampling, the procedure of training on all documents in the seed set and then testing every document in the sample space is a slow task. Batch learning can speed up this process at the cost of increased document redundancy; I choose not to use it because I want a diverse collection of annotated documents.

Active learning's advantage is that new documents are more likely to contain diverse (and thus interesting) combinations of entities and references, which annotators noticed during the annotation process. Documents selected by the active learning process were dissimilar to previously-selected questions in both content and structure.

3.1.3 A Simple Mention Detection and Clustering Model

First I run the Berkeley coreference system (Durrett and Klein, 2013) on our dataset to show that models trained on newswire data cannot effectively resolve coreference in quiz bowl data, and thus the domains are sufficiently different. Training and evaluating the Berkeley system on quiz bowl data also results in poor performance.[11] This result motivates us to build a simple end-to-end coreference resolution system that includes a data-driven mention detector (as opposed to Berkeley's rule-based one) and a simple pairwise classifier. Using our mentions and only six feature types, this method is able to outperform the Berkeley system on this data. Finally, I explore the linguistic phenomena that make quiz bowl coreference so hard and draw insights from our analysis that may help to guide the next generation of coreference systems.

3.1.3.1 Evaluating the Berkeley System on Quiz Bowl Data

I use two publicly-available pretrained models supplied with the Berkeley coreference system, *Surface* and *Final*, which are trained on the entire OntoNotes 4.0 dataset. The difference between the two models is that *Final* includes semantic features. This work reports results with both models to see if the extra semantic features in *Final* are expressive enough to capture quiz bowl's inherently difficult coreferences. I also train the Berkeley system on quiz bowl data and compare the performance of these models to the pretrained newswire ones in Table 3.3. Our re-

[11] This work uses default options, including hyperparameters tuned on OntoNotes 4.0

sults are obtained by running a five-fold cross-validation on our dataset. The results show that newswire is a poor source of data for learning how to resolve quiz bowl coreferences and prompted us to see how well a pairwise classifier does in comparison. To build an end-to-end coreference system using this classifier, the method needs to know which parts of the text are "mentions", or spans of a text that refer to real world entities. In the next section I talk about the mention detection system.

3.1.3.2 A Simple Mention Detector

As mentioned before the Berkeley system does rule-based mention detection to detect every NP span, every pronoun, and every named entity, which leads to many spurious mentions. This process is based on an earlier work of Kummerfeld et al. (2011), which assumes that every maximal projection of a noun or a pronoun is a mention and uses rules to weed out spurious mentions. Instead of using such a rule-based mention detector, our system follows the lead of much earlier work like Florian et al. (2006) and detects mentions via sequence labelling, as detecting mentions is essentially a problem of detecting start and stop points in spans of text. This also works for nested mentions, as instead of just detecting start and stop points, it ends up detecting combinations of states and stop, treating each combination as a sequence label. This work solves this sequence tagging problem using the MALLET (McCallum, 2002) implementation of conditional random fields (Lafferty et al., 2001) which is described in Chapter 2 Section 2.5.1. The labelling technique I use with the sequence labels is analogous to one used for BIO markers (Ratinov

Identify	VB	*	root-ROOT	lvl0
this	DT	*	det-poem	lvl1-Start
poem	NN	*	dobj-identify	lvl1
in	IN	*	prep-imagines	lvl1
which	WDT	*	pobj-in	lvl1
Yevgeniy	NNP	**PERSON**	nsubj-imagines	lvl2-Singleton
imagines	VBZ	*	rcmod-poem	lvl1
the	DT	*	det-statue	lvl2-Start
title	NN	*	nn-statue	lvl2
equestrian	NN	*	nn-statue	lvl2
statue	NN	*	dobj-imagines	lvl2

Figure 3.5: Features and BIO labels for our mention detector

and Roth, 2009). The features it uses, as shown in Figure 3.5, which are similar to those used in Kummerfeld et al. (2011), are:

- the token itself
- the part of speech
- the named entity type
- a dependency relation concatenated with the parent token[12]

Using these simple features, this work obtains surprisingly good results. When comparing the detected mentions to gold standard mentions on the quiz bowl dataset using exact matches, the results are 76.1% precision, 69.6% recall, and 72.7% F_1 measure. Now that high-quality mentions have been detected, each pair of mentions is fed into a pairwise mention classifier.

[12]These features were obtained using the Stanford dependency parser (De Marneffe et al., 2006).

		MUC		
System	Train	P	R	F_1
Surface	OntoN	47.22	27.97	35.13
Final	OntoN	50.79	30.77	38.32
Surface	QB	**60.44**	31.31	41.2
Final	QB	60.21	**33.41**	**42.35**

Table 3.3: The top half of the table represents Berkeley models trained on OntoNotes 4.0 data, while the bottom half shows models trained on quiz bowl data. The MUC F_1-score of the Berkeley system on OntoNotes text is 66.4, which when compared to these results prove that quiz bowl coreference is significantly different than OntoNotes coreference.

However, there is a massive scope of improvement here. Our mention detector is missing out of almost 30% mentions which are anaphoric, and the only reason it ties with the rule based system is because the rule based system doesn't attempt to do better on recall. Since our system cannot find such a large number of mentions which have coreferents the downstream classifier can not evaluate on those, and thus potentially could have done better. There is also potential work in this sub-problem to 'hallucinate" mentions in the Ontonotes dataset which have not been annotated, i.e. the singleton mentions using something like scheduled sampling. A method like that might make it possible to have enough singleton and anaphoric mentions (though noisy) from a large dataset to design a more robust mention detector.

3.1.3.3 A Simple Coref Classifier

I follow previous pairwise coreference systems (Ng and Cardie, 2002; Uryupina, 2006; Versley et al., 2008) in extracting a set of lexical, syntactic, and semantic

features from two mentions to determine whether they are coreferent. For example, if *Sylvia Plath*, *he*, and *she* are all of the mentions that occur in a document, our classifier gives predictions for the pairs *he*—*Sylvia Plath*, *she*—*Sylvia Plath*, and *he*—*she*.

Given two mentions in a document, m_1 and m_2, this method generates the following features and feed them to a logistic regression classifier:

- binary indicators for all tokens contained in m_1 and m_2 concatenated with their parts-of-speech
- same as above except for an n-word window before and after m_1 and m_2
- how many tokens separate m_1 and m_2
- how many sentences separate m_1 and m_2
- the cosine similarity of `word2vec` (Mikolov et al., 2013) vector representations of m_1 and m_2; these vectors are obtained by averaging the word embeddings for all words in each mention. This work uses publicly-available 300-dimensional embeddings that have been pretrained on 100B tokens from Google News.
- same as above except with publicly-available 300-dimensional `GloVe` (Pennington et al., 2014) vector embeddings trained on 840B tokens from the Common Crawl

The first four features are standard in coreference literature, while the word embedding similarity scores increase our F-measure by about 5 points on the quiz

bowl data. Since they have been trained on huge corpora with billions of words, the word embeddings allow us to infuse some sense of world knowledge into our model; for instance, the vector for *Russian* is more similar to *Dostoevsky* than *Hemingway*. Given enough text data to train these word embeddings on, they start reflecting some knowledge about the world.

		MUC			BCUB			CEAFE		
Model	Mentions	P	R	F1	P	R	F1	P	R	F1
QB Final	Berkeley	60.2	33.4	42.4	56.9	21.1	30.7	56.9	13.0	21.2
QB Final	CRF	56.9	29.6	38.9	50.6	18.7	27.2	51.7	12.7	20.4
LR	CRF	67.0	**72.4**	**67.8**	59.2	**78.6**	63.9	58.7	48.6	49.2
QB Final	Gold	**70.2**	40.2	49.6	**88.5**	64.7	**74.2**	56.5	**80.0**	65.7
LR	Gold	58.8	56.8	57.8	68.1	74.8	70.4	**73.3**	76.1	**74.2**

Table 3.4: Comparison of the LR pairwise classifier to the Berkeley *QB Final* system. The bolded values are the highest in every column. All models are trained and evaluated on quiz bowl data via five fold cross validation on F_1, precision, and recall. Berkeley/CRF/Gold refers to the mention detection used, LR refers to our logistic regression model and *QB Final* refers to the Berkeley model trained on quiz bowl data. Our model outperforms the Berkeley model on every metric when using our detected CRF mentions. When given gold mentions, LR outperforms Berkeley *QB Final* in five of nine metrics.

Table 3.4 shows that our logistic regression model (LR) outperforms the Berkeley system on numerous metrics when trained and evaluated on the quiz bowl dataset. Precision, recall, and F_1, metrics are used applied to MUC, BCUB, and CEAFE measures used for comparing coreference systems.[13] The results show that this LR model outperforms Berkeley by a wide margin when both are trained on the

[13]The MUC (Vilain et al., 1995) score is the minimum number of links between mentions to be inserted or deleted when mapping the output to a gold standard key set. BCUB (Bagga and Baldwin, 1998) computes the precision and recall for all mentions separately and then combines them to get the final precision and recall of the output. CEAFE (Luo, 2005) is an improvement on BCUB and does not use entities multiple times to compute scores.

mentions found by our mention detector (CRF). For four metrics, the CRF mentions actually improve over training on the gold mentions.

Why does the LR model outperform Berkeley when both are trained on our quiz bowl dataset? I hypothesize that some of Berkeley's features, while helpful for sparse OntoNotes coreferences, do not offer the same utility in the denser quiz bowl domain. Compared to newswire text, our dataset contains a much larger percentage of complex coreference types that require world knowledge to resolve. Since the Berkeley system lacks semantic features let alone any feature to reflect world knowledge, it is unlikely to correctly resolve these instances, whereas the pretrained word embedding features give our LR model a better chance of handling them correctly. Another difference between the two models is that the Berkeley system ranks mentions as opposed to doing pairwise classification like our LR model, and the mention ranking features may be optimized for newswire text.

3.1.4 Why Quiz Bowl Coreference is Challenging and Interesting

While models trained on newswire falter on these data, is this simply a domain adaptation issue or something deeper? Let us look at some examples that the *Final* model of Berkeley coref gets wrong.

> *This writer* depicted a group of samurai's battle against an imperial.
> For ten points, name *this Japanese writer of A Personal Matter and The Silent Cry.*

While the model identifies most of pronouns associated with Kenzaburo Oe

(the answer), it cannot recognize that the theme of the entire paragraph is building to the final reference, "this Japanese writer", despite the many Japanese-related ideas in the text of the question (e.g., Samurai and emperor). The model simply has no mechanism to learn the association between these phrases, as none of its features captures "Japanese-ness". Unless the dataset has these kinds of problems, it will not be revealed that models need mechanisms to learn these kinds of features.

Final also cannot reason effectively about coreferences that are tied together by similar modifiers as in the below example:

> That *title character* plots to secure a "beautiful death" for Lovberg by burning his manuscript and giving him a pistol. For 10 points, name this play in which *the titular wife of George Tesman* commits suicide.

While a reader can connect "titular" and "title" to the same character, Hedda Gabler, the Berkeley system fails to make this inference cause of its reliance on hard lexical features.

These data are a challenge for all systems, as they require extensive world knowledge. For example, in the following sentence, a model must know that the story referenced in the first sentence is about a dragon and that dragons can fly.

> The protagonist of *one of this man's works* erects a sign claiming that that story's title figure will fly to heaven from a pond. Identify this author of *Dragon: the Old Potter's Tale*

While word embeddings do capture this kind of knowledge, the method relies on something primitive like co-occurrence, and it can be observed why there is a

need for better machine learning methods to learn such specialised knowledge and representations for using such knowledge. These methods can work alongside non ML solutions like knowledge sources.

Humans solve cases like these using a vast amount of external knowledge, but existing models lack information about worlds (both real and imaginary) and thus cannot confidently mark these coreferences. In the next section I discuss similarly hard text reference problems, this time dealing with associations instead of identities.

3.2 Investigating Bridging Anaphora

A discussion on complex reference resolution in text is incomplete without describing the problem of bridging resolution and the problems encountered in solving it. As mentioned in the previous chapter, resolution in text could be about identity, which is coreference resolution, but it can also be about association, i.e. semantic relations other than "is-a" relation. These are bridging relations. This concept was first stated by Clark (1975) whose work made a distinction between definite descriptions referring to previously mentioned entities. It also distinguished these relations as direct or indirect. These definitions are now considered inadequate as the things being referred to need not be an entity in the first place.

For the purposes of this discussion, I take the definition from the work of Hou et al. (2013b) that any text reference relation that is not coreference or comparison is bridging. This could cover a large range. For example, an anaphor in a bridging relationship could be non definite:

I am going to cook lunch. I will fry some vegetables

In this text lunch is in a bridging relationship with some vegetables but not a coreference. In addition "some vegetables" is not a definite NP. The problem is made further complex by the fact that the antecedent need not be an entity. For example:

I am going to cook. The pot is on the stove

In this text the pot is in a bridging relationship with the verb cook. For the purposes of this discussion antecedents are restricted to entities. The rest of the structure of this problem is the same as coreference, i.e. there are anaphora and antecedents. Unlike coreferece, cataphora are not encountered. This problem is much more difficult than coreference as there are no clear syntactic clues to demonstrate clearly the existence of these relations, unlike coreference where good results can be obtained from only syntactic surface features as demonstrated in the *Surface* variant of Durrett and Klein (2013).

Because of the nature of bridging, and unlike coreference, there are significant differences between the anaphora and the antecedents, which prompt us to address them separately. Thus, the procedure for discovering bridging relations is often:

- find all the spans of text which are mentions (this is the same as coreference resolution)
- find which mentions are anaphors, by doing a binary classification on each
- for every anaphor thus detected, find which of the antecedent candidates before

it has a bridging link to it

3.2.1 Current Datasets and Methods for Bridging Anaphora

There is a lack of datasets in the study of bridging. The first dataset with annotations for bridging is the Vieira/Poesio dataset (Poesio et al., 2002) which has thirty three articles from the Penn Treebank corpus (Marcus et al., 1993) annotated. However, future authors have disagreements with the annotation in this dataset as hard coreferences are also annotated as bridging. Another dataset is the GNOME corpus (Poesio, 2004) which has only five hundred sentences from two domains. However, the authors of this work limited the kinds of bridging relationships they wanted to annotate in order to preserve inter annotator agreement, and thus only three kind of bridging relationships were annotated, leading to only 153 instances in the dataset which is too small for most machine learning models. Then there is the Switchboard Corpus (Godfrey et al., 1992) a portion of which annotated for Information Status by Nissim et al. (2004), which had among its types four subtypes related to bridging. A major flaw of this dataset is that the antecedent information of the anaphors is not annotated, and thus it cannot be tested on. There are some non-English datasets like DIRNDL (Björkelund et al., 2014) with around three thousand sentences and the large Prague Dependency Treebank (Hajič et al., 2017), which has been annotated for bridging relations by Nedoluzhko et al. (2009) with over 8000 bridging annotations.

