MULTIMODAL MACHINE COMPREHENSION OF HOW-TO INSTRUCTIONS WITH IMAGES AND TEXT

GÖRÜNTÜ VE METİN İÇEREN ÇOK KİPLİ NASIL YAPILIR TALİMATLARININ MAKİNE İLE KAVRANMASI

SEMİH YAĞCIOĞLU

ASSOC. PROF. DR. MEHMET ERKUT ERDEM
Supervisor
ASSOC. PROF. DR. İBRAHİM AYKUT ERDEM
2nd Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Doctor of Philosophy
in Computer Engineering

2023

ABSTRACT

MULTIMODAL MACHINE COMPREHENSION OF HOW-TO INSTRUCTIONS WITH IMAGES AND TEXT

Semih Yağcıoğlu

Doctor of Philosophy , Computer Engineering

Supervisor: Assoc. Prof. Dr. Mehmet Erkut Erdem

2nd Supervisor: Assoc. Prof. Dr. İbrahim Aykut Erdem

2023, 160 pages

In the blink of an eye, we understand what we are looking at. Most of our brain is organized to process the visual information we receive; thus, replicating human intelligence requires a complete understanding of human vision. But, is understanding vision enough to understand human intelligence? Probably not. Besides our visual perception skills, language is an essential and unique ability and a natural way of communication for humans.

For thousands of years, humankind has been telling stories and giving instructions through spoken language. One of the earliest written forms of language is instructions, specifically food recipes. These instructions not only help us understand what the people of that time ate but also teach us how they used to live their lives. Instructions have been around us for centuries, be it in the form of recipes, or how-to guides, written on stone tablets or books, or else published on the web. How-to instructions with images and text are perfect candidates for understanding human intelligence, and understanding them is an important, intriguing research problem to solve. Modern how-to guides of any form almost always contain multimodal information, such as images, videos, and text. Instructions are key to understanding and replicating a process, and how-to guides are great sources of instruction, as we can replicate the same process by just following the guide. Furthermore, how-to instructions often involve a joint understanding of multiple modalities of information *e.g.* images and text. However, they are also very challenging to understand as they often contain multimodal information such as images and text, consist of multiple objects and entities as well as require a procedural understanding of actions and interactions between such entities often referred in from one modality into another. How-to guides such as cooking recipes, typically consist of multiple steps involving various objects and entities, most of which interact with each other through different actions. Considering an action as a combination of a verb and an object or entity, being able to generalize to unseen compositions of these action compounds pose a great challenge. In this regard, understanding how visually grounded textual instructions might help models' systematic generalization abilities remains an important research problem.

In this thesis, we examine multimodal machine comprehension of how-to instructions with images and text, review related literature, and point out current challenges. We also propose methods to address some of these challenges and ways to improve upon existing approaches. The main contributions of this thesis can be summarized as follows. We investigate machine comprehension and reasoning problems and review the previous literature to lay the grounds for understanding multimodal how-to instructions. We survey compositional generalization literature, highlight current research challenges, and discuss its relation to understanding multimodal how-to instructions. We introduce a multimodal benchmark how-to instructions dataset comprised of cooking recipes with images and text. We propose novel methods for understanding multimodal procedures. Finally, we present a challenging multimodal compositional generalization setup and propose methods to benchmark and show multimodality's contribution to significantly improve the current state of the art in understanding multimodal how-to instructions and conclude with future research directions and discuss open challenges.

Keywords: machine comprehension and reasoning, how-to instructions, multimodality, procedural understanding, compositionality

ÖZET

GÖRÜNTÜ VE METİN İÇEREN ÇOK KİPLİ NASIL YAPILIR TALİMATLARININ MAKİNE İLE KAVRANMASI

Semih Yağcıoğlu
Doktora, Bilgisayar Mühendisliği
Danışman: Doç. Dr. Mehmet Erkut Erdem
Eş Danışman: Doç. Dr. İbrahim Aykut Erdem
2023, 160 sayfa

Göz açıp kapayıncaya kadar, neye baktığımızı anlıyoruz. Beynimizin büyük bir kısmı, aldığımız görsel bilgileri işlemek için organize edilmiştir; bu nedenle, insan zekasını taklit etmek, görmenin tam olarak anlaşılmasını gerektirir. Ancak görmeyi anlamak, insan zekasını anlamak için yeterli midir? Muhtemelen değil. Dil, görsel algı becerimizin yanı sıra, insanlar için vazgeçilmez ve eşsiz bir yetenek ve doğal bir iletişim biçimidir.

İnsanoğlu binlerce yıldır konuşma diliyle hikayeler anlatmakta ve talimatlar vermektedir. Dilin en eski yazılı biçimlerinden biri talimatlar olup özellikle yemek tarifleri talimatların arasında ön plana çıkmaktadır. Bu talimatlar bizlere sadece o zamanın insanlarının ne yediğini anlamamıza yardımcı olmakla kalmaz, aynı zamanda hayatlarını nasıl yaşadıklarını da öğretir. İster tarifler şeklinde, ister nasıl yapılır yönergeleri şeklinde, ister taş tabletlere veya kitaplara yazılmış olsun, ister web'de yayınlanmış olsun, talimatlar yüzyıllardır etrafımızdadırlar. Görüntü ve metin içeren nasıl yapılır yönergeleri, insan zekasını anlamak için mükemmel adaylar olmakla birlikte, bunları anlamak, çözülmesi gereken önemli, merak uyandıran bir araştırma problemidir. Herhangi bir biçimdeki modern nasıl yapılır

yönergeleri, neredeyse her zaman resimler, videolar ve metin gibi çok kipli bilgiler içerir. Talimatlar, bir süreci anlamanın ve çoğaltmanın anahtarıdırlar ve nasıl yapılır yönergeleri, yalnızca yönergeleri izleyerek aynı süreci tekrarlayabileceğimiz için harika talimat kaynaklarıdır. Ayrıca, nasıl yapılır yönergeleri genellikle, örneğin görüntü ve metin gibi birden çok kipin ortak bir şekilde anlaşılmasını içerir. Bununla birlikte, bu talimatlar, genellikle görüntüler ve metin gibi çok kipli bilgiler içerdiklerinden, birden çok nesne ve varlıktan oluştuklarından ve genellikle bir kipten diğerine atıfta bulunulan bu tür varlıklar arasındaki eylemler ve etkileşimlerin yordamsal olarak anlaşılmasını gerektirdiğinden, anlaşılması da oldukça zordur. Yemek tarifleri gibi nasıl yapılır yönergeleri, tipik olarak, çoğu farklı eylemler yoluyla birbiriyle etkileşime giren çeşitli nesneleri ve varlıkları içeren birden çok adımdan oluşur. Bir eylemi, bir fiil ile bir nesne veya varlığın birleşimi olarak ele aldığımızda, bu eylemi oluşturan parçaların daha önce gözlemlenmemiş bileşimlerine genelleme yapabilmek büyük bir zorluk teşkil etmektedir. Bu bağlamda, görsel temelli metinsel yönergelerin, makine öğrenmesi modellerinin sistematik genelleme becerilerine nasıl yardımcı olabileceğini anlamak önemli bir araştırma sorunu olmaya devam etmektedir.

Son yıllarda yapay zeka araştırmalarına giderek artan bir ilgi oluşmuştur. Özellikle, büyük ölçekli veri kümelerinin önerilmesi, araştırmacıları daha karmaşık modeller geliştirmeye motive etmiştir. Bilgisayarlı görü ve doğal dil işlemele alanlarında, örneğin görüntülerin tasviredilmesi ve görsel soruları yanıtlama gibi görevler bir çok araştırmacının ilgisini çekmiş ve bu problemler üzerinde pek çok çalışma yapılmıştır. Modellerin kalitesi zaman içinde istikrarlı bir şekilde iyileşse de, bu görevlerin doğası gereği, hem görüntü hem de metin alanlarında sorunların ortaklaşa çözülmesini gerektiren çeşitli zorluklar ortaya çıkmıştır.

Bu doğrultuda ön plana çıkan ve araştırmacıların ilgisini çeken konuların başında kavrama ve muhakeme problemleri dikkat çekmektedir. Geniş bir bağlamda bakacak olursak kavrama, bir şeyin ne anlama geldiğini idrak etme yeteneğini, muhakeme ise bilinen gerçeklerden sonuç çıkarmayı ifade etmektedir. Kavrama ve muhakeme yetenekleri üst düzey bilişsel beceriler olmakla birlikte, sadece makineler için değil insanlar için de oldukça zorlayıcı

problemler olarak kabul görmektedir. Her halükarda, makinelerin insan düzeyinde kavrama ve akıl yürütmeye ulaşması için halihazırda oldukça önemli eksikler bulunmaktadır.

Bu tez, hem görseller, hem de metinler için makine kavraması ve makine muhakemesi problemlerini incelemekte ve çok kipli muhakame ve kavrama problemlerini kapsamlı bir şekilde ele almaktadır. Ayrıca görseller ve metinleri anlamak ve akıl yürütmek için literatürde öne çıkan veri kümelerini incelemekle birlikte, daha önce önerilmiş görevleri genel bir bakışla ele almaktadır. Bu tezde, özellikle, makinelere okumayı öğretmeyi ve ardından bu konuda sorular sorarak okuduklarını anlamalarını sorgulamayı amaçlayan makine okuması ve anlaması görevlerini incelemekte, aynı zamanda görsel akıl yürütme alanında muhakeme ve çıkarım üzerine odaklanmakta, ayrıca çok kipli nasıl yapılır talimatlarının makine muhakemesi ve makine kavranması ile idrak edilmesi üzerinde çalışmaktayız. Bu problemlerin çözümü, tek kip ile dahi düşünüldüğünde oldukça zorlu problemler teşkil etmektedir. Örneğin, makine okuduğunu anlama testleri, makinelerin verilen metni ne kadar iyi anladıklarını, soruları ne kadar doğru yanıtladıklarını ölçerek değerlendirmeyi amaçlar. Görsel akıl yürütme, sahne, aktörler, varsa eylemler, bağlam, nesneler ve bunların bir biri ile olan etkileşimleri gibi görsel verilerin derinlemesine anlaşılmasını gerektirir. Görsel verileri anlama konusunda onlarca yıllık araştırma yapılmıştır. Görsel muhakeme ve anlama tipik olarak insanlarla, yani doğal dil yoluyla etkileşimi gerektirdiğinden, bu görev pratik olarak görüntüler veya videolar ve metin gibi çok kipli verileri aynı anda ele almayı gerektirmektedir. Tipik bir görsel muhakeme görevinde, bir makinenin görsel verileri anlaması, bu görsel ile ilgili sorulan soruyu kavraması ve son olarak görsel veriler ve soru bağlamında doğru bir cevap vermesi beklenir. Bu bağlamda, görüntü ve metinlerden oluşan nasıl yapılır talimatlarının makine kavranması ve muhakemesi oldukça önemli ve bir o kadar zor problemler olarak öne çıkmaktadır.

İnsanlar önceden bildikleri kavramların yeni bileşimleri ile ilk kez karşılaşsalar bile kolayca anlayabilmekte, bildikleri kavram ve nesneleri zahmetsizce bir araya getirerek yeni bileşimler oluşturabilmektedirler. Bu bağlamda, bileşimsel genelleme son yıllarda araştırmacıların ilgisini çekmekle birlikte, bu problem, çok kipli nasıl yapılır talimatlarının kavranması ve muhakemesi için oldukça önem arz etmektedir. Bu tezde, dilbilimsel bileşimler üzerine mevcut çalışmaların kapsamlı bir incelemesini sunmakta, mevcut görevleri ve veri kümelerini sınıflandırarak tartışmaktayız. Ayrıca, sinir ağı mimarileri ve bileşimsel genelleme için önerilen öğrenme stratejilerini incelemekte, ve mevcut görevleri, veri kümelerini, yöntem ve öğrenme stratejilerini tartışarak bu alanda önerilmiş çalışmaları kapsamlı bir şekilde ele almaktayız. Bununla birlikte, sistematik genelleme alanındaki mevcut kısıtlara vurgu yaparak ve gelecekteki muhtemel araştırma istikametlerini tartışıyor ve çok kipliliğin sistematik bileşime katkısını inceliyoruz.

Yemek tariflerini anlamak ve akıl yürütmek, makinelerin yordamsal metinleri yorumlamasını sağlamaya yönelik önemli bir araştırma alanıdır. Bu doğrultuda, bu tezde ayrıca, yemek tariflerinin çok kipli olarak anlaşılması için yeni bir veri kümesi olan RecipeQA veri kümesini sunmaktayız. Sunduğumuz bu veri kümesi, kendi içerisinde başlıklar, açıklamalar ve hizalanmış görüntüler gibi birden çok kipe sahip eğitici yemek tariflerini içermektedir. RecipeQA veri kümesi üzerinde hem tek kipli hem de çok kipli modeller önermekle birlikte, metinsel boşluk doldurma, görsel boşluk doldurma, görsel sıralama, görsel uyum gibi çeşitli muhakeme ve kavrama görevleri üzerinde farklı modeller ile deneyler gerçekleştirmekteyiz. Elde ettiğimiz sonuçlar, RecipeQA veri kümesinin zorlu bir test ortamı ve makine anlama sistemlerini değerlendirmek için ideal bir kıyaslama veri kümesi olarak hizmet edeceğini göstermektedir.

Yemek tariflerinin yordamsal olarak anlaşılması, nesnelerin kavranmasını, durum değişikliklerinin izlenmesini ve zamansal ve nedensel ilişkilerin anlaşılmasını gerektirdiğinden oldukça zorlu bir görevdir. RecipeQA veri kümesini tanıttıktan sonra, farklı bir problem olarak yordamsal ortak akıl bilgisini anlama problemini araştırmaktayız. Özellikle, RecipeQA veri kümesinden yararlanan çok kipli yordamsal bilgiyi anlamak için Yordamsal Muhakeme Ağlarını (PRN) öneriyor, buna ek olarak, metinsel kiplere tamamlayıcı bir anlamsal sinyal sağlamak için görsel kiplerden nasıl faydalanılabileceği sorusunu araştırıyoruz. Önermekte olduğumuz bu model, metin talimatlarını okurken birbiriyle ilişkili varlık durumlarını dinamik olarak güncellemeyi öğrenmektedir. Ayrıca, daha önce önerdiğimiz RecipeQA veri kümesindeki görsel muhakeme görevleri üzerine bir analiz sunmakta, yordamsal çok kipli nasıl yapılır talimatlarının anlaşılması için sunduğumuz bu yaklaşımımızın, daha önce elde elde ettiğimiz sonuçların doğruluğunu büyük bir farkla geliştirdiğini görmekteyiz.

Sinir ağı modelleri, pek çok farklı görevde etkileyici bir şekilde iyi performans gösterir, ancak genellikle daha önce gözlemlenmemiş kavramların bileşimsel olarak genelleştirmesinde başarısız olmaktadırlar. Bu doğrultuda, günlük ev görevlerinden oluşan bir veri kümesini kullanarak görsel ve metinsel bilgiye dayalı talimatlardan oluşan bir bileşimsel ve sistematik veri seti olan EK-100-SYS veri setini önermekte, bu verini kullanarak, sistematik genelleme problemini kapsamlı bir şekilde ele almaktayız. Ayrıca, bu çalışma kapsamında, bilinen kavramlardan yeni bileşimler içeren bir eylemi tahmin etmeyi amaçlayan bir görev ve eylem sınıfını tahminleme görevlerinde birkaç tek kipli ve çok kipli modeller de sunmaktayız. Elde ettiğimiz bulgular, görsel ve işitsel sinyallerden yararlanan modellerin, salt metin tabanlı modellere göre belirtilen görevlerde daha iyi sonuçlar elde edilmesine imkan sunduğunu göstermekte, bu bağlamda çok kipliliğin bileşimsel genelleme probleminde önemli bir katkı sunabileceğini göstermekteyiz.

Son olarak, bu tezde görüntü ve metinlerden oluşan çok kipli nasıl yapılır talimatlarının kavranması ve muhakemesi konusunda temel kısıtları ele almakta, gelecekte bu alanda yapılacak çalışmalara yön vermek için muhtemel araştırmaları istikametlerini belirtmekteyiz.

Keywords: Makine muhakemesi ve makine kavraması, nasıl yapılır talimatları, çok kiplilik, yordamsal anlama, bileşimsellik

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

RNN	:	**R**ecurrent **N**eural **N**etworks
CNN	:	**C**onvolutional **N**eural **N**etworks
LSTM	:	**L**ong **S**hort **T**erm **M**emory
GRU	:	**G**ated **R**ecurrent **U**nits
DL	:	**D**eep **L**earning
RC	:	**R**eading **C**omprehension
MRC	:	**M**achine **R**eading **C**omprehension
VR	:	**V**isual **R**easoning
MMC	:	**M**ultimodal **M**achine **C**omprehension
NLP	:	**N**atural **L**anguage **P**rocessing
CV	:	**C**omputer **V**ision
NLQ	:	**N**atural **L**anguage **Q**uestions
NLU	:	**N**atural **L**anguage **U**nderstanding
RGBD	:	**R**ed **G**reen **B**lue **D**epth
MT	:	**M**achine **T**ranslation
NMT	:	**N**eural **M**achine **T**ranslation
PRN	:	**P**rocedural **R**easoning **N**etworks
QA	:	**Q**uestion **A**nswering
VQA	:	**V**isual **Q**uestion **A**nswering
AA	:	**A**ction **A**nticipation
FUP	:	**F**uture **U**tterance **P**rediction

1. INTRODUCTION

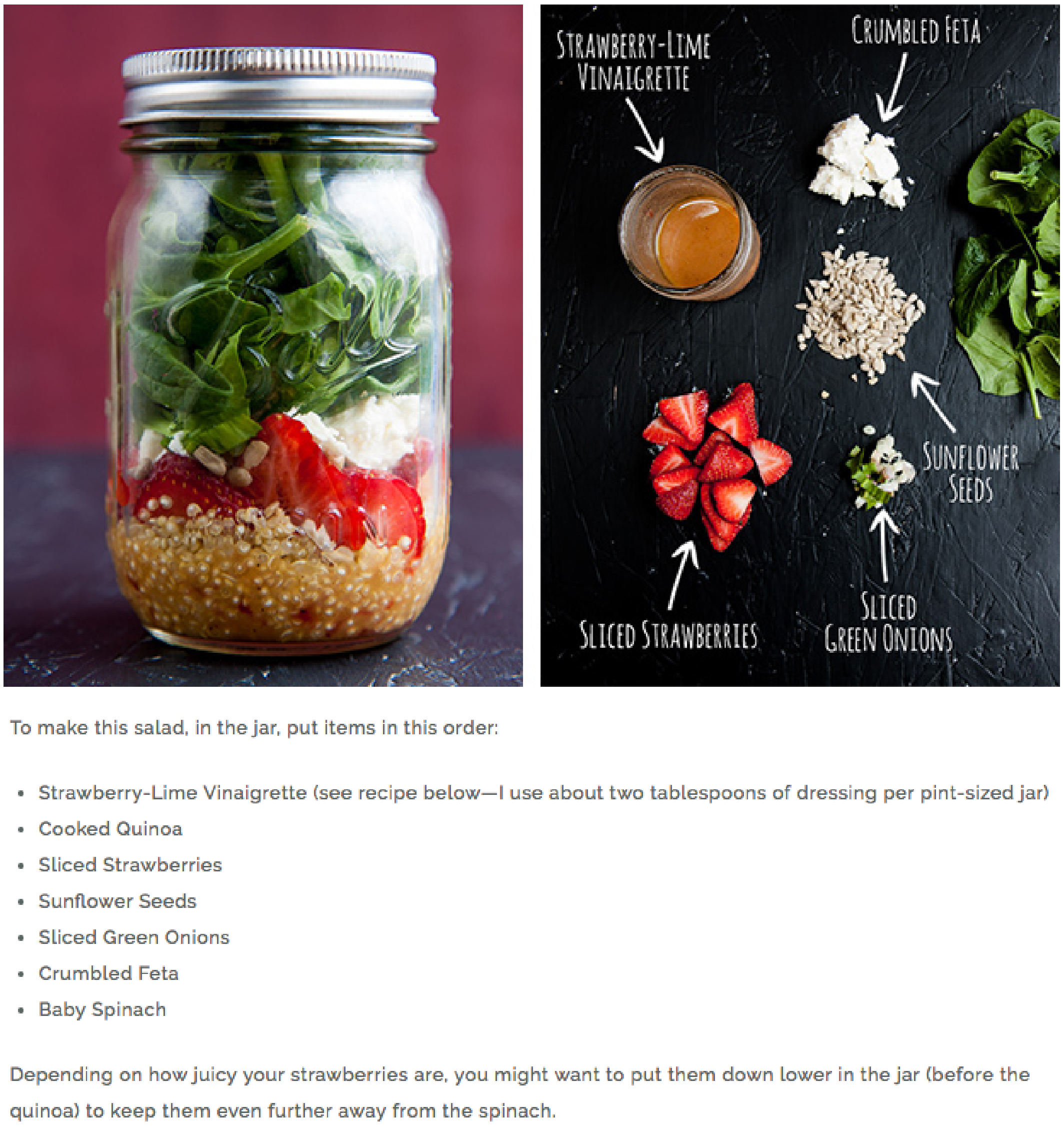
In order to understand what is going on around us, we just need to look for merely a second. In this short amount of time, the visual information flows through our eyes and through the lateral geniculate nucleus, a relay center located in the thalamus, and then reaches the visual cortex. The first cortical areas decode lines, edges, and contours. The parietal cortex takes the visual signals to locate where things are in 3-dimensional space. The inferotemporal cortex recognizes shapes, objects, and faces. In the blink of an eye, we understand what we are looking at. Humans, and in general, animals, are very good at understanding what they see. Seeing is merely a thing for us, we only need to look in order to see as it comes off the shelf. Nonetheless, this is not the case for machines. For them to recognize what they see, one obvious thing to do is to replicate animal vision. This, however, is not enough to understand our visual world but is an important milestone for machines. In order to understand what they see, machines have to learn, through vigorous training, of shapes, edges, geometry, scale, occlusion, time and context, and maybe and more importantly, their combinations, just like humans. There have been decades of work to understand the vision. One of the earliest works in understanding vision is Marr's approach to treating vision as an information processing system. In his work, Marr proposed three levels for a machine that carries out an information processing system, such as a vision task [1]. These three levels are computational theory, representation and algorithm, and hardware implementation, each of which refers to what a system does, how does it do what it does, and how the system is physically realized, respectively. Marr also proposed a representational framework for vision with three stages. In the first stage, that is, the primal sketch, edges, bars, ends, and blobs are represented. In the second stage, which is a 2.5D sketch, the textures are acknowledged. Finally, in the third stage, a continuous 3d map of the scene is used for visualization. Marr's theory kindled interest in understanding how our visual system works, along with a framework to process visual data. Visual perception and understanding how our visual system is important, as most of our brain is organized to process the visual information we receive, and thus, replicating human intelligence requires a complete understanding of the human vision. But is understanding vision enough to understand human intelligence? Probably not. Besides our visual perception skills, language is also an important and unique ability and a natural way of communication for humans. Vision is about how we see the world, but language is about how we communicate and interact with it. Among other things, language is maybe what makes the human species superior in the whole animal kingdom. Although there is no consensus on how, why, when, and where language might have emerged [2], there are about 7000 languages spoken around the world [3]. Languages emerge, live, and die with people. Shaped with historical events, languages capture almost every aspect of human life at the time they are being spoken. Although languages have been with us for thousands of years, humans started writing much later. With the invention of written language, humans have started to record everything around them. Through different mediums, we have produced and captured knowledge for thousands of years in written form. This leads to an accumulation of human knowledge, and today we are generating more than ever, adding up to that knowledge. Representing a language through graphic means, *i.e.* writing enables readers of that language to reconstruct the encoded contents. Reading, in this regard, is a very important tool for humans to acquire information and comprehend what is written. Understanding language is a very challenging task, as it often takes several years for an infant to learn a language from scratch. For thousands of years, mankind has been telling stories and giving instructions through spoken language. With the invention of written language, people started to write down what they deem to be important. Writing, in this regard, helped preserve human knowledge and subsequently to pass this knowledge from one generation to the next generation as well as to share it with different cultures. One of the earliest written forms of language is instructions, specifically food recipes. Nettle pudding, recently discovered in the UK is, to date, one of oldest recipes known in the world [4]. The findings indicate that the recipe was from circa 6000 BC and was possibly dating back to 8000 BC. This wild plant, often found in the woods, was an important source of food for the people of that era. The uncovered recipe gives directions to get rid of the famous stings of the nettle by boiling them in hot water and then preparing a mixture with barley.