The most recent datasets are ISNotes (Markert et al., 2012) annotated in 2012

and SciCorp (Rösiger, 2016) in 2016. SciCorp is made from 14 full length science papers from two genres with 890 and 599 instances of bridging in them respectively. ISNotes is made from a subset of OntoNotes, namely 50 texts taken from the Wall Street Journal portion of this dataset. Thus, it has only a very small fraction of the size of OntoNotes. Even compared to the quiz bowl dataset built in the last section, itself small and from specialised data, ISNotes has about a fifth of the mentions. It has a total of 663 bridging anaphors. Thus, there is an absence of any dataset with enough bridging instances to run a robust machine learning method on it, let alone data hungry deep learning methods.

Due to there not being enough data to train sophisticated learning systems, until 2015 rule based systems like (Hou et al., 2014) performed better on the task. Also, for the same reason, one should not learn models with all possible anaphor-antecedent candidate pairs, as there will be so many negative examples that the model will always give a negative result. In the work of Hou et al. (2013b) it is demonstrated that one can limit the number of antecedents of a particular anaphor by taking a window of only two sentences prior to the one the anaphor is on. In the ISNotes dataset this window covers 76.9% of anaphors having an antecedent in this range.

3.2.1.1 Anaphora Recognition

As mentioned earlier, the first step in detecting bridging relations is to find the mentions analogously to the coreference task. Then I find which of these mentions

are anaphors. A lot of early work in bridging (Lassalle and Denis, 2011; Markert et al., 2003; Poesio and Vieira, 1998; Poesio et al., 2004) assumed the gold data for this step as well. Early works (Rahman and Ng, 2012) in this sub problem reported different range of results as the Switchboard corpus and the ISNotes corpus have different definitions of what information state type even counts as bridging.

Various classification methods are used, of note is the Cascading Collective Classification method of Hou et al. (2013a) which is motivated by the fact that bridging anaphors are so rare, even among the "mediated" information state category of the ISNotes dataset, mediated mentions being mentions which are neither new, nor old (old mentions are coreferent). Mediated mentions have, aside from bridging anaphors, comparative mentions, world knowledge mentions, aggregate mentions, and function mentions.

The classification method used by Hou et al. (2013a) has binary classifiers for each of the five categories using SVMs, which predict labels in a row. If a class is true, it is assigned, if not the next binary classifier is used. If all of them report false, then a multi class framework is used. Their model used a wide variety of features which incorporated lexical features from previous works (Nissim, 2006) like mention matches, length, determiners, grammatical role, type of the noun phrase etc. Because the model is cascading they could use features for the other four classes to root out the bridging relations, for example a frequent proper name could indicate world knowledge, while a dependency on a changing verb could indicate a function. The authors use features from (Markert et al., 2012) as their baseline and show that this cascading model is better than a simple classifier on their feature set, which by

itself also outperforms their baseline.

3.2.1.2 Bridging Resolution

The second task in the pipeline is for each anaphor to select from the candidate antecedents the ones having a bridging relationship. As this is a difficult task, various authors restrict the problem to make it tractable, but that leads to the problem of not being able to evaluate and compare past systems. For ISNotes dataset, the data driven system which has the fewest constraints and achieves state-of-the-art is Hou et al. (2013b) while the rule based variant is Hou et al. (2014). Even for these systems the metric chosen is lax compared to the ones used in coreference, the metric measures how many anaphors are correctly connected to any correct antecedent. Also, these systems rely on all gold annotation for mention spans and other information. The Hou et al. (2013b) system uses 18 features which aside from lexical and syntactic features uses WordNet to get part-of semantic relations and semantic class, preposition patterns, verb patterns, and a score for saliency. Because detecting a bridging resolution is so hard, these features cannot be learnt from raw text, at least with the amount of annotated data present, and thus hand crafting is required. The model they used is based on a generative model called the Markov Logic Network (Domingos and Lowd, 2009). In short, MLNs are a way to combine first order logic with Probabilistic Graphical Models. Using an MLN it is possible to model bridging at the global level over the text. Every feature is associated with a "well formed formula" and each of these formulas is associated

with a weight.[14] These weights are learnt from the training data. With the use of some hard constraints like each anaphor having only one antecedent and the antecedents for the anaphor are only taken from the antecedent pool for it, the MLN based model is able to learn weights between antecedents themselves, and anaphors and antecedents at the global level. The accuracy metric for this model is 41.32% while similar features used in pairwise models give results in the range of 29.11% to 33.94% demonstrating the need for joint inference in these problems.

3.2.2 Bridging Anaphora and Word Embeddings

As mentioned before, bridging anaphora needs a significantly greater amount of shared world knowledge to solve compared to identity coreference resolution, and hence the difficulty of current models to find good results. In such a scenario can word embeddings still be useful? Here I perform two simple experiments to determine if word embeddings, learnt from suitably large corpus, give results comparable to existing models, in the same manner they helped in the coreference resolution model. The experiments cover the two stages of the process, namely detecting which mentions are anaphors and subsequently what antecedents link to those anaphors. To evaluate the embeddings vs. external knowledge sources with handcrafted features, no other external resource or semantic relations are used which are used in the state of the art systems (Hou et al., 2013a,b). Both experiments are done on the ISNotes dataset to compare with the state of the art methods.

[14] In first-order logic a well formed formula is a finite sequence of logical symbols formed from the alphabet of a formal language

In the first experiment to determine which mentions are bridging mentions, I treat it like a multi class classification problem using a simple LR model, with eight different classes, new mentions, old mentions, and the six types of mentions neither new nor old as defined in the ISNotes dataset. For baseline, aside from the state-of-the-art Hou et al. (2013a) which uses cascading classification, I also compare it with the older work of Nissim (2006). I use 10 fold cross validation.

For this task each mention is assigned one out of eight class labels and has these features:

- The length of the mention in tokens
- The averaged word vector over all tokens in the mention
- The averaged word vector over all tokens in the document
- binary indicators for all tokens contained in the mention span concatenated with their parts-of-speech
- same as above but with a window of five tokens before and after the mention

The results can be seen in table 3.5. The F_1 measures compare the state-of-the-art model with our method. While our model predictably is beaten by the state of the art, the results indicate that by just themselves word embeddings do capture some useful knowledge for this task.

I also designed a pairwise classification model for the next step, analogous to the one used in coreference. The candidate antecedents were selected only from two sentences prior to the one containing the anaphor to limit the negative training

Mention types	Nissim			Hou et al.			Our Method		
	P	R	F1	P	R	F1	P	R	F1
Old	85.1	82.7	83.9	82.2	87.2	**84.7**	75.0	72.5	73.1
Med/worldKnowledge	62.3	64.4	63.3	67.2	77.2	**71.9**	51.5	57.0	54.1
Med/syntactic	41.6	59.7	49.0	81.6	82.5	**82.0**	62.3	59.7	61.5
Med/aggregate	28.4	36.8	32.1	63.5	77.9	**70.0**	49.7	47.8	48.7
Med/function	0.0	NA	NA	67.7	72.1	**69.8**	68.9	30.7	42.5
Med/comparative	0.4	7.7	0.7	86.6	78.2	**82.2**	65.5	54.9	59.8
Med/bridging	4.4	23.0	7.4	44.9	39.8	**42.2**	34.5	26.5	30.0
New	82.7	62.3	71.1	83.0	78.1	**80.5**	71.1	78.5	74.6

Table 3.5: Comparison of the best cascading minority preference system bridging anaphora detection model with our LR model which uses `GloVe` (Pennington et al., 2014) word vectors as a feature

examples. Even then the ratio of negative training examples to positive was 18:1. The classification itself was done by a simple LR model. In this model, given that m_1 is the anaphor in question and m_2 the candidate antecedent, the features used are:

- binary indicators for all tokens contained in m_1 and m_2 concatenated with their parts-of-speech
- same as the above but with a window of 5 tokens on both sides of m_1 and m_2
- distance in tokens between m_1 and m_2
- distance in sentences between m_1 and m_2
- lengths in tokens for m_1 and m_2

As this is a harder task and there are absolutely no semantic features, or even any sophisticated lexical features present in the state-of-the-art model, the model

performed poorly compared to the last step, though better than random (1/18). While the F_1 measure of the negative examples in 96.4% the F_1 measure of positive examples is 0.095%. Then, to the same set of features with added three more features

- the cosine similarity of `GloVe` (Pennington et al., 2014) vector representations of m_1 and m_2
- the average over all tokens vector representation of m_1
- the average over all tokens vector representation of m_2

Just adding these word embeddings increases the F_1 measure of positive examples from 0.095% to a range of 15.6% to 21.1%, depending on the parameters of the classifier. Thus, while nowhere near the hand crafted features of the state-of-the-art model, even here word embeddings do capture a sense of world knowledge and are useful for discovering text references.

3.3 Summary

In this chapter I have discussed in detail the various sub problems which occur in coreference resolution, namely mention detection, and resolution, and allied problems like bridging anaphora. I described the current work which exists in these problems. I introduce a dataset which demonstrates the kind of hard and interesting coreference problems humans can solve and which benefits more from world knowledge sources like Wikipedia. I analysed why data sources from newswire are inadequate for the problem and demonstrate that such a dataset is sufficiently dif-

ferent from existing data and merits investigation. I also demonstrated that current models do not work well with such data, and I describe why this is so.

In this chapter I provided a small foray into using world knowledge by way of word embeddings which help our very simple model match performance with sophisticated ones. Then, I present some qualitative examples of what kinds of instances are present in our dataset which current models fail at. I also present a literature survey on the work done in the related problem of bridging anaphora resolution. I show that even for this task word embeddings are useful. I discuss how to extend this work into more generalised and human like coreference resolution, and how to better do the various subtasks, in Chapter 6.

Chapter 4: Analysing Atypical Images Via Multimodal References

In this chapter I first present our motivation for mapping references to entities present in text to regions in images. Then, I describe work on atypical images in which I use references to perform an understanding task (in the sense not just identities of visual artefacts but their locations, properties, etc. are inferred from text) structured as a ranking/retrieval problem in a dataset made of images of paintings which I have built.[1] This work addresses the unavailability of large datasets for atypical images (in this case paintings) by acquiring and using knowledge to interpret what is going on in the various regions of the paintings, in order to do the retrieval problem. Then, I describe another dataset of complex images with referential text, this time of comic books.[2]

[1]The painting dataset and the associated task in this chapter are from the published work called "A Distorted Skull Lies in the Bottom Center... Identifying Paintings from Text Descriptions" published in NAACL, Human-Computer QA Workshop, 2016. The authors of this work are Anupam Guha, Mohit Iyyer, and Jordan Boyd-Graber. In this work my contributions are primary and include designing the dataset and its annotation, the inference of visual classes from text embeddings without learning, which is a novel method, as well as the bipartite method used in the task

[2]The comic book dataset is from the published work called "The Amazing Mysteries of the Gutter: Drawing Inferences Between Panels in Comic Book Narratives" published in CVPR 2017. The authors of this work are Mohit Iyyer*, Varun Manjunatha*, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daumé III, and Larry Davis. In this work I'm a second author. My contributions, aside from aiding the primary authors to build the dataset, is devising a method to detect and remove advertisement pages from the dataset, and doing data analysis on intra and inter-panel transition metrics.

4.1 Why extend the problem of coreference resolution to vision?

In **Chapter 2** Section 2.4 I describe past and current work in computer vision when the methods involve human language. Why is language important? Humans solve complex vision problems using knowledge of contexts entities exist in, obtained via language, by continuously doing multi modal coreference resolution. What this means is, a human knows what to look for while engaging in vision tasks, and often this knowledge of what to expect is codified in *concepts* which are transmitted as shared world knowledge using language. Biological visual perception is a feedback driven system (Gilbert and Li, 2013) which refines its interpretation iteratively. Human vision, particularly, does not operate in isolation from global context or semantics. In its infancy, a human is a sensorimotor agent, it uses fast and dirty vision without abstract *concepts*, but within 8 months an infant is able to sequence discrete visual stimuli and associate it with auditory input sequences (Kirkham et al., 2002) and displays capacity to detect structure from vision. By 18 months language is used to form abstract categorical representations of spatial relations in vision (Casasola, 2005).

In contrast to the biological top down systems with mechanisms for semantic backflow existing, current machine learning vision models lack the backflow of information needed to augment their weights. For vision models, there is a need for a *conceptual scaffold* of world knowledge which can be used to refine them. For example, giraffes are not usually observed in kitchens, so if a state of the art category detector for images upon running on an image of a kitchen is showing a giraffe

in it, not only is it probably failing, it is failing "gracelessly". The problem is not that deep learning vision systems fail at times, the problem is they give incorrect results in a manner which does not make sense, if a human observer has certain world knowledge.

To illustrate this I take two publicly available vision APIs, CRFasRNN (Zheng et al., 2015), which is a semantic segmentation system, i.e. a system where each pixel is labelled with a category resulting in regions being discovered in the image corresponding to categories, and CLARIFAI (Sood, 2017) which simply tells which categories it detects in the image. I run the image of a tank in a an open field with camouflage in Figure 4.1 and observe the inaccuracies. Now a human might also do an error in understanding such a complex image, but I am more interested in the nature of the errors the systems makes. Observe the boat, cow, and flower-pot regions being discovered. I know from world knowledge it is unlikely for a boat, a cow, and a flowerpot to be in an image in those positions. Similarly, the category detector detects categories like "abandoned", "broken", "rusty", which do not go together with the semantic segmentation categories.

What can deep networks do to change this? I posit that research in this area needs to design deep neural networks which at training time suppress those parts of the competing abstractions of the signals at the lower layers of the network, which do not make sense with high level context. This cannot be done without the *knowledge* of what relations of categories make sense. I call this representation of knowledge a *conceptual scaffold*. In this thesis I claim that such world knowledge needed to create a conceptual scaffold can be inferred from unconstrained natural

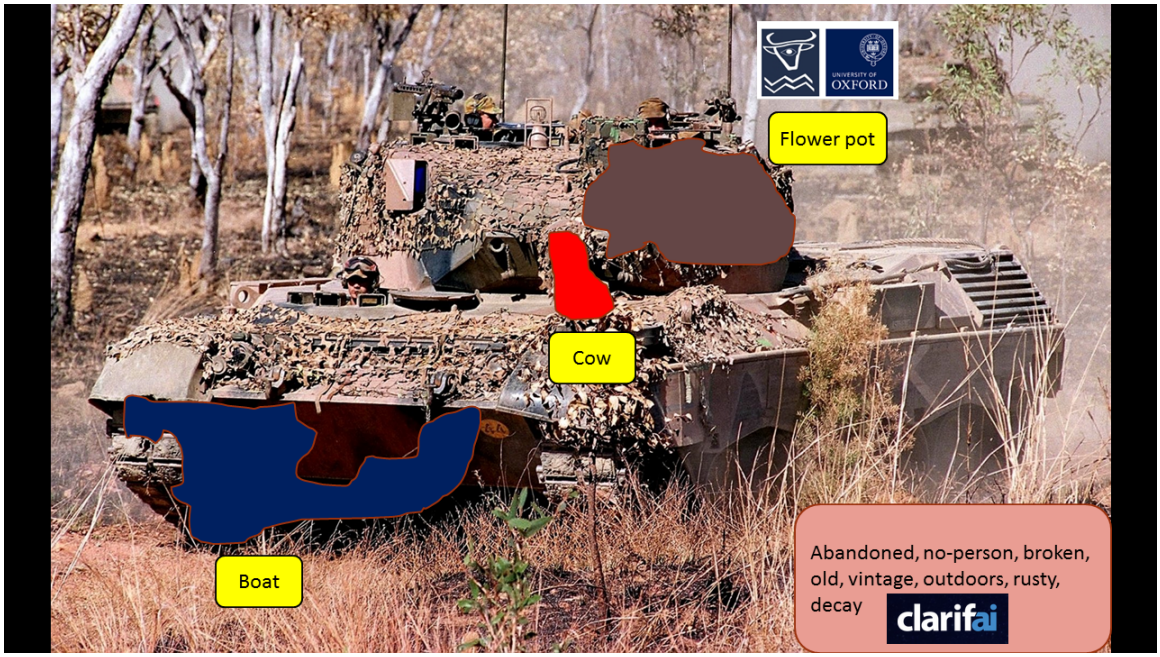

Figure 4.1: An image of a tank camouflaged run through the publicly available CRFasRNN semantic segmentation and the CLARIFAI visual category detection system

language, because natural language descriptions are easier to obtain than specific hard annotation labels. This is our motivation for acquiring text which describes what is going on in an image, and also which parts of the text refer to which parts of the image, thus extending the problem of referring concepts to become multi modal. Our goal in doing this is to influence vision research to use more knowledge to regularise their learning systems.

4.2 Hard Images: The Painting Dataset

As mentioned before the current methods depend completely on identification of correct object classes and recognition from images. This is a brittle methodology in several respects. It makes the system completely reliant on individual classifica-

tion for objects which may be incorrect. Humans do not use vision like this, they use context to identify objects in their field of vision (Torralba et al., 2006), i.e. humans are armed with the knowledge of what is expected to be observed in what context when observing a scene. They can also predict object motion using cognitive expectations (Kowler, 1989). From a pragmatic standpoint the flaw of this method is that both the closed set of categories needs to be large enough that retraining isn't often needed, and the images themselves be not too much of a domain mismatch to certain categories of natural images which the main large datasets are composed of. Annotating large image datasets with hundreds of thousands to millions of images is a non trivial task. So how does one solve a vision problem for an image which humans can understand despite not seeing many instances of something like it?

This work calls these complex images. An example of such complexity is in *paintings* where humans are able to identify extremely varied visual representations of real world images. These representations vary a lot painter to painter, and style to style, and sometimes look significantly different from the real world images of objects they are based on. In isolation humans might not able to identify individual objects present in these paintings but given the entire painting which presents those objects in relation to each other, spatially and otherwise, they are able to understand them, even if the style of the painting diverges from a realistic depiction. Other domains of such hard images are cartoons in comics (work described in Section 4.4) and sketches (Eitz et al., 2012). While humans generally have no problems in understanding these, current machine learning systems cannot generalise from real world images to these domains.

One method which humans use to learn usual co-occurrence of objects in vision and the contexts in which they appear is through language descriptions, i.e. humans use language to convey to each other what they are likely to see together in a complex image from one of the domains described above. This method is a visual analogue of the task of coreference resolution, because a human learns what part of a complex image, given the context around it should refer to what real world entity, even if the representation of it is not visually accurate. Hence, a domain is needed which is hard for present models to segment, understand, recognise objects in, etc. but possible to describe using language and then possible to do the aforementioned task using multi-modal coreferences between the images and languages. The dataset I build in this chapter is a spiritual sibling of (Kong et al., 2014) where a three dimensional scene dataset with descriptions is used. The domain I choose for this dataset is paintings. One of the reasons for choosing this domain, aside from the ones listed above, is the fact that quiz bowl, discussed in the last chapter, has a lot of questions on the topic of art, and thus it is relatively easy to get questions based on artwork which are descriptive .

Our dataset comprises of 128 unique paintings. These paintings are hand annotated using the labelme (Russell et al., 2008) webapp with contours of objects in them, and each object is assigned one out of a set of fixed classes which are organised in an hierarchical ontology. This ontology is based on the class tree of imagenet so that experiments can be done with networks pre trained on imagenet classes. Also, every painting is accompanied by one or more text descriptions. Spans of text which are coreferent to each other are first annotated and then these

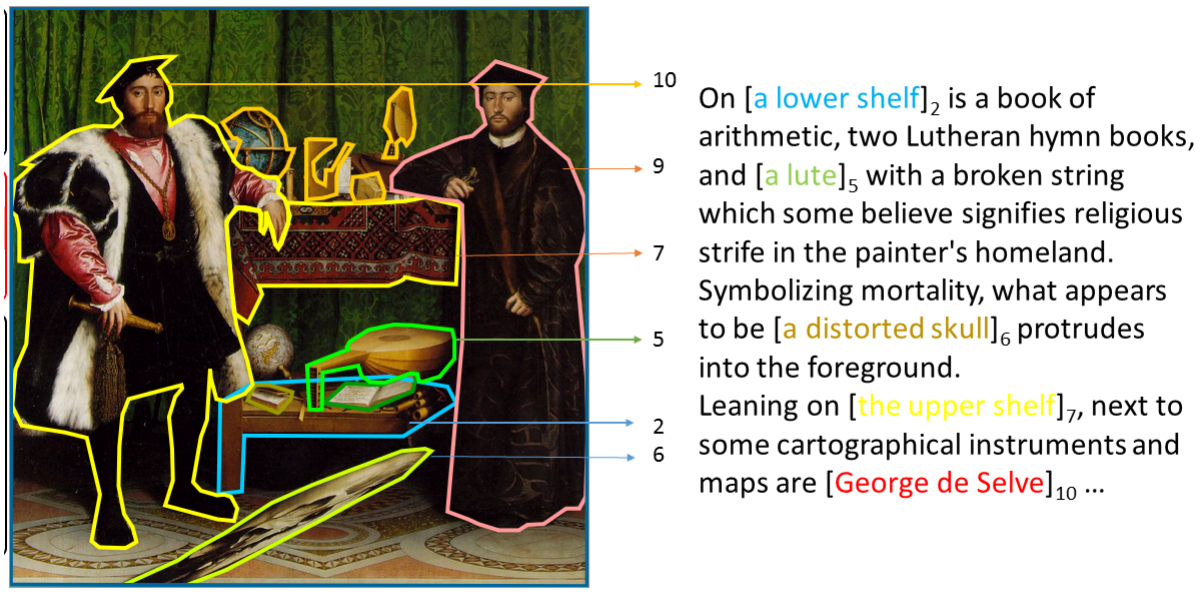

Figure 4.2: An annotated example of the painting The Ambassadors (1533) by Hans Holbein

coreference chains which refer to objects in the painting are annotated. Figure 4.2 shows an example of one annotated painting. These descriptions are also obtained from quiz bowl questions (same domain as in Chapter 3) and thus are full of non descriptive noisy text which do not refer to anything visual in the painting, like text talking about the painter or the time period. This leads to retrieving a correct painting from this dataset a very hard task if only raw text is used.

This dataset is useful to test vision tasks as it is harder and qualitatively different than datasets made of natural images. An example of what happens when a current CNN trained for the Imagenet 1000 class object detection problem is run on one of the paintings is shown in Figure 4.3.

From a human perspective the painting in Figure 4.3 is easy to understand

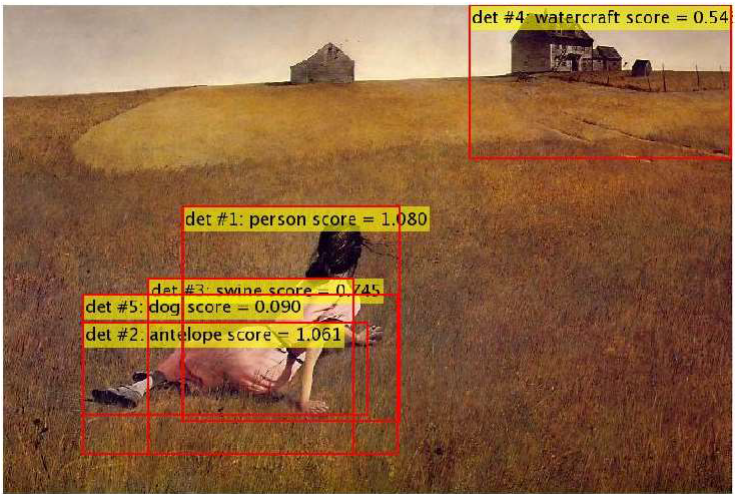

Figure 4.3: Running the RCNN system with an pretrained VGG architecture on the painting Christina's World (1948) by Andrew Wyeth

and the objects in it easy to recognise. This work is interested in developing vision architectures capable of doing standard tasks on such images, and for that it is necessary to take help of the semantics of the image.

4.3 Recognition Constrained on Semantics of an Ontology

I define the problem as one of retrieval and ranking which is a common image task. Given a set of descriptions of paintings, and having the images of the paintings, how to get the painting being closest to a description. To make the task even more difficult the textual descriptions contain a lot of noise and without a lot of training data and with a large number of possible paintings this is not an easy problem to solve as with a classifier. However, I solve this problem with via the classes of objects in the paintings which have similarities in the semantic domain to what the texts describe.

Our method assumes that the groups of coreferent text in the description might

describe an object in a painting as coreferent groups by definition refer to entities and in the case of paintings some of those entities are what is present visually in the paintings while other entities may be non-visual. If the ontology of object classes used to segment in the painting is constrained, then a bipartite mapping can be made between visual objects present in a painting, obtained via semantic segmentation, and the various groups of coreferent texts, if a unified vector representation could be made of these two. The beauty of bipartite matching lies in the fact that there can be groups of coreferent text that do not describe objects in the paintings (like text describing the painter) and that is okay as the bipartite matching is looking for the maximum number of matches between a candidate painting and a description, and the method will take the painting with the maximum number of matches in with a description as the solution for it.

Our method also matches not just object categories between the text and image, but also visual properties of these objects in relation to the image, like location, number, etc. Thus, while the ranking problem is solved, the method is also partially understanding the complex images, discovering not only object categories from descriptive text, but their visual properties.

4.3.1 Inferring Visual Properties from Coreference Chains and Bipartite Matching

In order to do bipartite matching between descriptions and candidate paintings, one needs two sets of lists from each, comprising of lists of class, location,

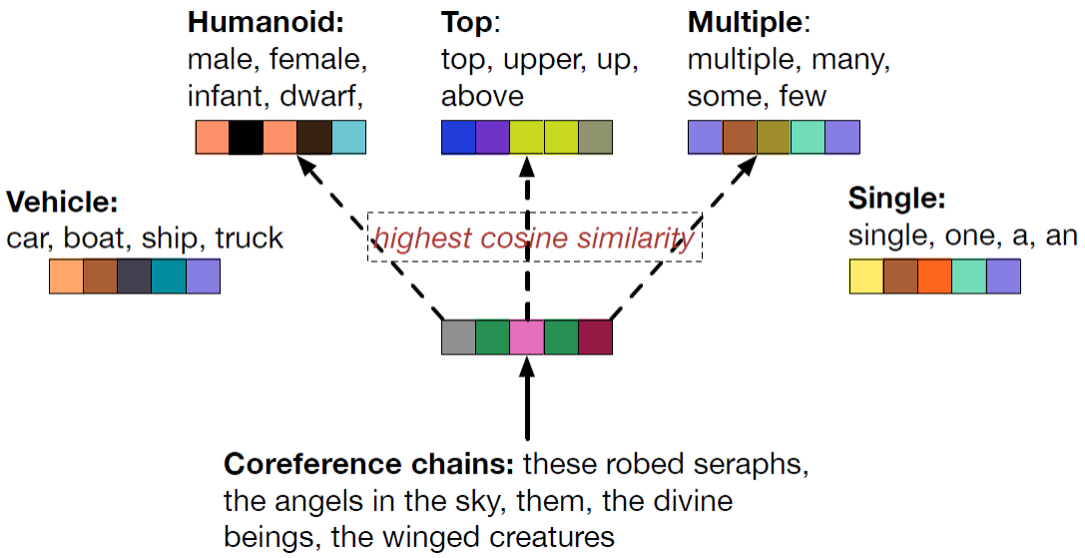

Figure 4.4: Using word vector representations from coreferent groupings in a description to deduce object class and attributes by cosine similarity

and number present in a description or a painting. To do that I first need to find what class matches a coreference chain in semantic vector space where they can be compared. I use word vector embeddings which have been described in Section 2.2 of Chapter 2. I use the publicly available 300 dimensional word2vec trained on Google news. The method averages over all the vector representations of words in a coreferent chain to get one vector per reference cluster in a piece of text, similarly, for every object class in the ontology I obtain a lexicon describing that object class via synonyms, hyponyms etc. (hyponyms are used so that the word vector is influenced by vectors of more specific classes, these are chosen according to the object category tree the paintings have) and average a word vector over it. Since distance between word vectors represent similarity this conversion of both visual and text data to word vectors presents an opportunity to first discover what visual class is most similar to a coreference chain, as demonstrated in Figure 4.4.

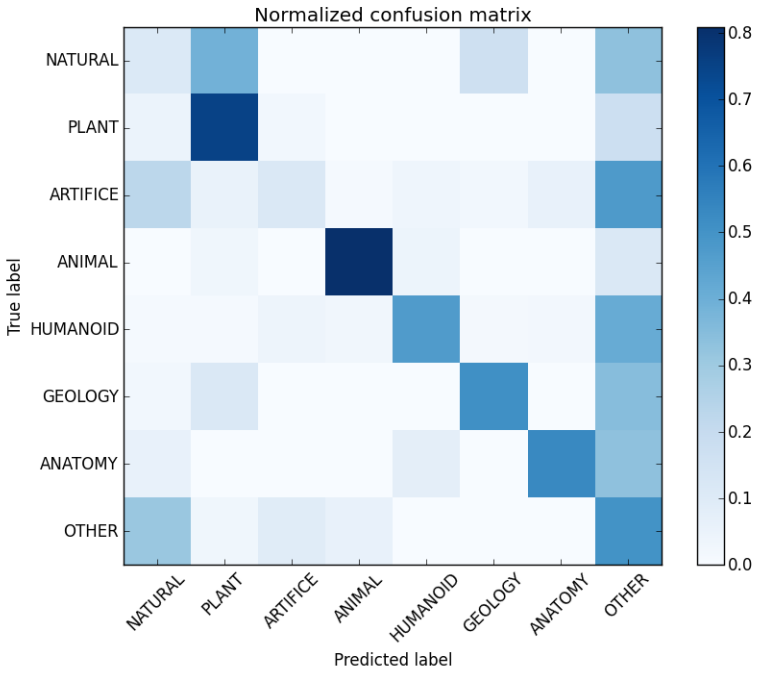

Figure 4.5: Using word vector representations of words in phrases describing objects to classify their coarse object classes.

The first experiment is to find if these vectors from text are good matches to the vectors for the visual classes. Using euclidean match between these two sets of vectors reveal that the they are surprisingly robust.