Food recipes are not the only instructional information humans have preferred to record in the early ages. They also wanted to record other aspects of their life. One of the earliest surviving manuals known to date is discovered in modern Turkey in 1906, which is about how to train Hittite chariot horses and is from circa 1350 BC [5]. In this text, Kikkuli, a master horse trainer from the land of Mitanni, gives detailed instructions to follow each day to raise the best chariot horses, such as the training program, the food and water rations, *etc*. These instructions not only help us understand what the people of that time eat or how they work but also give us how they used to live their lives by providing contextual information, how they collect food, and preserve it, or what methods they follow to train their horses and most importantly give us a piece of adequate information to how to replicate that process. Modern how-to instructions have been using both images and text for many years now, such as the recipes in cookbooks or the how-to guides shared across the web. These instructional guides often use visual information, such as images or videos, to aid textual directions. Although there are instructional guides with only one modality, be it text, videos, or images, the vast majority of modern how-to guides shared across the web are in a multimodal form that co-occurs with text and images, often with a visual that illustrates the process described in the text.

For instance, in Fig. 1.1, the end product of the recipe is illustrated on the top left, and ingredients are illustrated on the top right both as images and at the bottom the directions to follow as well as the ingredients are given. As seen from Fig. 1.1, different modalities are often used to aid the other modality.

1.1. Scope of the Thesis

Throughout this thesis, we will introduce the problem of comprehending multimodal how-to instructions from images and text. First and foremost, we will make an emphasis on why this problem is important and worth solving as well as possible ways to address the main challenges about it. There are a handful of reasons why how-to instructions with images and text are interesting to work on, and why understanding them is an interesting research problem to solve. Instructions have been around us for centuries, be it in the form of recipes,



To make this salad, in the jar, put items in this order:

- Strawberry-Lime Vinaigrette (see recipe below—I use about two tablespoons of dressing per pint-sized jar)
- Cooked Quinoa
- Sliced Strawberries
- Sunflower Seeds
- Sliced Green Onions
- Crumbled Feta
- Baby Spinach

Depending on how juicy your strawberries are, you might want to put them down lower in the jar (before the quinoa) to keep them even further away from the spinach.

Figure 1.1 A recipe with both images and text tells almost everything one would need about how to prepare food. A Feta salad in a jar recipe taken from [6]. On the top left, the final state is illustrated, top right shows the ingredients annotated with text. At the bottom, we see the directions to follow.

or how-to guides, written on stone tablets or books, or else published on the web. Thus, there is a good chance that they will still be useful sources of information in the future as well. A second interesting aspect of understanding images and text in the form of instructions is that they explicitly contain the information to replicate a process from scratch. This is especially important for three reasons, first, the instructions are almost always written with a clear purpose, that is to replicate a process, therefore contain only the specific information to do so. Second, understanding the required steps of a process is an important milestone for machines because step-by-step instructions inherently divide a process into smaller chunks, with each chunk followed by a successor chunk. The third interesting aspect of this problem is that modern how-to guides of any form almost always contain multimodal information, such as images, videos, and text. In particular, images and text often follow each other in subsequent steps, with no particular order. This is challenging, but an interesting part of the problem as having multiple modalities might be useful to address some of the challenges that surface around this problem, yet this might give birth to newer challenges such as the possible ambiguities, biases, or semantic gaps that are introduced with modality shifts. Moreover, having multiple modalities brings another challenge, that is, understanding the coreferences made in the text might refer to either the predecessor or the successor image or images. Another problem might be to skip mentioning an instruction in one modality and expect the reader to fill in this logical gap by inferring what is missing from the information provided from another modality, *e.g.* having a chopped tomato image and failing to mention chopping the tomatoes in the text and just continuing with "put the tomatoes in the pan". For this particular case, we need to be able to make a coreference resolution, but the problem here can not be simply solved by just mapping the word "tomatoes" to the tomatoes in the images, as the reader is forced to make a reasoning that the tomatoes should have been chopped because the tomatoes are different from the previous tomato images. Thus they might have been transformed *i.e.* chopped. Nevertheless, failing to mention this in one modality forces the reader to fill in the semantic gap using only one modality.

One particular challenge we also observe in how-to instructions is that often objects are referred to with no states, such as a tomato in good shape being cut into smaller parts but

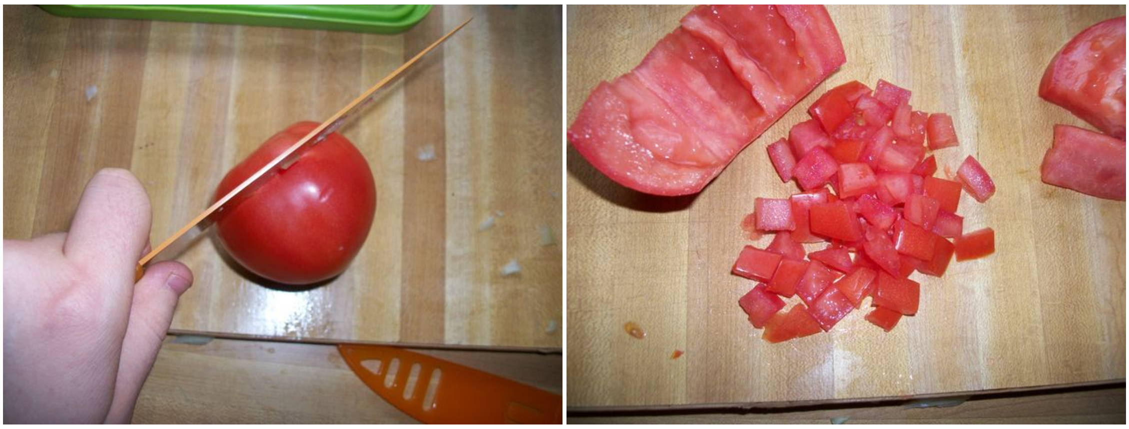


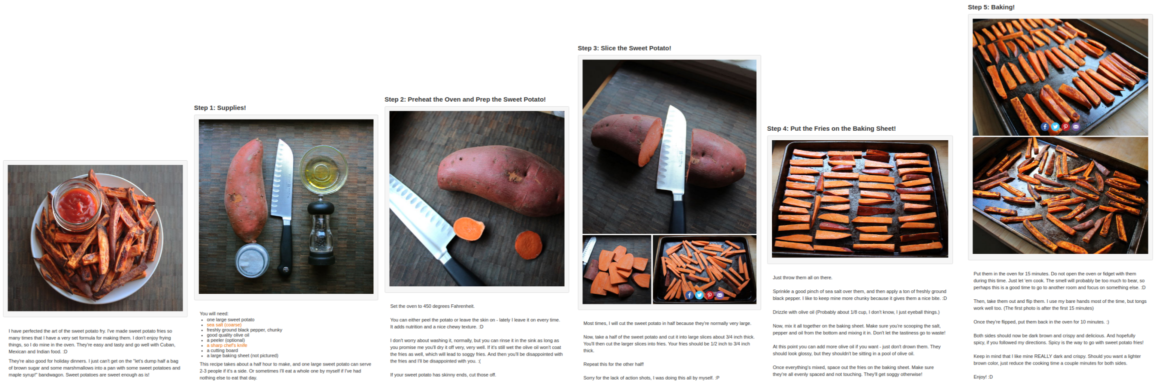
Figure 1.2 Different states of tomato, taken from Instructables.com website [7]

still mentioned as a tomato. This is because people seem to refer to objects even though they change states, nevertheless constituting a problem as the text does not completely describe the image, as also illustrated in Fig. 1.2.

Another challenge that needs to be addressed while tackling this problem is that how-to guides might have directions in them telling the same story, but in a different modality, although this might be seen as complementary for a direction in different modalities, the change in visual information and the change in textual information often can not be compared. For instance in a recipe, for an image that displays the ingredients as a first figure and as a list of those ingredients as the first textual direction, the next step might be to mix some of those ingredients, which would lead to a very different image from the previous image, but for the text modality, the direction might simply be “mix the pepper with the mince”. This is problematic for two reasons, first, the instructions in text modality do not simply describe the image, but the image might be a product of the actions described in the textual directions. Therefore, one needs to understand and infer that following a set of instructions would transform the image to the subsequent image, although the images are not their immediate successors, but might have textual steps between them.

Alignment between images and text is a big problem in understanding how-to guides. This is mainly because how-to instructions do not follow a fixed structure which makes them even harder to process, and finally leads to problems in aligning one modality with the other. In

Fig. 1.3, a how-to instruction, in the form of the cooking recipe is taught in 6 steps but the first set of images demonstrates the final state of the how-to but not the initial state. Thus, this alignment should be considered while dealing with the sequences in the tutorial.



I have perfected the art of the sweet potato fry. I've made sweet potato fries so many times that I have a very set formula for making them. I don't enjoy frying things, so I do mine in the oven. They're easy and tasty and go well with Cuban, Mexican and Indian food. :D

They're also good for holiday dinners. I just can't get on the "let's dump half a bag of brown sugar and some marshmallows into a pan with some sweet potatoes and maple syrup!" bandwagon. Sweet potatoes are sweet enough as is!

Step 1: Supplies!

You will need:

- one large sweet potato
- sea salt (coarse)
- freshly ground black pepper, chunky
- good quality olive oil
- a peeler (optional)
- a sharp chef's knife
- a cutting board
- a large baking sheet (not pictured)

This recipe takes about a half hour to make, and one large sweet potato can serve 2-3 people if it's a side. Or sometimes I'll eat a whole one by myself if I've had nothing else to eat that day.

Step 2: Preheat the Oven and Prep the Sweet Potato!

Set the oven to 450 degrees Fahrenheit.

You can either peel the potato or leave the skin on - lately I leave it on every time. It adds nutrition and a nice chewy texture. :D

I don't worry about washing it, normally, but you can rinse it in the sink as long as you promise me you'll dry it off very, very well. If it's still wet the olive oil won't coat the fries as well, which will lead to soggy fries. And then you'll be disappointed with the fries and I'll be disappointed with you. :(

If your sweet potato has skinny ends, cut those off.

Step 3: Slice the Sweet Potato!

Most times, I will cut the sweet potato in half because they're normally very large.

Now, take a half of the sweet potato and cut it into large slices about 3/4 inch thick. You'll then cut the larger slices into fries. Your fries should be 1/2 inch to 3/4 inch thick.

Repeat this for the other half!

Sorry for the lack of action shots, I was doing this all by myself. :P

Step 4: Put the Fries on the Baking Sheet!

Just throw them all on there.

Sprinkle a good pinch of sea salt over them, and then apply a ton of freshly ground black pepper. I like to keep mine more chunky because it gives them a nice bite. :D

Drizzle with olive oil (Probably about 1/8 cup, I don't know, I just eyeball things.)

Now, mix it all together on the baking sheet. Make sure you're scooping the salt, pepper and oil from the bottom and mixing it in. Don't let the tastiness go to waste!

At this point you can add more olive oil if you want - just don't drown them. They should look glossy, but they shouldn't be sitting in a pool of olive oil.

Once everything's mixed, space out the fries on the baking sheet. Make sure they're all evenly spaced and not touching. They'll get soggy otherwise!

Step 5: Baking!

Put them in the oven for 15 minutes. Do not open the oven or fidget with them during this time. Just let 'em cook. The smell will probably be too much to bear, so perhaps this is a good time to go to another room and focus on something else. :D

Then, take them out and flip them. I use my bare hands most of the time, but tongs work well too. (The first photo is after the first 15 minutes)

Once they're flipped, put them back in the oven for 10 minutes. :)

Both sides should now be dark brown and crispy and delicious. And hopefully spicy, if you followed my directions. Spicy is the way to go with sweet potato fries!

Keep in mind that I like mine REALLY dark and crispy. Should you want a lighter brown color, just reduce the cooking time a couple minutes for both sides.

Enjoy! :D

Figure 1.3 A sweet potato recipe with 6 steps, taken from Instructables.com website [8]

In Fig. 1.3, we see a recipe written in 6 steps. Each of these steps contains at least one or more images related to the story told in the text in that step, which leads to some structure and a degree of alignment between images and text. Nevertheless, considering the vast amount of recipes on the web, we observe varying structures between the websites, or even through each instruction. Thus, it is a challenge to understand which part of the recipe is related to which image or images as well as to extract ingredients and tools in the recipes.

Natural language understanding and visual understanding have been two major challenges in the artificial intelligence field, and there have been decades of work in both domains. Much recently, researchers in language and vision communities have started working more closely together to address problems that involve a collaborative understanding of both written language and visual data. Most mediums, such as the web, books, emails, *etc*. have both images and text nested in each other. Thus, understanding how-to instructions often involve a joint understanding of multiple modalities of information *e.g*. images and text.

Clearly designed instructions are key to understanding and replicating a process. How-to guides in this sense, are great sources of instruction as people can replicate the same process by just following the guide. Moreover, how-to guides are quite an interesting source of information as there are infinitely many instructions in the world, considering there is typically more than one way to do something, and in this regard, there might be infinitely many variations for almost any how-to guide. Instructional guides are also very challenging to understand. On the one hand, they often contain multimodal information such as images and text in them. On the other hand from the beginning to the end of a how-to often there is very little structure or no structure at all. Additionally, the images in the how-to's might show various forms of the objects, tools, or ingredients. Therefore, implicit and explicit knowledge of the text might be required to understand the transformation or relationship between each image. This is also true for the text as the author of the guide might have chosen to skip some of the vital information in the text but expect the reader to infer this information from the visual content. In general, there are several challenges to this problem, but this problem is quite interesting firstly because it is a real-world problem, *i.e.* almost everyone has read a how-to before or tried to replicate it from scratch, and second, how-to guides have instructional information for adults to replicate a process in an often multimodal form which makes it even more interesting as understanding instructions would be a crucial goal for machines to have a generalized intelligence.

How-to guides, such as cooking recipes, typically consist of multiple steps involving various objects and entities, most of which interact with each other through different actions. Considering an action as a combination of a verb and an object or entity, being able to generalize to unseen compositions of these action compounds pose a great challenge that remains an open research problem. In particular, having been exposed to primitive elements such as "cut" and "tomatoe", being able to comprehend a composition of those primitives *e.g.* "cut tomatoe", is a challenging research problem as this particular compound might have never been observed during training time. Therefore, understanding and testing models' compositional generalization abilities and exploring whether grounded textual instructions with images can help models to systematically generalize to novel compositions of actions remains an interesting challenge towards understanding multimodal how-to instructions. In this regard, understanding how visually grounded textual instructions might help models' systematic generalization abilities, as illustrated in Fig. 1.4 yet remains an important research problem we will be investigating.

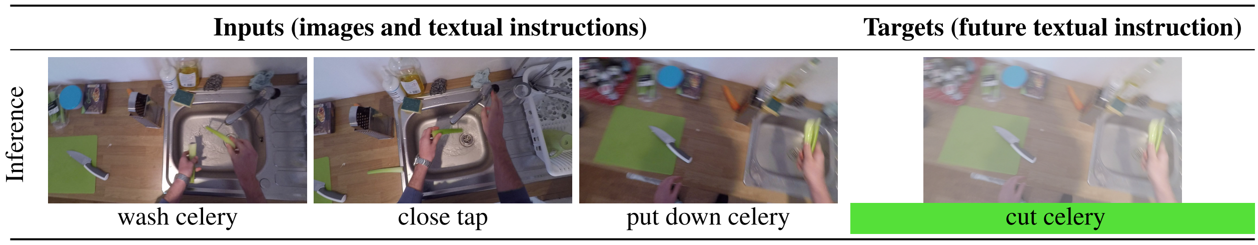

Inputs (images and textual instructions)
Targets (future textual instruction)
Inference
wash celery
close tap
put down celery
cut celery

Figure 1.4 Even though a model may be exposed to the primitives ‘wash’, ‘close’, ‘put down’, ‘cut’ and ‘celery’ during training, generalizing to the never before observed “cut celery” composition during inference pose a great challenge towards understanding how-to instructions, hence understanding whether multimodality can help models systematically generalize to unseen compositions yet remains an open research problem. (Image taken from EK100 [9].)

In summary, machine comprehension of how-to instructions with images and texts is a challenging but interesting real-life problem involving several tasks to be addressed. We briefly discussed a few of these challenges here and will be thoroughly addressing some of the main challenges related to this research problem throughout this thesis.

1.2. Contributions

In this thesis, we explore multimodal machine comprehension of how-to instructions with images and text and review related work in the literature and point out current challenges toward understanding multimodal how-to’s, and propose methods to address some of these challenges. In the following, we summarize the key contributions of this thesis.

- We investigate machine comprehension and reasoning problems and review the previous literature to lay the grounds for understanding multimodal how-to instructions.
- We survey compositional generalization literature and highlight current research challenges and discuss its relation to understanding multimodal how-to instructions.
- We introduce a multimodal benchmark how-to instructions dataset comprised of cooking recipes with images and text.
- We propose novel methods for understanding multimodal procedures.

- We introduce a challenging multimodal compositional generalization setup and propose methods to benchmark and show the contribution of multimodality to significantly improve the state-of-the-art in understanding multimodal how-to instructions.

1.3. Organization of the Thesis

In the following chapters, we will discuss how to teach machines to understand what they read and see and to comprehend and reason by using something that has been around us for thousands of years, how-to instructions.

The remaining chapters of the thesis are arranged in the following manner.

- Chapter 2 discusses the comprehension and reasoning problem.
- Chapter 3 reviews the compositional generalization problem and related studies.
- Chapter 4 introduces multimodal machine comprehension of cooking recipes.
- Chapter 5 demonstrates methods for procedural understanding of cooking recipes.
- Chapter 6 proposes methods for grounded compositional understanding of actions.
- Chapter 7 outlines the thesis and explores open research directions.

2. MACHINE COMPREHENSION AND REASONING

In recent years, we have seen a great deal of interest in artificial intelligence research. As larger-scale datasets became available, especially in the vision and language fields, it enabled researchers to attack problems that had never been proposed before. The abundance of large-scale data motivated researchers to develop more complex models which could benefit from these datasets, and the progress in deep learning methods has spurred the artificial intelligence communities to address a diverse range of problems. Several different tasks have been proposed that revolve around these datasets featuring multiple modes of information [10–13]. In the vision and language domain, especially image captioning, as well as visual question-answering tasks have garnered a lot of interest. In the image captioning task, the main goal is to describe an image with the best possible caption. The studies around this task mainly fall into two categories, that is generative-based models and retrieval models. The studies in the former category deal with generating new natural language descriptions from scratch [13, 14] whereas the studies that fall into the latter category aim to retrieve descriptive captions for the query image [15, 16]. Retrieval-based studies also deal with multi-modal retrieval by linking different modalities such as images and their related descriptions. The studies in this category use images as queries to retrieve a related caption among several candidates and a natural language description to retrieve a relevant image for that description. In visual question answering [12, 17, 18], the main goal is to answer a question given an image. Although the quality of the models improved steadily over time, due to the nature of these tasks, there have been several challenges that required solving problems jointly in both the image and text domains. For example, in the image captioning task it is problematic to measure how well the model performs. This is mainly because descriptions are hard to evaluate and how well the captions align with human judgment is a separate problem on its own. In visual question answering, although the problem is better formulated in terms of evaluation, casting the task as a classification raises other questions such as whether models actually learn to reason or just select answers which maximize a probability, even though they are out of context.

The question-answering problem has been studied in the natural language processing field [19] and is referred to as reading comprehension, where the aim is to read a story, understand it, and finally be able to answer questions about the story [20] to measure how well a reader understands the given story. Visual question answering (VQA) might be considered as an extension to this task where the narration in this setup is done via an image, whereas in reading comprehension narration is done via text. Understanding however is often measured in the same way. Asking a question and expecting a correct answer for the given context. Recently, comprehension and reasoning tasks have become very popular, especially in the text and image domain. Yet there is a huge gap between machines and humans for machines to achieve human-level comprehension and reasoning. Comprehension, in a broader context, refers to the ability to understand the meaning of something, and reasoning in the same regard refers to drawing conclusions from facts. These are very high-level problems and are considered to be challenging, not just for machines but even for humans. In the subsequent sections, we explore the comprehension and reasoning problem for both image and text domains and briefly describe prior efforts toward this direction, and lay the groundwork for our proposed research problem. We also provide an overview of the prominent datasets proposed in the literature for the comprehension and reasoning tasks in Table 2.1 which we will be discussing in this chapter.