Figure 4.5 shows the confusion matrix for classification for coarse object classes, i.e. classes on the top of the ontology whereas Figure 4.6 shows the confusion matrix for the fine object classes. The reason for considering both the coarse and fine object class is to make the bipartite matches of the next step more robust, i.e. even if the object detected from the coreference chain in the text is not specifically the one in the painting, if it is in its vicinity in the ontology tree, then the method still obtains a weak match.

Now that one can infer which object class each coreference chain corresponds to

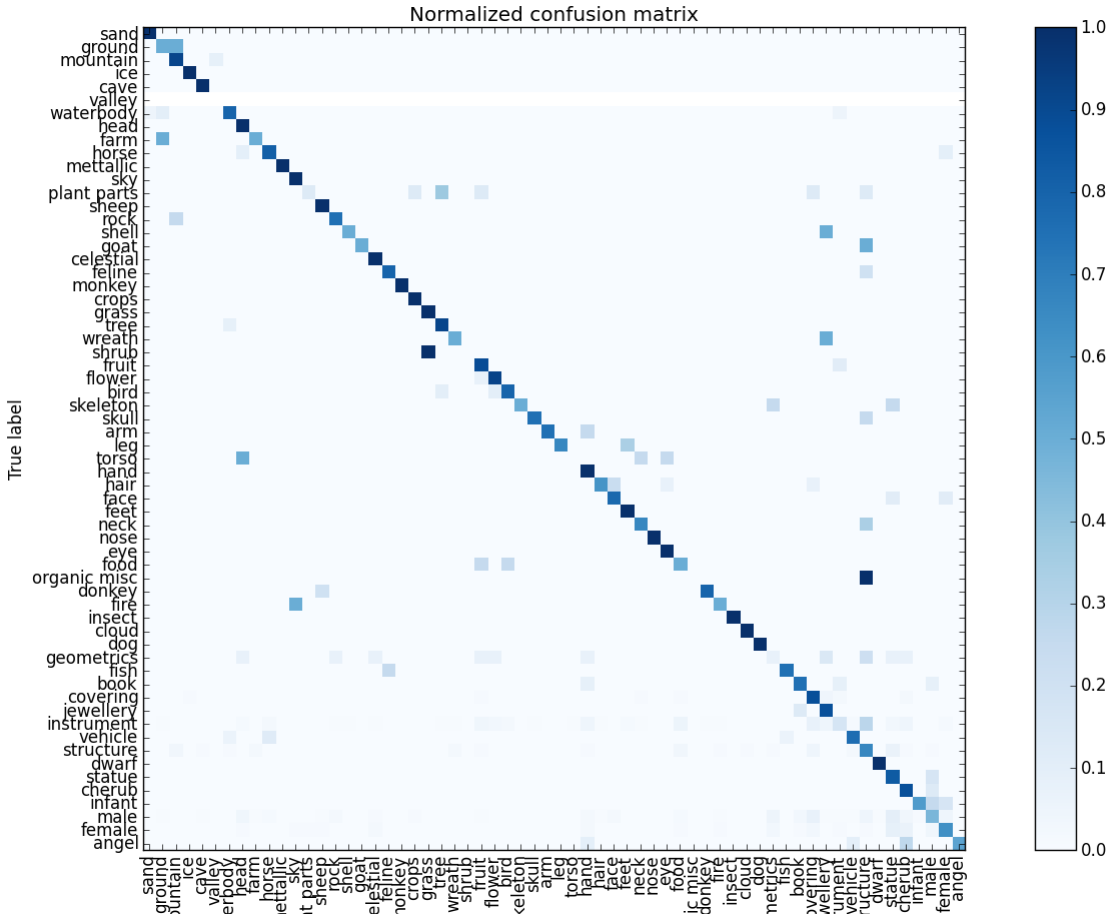

Figure 4.6: Using word2vec representations of words in a phrase, extracted from a description, to classify which visual object class, of fine granularity in this ontology, is being referred by it. Despite the specificity of some of these object classes and the vagueness of these descriptions most of the classes are detected correctly. The errors, like descriptions of farms being misclassified as ground, or those of shrubs as grass, also make sense.

it is possible to do bipartite matching as shown in Figure 4.7. A maximum bipartite match consists of finding the maximum number of matches between two lists such that no two matches share an endpoint. As the method has lists of objects for every painting, and lists of inferred objects from every description, it is possible to do bipartite matching to discover which painting gives the best match for a particular description. This is the second step of our experiment. When this is done the method obtains for 42% of the descriptions in the dataset the correct painting.

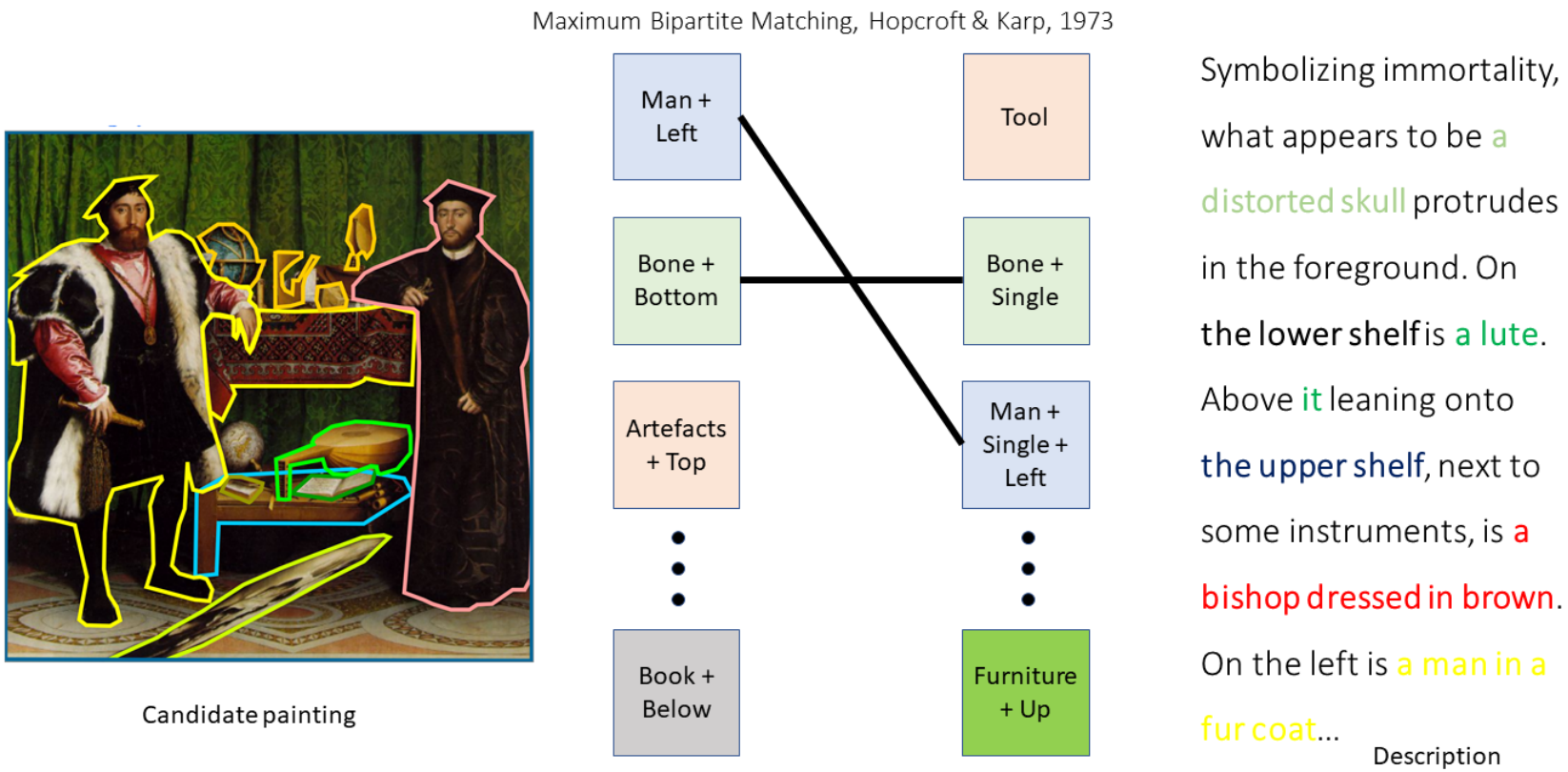

Figure 4.7: Bipartite matching between a description with that of a painting. From the painting annotations of objects are used to extract properties like their location and number while the gold object categories are used, from the description coreference chains are used to deduce class, location, and number using word embeddings. Matches may be bad individually as the descriptions may indicate a similar object class or a less specific object class in the ontology, but using multiple matches good retrieval results are obtained.

Feature	P	R	F1
Coarse object class	0.72	0.38	0.45
Fine object class	0.72	0.60	0.60
Object location	0.32	0.25	0.24
Object number	0.96	0.81	0.88

Table 4.1: Individual metrics of classes and features detected by word embeddings from coreference chains describing objects

However, these results can be improved. The painting annotations do not provide us just the class of the objects in them, it is possible to use the geometry of the object annotations to infer the location of it with respect to the painting, as well as the number of objects per class in the painting. These visual properties can also be inferred from the coreference chains in the text descriptions, and using these extended properties the bipartite matching procedure is improved. In order to discover spatial properties from embeddings, I make a list of words describing cardinal locations, like TOP, BOTTOM, RIGHT, LEFT, CENTRE, etc., and find word vectors corresponding to each of these labels, and on the description side the method deduces from the adjectives present in the coreferent group if there are any words describing its spatial direction with similar word embeddings. I take care to ensure that these direction words are relative to the whole painting and not relative to some other object in the painting and the method infers this using the IN POS tag and its proximity to another coreferent group describing an object. Similarly, it is possible to deduce from coreference chains, the whether the objects they describe are singular or plural. The Table 4.1 shows the classification results for both the object classes and their attributes using word embeddings of their textual descriptions.

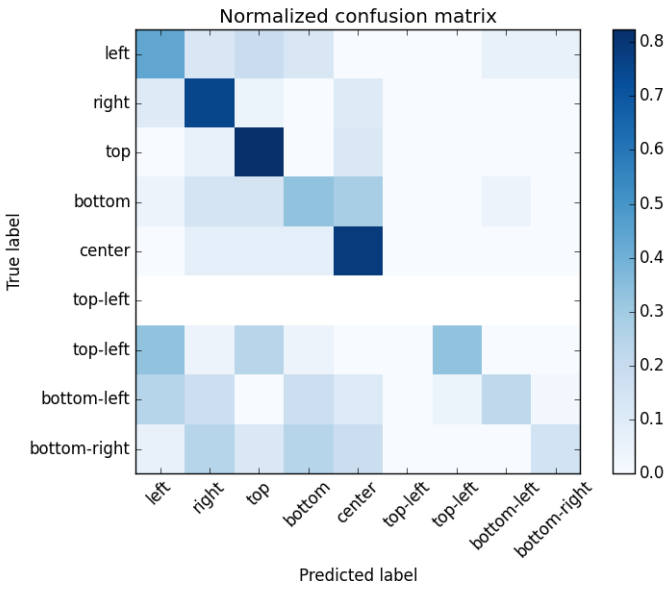

Figure 4.8: Using word vector representations of coreference chains to infer the location of the object being described by them compared to the true location of the object in the painting

The direction words obtained thus from the description prove to be extremely effective in judging the spatial location of the object for the four cardinal directions while due to the vagueness of descriptions (and often incorrectness) it is harder to deduce more specific directions like top-left, top-right etc (Figure 4.8 and Table 4.1). Word vectors obtained from the coreferent groupings are also robust enough to detect the number of objects as shown in Figure 4.9.

4.3.2 Performance in the Retrieval Task

Using these features in conjunction, over 61.7% of the descriptions are correctly identified as shown in Table 4.2 with their paintings. This is much higher than what can be achieved on such a small dataset with no training or fine tuning using a conventional method. To evaluate this result I use a baseline trained on extra text

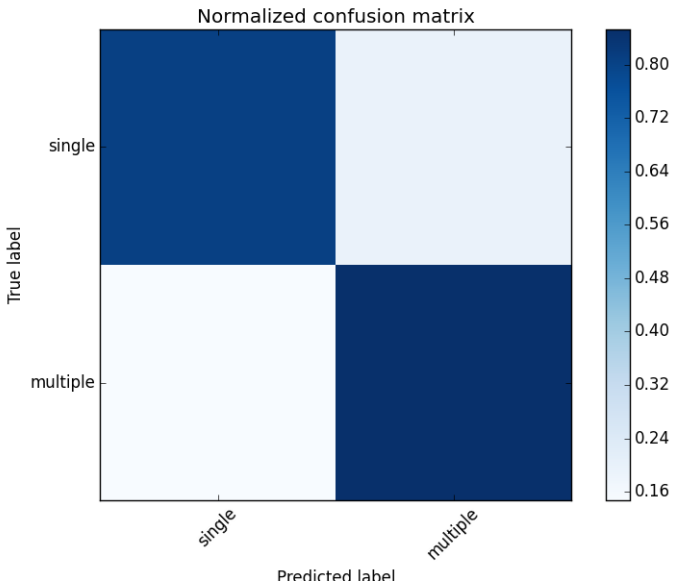

Figure 4.9: Using word vector representations of coreference chains to infer the number of objects being described by them compared to the visual number of objects

Method	**accuracy**
dan	59.4%
ArtMatch: fine objects	42.0%
ArtMatch: all objects	58.6%
ArtMatch: objects+attributes	61.7%
ArtMatch+dan	**65.7%**

Table 4.2: Our system vs the blind baseline. DAN is trained on 503 questions but has no visual information. ARTMATCH has visual features from paintings but no training data. Combining both leads to a significant increase in performance.

descriptions of paintings, but not vision (which is called the *blind* baseline). In this baseline I use 503 extra training descriptions of the same set of paintings to train a question answering model with the set of painting names as the possible answers. Thus, unlike our system which has no training data, I provide the blind baseline with text training dataset. The model I use for training this QA task is the Deep Averaging Network (Iyyer et al., 2015). A DAN is a neural network which averages the word embeddings of the various words in a sentence without considering their

order, thus it takes all the words as an unordered tuple, then passes them through a couple of hidden layers before predicting the category with a softmax layer. This model thus can be used for question answering if the answers are finite. Because a DAN uses word embeddings and it has extra data, the model outperforms our system, but when I combine the two systems the results further improve, indicating that the two models, one with no learning data but inferences from embeddings, and one which is blind to vision annotations but text training data are, are discovering different things.

I present a brief error analysis here. In 34 questions the DAN fails but ARTMATCH succeeds. For many of these, the DAN fails because it overfits to common clues. Given a test question about *Melencholia I*, the DAN answers *Madonna with the Long Neck*, as the training questions about both paintings repeatedly mention a female figure and cherubs. However, the question also mentions geometric figures, the spatial locations of which enable ARTMATCH to answer correctly. In 31 questions ARTMATCH fails but DAN succeeds. Some of these questions contain text constructs such as the painter's name that are repeated in both training and test questions, which makes it easy for the DAN to solve (e.g., "Identify this most famous work of Claude Monet"). In other cases, ARTMATCH answers incorrectly because of spurious matches due to substantial visual similarity between various objects in paintings. For example, in a question about *The Holy Trinity* by Masaccio, "St. John" is assigned the close but incorrect class of "statue" while "Jesus" is correctly identified as a person. Further confused with spatial similarities between the paintings, ARTMATCH's answer is *Supper at Emmaus*, which has Jesus but no St. John.

In other cases, peripheral similarity leads to the central mismatch being overlooked.

Here, I will discuss the limitations of the painting dataset. Due to the complex nature of annotation, as well as the rarity of unique quiz bowl questions referring to unique paintings this is an extremely small dataset, and while the annotations are of high quality, they are far too few to be used for data hungry deep learning models. Aside from this dataset there was no dataset of atypical or hard images (sufficiently different from natural images to be a challenge for existing multi modal frameworks) alongside text to be used for similar research in models operating between language and vision. And yet there is a need for such data, because humans can solve such problems with fluency. To address this, I present another dataset of atypical images alongside accompanying text described in the next section.

4.4 Comic Book Dataset

Here I introduce briefly another dataset of atypical complex images existing alongside text, namely one made from comic books called COMICS, which is relevant to this problem of references between text and images, and was built with my collaboration. One of the primary motivations for building this dataset was the difficulty which neural networks face for paucity of data for such complex images. Comic books solve the problem because, a) a lot of them exist in the public domain, b) they are by their very nature multi-modal with text alongside images, and c) a lot of information in comics is not directly portrayed visually, but is inferred "between" panels by the reader, or what is called the gutter. This makes for interesting and

complex experiments which do not get addressed because of lack of datasets. Have a look at one pair of panels from this dataset in Figure 4.10. This kind of reference problem is hard to solve.

Comics are sequential art (Eisner, 1985) which go through a narrative through a list of panels, each having images and/or text. Before the COMICS dataset was built, there were a few comic datasets (Guérin et al., 2013; Matsui et al., 2016) but they are too small to run any sophisticated machine learning experiments on. In contrast, COMICS is obtained from 3948 publicly available comic books, and their 198,000 pages, from which 1.2 million panels have been extracted using panel segmentation. This panel segmentation was done using deep learning in which 500 pages were manually annotated for panels. From each panel, the text boxes have been detected, again using a simple neural network. The primary authors and I annotated 1500 panels with text boxes to train. This resulted in almost 2.5 million text boxes. On these text boxes Optical Character Recognition (OCR), namely Google's Cloud Vision OCR, run on these textboxes. I also eliminated those pages from the comic books which had advertisements instead of comics by making a simple bag of words based advertisement detector from the OCR text and annotating a the adverts in a thousand pages. COMICS is one of the, if not the largest, multimodal dataset of images and corresponding text in existence.

The COMICS dataset also has some dataset analysis, namely panel type and panel transition information. In brief, panels in a comic book can be of certain types depending on what content they have, they could be dominated by the text, or by the image, or the text and the image could be independent, or they could be

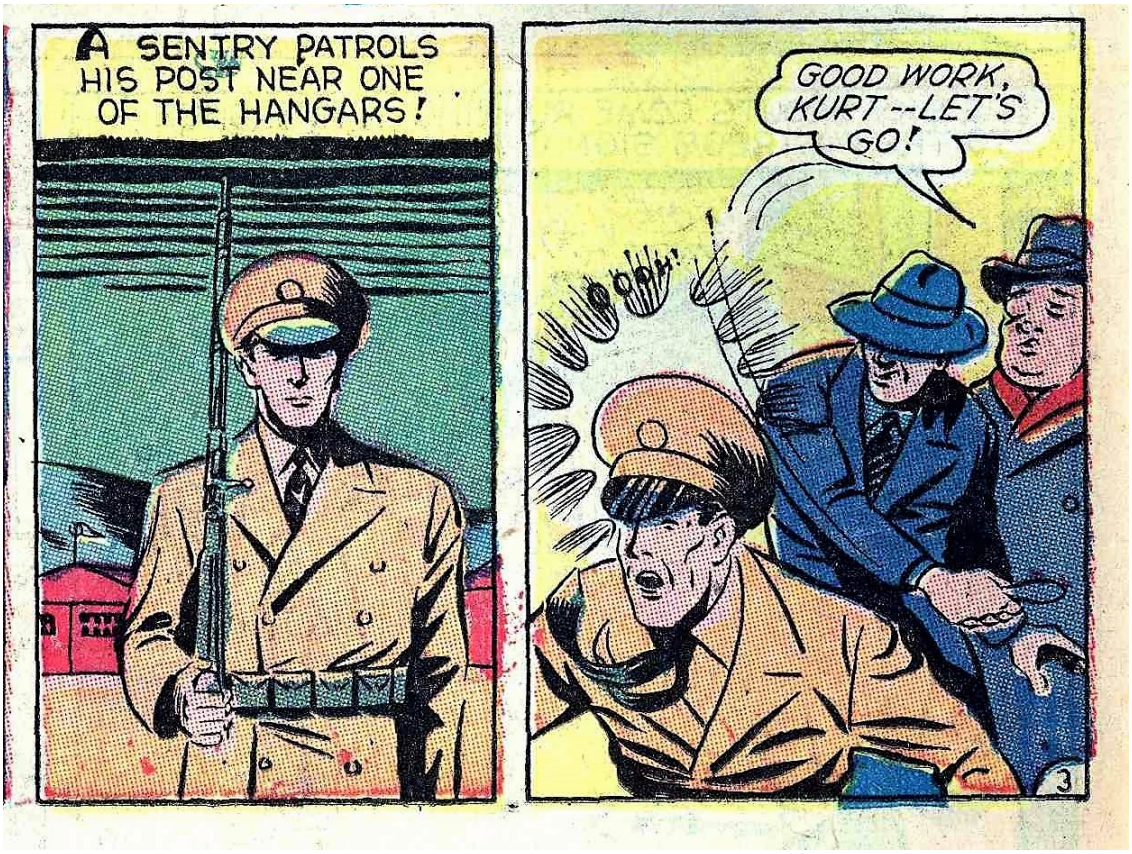

Figure 4.10: An example of a panel pair from the COMIC dataset. The panels have been detected by a segmenting neural network, as has the text boxes and OCR has been done on this. Using that data is it possible to infer who Kurt refers to in the second panel? Why do we know that the soldier is not Kurt? This is similar to the reference problem in paintings, where you need to match coreference chain entity with pixel blob, and it is impossible to correctly solve who Kurt is purely visually, without modelling sophisticated world knowledge.

interdependent. These categories are obtained from the work of McCloud (1993) which analyses comics. Similarly, the manner in which panels transit, i.e. how narrative goes from panel to panel can also be classified into certain categories. These transitions can be, for example, action to action, where the characters remain the same but the action differs, or they could be from scene to scene, where the entire scene changes, etc. 250 randomly panel pairs were annotated for their inter-panel transit and their intra-panel type, and found out while the transition types are more

evenly distributed, almost all intra-panel classes are those where text and image are interdependent, this signifying the relevance of this dataset to our work which needs a strong relationship between the images and the accompanying text. Because the text almost always is interdependent with what is going on in the image, future work can use this dataset to design hitherto unexplored experiments on multimodal reference.

4.5 Summary

In this chapter I have provided two complex vision datasets made out of paintings and comics, and for the first one, with the usage of word embeddings, described a simple method to refer visual properties to coreference chains. I described a ranking/retrieval task, and using bipartite matching between two lists of averaged out word vectors, one obtained from lexicons of object classes of the paintings, another obtained from the coreference chains of the description, retrieved the correct paintings. I thus motivate the kind of research work needed to use more world knowledge while performing vision task. I also introduce a large multimodal dataset of atypical images and accompanying text obtained from comic books. In Chapter 6 I talk about the future avenues for extending this work and how the COMICS dataset can be used to detect various kind of multimodal references.

Chapter 5: Discovering References to Prototypical Concepts in Movie Scripts

Till now, this thesis has dealt with concepts which can be referred to by contiguous text spans. However, I expand the conventional definition of references to associations which are more complicated: first, what is being referred to are not concrete entities but abstract concepts, and second, the references to concepts are not present in the text as contiguous spans but as discontinous distributions over larger spans in the body of text. Thus, these concepts are latent, analogous to topics in topic models described in Chapter 2, but different from something like LDA in the sense that their position in the text is relevant, the text isn't treated as a bag of words.

Consider themes in a narrative, like action, romance, sadness, etc. A large enough body of narrative text will have multiple oblique references to such themes present in every sentence, though not mentioned directly in words. In this work I expand the idea of reference like this, because for an intelligent agent it is necessary to know what abstract concepts are present in various spans in discourse to not just understand what is being talked about, but also correctly resolve more concrete references. Discovering these concepts and how they are referred to in important for

machine understanding of such complex text. Knowledge of what theme is under discussion in which sentences also serves as shared knowledge for downstream tasks. This chapter investigates the narrative text present in movie scripts, tries to discover what prototypical concepts may exist in them, and then tries to find which span refers to what concepts.

5.1 Current Approaches in Prototype Discovery

The idea of prototypical concepts in narrative text has long been around in computational linguistics, specifically in computational literary analysis, and recently, some datasets and methods have emerged in the area (Bamman et al., 2014a,b; Chaturvedi et al., 2016; Flekova and Gurevych, 2015). Often, these methods focus on solving characteristics of entities or events in a narrative. In previous work, classifiers have been proposed to learn character archetypes (Flekova and Gurevych, 2015), or relationships between pairs of characters (Chaturvedi et al., 2016) in a narrative. There has been work done to construct social networks from narratives as well (Elangovan and Eisenstein, 2015; Srivastava et al., 2016). Recently, there has been work which recognises that these prototypes might not remain constant over the entire text, for example, the relationship types between pairs of characters alters in time (Iyyer et al., 2016).

A close analogue of this problem is topic modelling (Blei et al., 2003). Topic models are machine learning methods to describe a collection of documents in terms of abstract "topics" wherein each word is assigned one of these latent topics. Topic

models in the past have been used to understand large text corpora (Dou et al., 2013; Gretarsson et al., 2012; Nguyen et al., 2014). Generally, topic models like the LDA (Blei et al., 2003), which learns a topic distribution per document, do not consider the sequential nature of text, but a class of topic models like the hidden topic Markov Model (Gruber et al., 2007) does that. The HTMM (described in Section 2.6.1 of Chapter 2) has a temporal component which makes sure that the topic of the current span is similar to the topic of the last one, and thus, the topics change smoothly over the document. As this work is interested in narrative text, it is essential that any model has the assumption that whatever prototypical concept a span refers to, will not change abruptly in the next span.

5.2 Our baseline: Relationship Modelling Network

I define the task as characterising each abstract concept by a distribution over the vocabulary, and assign reference from each sentence/span to a concept or a distribution over the set of the concepts. One model that emerges close to what a task like this needs is the RMN.[1] This model is similar to deep recurrent autoencoders like Li et al. (2015) and to neural topic models (Das et al., 2015). It is a model which takes as its input a set of documents, where each document is a subset of spans obtained from a narrative. The spans are those which refer

[1]The Relationship Modelling Network comes from the paper titled "Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships" published in NAACL 2016. The authors of this work are Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. In this work my contributions are designing experiments to test whether the topics can detect positive vs. negative, as well as in evaluating the results. The Joint Modelling Network which is an extension of the RMN is unpublished. In that work my contribution is primary, both in the model itself, adapting a dataset to work for it, and in evaluating the results The JMN work is done in collaboration with Dr. Ferhan Ture of Comcast Research

to two characters in the document, thus one document can be thought of as the history of the relationship of the pair of characters. The model assumes there are prototypical relationship types which every document is a sequence of. It discovers a) a list of these prototypical relationships over the dataset, and b) the prototypical relationship distribution per span. Thus, every relationship can be described as a trajectory of dynamically altering prototypes.

The model itself is a neural network which takes as its input a word vector representation of the span, the two characters in it, and the name of the book. These vectors are composed in the next layer, which gives it to a customised RNN layer. This layer, being recurrent, ensures the hidden state will get modified by the previous hidden state from the previous span (ensuring "smoothness", analogous to the HTMM). Then the hidden state is multiplied to a matrix initialised randomly, and the result is forced to be similar to the original vector. Thus, over time, the matrix has per row a vector which in word embedding space is a prototype of all the spans. The matrix, now a list of prototypes, is called the descriptor matrix. Because the model gives a distribution on this matrix for every hidden state, every span can be assigned one of the descriptors. Thus, each document can be represented as a descriptor trajectory. This process can be seen in Figure 5.1.

While in Iyyer et al. (2016) this model was used to discover relation prototypes for movies, it can be easily adapted for any other kind of prototypes in any other narrative text. After all if you replace spans with specifically two characters, to all the spans with only one character, you discover concepts which can, as a trajectory descrive all possible character arcs. If you take all possible spans in a narrative

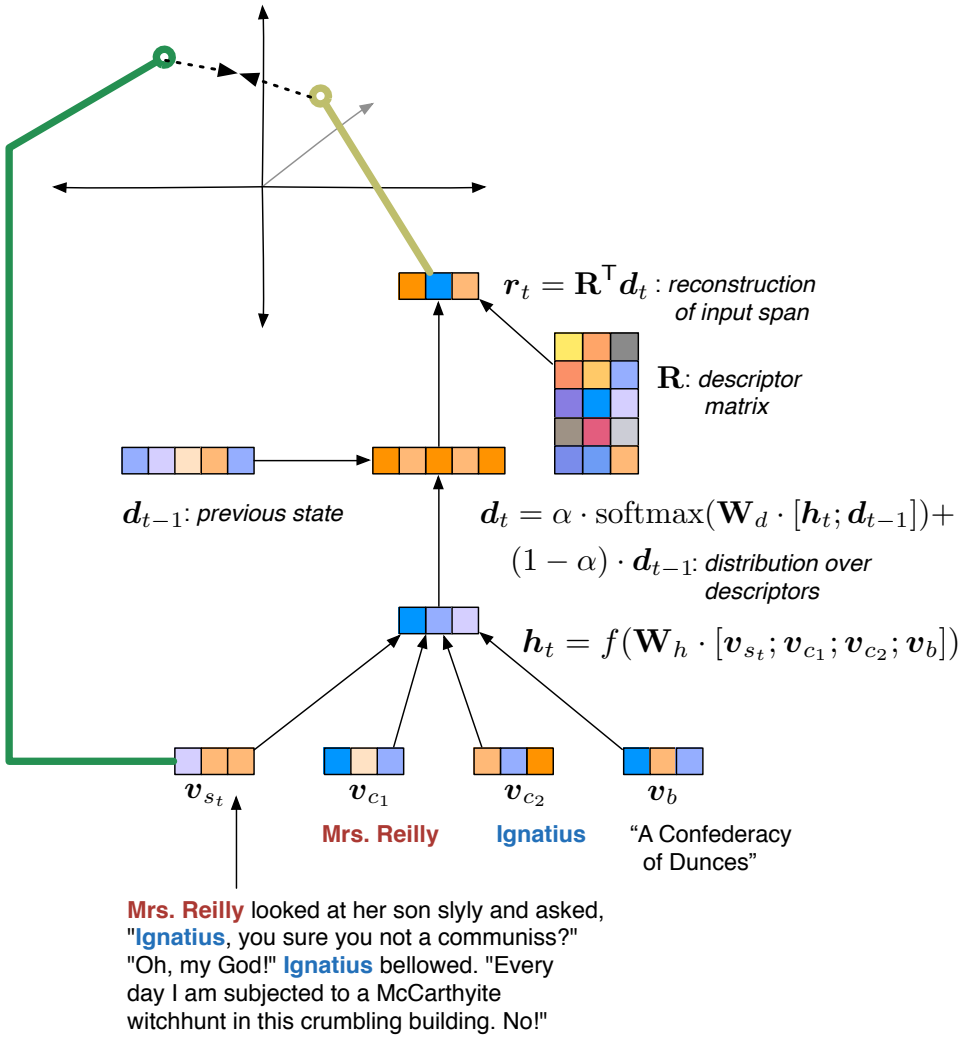

Figure 5.1: Relationship Modelling Network from Iyyer et al. (2016). In this model the input vectors get averaged, then go to the hidden layer which is influenced by the last stage, and then get multiplied by the descriptor matrix to get reconstructed. Each row of R trains to be a descriptor (credit: Mohit Iyyer)

without the character labels, you discover concepts which describe the narrative trajectory itself. However this model has one flaw. It assumes that these prototypes occur in isolation. In the relationship discovery case for example, this model completely ignores all other sentences of the narrative. I present an extension of this model, which discovers multiple classes of prototypes at the same time, and does

away with the independence assumption. I use that model to analyse movie scripts.