Table 2.1 List of Comprehension and Reasoning Datasets

Dataset	Image Source	Question Source	Formulation	#Images	#Questions	Modality
SQuAD	-	Human	RC	-	100K	T
MCTest	-	Human	RC	-	2640	T
WikiQA	-	Query Logs	IR	-	3047	T
TREC-QA	-	Query Logs	IR	-	1479	T
CNN/Daily Mail	-	Summary + Cloze	RC	-	1.4M	T
CBT	-	Cloze	RC	-	688K	T
DAQUAR	NYU-Depth V2	Human	VQA	1,449	12,468	I
Visual Madlibs	COCO	Human	VQA	10,738	360,001	I
VQA	COCO	Human	VQA	204,721	614,163	I
COCO-QA	COCO	Syntetic	VQA	117,684	117,684	I
FM-IQA	COCO	Human	VQA	158,392	14,944	I
Visual7W	COCO	Human	VQA	47,300	327,939	I
MovieQA	Movies	Human	Movie QA	408 (Movies)	14,944	I/V/T
TQA	CK12	Human	TQA	3,455	26,260	I/T
CLEVR	Rendered	Syntetic	VQA	100,000	999,968	I

T: Text, I: Image, V: Video

2.1. Reading Comprehension

The objective of the reading comprehension task is twofold: firstly, to train machines on how to read, and secondly, to evaluate their comprehension by posing questions related to the material they have read. Machine reading comprehension, in this regard, aims to evaluate how well the machines comprehend the given text by measuring how accurate the machines are to answer questions, similar to the measures humans evaluate reading comprehension for human students. In this regard, using comprehension tests is appealing mainly because the performance of the reader is objectively gradable. Reading comprehension task goes back to Hirschman *et al.* [19], in which they collected a dataset of 115 reading comprehension tests that encompasses four grade levels, from third grade to sixth grade. Stories in this dataset are in the form of a newspaper article, and for each story, there are five questions among the "who, what, when, where, why" types. In this work, Hirschman *et al.* showed that the pattern matching approach could be used to select answers for "who, what, when, where, why" questions. This dataset sparked interest among researchers, and their baseline was later improved by Riloff and Thelen [21], and Charniak *et al.* [22] using rule-based systems, and by Ng *et al.* [23] using a logistic regression based system. Wang *et al.* [24] used a neural method on the same dataset but could not improve upon the rule-based baseline. More recently, Wang *et al.* [25] created QASENT dataset utilizing TREC-QA, wherein the questions are selected from TREC 8-13 QA tracks, and sentences are chosen from questions with overlapping non-stopwords. Richardson *et al.* [26] collected a dataset consisting of 660 fictional stories for open-domain question answering. In this dataset, each story has 4 questions where the answer to each question exists in the story and the stories are limited to a level where young children would understand them, to reduce the common world knowledge required by the task. Yang *et al.* [27] created WikiQA for question answering problems in open-domain setup. In this dataset, there are 3047 questions sampled from query logs from Bing, and Wikipedia passages are used to source answers. Weston *et al.* [28] created bAbI, a synthetic dataset for reading comprehension for a set of toy tasks. Much recently, Rajpurkar *et al.* proposed SQUAD dataset which consists of more than 100K questions posed over Wikipedia articles. Trischler *et al.* [29] created NewsQA dataset from 12K CNN articles with 120K question-answer pairs where some of the questions might not have the correct answer in the related article.

Context:

(CNN) Dolce & Gabbana went familial for fall at its fashion show in Milan on sunday , dedicating its collection to ” mamma ” with nary a pair of ” mom jeans ” in sight . Dolce & Gabbana, who are behind the Italian brand , sent models down the runway in decidedly feminine dresses and skirts adorned with roses , lace and even embroidered doodles by the designers ’ own nieces and nephews . many of the looks featured saccharine needlework phrases like ” i love you , mamma ” and ” Per la mamma più bella del mondo ” (for the most beautiful mother in the world) as a tableau vivant of moms and daughters stood and posed as a backdrop for the runway . even the usually stoic - faced front row could n’t help but applaud and smile as a few models carried their own high - fashion progeny down the runway .

Query:

@placeholder dedicated their fall fashion show to moms

Answer:

Dolce & Gabbana

Figure 2.1 An example story and question-answer pair taken from CNN/Daily Mail dataset [30]

Reading comprehension task is by nature challenging, mainly because of ambiguities, and biases, and require a higher level of understanding and often common world knowledge to comprehend what is being read, as well as what is being asked. In Fig. 2.1, a typical setting is illustrated for the reading comprehension task where the story is given along with the query posed to the reader and the correct answer for that query. In order to answer the posed question correctly, a reader should be able to understand who the producer is, who attacked whom, and what the lawyer said, *etc*. Therefore, simple pattern-matching approaches would not suffice for this challenging task.

2.2. Visual Reasoning

Visual reasoning is a broad topic that takes its roots back from the well-known Summer Vision Project[1], aiming to identify objects by matching them with a vocabulary of known objects. The project is widely considered as the birth of artificial intelligence as a research field. In this regard, visual reasoning has been a major goal of computer vision from maybe the very beginning which was kindled with the Summer Vision Project. Subsequently, visual reasoning requires an in-depth understanding of visual data, such as what is the scene, the actors in it, the actions, if there are any, the context, the objects, and their compositions, the relations, and interactions between the actors and the objects, size, shape, orientation, geometry, scale, *etc*.

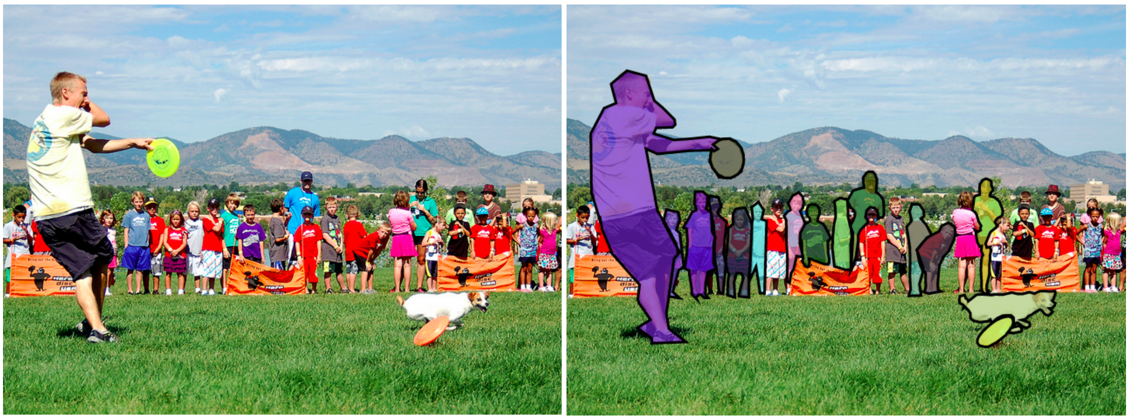
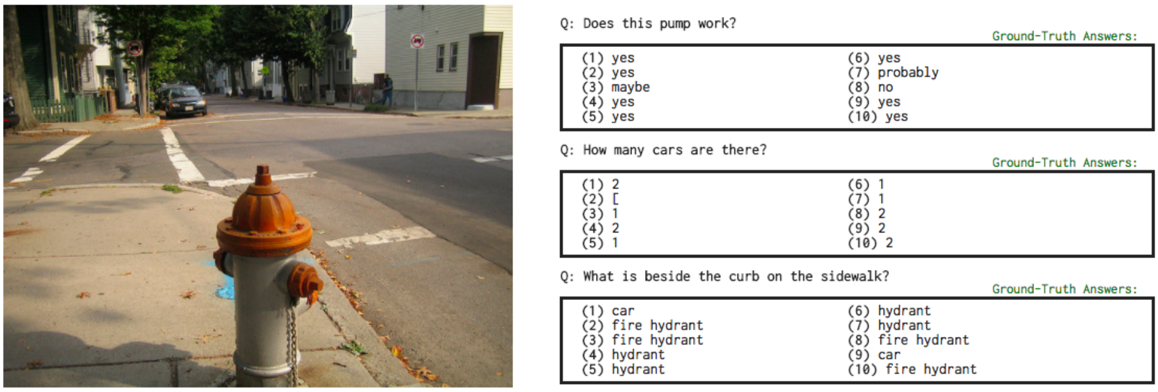


Figure 2.2 Semantic segmentation of pixels translates to understanding the correct labels of visual data. On the left, the original image is taken from the MSCOCO dataset [11], and on the right semantic annotations are drawn onto the same image.

There have been decades of research done in understanding visual data. Visual understanding, in this regard often referred as to understanding the correct labels of pixels through semantic annotations as also illustrated in Fig. 2.2. But, to completely understand a scene we should be able to go beyond semantic labeling. Moreover, how we measure the degree of visual understanding is a problem on its own. One possible way to do that is to be able to answer any question about it [17]. A large body of work has been amassed to date for question answering, an established task in the language domain. Recently, through

[1] https://dspace.mit.edu/handle/1721.1/6125

visual question answering, this task has been extended to the visual domain by replacing the passage or story in textual counterpart with an image which translates to changing the modality of the context from text to visual data [17]. Latest efforts in this direction sparked a lot of interest in visual reasoning and comprehension [12, 17, 18]. As visual reasoning and comprehension typically require interaction with humans, that is through natural language, this task practically requires dealing with multimodal data such as images or videos and text at the same time. In a typical visual reasoning task, a viewer is expected to understand the visual data as well as to comprehend the question about it, and finally be able to provide an answer within the context of the visual data and the question. Such a scenario is illustrated in Fig. 2.3.



Q: Does this pump work?

Ground-Truth Answers:

(1) yes
(2) yes
(3) maybe
(4) yes
(5) yes
(6) yes
(7) probably
(8) no
(9) yes
(10) yes

Q: How many cars are there?

Ground-Truth Answers:

(1) 2
(2) [
(3) 1
(4) 2
(5) 1
(6) 1
(7) 1
(8) 2
(9) 2
(10) 2

Q: What is beside the curb on the sidewalk?

Ground-Truth Answers:

(1) car
(2) fire hydrant
(3) fire hydrant
(4) hydrant
(5) hydrant
(6) hydrant
(7) hydrant
(8) fire hydrant
(9) car
(10) fire hydrant

Figure 2.3 A natural image on the left taken from MSCOCO dataset [11] along with questions and their ground-truth answers from VQA dataset [12].

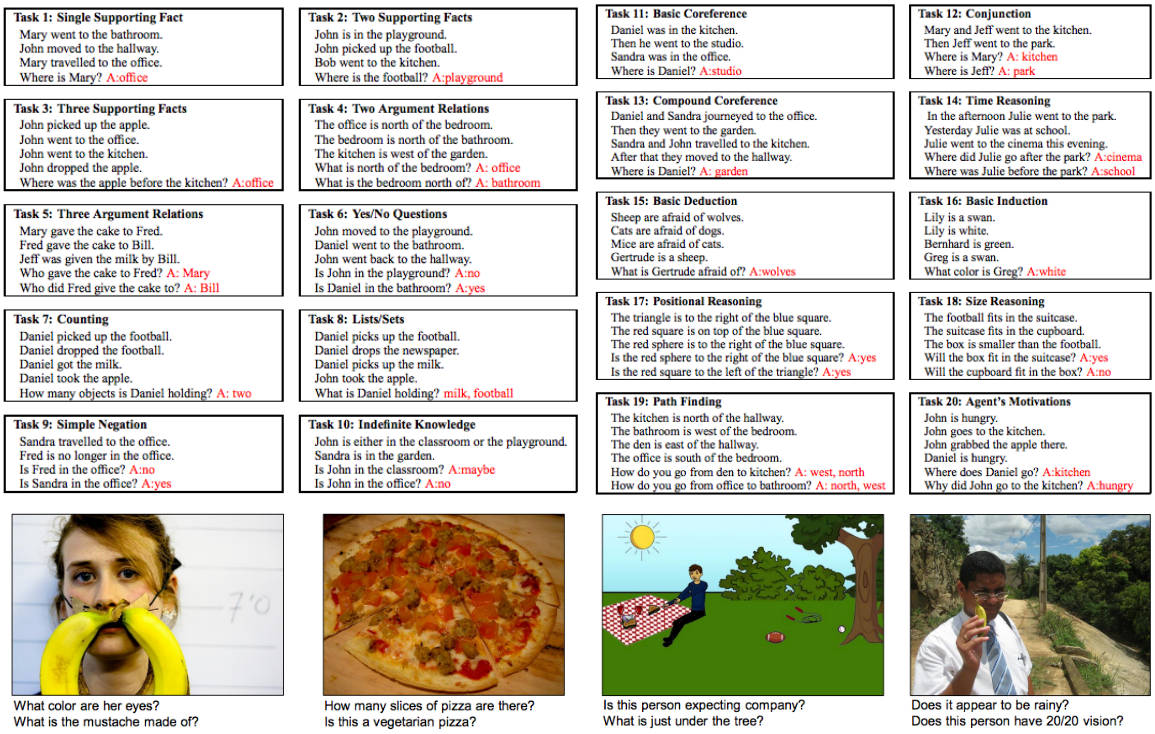
2.3. Tasks

Reading comprehension tasks require understanding the given context, be it a story, a dialogue, or a paragraph, and subsequently being able to query that understanding by providing answers for what is being questioned. In particular, the query performed over the previous understanding would require the reader's understanding of the given context in a variety of tasks such as assessing the ability to count, list, or else reasoning about the position and size *etc*. Visual reasoning is a challenging concept wherein a viewer needs to understand all the parts of the visual data, but also to understand the semantics of compositions of scenes, objects, entities, and events create. For the reading comprehension and visual reasoning tasks, a reader/viewer might need different forms of reasoning. We adopted and stratified a set of reasoning types from Trischler *et al.* [29] and Chen *et al.* [31] for the text domain and extended and transformed these types for multimodal comprehension of images and text.

- **Exact Match:** The answer is self-evident and can be found in the given context by just matching the surrounding words.
- **Paraphrasing:** The answer can be easily identified by paraphrasing the question.
- **Inference:** The answer can not be found easily by matching the available semantic information but must be inferred from the conceptual overlap of partial clues in the context.
- **Coreferencing:** The answer can not simply be matched but must be coreferenced from the available information.
- **Synthesis:** The answer can not be extracted from the context easily but must be synthesized from the available components.
- **Non-existent:** The answer can not be derived from the given context and common world knowledge is required.

In Fig. 2.4, a list of comprehension and reasoning tasks are given that assess a variety of abilities, such as reasoning about the size of things, making inductions between actions, inferring when they happen, referring to them and making deduction of what a good sight means, *etc*.

Comprehension and Reasoning as we discussed so far, are challenging tasks and due to the nature of this problem, for both image and text domains, using RNNs including their variants such as LSTM or GRU and CNNs does not simply work well in practice, for this task. In particular, comprehension and reasoning as we discussed need to deal with complex information and use different types of reasoning, thus we need specific mechanisms to account for the abilities a machine would need to answer such questions which we will be discussing under relevant sections.



Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

Task 7: Counting
Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 8: Lists/Sets
Daniel picks up the football.
Daniel drops the newspaper.
Daniel picks up the milk.
John took the apple.
What is Daniel holding? milk, football

Task 9: Simple Negation
Sandra travelled to the office.
Fred is no longer in the office.
Is Fred in the office? A:no
Is Sandra in the office? A:yes

Task 10: Indefinite Knowledge
John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

Task 11: Basic Coreference
Daniel was in the kitchen.
Then he went to the studio.
Sandra was in the office.
Where is Daniel? A:studio

Task 12: Conjunction
Mary and Jeff went to the kitchen.
Then Jeff went to the park.
Where is Mary? A: kitchen
Where is Jeff? A: park

Task 13: Compound Coreference
Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 14: Time Reasoning
In the afternoon Julie went to the park.
Yesterday Julie was at school.
Julie went to the cinema this evening.
Where did Julie go after the park? A:cinema
Where was Julie before the park? A:school

Task 15: Basic Deduction
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 16: Basic Induction
Lily is a swan.
Lily is white.
Bernhard is green.
Greg is a swan.
What color is Greg? A:white

Task 17: Positional Reasoning
The triangle is to the right of the blue square.
The red square is on top of the blue square.
The red sphere is to the right of the blue square.
Is the red sphere to the right of the blue square? A:yes
Is the red square to the left of the triangle? A:yes

Task 18: Size Reasoning
The football fits in the suitcase.
The suitcase fits in the cupboard.
The box is smaller than the football.
Will the box fit in the suitcase? A:yes
Will the cupboard fit in the box? A:no

Task 19: Path Finding
The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations
John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is hungry.
Where does Daniel go? A:kitchen
Why did John go to the kitchen? A:hungry

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 2.4 A set of reading comprehension tasks and their examples described on bAbI dataset, taken from [32] on the top five rows, and on the bottom row a few example question answer pairs given for a typical visual reasoning task taken from VQA dataset [12].

2.4. Question Types

In this section, we briefly go over the types of questions used commonly in comprehension and reasoning tasks. Each of these question types is often posed to query various forms of reasoning as well as abilities as we discuss in the previous Chapter 2.3..

True/False Questions. In True/False type questions the reader/viewer is expected to provide an answer by stating whether the facts in the question form a true statement or else false. Another variation of True/False type questions is Yes/No questions. In the case of a Yes/No type question, the reader/viewer is given a context (text/visual) and asked to assess whether the statement is true or false but the formulation of the question is slightly different from the original true/false statement as the answer should be in yes/no form.

Multiple Choice. One common way to asses a reader's/viewer's understanding of a context is to use multiple-choice tests. In multiple-choice type questions, the reader is given a set of

John is in the playground.
Daniel picks up the milk.

Q: Is John in the classroom?
A: No
Q: Does Daniel have the milk?
A: Yes

Figure 2.5 Yes/No question and answer pairs taken from bAbI dataset [33]

options to choose from. In such a question, the objective is to choose the right answer from the pre-selected answers [34–36].

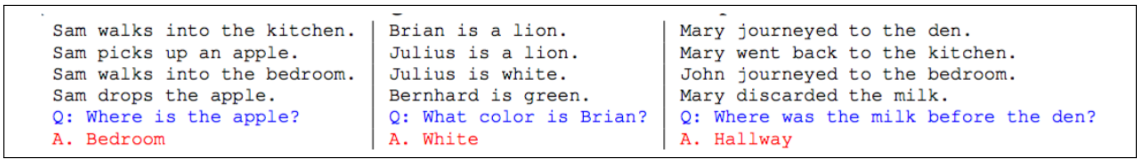
Question: What is Marsha's noodle made out of?

A) Spaghetti
B) plastic bag
C) mom
D) Macaroni

Figure 2.6 A multiple choice question answer pair selected from MCTest dataset [26]

These types of questions are relatively easier to answer as there are only a few options. Thus, as a precaution, the reader's ability to comprehend might be compared against random selection.

W Questions. In this type of question, which is often referred to as W questions, the questions are cast as Who, Where, What, When, Why, Which, and How, and the reader is asked to provide an answer that requires understanding a story, and actors in that story, as well as their relations while keeping track of different states in the given story. In W-type questions, typically Who questions assess the ability to find the related actor in a context, be it textual or visual. Such as 'who went to the kitchen', or 'who has a mustache in the image'. Where questions assess positional understanding, 'What' type of questions demand an understanding of the objects or entities in the context, 'When' type of questions evaluates the temporal understanding. 'How' type of questions are typically harder to answer as one needs to understand the relations and compositions in the context but also needs to make inferences using them.



```
Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.
Q: Where is the apple?
A. Bedroom
```

```
Brian is a lion.
Julius is a lion.
Julius is white.
Bernhard is green.
Q: What color is Brian?
A. White
```

```
Mary journeyed to the den.
Mary went back to the kitchen.
John journeyed to the bedroom.
Mary discarded the milk.
Q: Where was the milk before the den?
A. Hallway
```

Figure 2.7 Three W question and answer pairs taken from bAbI dataset [33]

Open Ended Questions In the open-ended type of questions, the answer might be in a free form and thus has no restrictions. The answers in open-ended question types might vary from one word to several words and thus might be considered more realistic and challenging. In Fig. 2.8 an open-ended type question-answer pair is illustrated.

Passage: The German measles are contagious for 7 days before to 7 days after the rash appears, as noted by the New York State Department of Health. Rubella is another term for the German measles, and the disease itself is highly contagious and can be seen in patients through a rash, swollen glands a fever. Know More.

Query: how long is german measles contagious

Answer: 7 days before to 7 days after the rash appears

Figure 2.8 An open-ended question with the passage from MS MARCO dataset, with query id 10555 taken from development set [37]

Cloze Style Questions. Cloze-style questions fill in the blank type of questions. It is often obtained by randomly removing a word from a sentence and asking the reader to find what is missing given the context. These types of questions are not considered Natural Language Questions (NLQ) as they are not really questions posed by humans, but only transformed sentences in fill-in-the-blank form.

Visual cloze style question type is similar to the textual counterpart. In this question type, the reader should select the missing visual content from a set of candidates. In Fig. 2.10 a visual cloze style question is illustrated.

Pointing Type Questions. In pointing-type questions, the machine is asked to point to a location in the given context. This type of question is mostly suitable for visual questions

Sentence: Earth has warmed one degree in past 100 years . Majority of scientists say greenhouse gases are causing temperatures to rise . Some critics say planets often in periods of warming or cooling .

Question: Earth has warmed one degree in past 100 years . Majority of scientists say ______________ are causing temperatures to rise . Some critics say planets often in periods of warming or cooling .

Answer: greenhouse gases

Figure 2.9 An illustrative cloze style question-answer pair from CNN/Daily Mail dataset [30]

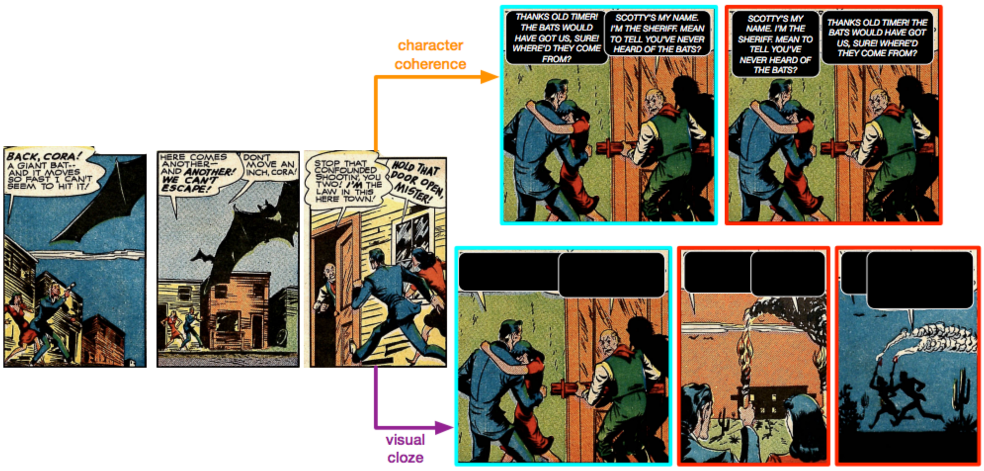
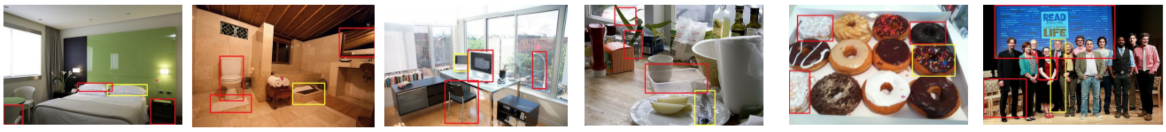

BACK, CORA! A GIANT BAT-- AND IT MOVES SO FAST I CAN'T SEEM TO HIT IT!
HERE COMES ANOTHER-- AND ANOTHER! WE CAN'T ESCAPE!
DON'T MOVE AN INCH, CORA!
STOP THAT CONFOUNDED SHOOTIN', YOU TWO! I'M THE LAW IN THIS HERE TOWN!
HOLD THAT DOOR OPEN, MISTER!
character coherence
THANKS OLD TIMER! THE BATS WOULD HAVE GOT US, SURE! WHERE'D THEY COME FROM?
SCOTTY'S MY NAME. I'M THE SHERIFF. MEAN TO TELL YOU'VE NEVER HEARD OF THE BATS?
SCOTTY'S MY NAME. I'M THE SHERIFF. MEAN TO TELL YOU'VE NEVER HEARD OF THE BATS?
THANKS OLD TIMER! THE BATS WOULD HAVE GOT US, SURE! WHERE'D THEY COME FROM?
visual cloze

Figure 2.10 A visual cloze style question is taken from [38], in which the reader is expected to select the correct panel that would not break the sequence of panel semantically.

where the machine might locate a point, but they might be adapted to other modalities as well.



Q: Which pillow is farther from the window?

Q: Which step leads to the tub?

Q: Which is the small computer in the corner?

Q: Which item is used to cut items?

Q: Which doughnut has multicolored sprinkles?

Q: Which man is wearing the red tie?

Figure 2.11 A set of examples for a pointing type question taken from Visual7W dataset [39]

For the visual context, pointing-type questions are typically asked by which question as in [39], but in general pointing type of questions might be used for circumstances one can point to, given the context is appropriate. A set of pointing-type questions is illustrated in Fig. 2.11.

2.5. Evaluation

Evaluation of machine comprehension and reasoning is extremely important as without a measure it is hard to say how well a model is doing. Evaluation of comprehension and reasoning tasks in this regard requires special attention as there are a variety of subtasks that comprehension and reasoning models should solve and measuring each of these tasks would need special care. Moreover, the nature of the context and how the models' understanding is questioned changes what is being measured. In the following, we will briefly describe the common evaluation methods for comprehension and reasoning-related tasks.

- **Accuracy:** Accuracy is an extensively utilized metric to assess comprehension and reasoning tasks performance. Accuracy is thus calculated by the ratio of correct answers to the total questions. Typically this metric is used in True/False question types, and multiple choice question types.
- **Word Match:** Word level matching aims to look at the exact string match between the answer and ground truth where only the correct matchings are used [40].
- **Automatic Metrics:** In this type of question, the evaluation is done by comparing how well the reader's answer match with the ground-truth answer by using automatic evaluation metrics for machine translation such as BLEU [41] and METEOR [42].
- **WUPS:** Wu-Palmer metric [43] looks at how two words are similar by considering the depths of their overlapping subsequences found in a taxonomy tree. In this metric, correctness is determined by how similar the answer is to the groundtruth using a threshold.
- **Human Judgment:** In this type of evaluation, the context, question, and answer pairs are presented to human subjects. Afterward, human subjects are asked to evaluate the

results as in the Turing test, whether the results are coming from a machine or human [44].

2.6. Reading Comprehension Datasets

There are a variety of datasets proposed in the literature for reading comprehension tasks. Each dataset has its own merits as well as weaknesses. In the following, we will briefly discuss the publicly available reading comprehension datasets.

QASent. QASent dataset is proposed by [25] and is based on data coming from TREC-QA. The sentences in this dataset are chosen in a way where the questions and answers share overlapping non-stopwords.

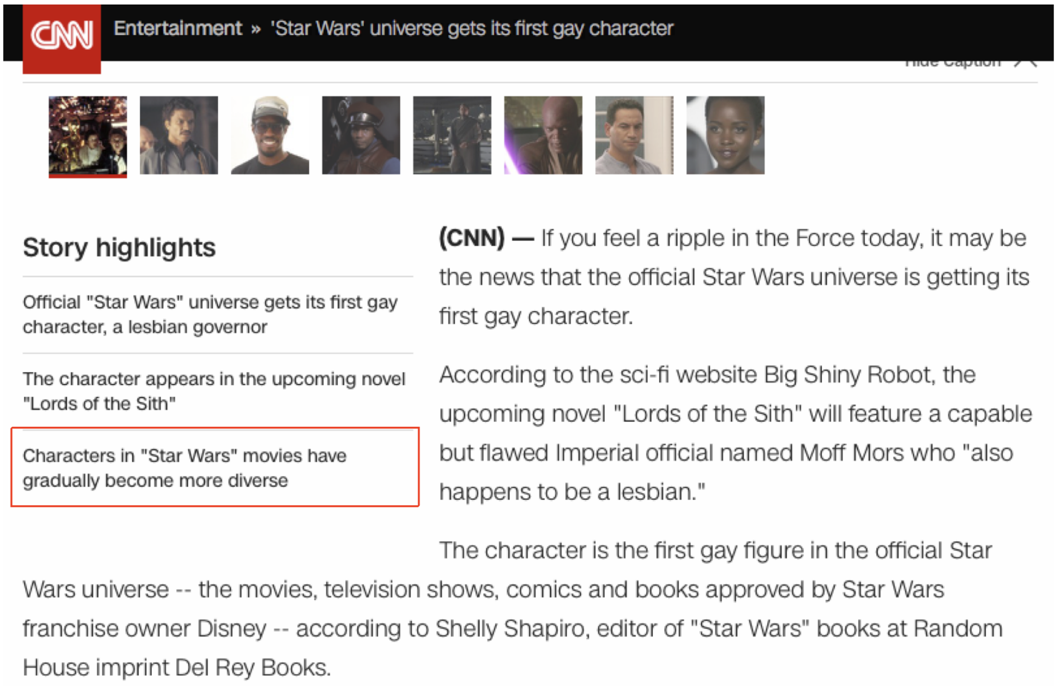
MCTest. MCTest is a challenging dataset consisting of 660 stories and each of these stories has 4 natural questions asked by humans wherein each question has 4 answers [26].

The dataset was constructed by crowd workers, with each story being written by a distinct individual. Furthermore, each story is self-contained and unrelated to the others. To address any inaccuracies that may have resulted from crowdsourcing, the dataset has been partitioned into two parts: MC160 and MC500, with the former being a smaller dataset that has undergone manual correction. One caveat of this dataset is that it is a small dataset to train deep learning models.

CNN/Daily Mail. CNN/Daily Mail datasets are collected from news web pages and contain real-life news articles [30]. CNN dataset contains over 90K news articles from CNN website, and on average it has 4 queries for each article resulting 380K question-story pairs. Daily Mail dataset has around 197K news articles wherein each article has 4 questions adding up to 880K story-question pairs.

Children's Book Test (CBT). CBT dataset [45] contains stories taken from child books. The dataset is created based on books that are available for free on Project Gutenberg [2]. In

[2] https://gutenberg.org



Entertainment » 'Star Wars' universe gets its first gay character

Story highlights

Official "Star Wars" universe gets its first gay character, a lesbian governor

The character appears in the upcoming novel "Lords of the Sith"

Characters in "Star Wars" movies have gradually become more diverse

(CNN) — If you feel a ripple in the Force today, it may be the news that the official Star Wars universe is getting its first gay character.

According to the sci-fi website Big Shiny Robot, the upcoming novel "Lords of the Sith" will feature a capable but flawed Imperial official named Moff Mors who "also happens to be a lesbian."

The character is the first gay figure in the official Star Wars universe -- the movies, television shows, comics and books approved by Star Wars franchise owner Disney -- according to Shelly Shapiro, editor of "Star Wars" books at Random House imprint Del Rey Books.

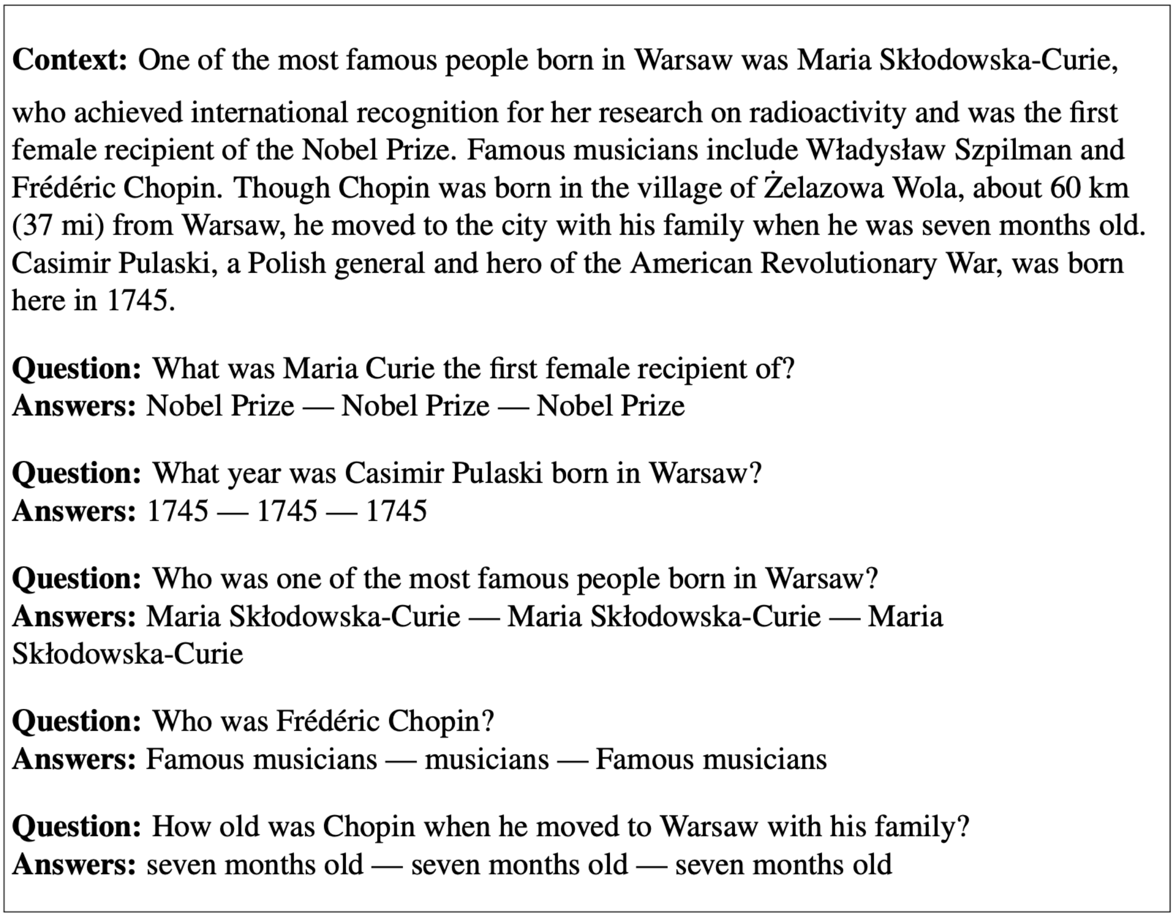
Figure 2.12 An example story taken from CNN/Daily Mail datasets where a question is selected from the highlights on the left marked as red box as cloze style [30]

each of the stories, there are 20 consecutive sentences and the 21st sentence is transformed into a question by removing a word from that sentence.

There are 4 splits in the dataset, each classified as the type of the removed word from the question, which are Named Entities, Common Nouns, Verbs, and Prepositions. For each question, 10 answers are selected from the story, each having the same POS tag with the correct answer.

Stanford Question Answering Dataset (SQuAD). The SQuAD dataset [46] was derived from 536 articles and, in total, comprises roughly 100,000 pairs of questions and answers. Every story in the dataset is a single paragraph that has been extracted from the aforementioned articles, and the questions related to each story have been gathered through crowdsourcing. The answers are also collected from crowd workers, and there is more than one answer for each question. One strength of this dataset is that the questions are NLQ. The availability of multiple responses from crowd workers collected for each question provides human agreement information for answers. Moreover, the answers are open-ended, thus more challenging and realistic for reading comprehension tasks. One weakness of this dataset is the small size of the articles. Another caveat is that it uses short paragraphs for stories, which typically contain 4 to 5 sentences.

Fig. 2.13 depicts a particular story from the SQuAD dataset, along with a corresponding group of question-answer pairs that pertain to that story.



Context: One of the most famous people born in Warsaw was Maria Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize. Famous musicians include Władysław Szpilman and Frédéric Chopin. Though Chopin was born in the village of Żelazowa Wola, about 60 km (37 mi) from Warsaw, he moved to the city with his family when he was seven months old. Casimir Pulaski, a Polish general and hero of the American Revolutionary War, was born here in 1745.

Question: What was Maria Curie the first female recipient of?
Answers: Nobel Prize — Nobel Prize — Nobel Prize

Question: What year was Casimir Pulaski born in Warsaw?
Answers: 1745 — 1745 — 1745

Question: Who was one of the most famous people born in Warsaw?
Answers: Maria Skłodowska-Curie — Maria Skłodowska-Curie — Maria Skłodowska-Curie

Question: Who was Frédéric Chopin?
Answers: Famous musicians — musicians — Famous musicians

Question: How old was Chopin when he moved to Warsaw with his family?
Answers: seven months old — seven months old — seven months old

Figure 2.13 Example story and a set of question-answer pairs related to that story is shown from the SQuAD dataset [46]

NewsQA. NewsQA is a challenging benchmark dataset consisting of 12K articles collected from CNN and around 120K question-answer pairs. The question and answer pairs are written by humans in natural language. Questions in this dataset may not be answered as some of the questions do not have an answer in the related story. The answers may be multiple-word passages from the related story.

Title: Soyuz Crew Endures Severe G-Forces on Re-entry

Context:

MOSCOW, Russia (CNN) – Russian space officials say the crew of the Soyuz space ship is resting after a rough ride back to Earth.
A South Korean bioengineer was one of three people on board the Soyuz capsule.
The craft carrying South Korea's first astronaut landed in northern Kazakhstan on Saturday, 260 miles (418 kilometers) off its mark, they said.
Mission Control spokesman Valery Lyndin said the condition of the crew – South Korean bioengineer Yi So-yeon, American astronaut Peggy Whitson and Russian flight engineer Yuri Malenchenko – was satisfactory, though the three had been subjected to severe G-forces during the re-entry.
Search helicopters took 25 minutes to find the capsule and determine that the crew was unharmed.
Officials said the craft followed a very steep trajectory that subjects the crew to gravitational forces of up to 10 times those on Earth.
Interfax reported that the spacecraft's landing was rough.
This is not the first time a spacecraft veered from its planned trajectory during landing.
In October, the Soyuz capsule landed 70 kilometers from the planned area because of a damaged control cable. The capsule was carrying two Russian cosmonauts and the first Malaysian astronaut. E-mail to a friend

Q1) Who was the capsule carrying?
Answer 1: two Russian cosmonauts and the first Malaysian astronaut.

Q2) Where did the Soyuz capsule land?
Answer 1: northern Kazakhstan

Q3) What distance from the target was the capsule?
Answer 1: 260 miles (418 kilometers)

Q4) What happened during the first landing?
Answer 1: landed in northern Kazakhstan on Saturday, 260 miles (418 kilometers) off its mark,
Answer 2: damaged control cable.

Q5) What was the target?
Answer not in story.

Figure 2.14 An example story and question answer pairs from NewsQA dataset [29].

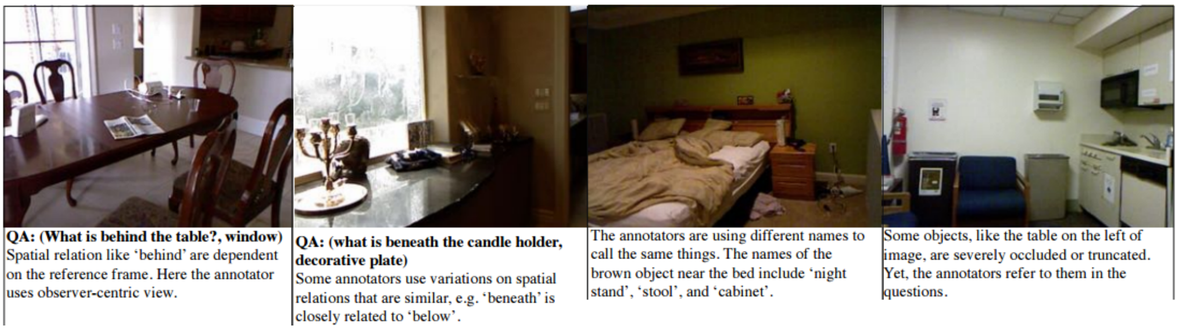
In Fig. 2.14, for a story collected from CNN, we see a headline, an article, and related questions and answers.

2.7. Visual Reasoning Datasets

In the following, we discuss the most relevant datasets in the visual domain for comprehension and reasoning tasks.

DAQUAR. This dataset is one of the earliest question-answering datasets in the visual domain [17]. The dataset contains over 12K question-answer pairs of RGBD images generated by humans. DAQUAR dataset uses 1449 RGBD indoor images from NYU-Depth V2 dataset [47] with depth information and annotated semantic segmentations of 894 object classes.

In Fig. 2.15, a set of challenging examples is illustrated along with the question and answer pairs.



QA: (What is behind the table?, window) Spatial relation like 'behind' are dependent on the reference frame. Here the annotator uses observer-centric view.

QA: (what is beneath the candle holder, decorative plate) Some annotators use variations on spatial relations that are similar, e.g. 'beneath' is closely related to 'below'.

The annotators are using different names to call the same things. The names of the brown object near the bed include 'night stand', 'stool', and 'cabinet'.

Some objects, like the table on the left of image, are severely occluded or truncated. Yet, the annotators refer to them in the questions.

Figure 2.15 Example images and question-answer pairs denoted as QA and questions denoted as Q from DAQUAR dataset, taken from [17]

Visual MadLibs. Visual Madlibs dataset [18] consists of 10,738 images from MSCOCO dataset [11] with 360K focused image descriptions collected automatically with fill-in-the-blank questions.

In Fig. 2.16, a set of examples are illustrated along with their descriptions. Descriptions in each row refer to a different type of question in the dataset.

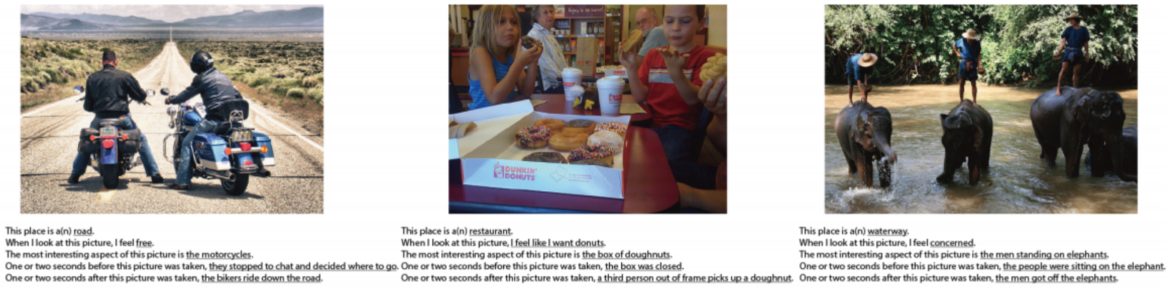

This place is a(n) road.
When I look at this picture, I feel free.
The most interesting aspect of this picture is the motorcycles.
One or two seconds before this picture was taken, they stopped to chat and decided where to go.
One or two seconds after this picture was taken, the bikers ride down the road.
This place is a(n) restaurant.
When I look at this picture, I feel like I want donuts.
The most interesting aspect of this picture is the box of doughnuts.
One or two seconds before this picture was taken, the box was closed.
One or two seconds after this picture was taken, a third person out of frame picks up a doughnut.
This place is a(n) waterway.
When I look at this picture, I feel concerned.
The most interesting aspect of this picture is the men standing on elephants.
One or two seconds before this picture was taken, the people were sitting on the elephant.
One or two seconds after this picture was taken, the men got off the elephants.

Figure 2.16 Example images and descriptions from Visual Madlibs dataset, taken from [18]

VQA. Visual Question Answering dataset [12], consists of around 205K images taken from MS COCO [11] dataset which are real images, and 50K abstract scenes from [48, 49]. The dataset contains around 615K questions for real images and 150K questions for abstract scenes. There are around 10M answers in the first version of the VQA dataset in total.

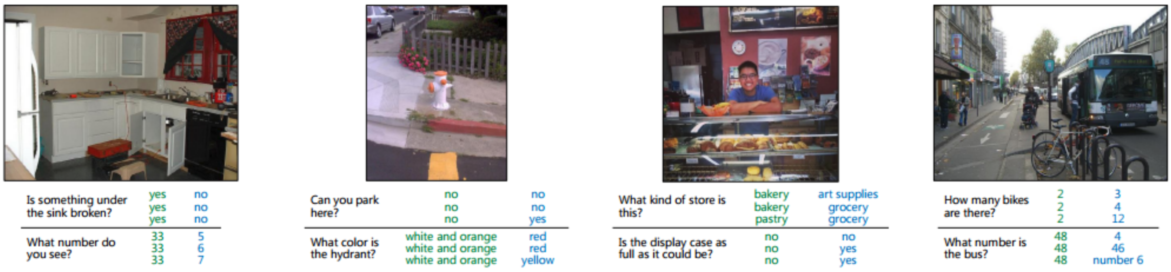

Is something under the sink broken?
yes yes yes
no no no
What number do you see?
33 33 33
5 6 7
Can you park here?
no no no
no no yes
What color is the hydrant?
white and orange white and orange white and orange
red red yellow
What kind of store is this?
bakery bakery pastry
art supplies grocery grocery
Is the display case as full as it could be?
no no no
no yes yes
How many bikes are there?
2 2 2
3 4 12
What number is the bus?
48 48 48
4 46 number 6

Figure 2.17 Example images from VQA dataset, taken from [12]. For each row, images are provided at the top, and below them, questions are provided in black, green, and blue responses referring to whether the answer is given by looking at the image or not, respectively.

In Fig. 2.17, a set of examples are illustrated utilizing the real image and abstract scene data from the VQA dataset.

FM-IQA.

FM-IQA dataset [44] contains 150K images from MSCOCO dataset [11] and 310K freestyle question and answer pairs in Chinese, as well as their translations into English for the purpose of multilingual image question-answering.