5.3 Joint Modelling Network

A narrative is an interplay of interdependent prototypes. For example, a narrative may have some global themes, i.e. concepts which are referred to by all the sentences of all the documents in the dataset being investigated. It may also have character arc prototypes, i.e. concepts being referred to only by those sentences which refer to one character in the narrative. If it has the global theme of "war" (i.e. general sentences referring to the "war" concept) and the character prototype of "tragedy" (i.e. sentences particular to one character referring to the "tragedy" concept), these prototypes will influence each other's occurrence. Similarly, the character arc prototypes may affect the relationship descriptors, the kind which the RMN discovers. If a character is in a "tragedy", their relationship descriptor with another character at that moment in the text will get suitably influenced. This idea is the basis of the joint neural network and is illustrated in Figure 5.2.

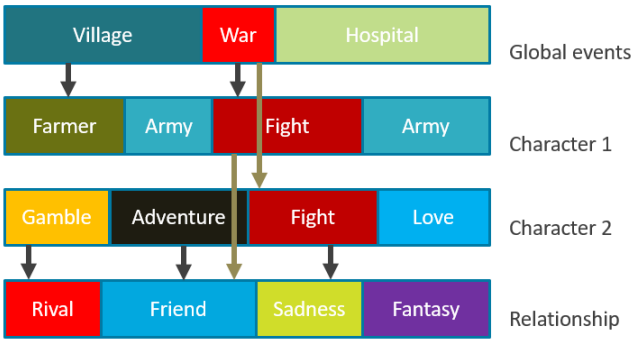

Figure 5.2: Example of sequential prototypical concepts of different types that occur in a movie script and how they influence each other

First, I alter the RMN to work for different kinds of prototypical concepts. One RMN is altered to a Universal Modelling Network, i.e. it takes all sentences from all narratives in the dataset instead of selected spans for relationship arcs like the RMN did, finding global themes across complete narratives and assigning each sentence in them with a global theme. Another RMN is changed to a character arc modelling network. It is fed sets of sentences which contain only one character reference, and this learning concepts which define character arcs. The third RMN is unaltered. Now these models are connected. The hidden layer of the universal model becomes the "history layer" for the character model. This means, the reconstructed vector from the universal model at that point in the narrative, is the history of all global themes that occurred till then. When running the character model, when it runs an iteration for a certain character, it takes that history vector as an input to its hidden layer.

Similarly, the reconstruction vectors the character model generates, become character history, and when the relationship model operates for a pair of characters, it takes those two vectors representing the history of the character till that point, as additional inputs. The joint model is illustrated in Figure 5.3

There is another small change from the RMN. The three components of the JMN are deeper, the universal network has three extra hidden layers, while the other two networks have two extra layers each. This is needed as the data obtained from the movie scripts is less coherent than the literature data the RMN is designed for, on deepening the model the results improve but training becomes slower. All the layers have dropout (randomly dropping units from the network during training)

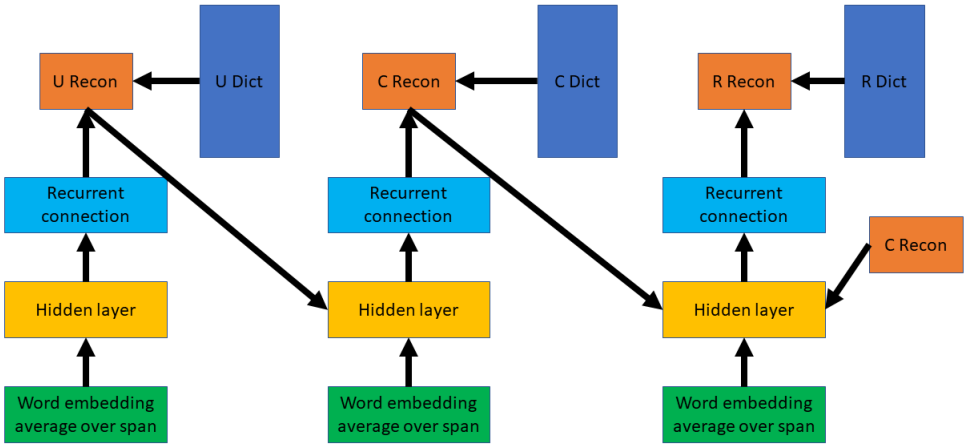

Figure 5.3: Connecting three RMNs into a Joint Modelling Network where the reconstruction vectors of the more general model serves as the history vector of the more specific one. To ease training (as the models use different subsets of the spans in a narrative), the reconstruction vectors are cached while training the more general models, and then reused while training the specific ones.

which prevents this network from overfitting. I do not alter the loss function from what is used in the RMN but the hyperparameters need to be empirically altered as the text in movie scripts is different from that of novels which the RMN was trained on. Also, I built a more sophisticated version of the JMN in which the descriptor dictionary is a 3d tensor rather than 2d. The model does not compute a distribution over a set of embeddings as it did in the RMN, but two distributions, one which selects a slice of the 3d dictionary and another which selects the row. The altered dictionary, instead of learning a list of prototypes, learns a list of prototype groupings, with each group having sub prototypes. For example topics related to "police", "military", and "war" might fall under one large prototype with three separate sub-prototypes. The way this works in the model is that the output of

the RNN layer (a softmax) gets converted to two dense layers (both softmax, so we get two distributions from one) one of which picks the slice (the depth) of the 3d dictionary, and the other picks the row (the breadth) from that slice.

5.4 Experiments and Results

The dataset I built for this task is based on the UCSC movie dataset (Hu et al., 2013; Lin and Walker, 2011; Walker et al., 2011) which contains 962 movie scripts. This dataset was extremely noisy as it split the dialogue part of the script out of the descriptive part via a rule based script which only looked at stylistic clues like whitespaces. While I do want the data to be devoid of dialogue text, the disorganised nature of script formats mean that often errors will seep in, compounding the noise heavy nature of movie script text which is filled with jargon phrases which doesn't have anything to do with the narrative. Heavy preprocessing was done to reduce the noise. After this, all stopwords, words which are too rare (appearing in less than 5% spans), etc. are removed from the text. As with the RMN, the Book-NLP (Bamman et al., 2014b) pipeline is run to identify characters and which sentences have what characters. Then all the data is divided into evenly sized spans with character information. This dataset is further trimmed into the three different kind of data to be fed into the three parts of the model. Again, filtering is used to remove spans which have low information, or which have characters/relationships occurring too few times, any character occurring less than five times is not considered for the character arc data, and any pairwise relationship occurring less than five times is

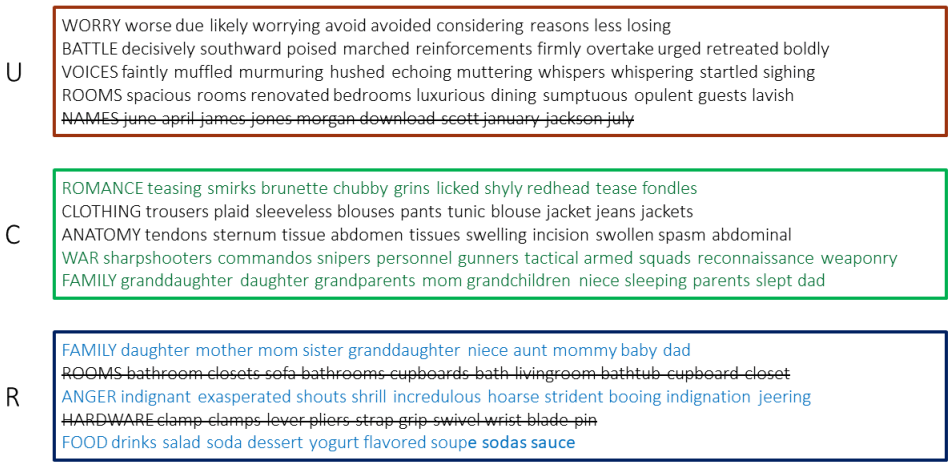

Figure 5.4: Universal prototypes, character arc prototypes, and relationship descriptors obtained from the JMN. Five top prototypes are shown. The crossed out prototypes are the ones which do not make good descriptors.

not considered for the relationship data. Due to the small size of the dataset, such text spans consisting of fleeting characters or relationships will lead the model to overfit and thus should be removed.

I run the JMN on this data. The top five descriptors by probability of the three dictionaries learnt are shown in Figure 5.4. As can be seen, the character prototypes are more personal than the universal prototypes as expected. Some non-relationship like themes are also obtained in the relationship prototypes. Also, because of the nature of the data, it is impossible to filter out all names of characters, leading to it forming its own prototype group.

Because this work is the first of its kind in movie script understanding, it is hard to evaluate empirically, but since the claim is that these prototypes can be used to obtain an understanding of movies based on prototypes, I built a simple recommendation experiment to see if the themes discovered can be used to recom-

mend movies, by doing sequence analysis. If two movies have the longest common local sub-sequences of prototypes, obtained via the Smith-Waterman algorithm, (a weighted sum over the three kinds of prototypes), then it must be similar to another movie based on the content of the script.

Recommendation systems are a large area of research, and are largely of two types, content based (Basu et al., 1998; Pazzani and Billsus, 2007; Van den Oord et al., 2013) and collaborative filtering (Herlocker et al., 2000; Huang et al., 2007; Sarwar et al., 2001). Content based systems base their recommendations on the profile of the user (actions, view history etc.) and the profile of the item (movie tags). Collaborative filtering (CF) methods on the other hand make predictions about the interest of one user based on data of other users. Historically, these systems use millions of user ratings to recommend items to users. Some of the datasets used for movie recommendation tasks are the MovieLens dataset (Harper and Konstan, 2016; Sarwar et al., 2000) which has 26 million ratings for 45,000 movies by 270,000 users, and the Netflix dataset (Bennett et al., 2007) which has more than a 100 million ratings. CF involves working with a two dimensional space of users and movies populated by ratings. The experiments can be predicting a rating for a movie for a given user, or for one user to use the rating data to generate top N recommendations per movie using some kind of similarity metric like cosine similarity or Pearson correlation (Benesty et al., 2009). These tasks are testable as the datasets used for them have rating information per user, so some random rows of the table can be used as a test set, also by removing some movies per user from the train set, it can be found whether the top N prediction for that user has that

movie or not.

Instead of top N movies per user using the ratings data, it is also possible to compute top N movies similar to a movie, simply by considering the distance between two movies as a function of the various user ratings. Given the millions of ratings present in these datasets the results from this are high quality. This is called Item-Item collaborative filtering (Deshpande and Karypis, 2004; Linden et al., 2003; Sarwar et al., 2001). Variations of the KNN algorithm (Koren, 2010) can be used to compute this with various similarity metrics. A point must be remembered here, this measure of similarity based on user ratings, is by convention, as two movies can be widely dissimilar in its contents and yet be highly rated by the same kind of audience.

However, a significant problem in this direction of research is the cold start problem, i.e. if all you have is the script of a movie, but no user ratings yet, let alone millions of ratings, is it possible to recommend it usefully? This is a significant problem as recommendation systems must deal with new movies. In that scenario a partial content based solution is to measure similarity over robust tag information. The MovieLens dataset for example, uses machine learning generated weights for a set of 1128 tags called genome tags (Vig et al., 2012) for a large subset of movies it has, and each movie can be described as a 1128 dimensional vector, and a similarity metric between two movies can be computed. The performance of this is near rating based methods. The genome tag model and similar extensions of traditional tagging involves getting thousands of users to carefully tag movies and is only a partial solution to the cold start problem.

It is evident that the kind of similarity based on the script which JMN finds is not the same as the kind of similarity collaborative filtering methods find. However, I posit that there is some correlation between the two that JMN will give better than naive results, useful in cold start. To compare JMN against rating based systems as well as a tag based system I take from the Movie Lens dataset rating data for those movies which are present in my dataset, and obtain an overlap of 943 movies. I also take genome tag information from the Movie Lens dataset and obtain an overlap of 845 movies there. As there are no "gold recommendations" I assume that the recommendations given by gold rating data as the one to compare against. Using the gold rating information, I use a modified KNN algorithm (Koren, 2010) with Pearson's correlation, to obtain top 10 movies per movie. This I consider the true recommendations based on ratings and measure against the top 10 movie list obtained via JMN.

As there are cases in which one has ratings for a movie by some but not all users, (a partially blind scenario), I also compare against top 10 results per movie using CF based methods, in which I do 10 fold cross validation. This is to see if our rating blind method can be useful as compared to partially blind sophisticated recommendation methods using predicted ratings instead of gold ratings. The methods I use are two baseline methods present in Koren (2010), which are similarity measures trying to predict rating of a user and movie, minimising the objective function $\sum_{r_{ui} \in R_{train}} (r_{ui} - (\mu + b_u + b_i))^2 + \lambda (b_u^2 + b_i^2)$ where μ is the average rating, $b_u + b_i$ are the sum of user and movie bias, using SGD (Gardner, 1984) and ALS (Takane et al., 1977) respectively. Aside from the baselines I compare against improved ver-

sion of SVD (Mnih and Salakhutdinov, 2008) which can be seen in Koren et al. (2009), a collaborative filtering algorithm based on Non-negative Matrix Factorization (NMF) (Zhang et al., 2006), and I also compare these results against predicting top 10 movie similarity using genome tags described earlier.