In Fig. 2.18, a set of examples are taken from the FM-IQA dataset, where each image has both Chinese and English translations for question-answer pairs.

Visual7W. This is a visual question-answering dataset with object groundings [39] collected

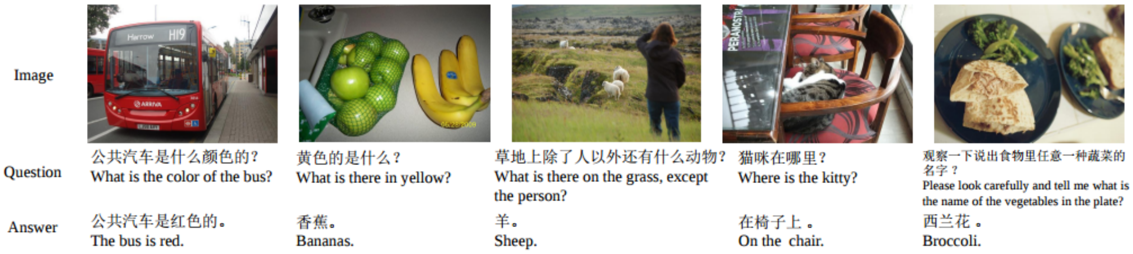


Image
Question
Answer
公共汽车是什么颜色的？
What is the color of the bus?
公共汽车是红色的。
The bus is red.
黄色的是什么？
What is there in yellow?
香蕉。
Bananas.
草地上除了人以外还有什么动物？
What is there on the grass, except the person?
羊。
Sheep.
猫咪在哪里？
Where is the kitty?
在椅子上。
On the chair.
观察一下说出食物里任意一种蔬菜的名字？
Please look carefully and tell me what is the name of the vegetables in the plate?
西兰花。
Broccoli.

Figure 2.18 Example images and question-answer pairs in Chinese and their translations in English from FM-IQA dataset, taken from [44]

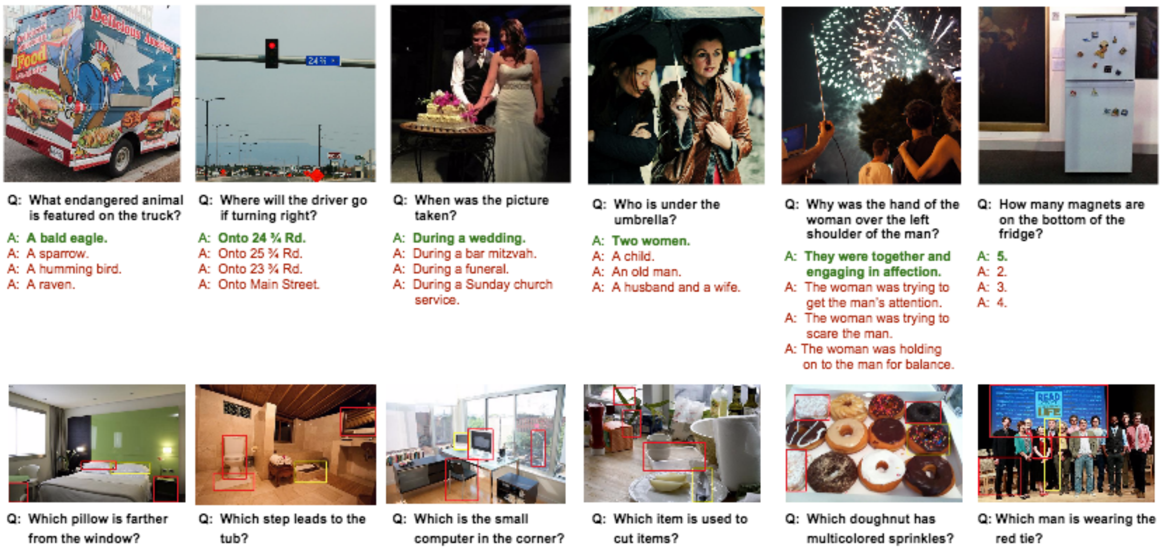

Q: What endangered animal is featured on the truck?
A: A bald eagle.
A: A sparrow.
A: A humming bird.
A: A raven.
Q: Where will the driver go if turning right?
A: Onto 24 ¾ Rd.
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.
Q: When was the picture taken?
A: During a wedding.
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church service.
Q: Who is under the umbrella?
A: Two women.
A: A child.
A: An old man.
A: A husband and a wife.
Q: Why was the hand of the woman over the left shoulder of the man?
A: They were together and engaging in affection.
A: The woman was trying to get the man's attention.
A: The woman was trying to scare the man.
A: The woman was holding on to the man for balance.
Q: How many magnets are on the bottom of the fridge?
A: 5.
A: 2.
A: 3.
A: 4.
Q: Which pillow is farther from the window?
Q: Which step leads to the tub?
Q: Which is the small computer in the corner?
Q: Which item is used to cut items?
Q: Which doughnut has multicolored sprinkles?
Q: Which man is wearing the red tie?

Figure 2.19 Examples from the Visual7W dataset taken from [39]. For each instance, an accompanying contextual image is provided, along with a set of questions and answers. The correct answers are indicated in green, while the incorrect answers are marked in red. The uppermost row pertains to 'Telling' type of questions, for which the answer modality is textual. At the bottom row, 'Pointing' type of questions is given where the answer is a bounding box in the image, for which the bounding box is yellow and red for the correct and wrong answers respectively.

on 47,300 images from MSCOCO [11] and has around 328K question-answer pairs with over 1.3M human-generated multiple-choices and over 561K object groundings from 36,579 categories. Each question in the dataset is a W question which is Who, Where, When, What, Why, How, and Which. In Fig. 2.19, examples are illustrated related to the aforementioned question types.

The main difference of this dataset is that it introduces a multimodal answer type for QA tasks, wherein the answer is a grounded image for the Which type of questions. For the rest of the questions, the answer is in textual modality.

MovieQA. MovieQA is a dataset that contains 15K multiple-choice questions and answers obtained from over 400 movies. The dataset aims to be a benchmark for story comprehension for both video and text [50].

Q: How does E.T. show his happiness that he is finally returning home?
A: His heart lights up

Q: Why do Joy and Jack get married that first night they meet in Las Vegas?
A: They are both vulnerable and totally drunk

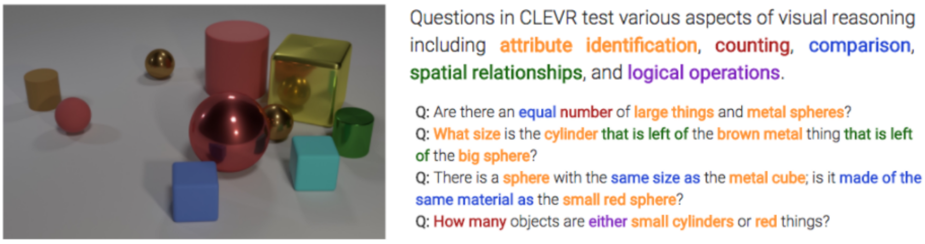
Q: Why does Forrest undertake a three-year marathon?
A: Because he is upset that Jenny left him

Q: How does Patrick start winning Kat over?
A: By getting personal information about her likes and dislikes

Figure 2.20 Examples from the MovieQA dataset taken from [50]. For each example, a snapshot from a movie clip is given, along with the questions and answers below the snapshot.

Each question in this dataset has 5 answers. Among them only one is correct. The dataset contains multiple sources of information such as movie clips, dialogs, and plots. Although some of the questions can be answered using only one modality, most of them require using multiple modalities, in particular, videos and text.

CLEVR. CLEVR is a visual reasoning dataset [51] with 100K computer-rendered images and around 1M generated questions. The objective of the questions in the CLEVR dataset is to assess different visual reasoning skills *e.g.* spatial relationship, identification of attributes, multi-attention, logical operations, comparison, and counting. In Fig. 2.21, a set of questions are provided for computer-rendered images.



Questions in CLEVR test various aspects of visual reasoning including attribute identification, counting, comparison, spatial relationships, and logical operations.

Q: Are there an equal number of large things and metal spheres?
Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
Q: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: How many objects are either small cylinders or red things?

Figure 2.21 An example image and generated questions taken from CLEVR dataset [51]

COMICS. COMICS dataset [38] consists of around 1.2M comic panels collected from

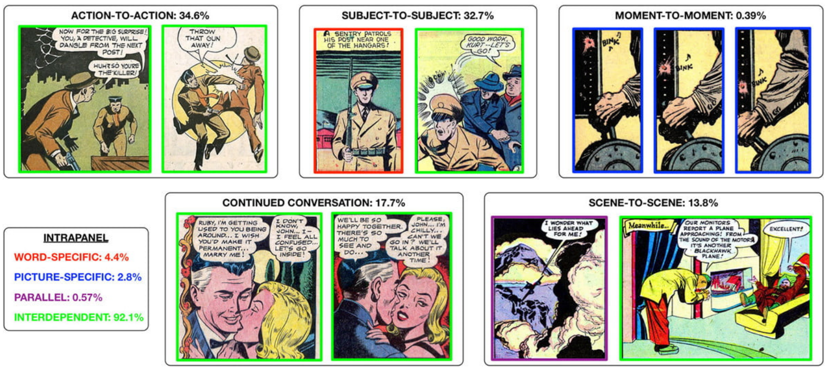

ACTION-TO-ACTION: 34.6%
NOW FOR THE BIG SURPRISE! YOU, A DETECTIVE, WILL DANGLE FROM THE NEXT POST!
HUH? SO YOU'RE THE KILLER!
THROW THAT GUN AWAY!
SUBJECT-TO-SUBJECT: 32.7%
A SENTRY PATROLS HIS POST NEAR ONE OF THE HANGARS!
GOOD WORK, KURT--LET'S GO!
MOMENT-TO-MOMENT: 0.39%
BINK
BINK
BINK
INTRAPANEL
WORD-SPECIFIC: 4.4%
PICTURE-SPECIFIC: 2.8%
PARALLEL: 0.57%
INTERDEPENDENT: 92.1%
CONTINUED CONVERSATION: 17.7%
RUBY, I'M GETTING USED TO YOU BEING AROUND... I WISH YOU'D MAKE IT PERMANENT... MARRY ME!
I DON'T KNOW, JOHN... I— I FEEL ALL CONFUSED... LET'S GO INSIDE!
WE'LL BE SO HAPPY TOGETHER. THERE'S SO MUCH TO SEE AND DO...
PLEASE, JOHN... I'M CHILLY... CAN'T WE GO IN? WE'LL TALK ABOUT IT ANOTHER TIME!
SCENE-TO-SCENE: 13.8%
I WONDER WHAT LIES AHEAD FOR ME!
Meanwhile...
OUR MONITORS REPORT A PLANE APPROACHING! FROM THE SOUND OF THE MOTORS IT'S ANOTHER BLACKHAWK PLANE!
EXCELLENT!

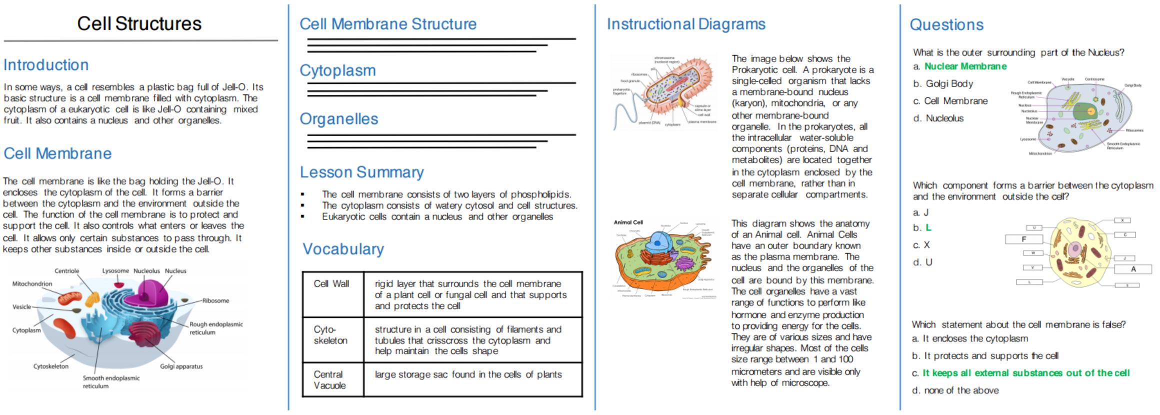
Figure 2.22 A few examples from Comics dataset taken from [38]

comic books published during the “Golden Age” of American comics which spans from 1938 to 1954 period. These comic books are now publicly available because of copyright expiration. The dataset is created from 4000 highest-rated comic books from DCM (http://digitalcomicmuseum.com). The comic books are broken into smaller components and panels are extracted from each comic book leading to 1.2M panels with images and text told in a narrative manner.

TQA. This dataset [52] was constructed using the open-source science curriculum of ck12 [3]. The dataset consists of 1076 lessons from science textbooks and has around 26K questions, of which around 12K of them with an accompanying diagram. The questions have multiple-choice answers ranging from 2 to 7 choices.

TQA is a challenging multi-modal dataset with various science concepts that aim to measure visual and textual data comprehension through the concepts taught in each lesson. The questions are divided into two categories as such, whether they have a diagram accompanying them or not. Thus, a question with a diagram can only be answered with the text and the accompanying diagram. No diagram questions can be answered using only relying on the provided textual context. Although they are not a part of the dataset, lessons have links to

[3] htt://ck12.org



Cell Structures

Introduction

In some ways, a cell resembles a plastic bag full of Jell-O. Its basic structure is a cell membrane filled with cytoplasm. The cytoplasm of a eukaryotic cell is like Jell-O containing mixed fruit. It also contains a nucleus and other organelles.

Cell Membrane

The cell membrane is like the bag holding the Jell-O. It encloses the cytoplasm of the cell. It forms a barrier between the cytoplasm and the environment outside the cell. The function of the cell membrane is to protect and support the cell. It also controls what enters or leaves the cell. It allows only certain substances to pass through. It keeps other substances inside or outside the cell.

Cell Membrane Structure

Cytoplasm

Organelles

Lesson Summary

- The cell membrane consists of two layers of phospholipids.
- The cytoplasm consists of watery cytosol and cell structures.
- Eukaryotic cells contain a nucleus and other organelles

Vocabulary

Cell Wall	rigid layer that surrounds the cell membrane of a plant cell or fungal cell and that supports and protects the cell
Cyto-skeleton	structure in a cell consisting of filaments and tubules that crisscross the cytoplasm and help maintain the cells shape
Central Vacuole	large storage sac found in the cells of plants

Instructional Diagrams

The image below shows the Prokaryotic cell. A prokaryote is a single-celled organism that lacks a membrane-bound nucleus (karyon), mitochondria, or any other membrane-bound organelle. In the prokaryotes, all the intracellular water-soluble components (proteins, DNA and metabolites) are located together in the cytoplasm enclosed by the cell membrane, rather than in separate cellular compartments.

This diagram shows the anatomy of an Animal cell. Animal Cells have an outer boundary known as the plasma membrane. The nucleus and the organelles of the cell are bound by this membrane. The cell organelles have a vast range of functions to perform like hormone and enzyme production to providing energy for the cells. They are of various sizes and have irregular shapes. Most of the cells size range between 1 and 100 micrometers and are visible only with help of microscope.

Questions

What is the outer surrounding part of the Nucleus?
a. **Nuclear Membrane**
b. Golgi Body
c. Cell Membrane
d. Nucleolus

Which component forms a barrier between the cytoplasm and the environment outside the cell?
a. J
b. **L**
c. X
d. U

Which statement about the cell membrane is false?
a. It encloses the cytoplasm
b. It protects and supports the cell
c. **It keeps all external substances out of the cell**
d. none of the above

Figure 2.23 Overview of multi-modal machine comprehension from TQA dataset, taken from [52]

instructional videos, which might help researchers extract additional information about the lessons.

In Fig. 2.23, an overview of multi-model comprehension is illustrated where a lesson is given as a story along with textual and visual information.

3. COMPOSITIONAL GENERALIZATION

In this chapter, we survey compositional generalization literature and highlight current research challenges to lay the grounds for its connection to understanding multimodal how-to instructions based on our work[4]. We only made minor changes to fit the text in the narrative of the thesis and made small corrections in the text.

As the world we live in is inherently structured, many believe that capturing this compositional structure is a key component of artificial intelligence. In recent years, compositional generalization has become increasingly important since neural models are largely incapable of generalizing to novel compositions never observed during training. Hence, researchers have been investigating how the generalization ability of these models can be improved by exploring different aspects of compositionality. In this chapter, we provide a systematic review of the existing works on linguistic compositionality. In particular, we categorize and discuss existing tasks and datasets and examine neural architectures and learning strategies proposed for compositional generalization. Finally, we conclude by pointing out some challenges and opportunities for future research.

Thanks to their compositional skills, humans can effortlessly understand and construct new utterances even though they encounter such novel compositions for the first time. For instance, once we understand the meaning of primitives such as “dax” and “twice”, we can easily infer what “dax twice” means [53]. Since its inception, compositionality has been studied in various aspects and sparked interest in linguistics, philosophy, and vision. As humans can potentially understand an unbounded number of novel utterance combinations by only being exposed to a limited number of them, compositionality is regarded as a key towards solving the generalization problem. In particular, the systematicity aspect of compositionality has been an active research area as a common solution to maintaining the meanings of complex expressions by depending on their simple parts and the way those parts are organized.

[4] Under review

In the last few years, neural models have shown astounding performance on many tasks [54]. Nevertheless, there has been an ongoing debate around neural models' ability to compositionally generalize to unseen instances. Due to their associative nature and inability to accurately represent systematic compositionality, researchers have been debating that neural models are not realistic representations of the human mind [55–58]. Furthermore, despite a wealth of empirical evidence, there is little agreement on whether and how much neural networks can generalize compositionally [53, 59]. Despite the recent successes of neural models, they often cannot capture the compositional structure; therefore, failing to generalize compositionally remains an open research challenge yet to be solved. This open research problem recently garnered much interest from different fields, such as language and vision. Recent work around compositional generalization sparked interest in the contribution of multimodality and grounded language processing toward systematic compositionality [60–62].

In this study, we survey the literature on compositional generalization by categorizing and discussing the existing tasks, datasets, neural models, and learning strategies proposed for improving linguistic compositionality and review related studies from the point of language generation. Moreover, we highlight current limitations and open research problems for future research directions. In particular, we review background work in Sec. 3.1.. Sec. 3.2. outlines the proposed tasks for measuring the compositional generalization abilities of learning systems. In Sec. 3.3., we underline the existing datasets in compositional generalization literature. In Sec. 3.4., we examine the prominent models and the learning strategies in the previous relevant research. Finally, in Sec. 3.5., we debate open research challenges and provide some discussions around the compositional generalization problem.

3.1. Background

Compositionality has been widely studied for many years. It studies ways to define a particular sentence's meaning as a function of its constituents' meaning as well as the rules used to put those parts together. The principle of compositionality, which was introduced by Frege, refers to the meaning of an expression being determined by the meanings of its parts [63]. It deals with describing the relationship between an unbounded number of sentences and a vast set of meanings from a finite set of rules. As a long-standing problem, compositionality has been investigated by the linguistic and philosophy communities and has been studied from different perspectives.

Compositionality was characterized by [64] from a syntactical point of view. From a different point of view, compositionality and systematic generalization have been viewed as standing problems to represent symbolic computation and human cognition [55, 65]. Since the beginning of the compositional generalization debate, researchers have studied the compositional abilities of neural models with mixed results (*e.g.* [53, 56, 66–73]) and little agreement on whether neural models can perform systematic compositionality or to what extent. Much recently, the compositionality problem has been investigated in various settings. [61] constructed the CLEVR-CoGenT dataset based on the CLEVR dataset to test models' ability for compositional generalization on visual reasoning tasks. [74] investigated systematic generalization in a VQA-like setting. [75] studied systematicity and compositionality with a human-like number of examples. [76] examined picking up new concepts and applying them in test time by coupling previously learned concepts with new concepts in a meta-learning setup. [77] explored compositional generalization, to compose unseen combinations of concepts in an image captioning setting. [78] inspected the capacity of artificial neural networks in linguistic compositionality. Though neural models have been shown to generalize well across biased dataset splits, they are criticized for learning surface statistics. Towards addressing this problem, [79] described a heuristic to create similar distributions of primitives and different distributions of combinations to measure models' compositional abilities on novel compositions. [62] explored a 2D grid world setting for situated language understanding using grounded instructions. [80] investigated compositionality in a novel word acquisition setting from narrated videos. [59] proposed an evaluation framework to test models' compositional ability in five dimensions.

3.2. Tasks

Previous studies explored a collection of tasks to test models' compositional ability. Below we highlight the prominent tasks from prior work to evaluate the compositional generalization properties of neural models.

Generalizing to Novel Compositions. The purpose of this task is to assess the compositional skills of models when all primitive components of the input have been observed during training, but the models are tested with novel compositions of primitive components – defined as *systematicity* by [59]. [53] introduced SCAN to measure the systematicity of models in a highly compositional textual navigation setting. They found that popular sequence-to-sequence models do not show systematic generalization skills. [81] inverted the SCAN task by swapping the input commands with action sequence outputs, introducing the NACS task. They confirmed the results of [53]. [79] explored a divergence-based approach and created MCD splits for SCAN, which the authors used to test models' systematicity performance on the MCD splits.

Generalizing to Longer Sequences. [53] introduced a testing framework to assess the ability of recurrent sequence-to-sequence models to generalize. Specifically, the models were trained on shorter action sequences and then evaluated on commands that required longer action sequences, using the SCAN dataset. Similarly, [59] tested the models' generalization abilities of the PCFG-SET in a setup where the aim is to generalize to an unbounded length of sequences.

Machine Translation. [53] experimented with a machine translation (MT) task using a simple setup in which they trained a simple seq2seq model on short sentence pairs where English is the source language, and French is the target language. The authors chose sentences that begin with English phrases and contractions of these phrases. The authors trained a new network with 1,000 repeats of the phrase "I am daxy" to assess compositionality with the addition of a new word. Their motivation was to show that an introduction of a new word in the vocabulary will lead to difficulty in the systematic generalization of models. [82] examined neural machine translation models to understand whether they can solve compositional tasks. As SCAN MT split only contains 8 words, it is limited in its scope. For this, they curated a 216K-sized English-Chinese translation dataset and trained a Transformer based model. They found that even if the model has good metrics under traditional metrics, they fail to generalize compositionally. [83] claim that compositionality in natural language is much more multi-dimensional than rigid, artificially generated datasets, and these datasets often focus on only strongly local, context-independent structures. As natural language involves *atomic compounds* (*e.g.* collocations, idioms) and global relations between words, a fine-tuned local-global approach to compositionality is indispensable. For this purpose, the authors formulate three different experimental setups for systematicity, substitutivity, and overgeneralization. Finally, the authors conclude that as natural language is not as algebraic as artificial data, a non-compositional word translation may not change the meaning of the whole. Hence, they state that MT is a more suitable task for compositionality than artificial transductions.

Table Lookup. [84] formulated the table lookup tasks on seq2seq models to measure their compositional generalization abilities. In this task, the objective is to map bit string inputs to function outputs, hence learning how to apply *functions* in a compositional setup. The authors experimented on the CTL dataset that maps atomic functions with input and output bit strings. A sample lookup task is depicted in Figure 3.1.

Atomic			Atomic			Composed		
g			f			fg		
00	→	01	00	→	11	00	→	0110
01	→	00	01	→	10	01	→	0011
10	→	10	10	→	01	10	→	1001
11	→	11	11	→	00	11	→	1100

Figure 3.1 Two atomic 2-bit lookup tasks f and g and their composition fg. Note that the composition function output starts with the output of the function g.

Image Captioning. The objective of this task is, given an input image, to generate a natural description [85]. Previous work on compositionality in image captioning research focused on triplet prediction [86], where each triplet is defined as subject-relation-object (SRO) present in a visual scene, and the task is predicting unseen SRO combinations successfully in test time. [77] and [87] also focused on only systematicity, but the task is composing unseen combinations of concepts by generating natural language captions. Recently, [88] also investigated models with a broader spectrum of compositional aspects such as productivity, substitutivity, and systematicity. [86] investigated whether state-of-the-art image captioning models that perform well on IID splits can generalize to novel compositions in terms of triplet predictions. The authors created a compositional split from MSCOCO dataset [89] such that test set triplets have zero probability distribution under the train set, but the test set is composed of the same atoms (*e.g*. subjects) as in the train set. They found that state-of-the-art models [90] cannot generalize compositionally. [77] studied compositional generalization from the point of image captioning, focusing on systematicity. They created a new split of MSCOCO dataset composed of novel combinations of adjective-noun and noun-verb pairs [91] and investigated to what extent image captioning models can generalize systematically. The authors found that state-of-the-art image captioning models [90], [92] failed to generalize systematically, mainly due to language generation components based on textual distributions that cannot incorporate a compositional perspective. They achieve better results by creating a multi-task setup combining captioning and image-sentence ranking on top of different attention models [92].

Visual Question Answering. In this task, the reasoning and understanding capabilities of the models are examined by the answers they provide to the questions about visual scenes. Work on compositionality in VQA can be divided into two: where the image distribution in the dataset is changed, or the question distribution in the dataset is changed. Independent of the division, compositional VQA tasks test spatial reasoning, quantification, and comparison in general. CLEVR-CoGenT [61] is a compositional VQA task that requires the listed skills above from a model. The dataset measures compositional skills by using mutually exclusive colors for cubes and cylinders in the train set and swapping these colors in the test set. CLOSURE [93] is another VQA task derived from CLEVR. In CLOSURE, the change in image distribution is modified with a change in question distribution, which increases the complexity of the task by enabling multiple referring expressions. In VQA-CP,

[94] find that VQA models heavily rely on priors in data while producing correct results on non-compositional datasets, which inflates models' compositional understanding of the real-world phenomenon.

Cloze. In a compositional generalization setup, [80] investigated the acquisition of words from visual scenes. They measured how effectively the representations generalize to new verb and noun combinations that aren't part of the training set. They employed the cloze task for model evaluation but require the model to predict both a verb and a noun rather than just one word. The findings were then divided based on whether or not the compositions were visible during training. According to the authors, there is a significant performance difference between compositions that have been seen before and those that haven't. However, the gap is considerably lower because their model is explicitly trained for generalization (nearly twice as small). Additionally, their method significantly outperforms baselines for both known and unique compositions. Furthermore, even though their model is trained on three orders of magnitude less training data than pre-trained BERT [95], their approach can outperform or match BERT's performance.

3.3. Datasets

SCAN. [53] introduced SCAN, a synthetically generated dataset that consists of navigation commands associated with action sequences such that the command *"jump twice"* is mapped to actions `JUMP JUMP` and *"turn around right"* is mapped to `RTURN RTURN RTURN RTURN`. The focus of this paper is the SCAN tasks, which involve translating simplified natural language commands into a sequence of actions. The aim is to evaluate how well recurrent seq2seq models can generalize in this context.

CTL. [84] introduced a setup to measure compositional generalization of neural networks by artificially generating compositional table lookup (CTL) using a simple binary representation *e.g.* if $c(100) = 011$, $g(011) = 001$, $c(101) = 100$, $g(001) = 101$ and $g(100) = 010$, compositions of functions are then applied in the following manner $gc(100) = 001$, $cc(101) = 011$, $gcg(001) = 010$. Lookup tasks were formulated as, given the atomic functions such that g and c, the objective is to predict the output of their composition gc. In particular, the intermediate step's output is used as an input to the composition, where in this case, applying function c, then g to predict the final output gc.

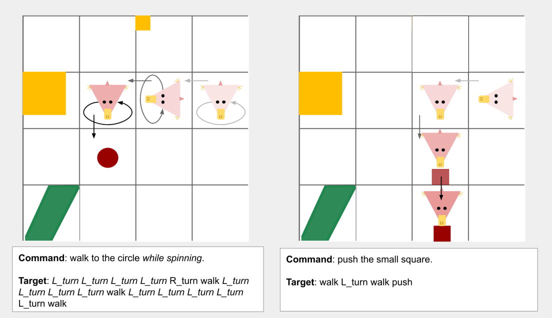
PCFG-SET. [59] introduced a synthetically generated dataset using a probabilistic context-free grammar (PCFG), which generates the sequences where output sequences correspond to the meanings of input sequences. There are three categories of terms in PCFG-SET: a term for unary operations (*e.g.* copy, reverse, swap) and binary operations (*e.g.* append, prepend, remove_first), and elements which the aforementioned operations apply to (*e.g.* A, B, A1, B1), as well as and a separator token (,) to set apart binary function arguments. In particular, for an input sequence *"repeat D E B"* the expected output is `D E B D E B` and for *"append swap C G, repeat H D"* the target is `G C H D H D`.

CFQ. [79] introduced *Compositional Freebase Questions* (CFQ), a procedurally generated dataset, where the objective is translating natural language questions into SPARQL queries against the Freebase. The main idea behind CFQ dataset creation is based on *distribution-based compositionality assessment* (DBCA), where the train and test split display similar atom (*e.g.* entities, question patterns) distributions while the distribution of compounds in train and test splits are maximized using *maximum compound divergence* (MCD) heuristic. It has been empirically shown that there is a significant negative correlation between the model accuracy and the compound divergence.

NACS. [81] introduced NACS by leveraging the SCAN dataset. In particular, they modified the SCAN dataset so that action sequences are translated into navigation commands rather than navigation commands being translated into action sequences, such as translating `LTURN` into "*turn left*", instead of vice versa. This simple modification makes the SCAN dataset more difficult in terms of compositional generalization for seq2seq models.

Mathematics. [96] presented the Mathematics dataset, comprised of mathematics question-answer pairs up to university level in textual form (*e.g.* `solve` $3(x-5) = x+3$ `for x`, where the answer is 9). Compared to other datasets like SCAN, Mathematics focuses on mathematical reasoning rather than language transduction to test the compositional skills of a model. This large, automatically generated dataset is divided into 56 modules (*e.g.* linear equations in two variables, addition or summation, differentiation) and has 2 million training examples with 10K test examples for each module. Moreover, the dataset includes extrapolation modules to test compositional generalization.

gSCAN. [62] presented gSCAN, a dataset based on SCAN [53] for measuring the compositional skills of models in a grounded learning setting. In natural languages the interpretation of linguistic data is grounded in the real world, introducing the contextual sensitivity notion. However, SCAN imitates a navigation task only in textual form, where the goal is to learn an interpretation function with only the linguistic structure. Hence, SCAN lacks this type of contextual sensitivity. gSCAN alleviates this shortcoming by grounding textual information with visual information in a grid world setup (see Figure 3.2), measuring both the compositionality and contextual sensitivity of models.



Command: walk to the circle *while spinning*.

Target: *L_turn L_turn L_turn L_turn* R_turn walk *L_turn L_turn L_turn L_turn* walk *L_turn L_turn L_turn L_turn* L_turn walk

Command: push the small square.

Target: walk L_turn walk push

Figure 3.2 Illustration of a sample from gSCAN showing actions walking while spinning and pushing. Image is taken from [62].

For example, *Walk to the small circle* command and an accompanying visual representation of a grid world with the target object, several distracting objects, and the agent altogether form a data sample. All objects can have different sizes, shapes, and colors, which implies that *small circle* in a visual representation is context-dependent. The introduction of context sensitivity enables testing broader types of compositional generalization.

ReaSCAN. [97] presented ReaSCAN, a dataset based on gSCAN trying to alleviate its limitations. The authors argue that there are four potential downsides of the gSCAN dataset, namely irrelevance of word order, biased distractor object sampling, a small number of effective distractors, and its limits in testing linguistic compositionality. Irrelevance of word order makes bag-of-words models adequate for encoding gSCAN without any compositional understanding. In gSCAN, all compounds are not needed to generate the correct action sequence which raises the question of whether this can help models to attain linguistic compositional skills effectively. Hence, ReaSCAN has a more complex structure achieved by multiple relational clauses and that encourages models to display compositional skills (*e.g.* **gSCAN** → push the big yellow circle cautiously, **ReaSCAN** → push the big yellow circle that is in the same column as a blue square and in the same row as a red circle). Moreover, in ReaSCAN, commands become ambiguous when permuted, subsequently a random distractor sampling is used to make ReaSCAN more robust.

CLEVR-CoGenT. CLEVR [61] is a visual question-answering benchmark that measures a spectrum of visual reasoning skills. As CLEVR is artificially generated, it addresses the problem of inherent biases that can elicit in real-world data. The dataset consists of question-image-answer triplets, where each image is a scene from a 3D-rendered world with multiple objects. The questions require complex multi-step reasoning about the contents of rendered images. The objects are made of two different materials (matte "rubber" and shiny "metal") and can have 8 colors (*e.g.* yellow, red), 2 sizes (small and big), 3 shapes (sphere, cube, and cylinder), and an absolute position in the 3D scene. An example question-image-answer triplet could then be an image of a 3D visual scene (see Figure 3.3), a question such as "*What color is the cube behind the brown cylinder?*" and, the answer is "`yellow`".

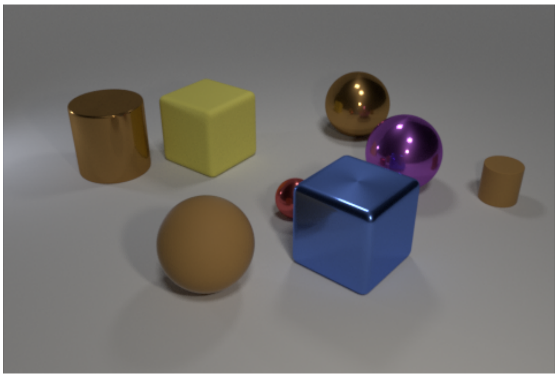


Figure 3.3 Illustration of a visual scene sample from CLEVR dataset. Image taken from [61].

CLEVR tests a range of reasoning abilities such as spatial reasoning, comparison, and quantification. In this dataset, authors proposed the CLEVR-CoGenT split, created to test the compositional generalization abilities of models. In the CLEVR-CoGenT train split, all cubes are can have predetermined colors, whereas cylinder color does not overlap with cube color to measure compositional generalization, and colors between cubes and cylinders are swapped in the test set, while sphere colors can be any color from the given color palette.

CLOSURE. [93] constructed CLOSURE, a visual question-answering dataset based on CLEVR [61], which tests spatial, logical, and quantification skills of a model. While CLEVR-CoGenT tests models' generalization skills under changing image distributions, on the contrary, CLOSURE tests models' generalization skills under changing question distributions caused by the introduction of novel question compounds. The authors argue that the main complexity of CLEVR is how the present objects are referred to and to augment this complexity, they defined 7 different CLOSURE tests. In this scope, authors used three types of *referring expression* (RE) from CLEVR (*e.g.* simple, complex, and logical REs) where RE is a noun phrase referring to one or more objects. Then, by composing novel questions similar to CLEVR with matching object properties (*e.g.* small sphere that is the same color as the big cube) they constructed the CLOSURE dataset. The questions in this dataset follow the same structure as the CLEVR questions. However, the CLOSURE questions have zero probability distribution under CLEVR data.

SQOOP. SQOOP [74] is a synthetic visual question-answering dataset to test the compositional skills of models in terms of spatial reasoning. The dataset is composed of image-question-answer triplets, where each 64x64 image contains 5 distinct alphanumerical characters, and each question has a binary answer (yes/no). Each question has the following structure: ($\texttt{CHAR}_a$ SP_RELATION $\texttt{CHAR}_b$) where CHARs are distinct alphanumeric characters, and SP_RELATION is an element from the set {LEFT_OF, RIGHT_OF, ABOVE, BELOW}. These properties make SQOOP simple and maybe not representative of the real world, but they measure the spatial reasoning of models in an unbiased and isolated manner, which is beneficial.

COGS. [98] introduced a procedurally generated semantic parsing dataset (COGS) based on a subset of English. COGS examples consist of a natural language sentence and a corresponding logical form (*e.g.* A frog hopped. → `frog(x1) AND hop.agent(x2, x1))`. COGS embodies a large spectrum of syntactic structures and semantic representations (by using lambda calculus) that emerge in natural languages. Hence, the authors suggest that COGS can be a better benchmark for compositionality compared to SCAN, as it encompasses more systematic subcomponents that are present in natural languages. The evaluation part of COGS contains several systematic gaps that can only be addressed by compositional generalization such as new combinations of known syntactic structures or new combinations of known words. The dataset consists of three sets: a train set, an independent and identically distributed (IID) validation set, and an out-of-distribution (OOD) test set.

EPIC-Kitchens. [99] introduced an egocentric video benchmark dataset recorded by 32 participants in their native kitchen environments capturing non-scripted daily household activities such as cooking, cleaning, *etc*. This dataset is then extended by [100] to 100 hours of videos captured by 45 participants with denser annotations and fine-grained actions (see Fig. 3.4 for an overview of the dataset).

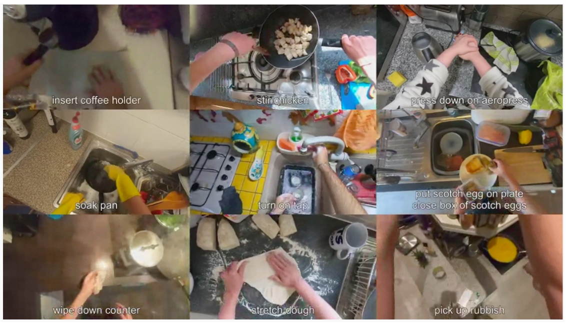


Figure 3.4 Illustration of EPIC-KITCHENS dataset. Image is taken from [100].

Even though the EPIC-Kitchens dataset was introduced mainly for object detection, action recognition, and action anticipation tasks, it was later utilized for measuring the compositional abilities of neural networks on a word acquisition task [80] leveraging its multimodal nature.

VQA-CP. [94] constructed a real-world visual question-answering dataset from the point of compositional generalization where the types of questions in the test split are of different distribution than the types of questions found in the train split. The authors discovered that VQA models exploit the biases in real-world data while achieving near-perfect results on non-compositional datasets and do not show any compositional understanding of underlying phenomena.

3.4. Methods

Through this section, we discuss the prominent models proposed for addressing compositional generalization and highlight learning strategies used in the existing work.

3.4.1. Models

Some researchers have tackled the compositional generalization problem with different neural architectures. In the following, we highlight the most prominent models investigated in the existing studies.

Neural Sequence Models. Recurrent models have been extensively used for text scanning and widely leveraged in the literature for solving various NLP tasks. [84] experimented with vanilla recurrent neural networks (RNN) [101] with no special architectural constraints. [53] described a simple sequence-to-sequence framework to tackle the compositional generalization problem on the SCAN dataset.

Convolutional Neural Models. Although they were first developed for visual data, convolutional neural networks (CNNs) have proven to be highly effective in addressing a variety of problems related to vision [54]. After the early success of CNNs, convolutional architectures have been applied to solving various problems and adopted to process text input in NLP research [102]. [103] proposed a convolutional model for sequence-to-sequence learning where the authors discuss advantages over a recurrent neural model for the seq2seq setting. Recently, [59] explored the compositional generalization abilities of the aforementioned convolutional neural model as a baseline on PCFG SET tasks.

Neural Module Networks. [60] approached compositionality from a structural point of view and proposed a dynamic neural module network for answering queries provided with only triplets of (question, world, answer) as training data, where the model learns to combine neural networks on-the-spot from a catalog of neural models, and learns weights concurrently for these modules in a way that they can be assembled into novel structures. In a similar direction, [104] proposed an end-to-end neural modal for the VQA setup to address the compositional aspect of the task.

Transformers. The transformer architecture was proposed in [105] and has since been widely embraced by the natural language processing (NLP) community. It has been demonstrated to achieve very good results in a range of NLP tasks. The transformer architecture is built with a stack of encoder layers and also in a similar fashion, a stack of decoder layers. In these stacks of encoder and decoder layers, each stack has its corresponding embeddings. The model also utilizes a self-attention mechanism which works as a function for mapping query-key-value triples to an output. Positional encodings in the transformer model, coupled with the self-attention mechanism, allow the model to process input in parallel, which gives the model an advantage over recurrent models, where models need to process tokens in a sequential manner. Recent studies in compositional generalization literature explored different aspects of the Transformer model and its generalization capacity across different compositional tasks, and datasets [106–109]

3.4.2. Learning Strategies

In the literature, there has been a growing body of work in which researchers investigated a different aspect of the compositional generalization problem. Below we review different learning strategies explored in the compositional generalization works.

Meta-learning. [76] investigated systematicity in a meta-learning setup. In this study, the author proposed a seq2seq architecture as a backbone coupled with a memory unit to solve SCAN tasks and demonstrates that memory-augmented neural models can compositionally generalize compositionally in this setup. The proposed setup consists of the meta-training phase where models observe episodes of primitive instructions *e.g.* "jump", "turn left", "look", and "walk" and learn the corresponding meanings for these primitives `JUMP`, `L_TURN`, `LOOK`, `WALK` and in during test phase, support items are fed into the memory, and models are required to learn new meanings of primitives during test time to predict the target commands. Different from the regular seq2seq training setup, the models are provided support items in each training episode, which then are used to form a context to improve models' compositional generalization abilities while also utilizing a self-attention mechanism with an external memory. Another work that leverages meta-learning is [80] on the EPIC-Kitchens [99] and Flickr30K [110] datasets in which the authors provide reference primitives to aid the models both during training time and test time. In this setup, the models are not only trained with training data but also reference instances are provided to support the models. The same procedure is used in the test phase, where the models are tasked with predicting the masked target word while also being provided a set of reference instances that contains the target word. While the objective is to generalize to novel compositions, this study diverges from the usual compositional generalization problem as the main goal is to acquire words from the visual scene and to generalize to unseen compositions of these primitives while being supported with reference instances that the models have access to during both the training as well as the test time.

Supervised Learning. Much recently, [111] formulated the compositional generalization problem as a classification problem by transforming a seq2seq setup into a classification setup using a natural language dataset that requires compositional generalization abilities. In this study, the authors converted the CFQ dataset into a binary classification dataset, where a question and a SPARQL query are the inputs, then the objective is to decide whether the given two inputs have the same meaning or not. Diverging from most of the sequence-to-sequence tasks in the literature, the authors describe a negative sampling strategy and construct negative samples for every input to train and evaluate the models. Here, a random sampling strategy for constructing incorrect predictions was followed by another approach, where a pre-trained network is used to retrieve incorrect examples to choose hard examples aiming to make the task more challenging, hence requiring the models to leverage compositional generalization abilities to correctly predict the target.

Pre-training. Lately, the Transformer model [105] and its variants have been shown to work very well, particularly across many NLP tasks. Thanks to their generalization capacity, transformer-based pre-trained models such as BERT and its variants are hugely adopted in the literature and applied to solving different tasks. These pre-trained models are often trained on large datasets and are widely used for solving different tasks, such as semantic labeling, sentiment classification, and language generation. Some researchers leveraged these models as pre-trained encoders and applied them to solving various tasks in NLP. One of the recent works in this direction is proposed by [106], in which the authors investigated the models' generalization abilities in a compositional generalization setup. They explored how transformer models perform in compositional generalization tasks against different compositional generalization datasets and how the design decisions in transformers, such as position encoding, decoder type, and weight sharing, impact models' compositional generalization abilities. The authors reported that they achieve state-of-the-art for COGS as well as PCFG datasets.

Augmentation. Data augmentation is a strategy recently researchers explored from the point of improving the compositional generalization performance of models. A replacement-based strategy was employed by [112], where they explored a synthetic augmentation procedure to construct examples for downstream tasks such as semantic parsing using compositionality. In this approach, fragments of original data instances are replaced with fragments from other instances where samples are chosen with similar contexts. Another data augmentation strategy was introduced by [113]. In particular, they created new synthetic examples by randomly combining parts of two sentences from the training data to encourage models to rely on compositions of sentence segments to predict the output. [114] proposed a data augmentation procedure to improve compositional generalization in neural sequence models by recombining fragments of training instances to reconstruct other instances and resample model outputs to pick high-quality synthetic samples. Recently, [107] studied one-shot primitive generalization introduced by the SCAN benchmark. They show that typical seq-to-seq models can obtain near-perfect generalization performance by changing the training distribution in simple and intuitive manners. Furthermore, they carried out thorough empirical analyses demonstrating that the traditional seq-to-seq models' generalization capability is largely underestimated. A more general conclusion of their research is that, even though systematicity must be preserved when creating such benchmarks, it is crucial to carefully examine various setup parameters in order to make meaningful judgments about a model's generalization capabilities. The authors of this paper argue that their proposed approach retains the systematic distinctions between the training and testing sets while enhancing the performance of sequence-to-sequence models. Earlier research had disregarded this aspect while trying to improve the compositional generalization capabilities but break the systematicity aspect of the setup.

Prompting. [115] proposed the prompting technique to improve language models' performances for various tasks. In particular, the authors discuss that large language models can be used to solve complex mathematical reasoning tasks. In this work, the authors propose prompting with chain-of-thought, allowing models to use intermediate reasoning steps, hence demonstrating that prompting technique can lead to performance improvement for language models on a different task, in addition to providing better compositional generalization.