The evaluation metric I use to compare two sets of top N results is the Hit Rate (Deshpande and Karypis, 2004). The Hit Rate is simply the total number of times a recommendation for a movie present in the gold data is present in the recommendation set for the movie generated by the method divided by the total number of recommendations in the gold data. The hit rate does not take into account the position of the recommendation, so I use another metric called the Average Reciprocal Hit Rank (Deshpande and Karypis, 2004) which instead of increasing the count by 1 every time a hit occurs, increases it by a fraction representing the position of the hit in the top 10.

The results of this evaluation can be seen in Table 5.1. JMN beats the naive baselines and is comparable with the NMF, it cannot beat the more sophisticated SVD or the genome tags. Thus, the JMN is useful in the cold start scenario but not if there is partial rating data.

It is useful to have a look at the results being generated by JMN which are not recommended by the ratings based methods to see what it considers as similarity. Some interesting examples of movies the sequence similarity metrics considers close to a movie chosen can be seen in Table 5.2

These recommendations provide some interesting insights. For Fargo this model recommended two other Coen brothers movies without knowing the director,

Method	Hit Rate	ARHR
JMN	0.041	0.014
$Baseline_1$	0.038	0.012
$Baseline_2$	0.046	0.010
NMF	0.040	0.024
SVD	0.305	0.113
Genome	0.260	0.104

Table 5.1: Performance of JMN generated recommendation, blind to ratings, as measured by hit rate and average reciprocal hit rank, against gold ratings based KNN generated recommendations. Compared to this are four, 10 fold cross validated, ratings based methods, and genome tag based recommendations. JMN does better than the two baselines and the NMF method, but worse than the SVD and genome based method, indicating that it cannot compete with SoTA ratings based methods if ratings based recommendations are considered gold, but can be used in cold start scenarios.

Movie	Top 10 recommendations based on JMN prototypes
Fargo	Thunder Heart, Brick, Terminator, Bad Day at Black Rock, The Big Lebowski, The Lady Killer, Fatal Instinct, The Shipping News, Fight Club, Take Shelter
Prometheus	Lost in Space, Aliens, The Abyss, Mission To Mars, Avatar, Ghost Ship, Tron, Pitch Black, Alien, Transformers
Stranger on the Train	Someone Watch Over Me, New York Minute, Dog Day Afternoon, 12 Monkeys, Jimmy and Judy, His Girl Friday, Unknown, Gothika, Frances, Rear Window

Table 5.2: Three examples of top ten movies in the dataset having the longest common sub-sequences of prototypes with the given movie discovered by the joint modelling network

because movies produced by the duo have similar kinds of prototypes which indicates black comedy. The second row has a lot of dark space themed movies, and Prometheus is considered a sequel to the Alien franchise by Ridley Scott, a subtlety the model catches. In the third example, Hitchcock's Rear Window is suggested based on Strangers on a Train. Thus, this kind of discovery of prototypical concepts can be used to understand complicated discourse like a movie script. This work, including the recommendation experiments were done in collaboration with Dr. Ferhan Ture of Comcast Research.

Can JMN predict the tags themselves? This is in interesting question because while it is possible for all movies to be manually tagged for genres if the number is low, there might be certain plot specific qualities users might be interested in for which it will be hard to maintain a list. Movies with chase sequences for example could be a genre, which would not be otherwise tagged. I do another experiment using movie genre data which is obtained from human tagging for each movie in the dataset. We choose the eight most popular genre tags for testing on. Movies can have one or more genres out of these eight. The experiment is whether the archetypes predicted for each movie can correctly predict its genre by training a binary classifier for each. This is an interesting experiment as it can not only help with content based recommendation systems described earlier but also can help with tasks related to context or sentiment. The classifier I use is logarithmic regression along with feature selection, which uses F value between feature and label to choose which ones to use for the final set. The feature set itself is composed of unigrams, bigrams, and trigrams obtained from the descriptions of the movies in terms of

Feature set	F_1 measure per genre							
	Indy	**Drama**	**Thriller**	**Horror**	**Romance**	**Fantasy**	**Mystery**	**Comedy**
$Baseline_1$	0.602	0.774	0.583	0.703	0.568	0.770	0.515	0.613
$Baseline_2$	0.200	0.467	0.423	0.700	0.410	0.700	0.436	0.543
C prots.	**0.630**	**0.799**	0.651	0.754	0.717	0.812	0.618	0.721
R prots.	0.609	0.764	0.583	0.697	0.665	0.788	0.564	0.671
U prots.	0.597	0.790	**0.665**	**0.773**	**0.739**	**0.821**	**0.622**	**0.736**
R+C prots.	**0.630**	**0.799**	0.651	0.754	0.717	0.812	0.618	0.721
U+C prots.	**0.630**	**0.799**	0.651	0.754	0.715	0.816	0.618	0.721
U+R prots.	0.612	0.776	0.587	0.701	0.665	0.789	0.566	0.679
All prots.	**0.630**	**0.799**	0.651	0.754	0.715	0.812	0.619	0.721

Table 5.3: Performance of genre prediction for the various kinds of prototype features used. For a prototype or a set of prototypes in sequences, unigrams, bigrams, and trigrams are made, feature selection is used and the best ones are fed into a LR classifier. The prototypes used can be universal, character arc, or relationships, and any combinations of the three. Only the F_1 measure is used here. The universal prototypes are the best to predict movie genre.

sequences of prototypes, with different experiments for each of the prototype class taken individually or some combination of them. The results for genre prediction can be observed in Table 5.3.

A baseline is needed to compare against, and as there is none available for such a task I create two simple baselines. This first baseline, for a genre, takes its binary classification in the training set and applies it to the test set. That is, if most of movies in the training set are comedies, assume all test set movies are comedies. As the dataset is extremely imbalanced this simple baseline results in extremely high performance. The second baseline takes all the words in a script, filters them through the same process used to create the dataset, and uses all the filtered words as a bag of words model to predict whether or not the genre exists for the movie. For classification it uses an SVM with L1 regularisation with the

best possible parameters. This baseline gives worse results than the majority one. But the model I have created which is trained on n-grams of prototypes gives better results beating both the baselines for all genres. It can be observed that the universal prototypes by themselves are best to predict movie genres.

5.5 Summary

In this chapter I introduce the framework of prototypical concept discovery wherein each sentence in narrative text refers to some latent concept. This is analogous to topic modelling, but with the added constraint of temporal smoothness. I described the only model which does this discovery in a semi supervised manner, but only for one prototype class independent of any other, namely the Relationship Modelling Network, and I extended it into a joint model which uses the hidden layers of the network discovering one prototype class to serve as the history feature of another prototype class. Using this model I deconstructed movie scripts into chains of universal prototypes, character arc prototypes, and relationship prototypes. Using the results thus obtained I found which movies have longest common subsequences of these concepts, and manually examined these, and observed the capacity of these concepts to capture sophisticated notions of similarity. In Chapter 6 I suggest future work of using references to oblique concepts together with references to entities, in a joint manner.

Chapter 6: Conclusion

To conclude this thesis I will summarise all of my contribution and then give a few directions of potential future research.

6.1 Summary

This thesis investigates the problem of reference, in different settings, by using various data sources, machine learning systems and word embeddings. Reference can be about spans of text referring some common entity, the most common kind of coreference resolution problem. Reference can also be about spans of text referring to entities not in an identity, but in an association relation. This is called bridging anaphora resolution. References can be about regions in images, analogous to the visual property of object-hood matching spans in their text descriptions, and finally, in this thesis I expand upon the conventional definition of *reference* from contiguous text spans referencing concrete entities, to distributions of text referencing abstractions or concepts. These concepts are distributed artefacts like themes and prototypes, and discovering which span refers to what concept is also necessary for understanding discourse. In this thesis I cover briefly all these topics and devise data and methods to investigate them.

Hard reference problems of various kinds need some notion of world knowledge to be solved properly. For coreference resolution the existing datasets, being based on newswire, do not reflect these interesting problems which humans are able to solve. I address this by designing a dataset collected from a coreference rich domain which comprises of questions needing humans to do coreference on the fly to solve. I provide the annotation mechanism and the guidelines needed to do so. I address the sub problem of mention detection by using sequence labelling instead of the prevalent rule based approach. To incorporate world knowledge to this problem in a data driven manner, I take the aid of skip-gram learnt word embeddings and discover that it does perform well in this setting compared to existing systems. I also describe the bridging problem, describe existing methods for it, and test if word embeddings are useful for them.

Then, I build a multi-modal dataset of paintings annotated with object contours, which refer to spans of text in questions about them. I treat this like a ranking problem and use bipartite matching to discover which painting is referred to by which question, having obtained coreference chains in the text and gold annotations of objects in the images. I infer visual properties from coreference chains using word embeddings. I demonstrate how this usage of knowledge obtained from text sources in the vision setting is useful. I also describe a large dataset based on comic books with accompanying text. Next, I describe a method which discovers multiple prototypical concepts in a joint fashion by taking an existing neural network model and extending it. I use this model to find which span in a movie script refers to what latent themes of different classes. Having deconstructed the movies

into sequences of prototypes I analyse using subsequence analysis which movies are "similar" to one given, having the same sequence of themes. I also use these sequences to predict movie genre.

6.2 Future Directions of Research

The work in this thesis can be extended in several different directions, both in better methods and in more interesting problems to do with difficult reference.

6.2.1 Reference-Specific Word Embeddings

Throughout this thesis I have used word embeddings to get an idea of "similarity" without rigorously defining what this similarity is. While this worked for these problems, a solution of reference cannot be done without word vectors actually encoding world knowledge, which is only obliquely captured by the concept of neighbourhood. There has been work done recently (Hill et al., 2016) which re-evaluates the concept of similarity (and distinguishes it from the concept of association) for semantic models. This can be used to evaluate improved distributional semantic models. One possible method (Xu et al., 2014) of creating embeddings incorporating more world knowledge is to obtain relational and categorical knowledge from artefacts like knowledge graphs and use that to improve embeddings models. Also, it is possible to design word embeddings specifically for tasks like coreference resolution, by using a cheap rule based resolver on a large enough text corpus, then incorporating the coreference chains into the model to learn word embeddings. These

embeddings then should not be evaluated by similarity metrics but by a reference specific metric, dependent on the task being performed.

6.2.2 Joint Vision-Text Coreference for Atypical Images

The COMICS dataset introduced in this thesis is a source of another research direction. Since the text present in comic textboxes is simple and has direct references to objects in the panels, it should be used to aid comic understanding. The obvious reference task is to detect which are the characters in the comic in vision, segment them, and jointly discover which text box refers to which character (and is spoken by which character). This task is interesting as well as reasonably ambitious, there has been no work yet in multi-modal coreference in atypical images. However, this task becomes very complex because of two reasons. First, comic understanding for humans is very context heavy, as the artist often leaves out significant details between panels for the imagination of the reader. The text being dialogue will also often lack clues to point what entity/character is being referred, because it is commonsense knowledge for the reader. Secondly, the vision part of comics till now how proven somewhat resistant to deep neural networks, as the experiments in Iyyer et al. (2017) which used VGG-16 features demonstrated. The inconsistency in artistic style in the medium also contributes to this problem. To solve the first challenge, external sources of knowledge is required, and to solve the second (as well as any end-to-end system for other atypical images like paintings), models are needed which separate the style of the image from its semantic content. For this past

works on artistic style like Gatys et al. (2016) might be useful in conjunction with transfer learning, as image datasets exist with category/object/scene annotations and perhaps once artistic style is separated domain adaptation be possible.

6.2.3 End to End Reference on Raw Text

The tasks of coreference and bridging resolution are analogous, as they differ only in what kind of antecedents are matched with the anaphor mention. Similarly, there are other kinds of mentions in text as well, and all of these are interrelated. A joint model which does mention detection and at the same time assigns each mention to its category, as well as forms reference links would be better than models which do all these tasks piecemeal. To do this task, several things are required. First, a larger section of OntoNotes needs to be annotated with mediated links information than ISNotes has. This is essential in order to train any neural network model of sophistication, or really using embeddings in conjunction with lexico-semantic features. Second, the model as it will go from mention to mention must be a ranking model rather than a pairwise, and because some of the reference links will be sparse, it should be using cluster level global features like Wiseman et al. (2016). Third, as OntoNotes lacks singleton information, which would be needed to make the mention detection robust, a certain part of the dataset should be annotated with singletons, using which the rest can be detected. Lastly, this should be a joint model with a task like entity linking, in the manner of Durrett and Klein (2014) but unlike that system it should use embeddings to do the linking, thus leveraging Wikipedia for

more world knowledge. Here it should be mentioned that the older ACE dataset can be useful as it has mention class information which OntoNotes lacks. This direction of research is quite ambitious but would result in an end-to-end robust reference framework which is needed for many downstream tasks.

6.2.4 Prototypical Concepts in Text and Vision

In this thesis I built a model to discover which span in narrative text referred to what kinds of prototypical concepts and how those sets of concepts influence each others existence. This problem can be extended to the visual domain. Visual representations, aside from definite objects, also has prototypical concepts which are not apparent. For example, the sepia tone in movie frames indicates a scene from the past, the seemingly chaotic scribbles in a comic panel connotes sound effects or movements of objects in it, or the jagged edges in a piece of contemporary art denotes a specific concept the painter wants to convey. These things are hard to discover by themselves, let alone in a semi supervised fashion, but they can be discovered if the model also takes in text references. For example, if the frames of the movie have corresponding lines of script, or text from the subtitles, the frames themselves are converted to a vector, and along with the text vector are fed into a suitably altered variant of the JMN it is possible to obtain a dictionary of visual themes, each defined by a cloud of words. Similarly, themes from comic book panels can also be discovered given enough panel data and accompanying text. Thus, with the aid of text it may be possible to discover latent concepts hidden in frames or panels

which would aid understanding tasks. This is a novel direction for investigating creative media.

6.2.5 Investigations in Event Coreference

I end this thesis by talking about event coreference, a topic which has been very sparsely covered in the field compared to entity coreference. Event coreference involves finding links between text spans which refer to real world events (this is more of an artefact of datasets, the actual events need not be real world, but the current data is based on newswire). To be specific most of the task boils down to finding correspondence bewtween trigger phrases and the entire event in text. There is only one prominent event coreference dataset, the EventCorefBank (ECB) with 482 texts (and its extension ECB+ with 502 additional texts) (Bejan and Harabagiu, 2010; Cybulska and Vossen, 2014). These texts cover 43 real world topics. Parts of the ACE 2005 dataset and the english part of OntoNotes has event information annotated. More recently the KBP corpus (Mitamura et al., 2015) has been created to address some of the weaknesses of the previous datasets. State of the art event resolutions systems Lu and Ng (2016); Lu et al. (2016) on this corpus have CEAFE F_1 scores in the low 40s indicating significant scope of improvement.

From the point of view of this thesis, event coreference is an interesting route for future research because events, real world or not, are accompanied by multimodal forms of representation and require world knowledge to detect or understand. Also, on a smaller scale events are analogous to actions/activities in more constrained

settings, like events happening in cooking videos described via text, detecting which would be useful for the kind of reference tasks this thesis investigates. Event coreference also has significance for future research on the comic panels as well because certain concepts like event triggers carry over to that setting. Also, current research on event coreference lacks neural network approaches, and thus there is the opportunity of trying joint models with more mature architectures.