3.5. Research Challenges

In this section, we discuss the open research challenges toward understanding compositional generalization. In particular, we review these challenges in the following orthogonal dimensions: *natural datasets*, *violation compositionality*, *a general evaluation benchmark*, *multimodality*, *multilinguality*, and *interpretability*.

Natural Datasets. Much of the previous work investigated compositional generalization problem using artificially generated datasets due to their ease of creation with the predetermined rules and constraints. These synthetic dataset construction strategies followed by many researchers enable them to build larger datasets that adhere to a set of rules and constraints (*e.g*. achieving different distribution for compositions across different splits *etc*.). On the other hand, it remains a challenge to achieve a certain distribution for natural datasets due to the lack of labels or even the existence of certain data. Even though there has been a considerable amount of effort in understanding and assessing compositional generalization problem with natural data, and an interest in this direction, we consider this to be an important open research problem to be addressed in future work.

Violating Compositionality Constraints. We should emphasize that diverging from the regular systematic generalization setup, where the models are expected to generalize to novel compositions and are evaluated against compositional generalization constraints, the pre-training in transformer model training naturally does not adhere to these constraints as the models during their pre-training phase could have been already exposed to the compositions which they are evaluated in test time. Nevertheless, it is still worth investigating the compositional generalization abilities of these models and exploring how pre-trained transformer models perform in various compositional generalization tasks. This could be an interesting direction for transformer-based compositional generalization research for future work. In another direction, researchers have been investigating data augmentation strategies to alleviate the out-of-distribution challenge we observe in the compositional splits. We should highlight that, little has been done in previous studies to enforce the compositional nature of datasets. This particular problem of data augmentation without breaking the compositional aspect of the problem in compositional splits remains an open challenge and is a future research direction.

A General Evaluation Benchmark. As compositional generalization research has been rekindled in the past decade, there is no general evaluation benchmark to test the compositional skills of a model. The lack of well thought general benchmark results in a vicious loop, where a new benchmark is curated to remedy problems in an earlier benchmark, a new symbolic/neural model performs great on this specific dataset while previous models cannot, then a new benchmark is created that the new model does not perform well. For natural language understanding (NLU) and language generation (LG) tasks, there exist de facto general evaluation benchmarks (*e.g.* GLUE for NLU and GEM for LG) [116], [117], [118]. By following a similar path, a systematic way to follow the progress in compositional models can be created. A general evaluation benchmark quite likely would accelerate the research and increase the quality and robustness of future studies.

Multimodality. The contribution of multimodality to the compositional generalization problem has become an interesting research problem researchers recently started to explore. However, due to limitations with natural datasets explored for compositional generalization, whether multimodality can help models systematically generalize and how multimodality can help solve compositional generalization, yet remains an open research problem. Nevertheless, we believe exploring multimodality, in a grounded language understanding and generation setup could be a fruitful research direction as we humans leverage multimodal cues in the real world to generalize compositionally. Toward this goal, we investigate the contribution of multimodality in compositional generalization for how-to instructions in Chapter 6..

Multilinguality. Multilinguality and its connections to compositional generalization is another research direction that remains an under-explored topic yet. There has been little done in the past to understand the impact of models' compositional generalization abilities in multilingual settings, even though there has been a plethora of work related to multilinguality in the past years. From this point of view, this makes understanding models' compositional generalization skills on multilinguality an exciting research direction for future work.

Interpretability. Neural networks have been largely criticized in the literature because of their black-box nature and their lack of interpretability. A body of work focuses on interpretability in machine learning models, but we consider this a significant research problem for compositional generalization since little has been done to understand how models compositionally generalize and why. We consider this would be an interesting research problem to explore for future work.

4. MULTIMODAL COMPREHENSION OF COOKING RECIPES

The ability to understand cooking recipes and to reason about them is an important research area in the quest to enable machines to comprehend procedural knowledge. Throughout this chapter, we present RecipeQA, a benchmark for multimodal comprehension of cooking recipes, which is based on our work,

- RecipeQA: A Dataset for Multimodal Comprehension of Cooking Recipes, S. Yagcioglu, A. Erdem, E. Erdem, N. Ikizler-Cinbis. EMNLP 2018, Brussels, Belgium, Oct. 2018.

We only made minor changes to fit the text in the narrative of the thesis and made small corrections and changes in the text and figures.

RecipeQA is composed of approximately 20,000 instructional cooking recipes that incorporate various modalities, including recipe step titles, step descriptions, and corresponding sets of images. Using over 36,000 question-answer pairs that were automatically generated, we devised a series of comprehension and reasoning tasks that demand a comprehensive understanding of both text and images. These tasks capture the progression of events in temporal order and necessitate the ability to reason about procedural knowledge. Our initial findings suggest that RecipeQA presents a challenging and ideal benchmark for evaluating machine comprehension systems. [5]

There is a rich literature in natural language processing, and information retrieval on question answering (QA) [119], but recently deep learning has sparked interest in a special kind of QA, commonly referred to as reading comprehension (RC) [20]. The main goal of RC research is to build systems to read and make sense of natural language text and as well as to answer questions about the provided text [120]. These tests are attractive because they

[5] The leaderboard and the dataset can be accessed via http://hucvl.github.io/recipeqa.

demand a comprehensive grasp of the question and the related passage (or context) and can objectively evaluate various types of abilities [121].

Although recent years have seen remarkable advances in reading comprehension (RC), there is still a considerable discrepancy between the performance of deep neural models and human abilities. To further our comprehension of the potential and limitations of these approaches, researchers have developed new datasets. The variations in existing RC tasks can be broadly categorized into two aspects: (1) question-answer formats, which can be cloze (fill-in-the-blank), span selection, or multiple choice, and (2) text sources used, which can include news [29, 30], fictional stories [45], and Wikipedia articles [46, 122, 123] or other web sources [124]. A popular topic in computer vision closely related to RC is Visual Question Answering, wherein the context is an image instead of text in the reading comprehension task, such as [12, 18, 51, 125], to name a few.

More recently, research in QA has been extended to focus on the multimodal aspects of the problem where different modalities are being explored. Tapaswi *et al.* [50] introduced MovieQA where they concentrate on evaluating comprehension of stories from both text and video in an automatic manner. In COMICS, Iyyer *et al.* [38] turned to comic books to test understanding of closure, and transitions in the narrative from one panel to the next. In AI2D [126] and FigureQA [127], the authors addressed comprehension of scientific diagrams and graphical plots. Last but not least, Kembhavi *et al.* [52] has proposed another comprehensive and challenging dataset named TQA, which is comprised of middle school science lessons of diagrams and texts.

In this study, we focus on *multimodal machine comprehension of cooking recipes* with images and text. In that regard, we introduce a novel question-answering dataset called *RecipeQA* that consists of recipe instructions and related questions (see Fig. 4.1 for an example text cloze style question).

There are a handful of reasons why understanding and reasoning about recipes is interesting. Recipes are written with a specific goal in mind, which is to teach others how to prepare a particular food. Hence, they contain immensely rich information about the real world.

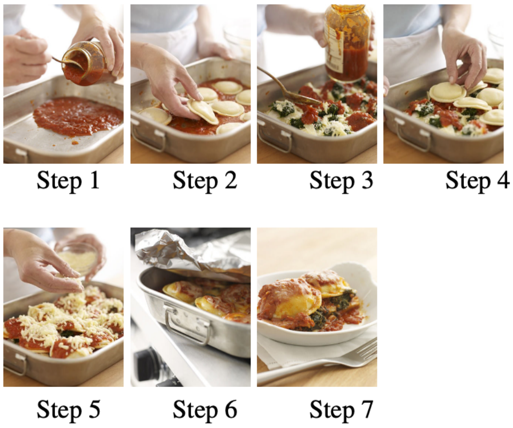
Text Cloze Style Question	Context Modalities: Images and Descriptions of Steps
Recipe: Last-Minute Lasagna 1. Heat oven to 375 degrees F. Spoon a thin layer of sauce over the bottom of a 9-by-13-inch baking dish. 2. Cover with a single layer of ravioli. 3. Top with half the spinach half the mozzarella and a third of the remaining sauce. 4. Repeat with another layer of ravioli and the remaining spinach mozzarella and half the remaining sauce. 5. Top with another layer of ravioli and the remaining sauce not all the ravioli may be needed. Sprinkle with the Parmesan. 6. Cover with foil and bake for 30 minutes. Uncover and bake until bubbly, 5 to 10 minutes. 7. Let cool 5 minutes before spooning onto individual plates.	 Step 1 Step 2 Step 3 Step 4 Step 5 Step 6 Step 7
Question	Choose the best text for the missing blank to correctly complete the recipe Cover. ____________. Bake. Cool, serve.
Answer	**A. Top, sprinkle** B. Finishing touches C. Layer it up D. Ravioli bonus round

Figure 4.1 An illustrative text cloze style question (context, question and answer triplet). The context is comprised of recipe descriptions and images where the question is generated using the question titles. Each paragraph in the context is taken from another step, as also true for the images. The bold answer is the correct one.

Recipes consist of instructions, wherein one needs to follow each instruction to successfully complete the recipe. As a classical example in introductory programming classes, each recipe might be seen as a particular way of solving a task and in that regard can also be considered an algorithm. We believe that recipe comprehension is an elusive challenge and might be seen as an important milestone in the long-standing goal of artificial intelligence and machine reasoning [128, 129].

Among previous efforts towards multimodal machine comprehension [38, 50, 52, 126, 127], our study is closer to what Kembhavi *et al*. [52] envisioned in TQA. Our task primarily differs in utilizing a substantially larger number of images found in the dataset. In RecipeQA, on average we have 12 images per recipe, whereas TQA has only 3 images per question on average. Moreover, in our case, each image is aligned with the text of a particular step in the corresponding recipe. Another important difference is that TQA contains mostly diagrams or textbook images whereas RecipeQA consists of natural images taken by users in unconstrained environments.

Some of the important characteristics of RecipeQA are as follows:

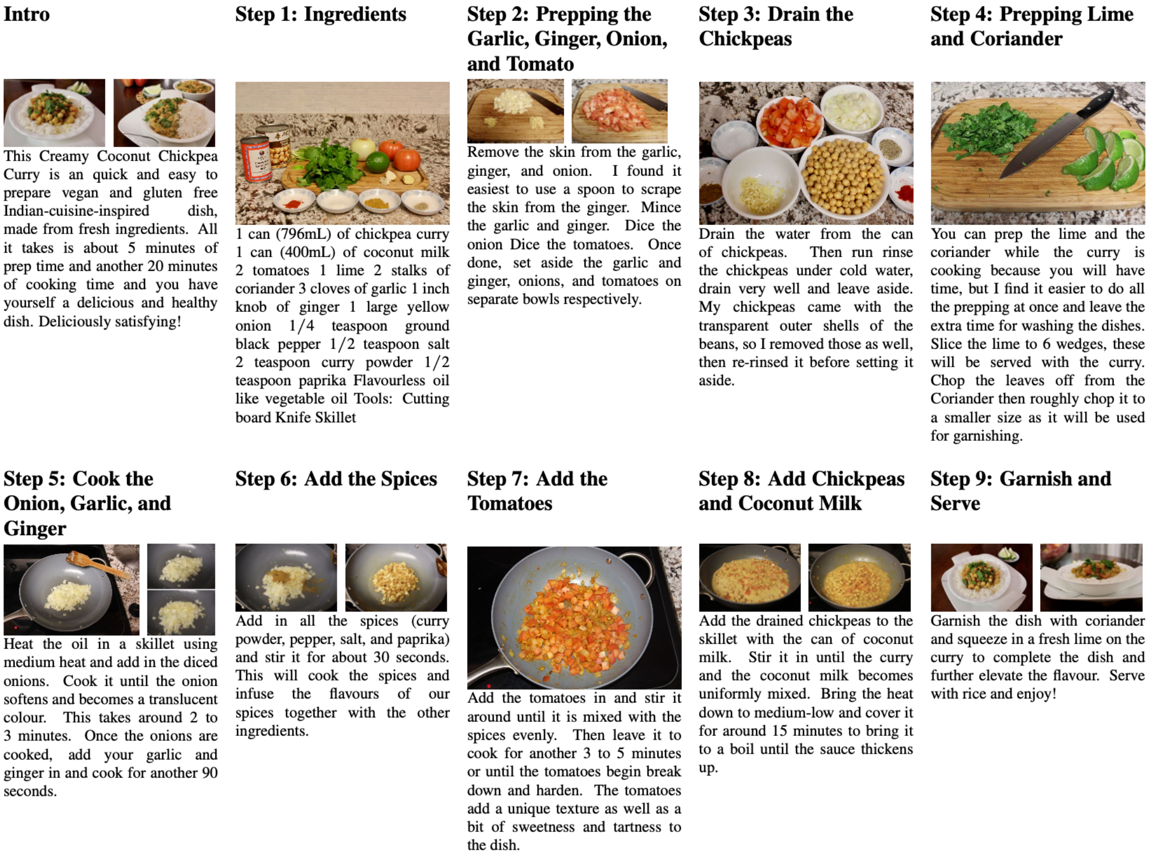
- There are arbitrary numbers of steps in recipes and images in steps, respectively.
- There are different question styles, each requiring a specific comprehension skill.
- The contexts, questions, and answers show a significant level of variation in terms of both word choice and sentence structure.
- The comprehension of procedural language, especially in terms of monitoring entities and/or actions and how they change state, is necessary to provide answers.
- Answers may need information coming from multiple steps (*i.e.* multiple images and multiple paragraphs).
- Answers inherently involve a multimodal understanding of image(s) and text.

In summary, we believe RecipeQA is a challenging dataset that will be useful as a benchmark for assessing the performance of multimodal comprehension systems.

4.1. RecipeQA Dataset

RecipeQA is a challenging multimodal question-answering dataset that evaluates reasoning over real-life cooking recipes [130]. Approximately 20K recipes from 22 different food categories are included in the dataset, along with over 36K questions.

Fig. 4.2 shows an illustrative cooking recipe from our dataset. Each recipe in RecipeQA comprises of multiple steps that comprise both textual and visual components. In particular, each step of a recipe is accompanied by a 'title', a 'description', and a set of illustrative 'images' that are aligned with the title and the description. Each of these elements can



Intro

This Creamy Coconut Chickpea Curry is an quick and easy to prepare vegan and gluten free Indian-cuisine-inspired dish, made from fresh ingredients. All it takes is about 5 minutes of prep time and another 20 minutes of cooking time and you have yourself a delicious and healthy dish. Deliciously satisfying!

Step 1: Ingredients

1 can (796mL) of chickpea curry 1 can (400mL) of coconut milk 2 tomatoes 1 lime 2 stalks of coriander 3 cloves of garlic 1 inch knob of ginger 1 large yellow onion 1/4 teaspoon ground black pepper 1/2 teaspoon salt 2 teaspoon curry powder 1/2 teaspoon paprika Flavourless oil like vegetable oil Tools: Cutting board Knife Skillet

Step 2: Prepping the Garlic, Ginger, Onion, and Tomato

Remove the skin from the garlic, ginger, and onion. I found it easiest to use a spoon to scrape the skin from the ginger. Mince the garlic and ginger. Dice the onion Dice the tomatoes. Once done, set aside the garlic and ginger, onions, and tomatoes on separate bowls respectively.

Step 3: Drain the Chickpeas

Drain the water from the can of chickpeas. Then run rinse the chickpeas under cold water, drain very well and leave aside. My chickpeas came with the transparent outer shells of the beans, so I removed those as well, then re-rinsed it before setting it aside.

Step 4: Prepping Lime and Coriander

You can prep the lime and the coriander while the curry is cooking because you will have time, but I find it easier to do all the prepping at once and leave the extra time for washing the dishes. Slice the lime to 6 wedges, these will be served with the curry. Chop the leaves off from the Coriander then roughly chop it to a smaller size as it will be used for garnishing.

Step 5: Cook the Onion, Garlic, and Ginger

Heat the oil in a skillet using medium heat and add in the diced onions. Cook it until the onion softens and becomes a translucent colour. This takes around 2 to 3 minutes. Once the onions are cooked, add your garlic and ginger in and cook for another 90 seconds.

Step 6: Add the Spices

Add in all the spices (curry powder, pepper, salt, and paprika) and stir it for about 30 seconds. This will cook the spices and infuse the flavours of our spices together with the other ingredients.

Step 7: Add the Tomatoes

Add the tomatoes in and stir it around until it is mixed with the spices evenly. Then leave it to cook for another 3 to 5 minutes or until the tomatoes begin break down and harden. The tomatoes add a unique texture as well as a bit of sweetness and tartness to the dish.

Step 8: Add Chickpeas and Coconut Milk

Add the drained chickpeas to the skillet with the can of coconut milk. Stir it in until the curry and the coconut milk becomes uniformly mixed. Bring the heat down to medium-low and cover it for around 15 minutes to bring it to a boil until the sauce thickens up.

Step 9: Garnish and Serve

Garnish the dish with coriander and squeeze in a fresh lime on the curry to complete the dish and further elevate the flavour. Serve with rice and enjoy!

Figure 4.2 ‘Creamy Coconut Chickpea Curry’ recipe with 9 steps from Instructables website [131].

be considered as a different modality of the data. The questions in RecipeQA explore the multimodal aspects of the step-by-step instructions available in the recipes through a number of specific tasks that are described in Sec. 4.2.. In particular, we have 4 main tasks, which are *visual ordering*, *visual coherency textual cloze*, *visual cloze*.

4.1.1. Data Collection

We consider cooking recipes as the main data source for our dataset. These recipes were collected from Instructables [6], which is a how-to website where users share all kinds of instructions including but not limited to recipes.

[6] All materials from http://instructables.com were downloaded in April 2018.

We employed a set of heuristics that helped us collect high-quality data in an automatic manner. For instance, while collecting the recipes, we downloaded only the most popular recipes by considering their popularity as an objective measure for assessing the quality of a recipe. Our assumption is that the most viewed recipes contain less noise and include easy-to-understand instructions with high-quality illustrative images.

In total, we collected about 20K unique recipes from the food category of Instructables. We filtered out non-English recipes using a language identification [132] and automatically removed the ones with unreadable contents such as the ones that only contain recipe videos. Finally, as a post-processing step, we normalized the description text by removing non-ASCII characters from the text.

4.1.2. Questions and Answers

For machine comprehension and reasoning, forming the questions and the answers is crucial for evaluating the ability of a model in understanding the content. Prior studies employed natural language questions either collected via crowdsourcing platforms such as SQuAD [46] or generated synthetically as in CNN/Daily Mail [30]. Using natural language questions is a good approach in terms of capturing human understanding, but crowdsourcing is often too costly and doesn't scale well as dataset sizes keep growing. Synthetic question generation is a low-cost solution, but the quality of the generated questions is subject to question.

The structured data of cooking recipes in RecipeQA provides step-by-step instructions, which enables us to automatically generate questions with high quality.Our questions test the semantics of the instructions of the recipes from different aspects through the tasks described in Sec. 4.2.. In particular, we generate a set of multiple-choice questions (the number of choices is fixed as four) by following a simple procedure that applies to all of our tasks with slight modifications.

In order to generate question-answer-context triplets, we first filtered out recipes that contain less than 3 steps or more than 25 steps. We also ignored the initial step of the recipes as our preliminary analysis showed that the first step of the recipes is almost always used by the authors to provide a narrative, *e.g.* why they love making that particular food or how it makes sense to prepare food for some occasion and often is not relevant to the recipe instructions. In addition, we automatically removed some indicators such as step numbers, that explicitly emphasize temporal order from the step titles while generating questions.

Given a task, we first randomly select a set of steps from each recipe and construct our questions and answers from these steps according to the task at hand. In particular, we employ the modality that the comprehension task is built upon to generate the candidate answers and use the remaining content as the necessary context for our questions. For instance, if the step titles are used within the candidate answers, the context becomes the descriptions and the images of the steps. As the average number of steps per recipe is larger than four, using this strategy, we can generate multiple context-question-answer triplets from a single recipe.

Candidate answers can be generated by selecting the distractors at random from the steps of other recipes. To make our dataset more challenging, we employ a different strategy and select the distractors from the relevant modalities (titles, descriptions, or images) which are not too far or too close to the correct answer. Specifically, we employ the following simple heuristic. We first find k nearest neighbors ($k = 100$) from other recipes. We then define an adaptive neighborhood by finding the closest distance to the query and removing the candidates that are too close. The remaining candidates are similar enough to be adversarial but not too similar to semantically substitute for the groundtruth. Finally, we randomly sample distractors from that pool. Details of the question generation procedure for each of the tasks are given in Sec. 4.2..

4.1.3. Dataset Statistics

RecipeQA dataset contains approximately 20K cooking recipes and over 36K question-answer pairs divided into four major question types reflecting each of the tasks at hand. The data is split into non-overlapping training, validation, and test sets so that one set does not include a recipe and/or questions about that recipe which are available in other sets. There are 22 different food categories across our dataset whose distribution is shown in Fig. 4.3. While splitting the recipes into sets, we take into account these categories so that all the sets have a similar distribution of recipes across all the categories.

Table 4.1 RecipeQA dataset statistics.

	train	**valid**	**test**
# of recipes	15847	1963	1969
…avg. # of steps	5.99	6.01	6.00
…avg. # of tokens (titles)	17.79	17.40	17.67
…avg. # of tokens (descr.)	443.01	440.51	435.33
…avg. # of images	12.67	12.74	12.65
# of question-answers	29657	3562	3567
…cloze (textual)	7837	961	963
…cloze (visual)	7144	842	848
…coherence (visual)	7118	830	851
…ordering (visual)	7558	929	905

In Table 4.1, we show detailed statistics about our RecipeQA dataset. Moreover, to visualize the token frequencies, we also provide the word clouds of the titles and the descriptions from the recipes in Fig. 4.4.

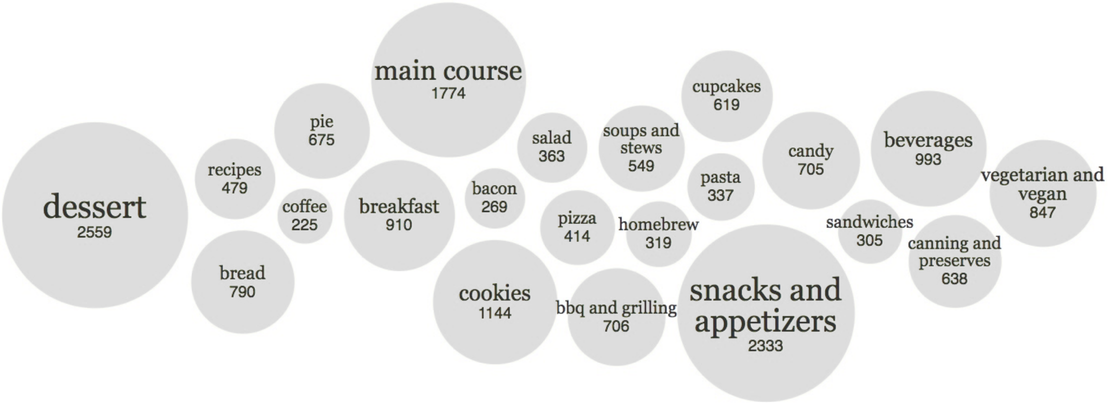


Figure 4.3 Distribution of food categories across RecipeQA.