Bibliography

Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *International Language Resources and Evaluation*. Citeseer, 1998.

David Bamman, Brendan O'Connor, and Noah A Smith. Learning latent personas of film characters. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 352, 2014a.

David Bamman, Ted Underwood, and Noah A Smith. A bayesian mixed effects model of literary character. In *ACL (1)*, pages 370–379, 2014b.

Mohit Bansal and Dan Klein. Coreference semantics from web features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 389–398. Association for Computational Linguistics, 2012.

Chumki Basu, Haym Hirsh, William Cohen, et al. Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720, 1998.

Cosmin Adrian Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422. Association for Computational Linguistics, 2010.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

Anders Björkelund and Richárd Farkas. Data-driven multilingual coreference resolution using resolver stacking. In *Conference on Computational Natural Language Learning*, 2012.

Anders Björkelund and Jonas Kuhn. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the Association for Computational Linguistics*, 2014.

Anders Björkelund, Kerstin Eckart, Arndt Riester, Nadja Schauffler, and Katrin Schweitzer. The extended dirndl corpus as a resource for automatic coreference and bridging resolution. In *International Language Resources and Evaluation*, pages 3222–3228, 2014.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

A. Boyd, P. Stewart, and R. Alexander. *Broadcast Journalism: Techniques of Radio and Television News*. Taylor & Francis, 2008. ISBN 9781136025853. URL http://books.google.com/books?id=523XNicUFl8C.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daume III. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of Empirical Methods in Natural Language Processing*, 2012.

Daniel Büring. *Binding theory*. Cambridge University Press, 2005.

Claire Cardie, Kiri Wagstaff, et al. Noun phrase coreference as clustering. In *Proceedings of the Joint Sigdat Conference on empirical methods in natural language processing and very large corpora*, pages 82–89, 1999.

Marianella Casasola. Can language do the driving? the effect of linguistic input on infants' categorization of support spatial relations. *Developmental psychology*, 41 (1):183, 2005.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. Modeling evolving relationships between characters in literary novels. In *Association for the Advancement of Artificial Intelligence*, pages 2704–2710, 2016.

Xinlei Chen and C Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Herbert H Clark. Bridging. In *Proceedings of the 1975 workshop on theoretical issues in natural language processing*, pages 169–174. Association for Computational Linguistics, 1975.

Herbert H Clark and Catherine Marshall. Reference diaries. In *Proceedings of the 1978 workshop on Theoretical issues in natural language processing*, pages 57–63. Association for Computational Linguistics, 1978.

Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *International Language Resources and Evaluation*, pages 4545–4552, 2014.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *ACL (1)*, pages 795–804, 2015.

Hal Daumé III and Daniel Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 97–104. Association for Computational Linguistics, 2005.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *LREC*, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

Xiaowen Ding and Bing Liu. Resolving object and attribute coreference in opinion mining. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 268–276. Association for Computational Linguistics, 2010.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The automatic content extraction (ACE) program-tasks, data, and evaluation. In *International Language Resources and Evaluation*, 2004.

Pedro Domingos and Daniel Lowd. Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155, 2009.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014.

Wenwen Dou, Li Yu, Xiaoyu Wang, Zhiqiang Ma, and William Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2002–2011, 2013.

Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of Empirical Methods in Natural Language Processing*, 2013.

Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2014.

Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3): 361–365, 1996.

Will Eisner. Comics & sequential art. 1985.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on Graphics (Proceedings of Special Interest Group on Computer GRAPHics and Interactive Techniques, SIGGRAPH)*, 31(4):44:1–44:10, 2012.

Vinodh Krishnan Elangovan and Jacob Eisenstein. You're Mr. Lebowski, I'm the Dude: Inducing address term formality in signed social networks. In *Conference of the North American Chapter of the Association for Computational Linguistics*. The Association for Computational Linguistics, 2015.

George Engelbretsen. Denotation and reference. *Philosophical Studies*, 27:229–236, 1980.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of Empirical Methods in Natural Language Processing*, 2012.

Lucie Flekova and Iryna Gurevych. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1805–1816, 2015.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, H Nicolov, and Salim Roukos. A statistical model for multilingual entity detection and tracking. Technical report, IBM Thomas J Watson Rsesearch Center Yorktown Heights NY, 2004.

Radu Florian, Hongyan Jing, Nanda Kambhatla, and Imed Zitouni. Factorizing complex models: A case study in mention detection. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics, 2006.

William A Gardner. Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal processing*, 6(2):113–133, 1984.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

Peter Geibel and Fritz Wysotzki. Perceptron based learning with example dependent and noisy costs. In *Proceedings of the International Conference of Machine Learning*, pages 218–225, 2003.

Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature reviews. Neuroscience*, 14 5:350–63, 2013.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520. IEEE, 1992.

N. Goldstein and A. Press. *The Associated Press Stylebook and Briefing on Media Law*. Associated Press Stylebook and Briefing on Media Law. Basic Books, 2004. URL http://books.google.com/books?id=8wtW0p2wZVoC.

Brynjar Gretarsson, John Odonovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):23, 2012.

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic markov models. In *Artificial intelligence and statistics*, pages 163–170, 2007.

Clément Guérin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karell Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier, and Arnaud Revel. ebdtheque: a representative database of comics. In *Document Analysis and Recognition (ICDAR), 2013 12th international conference on*, pages 1145–1149. IEEE, 2013.

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. Removing the training wheels: A coreference dataset that entertains humans and challenges computers. In *North American Association for Computational Linguistics*, 2015.

Anupam Guha, Mohit Iyyer, and Jordan Boyd-Graber. A distorted skull lies in the bottom center: Identifying paintings from text descriptions. In *NAACL Human-Computer Question Answering Workshop*, 2016.

Jan Hajič, Eva Hajičová, Marie Mikulová, and Jiří Mírovskỳ. Prague dependency treebank. In *Handbook of Linguistic Annotation*, pages 555–594. Springer, 2017.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2016.

Jerry R Hobbs. World knowledge and word meaning. In *Proceedings of the 1987 workshop on Theoretical issues in natural language processing*, pages 20–27. Association for Computational Linguistics, 1987.

Yufang Hou, Katja Markert, and Michael Strube. Cascading collective classification for bridging anaphora recognition using a rich linguistic feature set. In *Proceedings of Empirical Methods in Natural Language Processing*, 2013a.

Yufang Hou, Katja Markert, and Michael Strube. Global inference for bridging anaphora resolution. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2013b.

Yufang Hou, Katja Markert, and Michael Strube. A rule-based system for unrestricted bridging resolution: Recognizing bridging anaphora and finding links to antecedents. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2082–2093, 2014.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90% solution. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2006.

Zhichao Hu, Elahe Rahimtoroghi, Larissa Munishkina, Reid Swanson, and Marilyn A Walker. Unsupervised induction of contingent event pairs from film scenes. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 369–379, 2013.

Zan Huang, Daniel Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 22(5), 2007.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139. Association for Computational Linguistics, 2007.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A neural network for factoid question answering over paragraphs. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*, 2015.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *North American Association for Computational Linguistics*, 2016.

Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daumé III, and Larry Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. Construction of a japanese relevance-tagged corpus. In *International Language Resources and Evaluation*, 2002.

Natasha Z Kirkham, Jonathan A Slemmer, and Scott P Johnson. Visual statistical learning in infancy: Evidence for a domain general learning mechanism. *Cognition*, 83(2):B35–B42, 2002.

Chen Kong, Dahua Lin, Mayank Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3558–3565. IEEE, 2014.

Yehuda Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1, 2010.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

Eileen Kowler. Cognitive expectations, not habits, control anticipatory smooth oculomotor pursuit. *Vision research*, 29(9):1049–1057, 1989.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Jonathan K Kummerfeld, Mohit Bansal, David Burkett, and Dan Klein. Mention detection: heuristics for the ontonotes annotations. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 102–106. Association for Computational Linguistics, 2011.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Thomas K Landauer. *Latent semantic analysis*. Wiley Online Library, 2006.

Emmanuel Lassalle and Pascal Denis. Leveraging different meronym discovery methods for bridging resolution in french. *Anaphora Processing and Applications*, pages 35–46, 2011.

Florian Laws, Florian Heimerl, and Hinrich Schütze. Active learning for coreference resolution. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2012.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Conference on Computational Natural Language Learning*, 2011.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2657–2664. IEEE, 2014.

Grace I Lin and Marilyn A Walker. All the world's a stage: Learning character models from film. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainm*, 2011.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

Jing Lu and Vincent Ng. Event coreference resolution with multi-pass sieves. In *International Language Resources and Evaluation*, 2016.

Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. Joint inference for event coreference resolution. In *Proceedings of International Conference on Computational Linguistics*, pages 3264–3275, 2016.

Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of Empirical Methods in Natural Language Processing*, 2005.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

Katja Markert, Malvina Nissim, and Natalia N Modjeska. Using the web for nominal anaphora resolution. In *EACL Workshop on the Computational Treatment of Anaphora*, volume 3946, 2003.

Katja Markert, Yufang Hou, and Michael Strube. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 795–804. Association for Computational Linguistics, 2012.

Sebastian Martschat and Michael Strube. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418, 2015.

Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, pages 1–28, 2016.

Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*, 2012.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

Scott McCloud. Understanding comics: The invisible art. *Northampton, Mass*, 1993.

Alexius Meinong. On the theory of objects (translation of'über gegenstandstheorie', 1904). 1960.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Timothy A Miller, Dmitriy Dligach, and Guergana K Savova. Active learning for coreference resolution. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*. Proceedings of the Association for Computational Linguistics, 2012.

Shachar Mirkin, Ido Dagan, and Sebastian Padó. Assessing the role of discourse references in entailment inference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1209–1219. Association for Computational Linguistics, 2010.

Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. Overview of tac kbp 2015 event nugget track. In *Text Analysis Conference*, 2015.

Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

Thomas S Morton. Using coreference for question answering. In *Proceedings of the Workshop on Coreference and its Applications*, pages 85–89. Association for Computational Linguistics, 1999.

MUC-6. Coreference task definition (v2.3, 8 sep 95). In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

MUC-7. Coreference task definition (v3.0, 13 jun 97). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1997.

Judith Muzerelle, Anaïs Lefeuvre, Emmanuel Schang, Jean-Yves Antoine, Aurore Pelletier, Denis Maurel, Iris Eshkol, and Jeanne Villaneau. Ancor_centre, a large free spoken french coreference corpus: description of the resource and reliability measures. In *International Language Resources and Evaluation*, 2014.

Anna Nedoluzhko, Jiří Mírovskỳ, Radek Ocelák, and Jiří Pergler. Extended coreferential relations and bridging anaphora in the prague dependency treebank. In *Proceedings of the 7th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2009), Goa, India*, pages 1–16, 2009.

Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the Association for Computational Linguistics*, 2010.

Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the Association for Computational Linguistics*, 2002.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, Deborah A Cai, Jennifer E Midberry, and Yuanxin Wang. Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, 95(3):381–421, 2014.

Malvina Nissim. Learning information status of discourse entities. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 94–102. Association for Computational Linguistics, 2006.

Malvina Nissim, Shipra Dingare, Jean Carletta, and Mark Steedman. An annotation scheme for information status in dialogue. In *International Language Resources and Evaluation*, 2004.

Maciej Ogrodniczuk, Katarzyna Głowińska, Mateusz Kopeć, Agata Savary, and Magdalena Zawisławska. Polish coreference corpus. In *Language and Technology Conference*, pages 215–226. Springer, 2013.

Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. A joint framework for coreference resolution and mention head detection. *Conference on Computational Natural Language Learning*, 51:12, 2015.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

Massimo Poesio. The mate/gnome proposals for anaphoric annotation, revisited. In *SIGDIAL Workshop*, pages 154–162, 2004.

Massimo Poesio and Renata Vieira. A corpus-based investigation of definite description use. *Computational linguistics*, 24(2):183–216, 1998.

Massimo Poesio, Tomonori Ishikawa, Sabine Schulte Im Walde, and Renata Vieira. Acquiring lexical knowledge for anaphora resolution. In *International Language Resources and Evaluation*, 2002.

Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 143. Association for Computational Linguistics, 2004.

Thierry Poibeau and Leila Kosseim. Proper name extraction from non-journalistic texts. *Language and computers*, 37(1):144–157, 2001.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. CoNLL-2011 shared task: Modeling unrestricted coreference in Ontonotes. In *Conference on Computational Natural Language Learning*, 2011.

Sameer S Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing*, 1(04), 2007.

Willard Van Orman Quine, Patricia S Churchland, and Dagfinn Føllesdal. *Word and object*. MIT press, 2013.

Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 968–977. Association for Computational Linguistics, 2009.

Altaf Rahman and Vincent Ng. Learning the fine-grained information status of discourse entities. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 798–807. Association for Computational Linguistics, 2012.

Vignesh Ramanathan, Percy Liang, and Li Fei-Fei. Video event understanding using natural language descriptions. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 905–912. IEEE, 2013.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Conference on Computational Natural Language Learning*, 2009.

Ina Rösiger. Scicorp: A corpus of english scientific articles annotated for information status analysis. In *International Language Resources and Evaluation*, 2016.

Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, Minnesota Univ Minneapolis Dept of Computer Science, 2000.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

Hinrich Schütze. Word space. In *Advances in neural information processing systems*, pages 895–902, 1993.

Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

Gaurav Sood. *clarifai: R Client for the Clarifai API*, 2017. R package version 0.4.2.

Shashank Srivastava, Snigdha Chaturvedi, and Tom M Mitchell. Inferring interpersonal relations in narrative summaries. In *Association for the Advancement of Artificial Intelligence*, pages 2807–2813, 2016.

Yoshio Takane, Forrest W Young, and Jan De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. Ancora: Multilevel annotated corpora for catalan and spanish. In *International Language Resources and Evaluation*, 2008.

Heike Telljohann, Erhard Hinrichs, Sandra Kübler, and Ra Kübler. The tüba-d/z treebank: Annotating german with a context-free backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Citeseer, 2004.

Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

Olga Uryupina. Coreference resolution with and without linguistic knowledge. In *International Language Resources and Evaluation*, 2006.

Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. Bart: A modular toolkit for coreference resolution. In *Proceedings of the Association for Computational Linguistics*, 2008.

Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):13, 2012.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the conference on Message understanding*, pages 45–52, 1995.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

Marilyn Walker, Ricky Grant, Jennifer Sawyer, Grace Lin, Noah Wardrip-Fruin, and Michael Buell. Perceived or not perceived: Film character models for expressive nlg. *Interactive Storytelling*, pages 109–121, 2011.

Sam Wiseman, Alexander M Rush, Stuart M Shieber, and Jason Weston. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the Association for Computational Linguistics*, 2015.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*, 2016.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1219–1228. ACM, 2014.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 702–709. IEEE, 2012.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from video described with sentences. In *ACL (1)*, pages 53–63, 2013.

Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 549–553. SIAM, 2006.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.