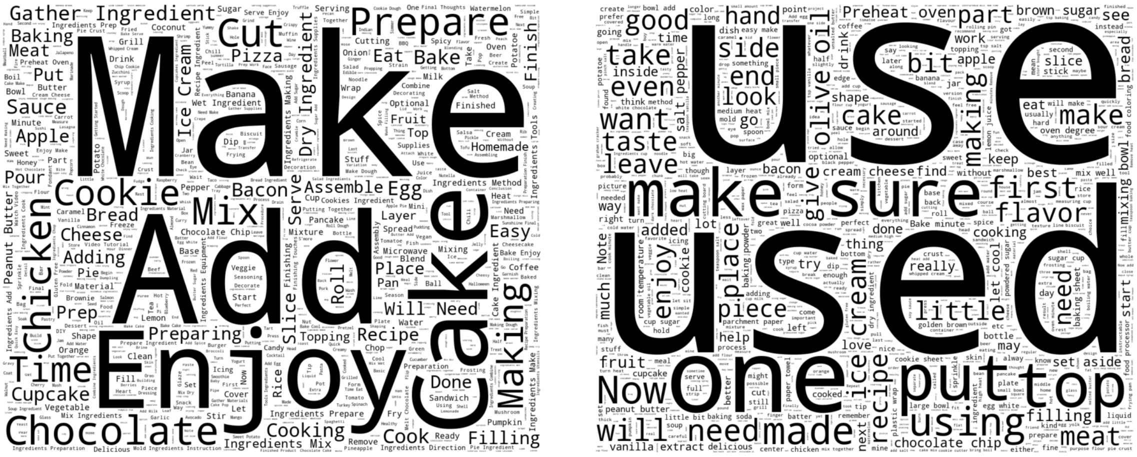


Figure 4.4 Word clouds of the tokens for the titles and the descriptions of the recipes from RecipeQA.

4.2. Tasks

RecipeQA includes four different types of tasks: (1) Cloze (textual), (2) Cloze (visual), (3) Coherence (visual), and (4) Ordering (visual). As discussed in [121], each proposed task requires different reasoning abilities and considers different modalities in their contexts and candidate answer sets. By modalities, we refer to the following pieces of information: (i) titles of steps, (ii) descriptions of steps, and (iii) illustrative images of steps. While generating the questions for these tasks, we rather employ fixed templates as will be discussed below, which helps us to automatically construct question-answer pairs from the recipes with no human intervention. Using these tasks, we can easily evaluate complex relationships between different steps of a recipe via their titles, their descriptions, and/or their illustrative images. Hence, our question-answers pairs are multimodal in nature. In the following, we present a thorough description of each task and discuss our strategies for how we selected candidate answers.

4.2.1. Textual Cloze

Textual cloze-style questions test whether models can infer the missing text either in the title or the description of the steps by taking into account the question's context which includes a set of illustrative images besides the text. When creating the question-answer pairs for RecipeQA, a random step is chosen from the available candidate steps in a recipe. The title and description of that step are hidden, and the question is generated by asking the user to identify the hidden text among several options from the other available modalities. To construct the distractor answers, we use the strategy in Sec. 4.1.2. that depends on the WMD [133] distance measure. In Fig. 4.1, we provide a sample text cloze question from RecipeQA generated automatically in this way.

4.2.2. Visual Cloze

The skill tested in visual cloze-style questions is similar to that of a textual cloze task, but the missing information in this type of task is in the visual domain instead of the text. In this type of task, similar to the textual cloze task, a step from a recipe is chosen at random, and its corresponding image is hidden. Then, the question is asked to identify the hidden image among several multiple-choice options. For this task, the context is solely textual, consisting of a sequence of titles and descriptions. To choose the distractor images, we measure the Euclidean distances between 2048-dimensional *pool5* features extracted from a ResNet-50 model [134] trained on the ImageNet classification task. We show a sample visual cloze style question in Fig. 4.5 (second row).

4.2.3. Visual Coherence

Visual coherence questions assess the ability to identify a visually incoherent image among a set of images that are arranged in accordance with the step titles and descriptions of the associated recipe context. This task requires understanding the sequential flow of events in the recipe and associating it with the corresponding visual content. Therefore, in order to perform well on this task, a system must not only comprehend the connections between the candidate steps, but also align and connect different modalities found in both the context and the answers. While generating the answer candidates for this task, we randomly select a single representative image from a single step and replace this image with a distractor image

Context Modalities: Titles and Descriptions of Steps for Recipe: Bacon Sushi

Step 1: **What You'll Need** This recipe makes enough bacon sushi to feed 2 - 4 people. 2 x 500g(1 lb.) packages of bacon (I chose an applewood smoked bacon, but any type would work)...

Step 2: **Cooking the Bacon** The bacon "nori" will have to be partially cooked before it can be rolled with the risotto filling. Preheat the oven to 350 degrees F. Lay half a package of bacon on...

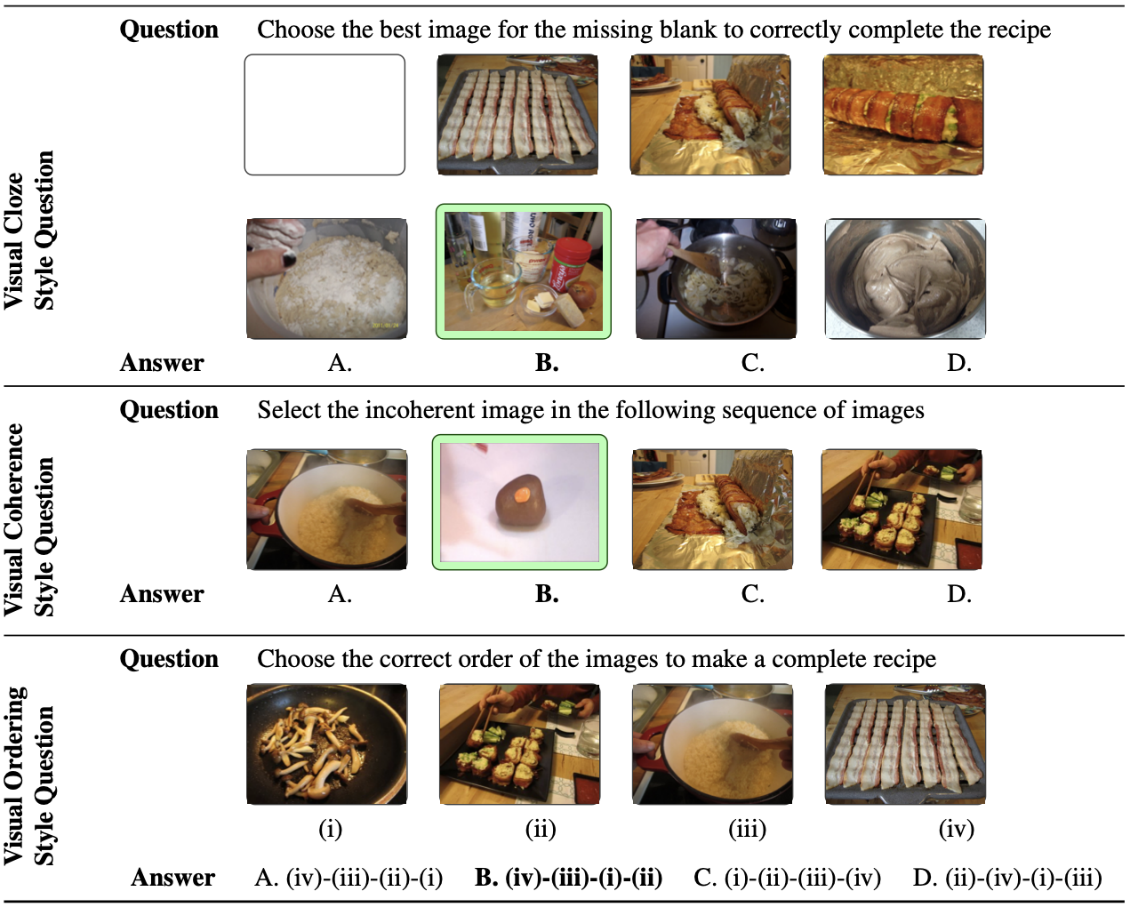
Step 3: **Making the Risotto Filling** I once made risotto with sushi rice, since I had no Arborio rice on hand, and I decided that the starchiness was similar in the two...

Step 4: **Jazzing Up the Risotto** Risotto is a wonderfully customizable dish, and a quick search on the internet will result in a multitude of variations. Here are two of my favorites: Asian mushroom risotto. 1 tbsp. oil. 1 package...

Step 5: **Rolling the Sushi** Cover the sushi rolling mat with a large piece of aluminum foil as protection from the risotto and bacon grease. (You don't want your next sushi dinner tasting like bacon. Or maybe you do...) Lay the stri...

Step 6: **Baking and Slicing** Preheat the oven to 350 degrees F. Place the aluminum foil-covered sushi rolls in the oven and bake for 20 minutes. This will warm all the ingredients and crisp the bacon a little more. It will also melt a...

Step 7: **And You're Done!** Serve the sushi with a light crispy vegetable side dish, such as refreshing cucumber sticks, or a green salad. White wine makes an excellent compliment to the meal, especially if it is the same wine used in ...



Visual Cloze Style Question

Question Choose the best image for the missing blank to correctly complete the recipe

Answer A. **B.** C. D.

Visual Coherence Style Question

Question Select the incoherent image in the following sequence of images

Answer A. **B.** C. D.

Visual Ordering Style Question

Question Choose the correct order of the images to make a complete recipe

Answer A. (iv)-(iii)-(ii)-(i) **B. (iv)-(iii)-(i)-(ii)** C. (i)-(ii)-(iii)-(iv) D. (ii)-(iv)-(i)-(iii)

Figure 4.5 Sample question types (context, question and answer triplet) taken from the RecipeQA training set. The answers that are indicated by green frames or bold formatting are correct answers.

by employing the distractor selection strategy used for the visual cloze task. In Fig. 4.5 (third row), we provide a sample question for this task.

4.2.4. Visual Ordering

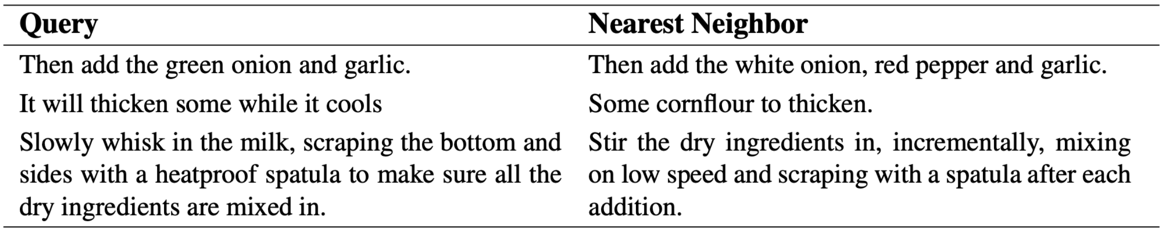
Visual ordering questions assess the capacity of a system to correctly arrange a series of images in a specific order, based on a set of images that have been disordered and represent a cooking recipe. As we mentioned earlier, the context for this task is composed of the step titles and descriptions for a particular recipe. To perform well in this task, the system must understand the temporal sequence of recipe steps and deduce the relationship between candidates from a temporal perspective, such as boiling water first and then adding the spaghetti. Consequently, the series of images should demonstrate an order that corresponds to the recipe. To generate answer choices, we simply use random permutations of the illustrative images in the recipe steps. In Fig. 4.5 (last row), we illustrate this visual ordering task through an example question. Here, we should note that a similar task has been previously investigated by [135] for visual stories where the task is to order a jumbled set of aligned image-description pairs.

4.3. Experiments

4.3.1. Data Preparation

Ingredient Detection. We employed the method proposed in [136] to detect recipe ingredients. To learn more effective word embeddings, we transformed the ingredients with compound words such as *olive oil* into single-word ingredients with a proper hyphenation as *olive_oil*.

Textual Embeddings. We trained a distributed memory model, namely Doc2Vec [137], and used it to learn word-level and document-level embeddings while encoding the semantic similarity by taking into account the word order within the provided context. In this way, we can represent each word, sentence, or paragraph by a fixed-sized vector. In our experiments, we employed 100-d vectors to represent all of the textual modalities (titles and descriptions). We made sure that the embeddings encode semantically useful information by exploring nearest neighbors (see Fig. 4.6 for some examples.)



Query	Nearest Neighbor
Then add the green onion and garlic.	Then add the white onion, red pepper and garlic.
It will thicken some while it cools	Some cornflour to thicken.
Slowly whisk in the milk, scraping the bottom and sides with a heatproof spatula to make sure all the dry ingredients are mixed in.	Stir the dry ingredients in, incrementally, mixing on low speed and scraping with a spatula after each addition.

Figure 4.6 Sample nearest neighbors from the embeddings by the trained Doc2Vec model.

Visual Features. We used the final activation of ResNet-50 [134] architecture trained on ImageNet [138] to extract 2048-d dense visual representations. Then, we further utilized an autoencoder to decrease the dimension of the visual features to 100-d so that they become compatible in size with the text embeddings.

4.3.2. Baseline Models

Neural Baselines. We modified the Impatient Reader model described in a previous study by Hermann et al. [30], which was initially designed for answering cloze-style text comprehension questions in the CNN/Daily Mail dataset, for our neural baseline models. In our implementation, we used a uni-directional stacked LSTM architecture with 3 layers, in which we feed the question context sequentially to the model. Particularly, we preserve the temporal order of the steps of the recipe while feeding it to the neural model by mimicking the most common reading strategy – reading from top to bottom. For the multimodal setting, since images are represented with vectors that are of the same size as the text embeddings, we also feed the images to the network in the same order they are presented in the recipe.

In order to account for different question types, we employ a modular architecture, which requires small adjustments to be made for each task. For instance, we place the candidate answers into the query for the cloze style questions or remove the candidate answer from the query for the visual coherence type questions. In training our Impatient Reader baseline model, we use a cosine similarity function and employed the *hinge ranking loss* [139] as follows:

$$L = \max\{0, M - cos(q, a_+) + cos(q, a_-)\} \tag{1}$$

in which M denotes a scalar that corresponds to a margin, a_+ represents the ground truth answer, and a_- corresponds to an incorrect answer which is sampled randomly from the whole answer space. For all of our experiments, we select M as 1.5 and employ a simple heuristic to prevent overfitting by following an early stopping scheme with patience set to 10 against the validation set accuracy after the initial epoch. We utilize ADAM optimizer and use $1e-3$ for the learning rate. The training took around 18 to 24 hours on GTX 1080Ti on a single GPU. We did not perform any hyperparameter tuning.

Simple Baselines. We adopt the Hasty Student baseline, which was initially introduced in [50], to work with RecipeQA. Unlike other models, the Hasty Student model does not take into account the provided context and answers questions by merely assessing the similarity between the candidate answers and questions. For the textual close task, each candidate answer is compared against the titles or descriptions of the steps by using WMD [133] distance, where such distances are averaged. Then, the choice closest to all of the question steps is selected as the final answer. For the visual cloze task, a similar approach is carried out by considering images instead of text using deep visual features. In visual coherence, as the objective is to identify an incoherent image from among other images, the answer is chosen as the most dissimilar one to the remaining images on average. Lastly, for the visual ordering task, first, the distances between each consecutive image pair in a candidate ordering of the jumbled image set are estimated. Then, each candidate ordering is scored based on the average of these pairwise distances, and the choice with the minimum average distance is set as the final answer. In all these simple baseline models, we utilize cosine similarity to rank the candidates.

4.3.3. Baseline Results

We report the performance of our baselines in Table 4.2 which presents the proportion of correct answers out of the total number of questions in the test. In other words, we use the accuracy metric for calculating the model performance on these tasks, considering these tasks are multiple choice questions each consisting of 4 candidate answers per question.

Table 4.2 Results for simple and neural models on the test set of RecipeQA dataset.

	Visual Cloze	Textual Cloze	Visual Coherence	Visual Ordering
Hasty Student	27.35	26.89	**65.80**	**40.88**
Impatient Reader (Text only)	–	28.03	–	–
Impatient Reader (Multimodal)	**27.36**	**29.07**	28.08	26.74

For the textual cloze, the comparison between text-only and multimodal Impatient Reader models shows that the additional visual modality helps the model to understand the question better and to provide more accurate answers. While for the cloze style questions, the Impatient Reader outperforms the Hasty student, for the visual coherence and visual ordering style questions Hasty student gives way better results. This demonstrates that better neural models are needed to be able to deal effectively with these kinds of questions. In Fig. 4.7 and in Fig. 4.8 we provide some qualitative examples.

4.4. Related Work

Question Answering has been studied extensively in the literature. With the success of deep learning approaches in question answering, comprehension, and reasoning aspects of the task has attracted researchers to investigate QA as a medium to measure intelligence. Various datasets and methods have been proposed for measuring different aspects of the comprehension and reasoning problem. Each dataset has its own merits as well as weaknesses. Recently, a thorough analysis by [31] revealed that the required reasoning and inference level was quite simple for CNN/Daily Mail dataset [30]. To make reasoning task

Context Modalities: Images and Descriptions of Steps

Recipe: Grans-Green-Tomato-Chutney

Step 1: Ingredients: 2.5kg green tomatoes, roughly chopped 0.5kg onions, finely sliced 4 tsp / 30g salt1L malt vinegar 0.5kg soft light brown sugar 250g sultanas, 1 1...

Step 2: Finely slice your onions and washed green tomatoes, cutting out any bad bits. Add to a large bowl and stir. Add the 4 teaspoons of salt, stir again and then cover with food wrap or a large plate and leave overnight. This will draw out lots of the tomato juices and...

Step 3: The next day...Place the litre of vinegar into a large pan. Add the 500g of light brown soft sugar and stir over a medium heat until all the sugar has dissolved. Bring to the boil....

Step 4: ...

⋮

Step 9: While the jars cool, write some labels showing the date, content and maker. Once cool, add the lids and stick on the labels. ...

Textual Cloze Style Question

Question Choose the best text for the missing blank to correctly complete the recipe

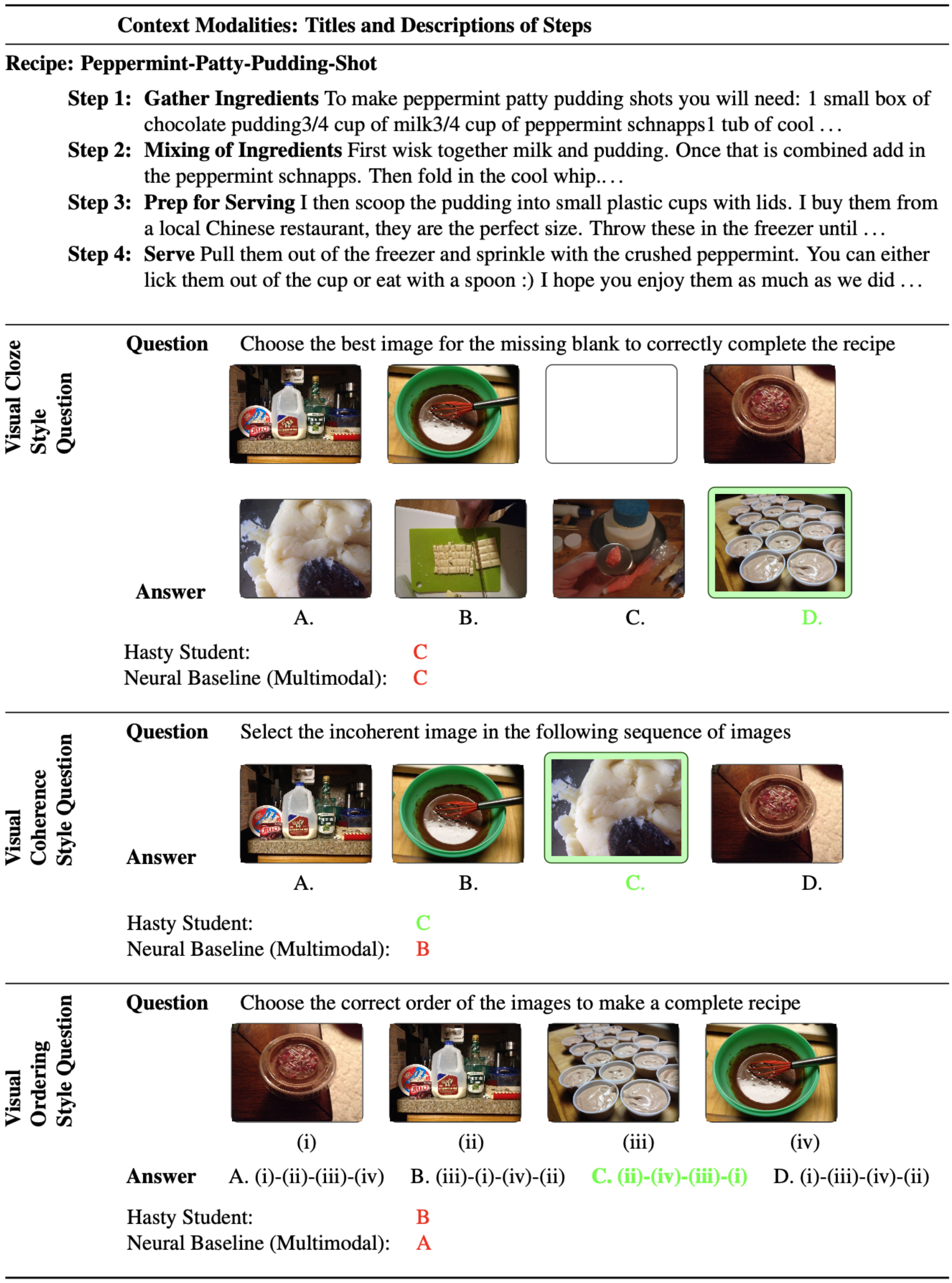
Ingredients. _________. Drain and Add the Tomatoes and Onions. Preparing Your Jars.

Answer

A. Sultanas B. Spicy Tomato Chutney. C. Cover and Slice. D. Enjoy.

Hasty Student:	Cover and Slice
Neural Baseline (Text only):	Sultanas
Neural Baseline (Multimodal):	Sultanas

Figure 4.7 Sample groundtruth and model prediction results for a textual cloze style question (context, question and answer triplet) taken from the RecipeQA test set (Question Id: 1000-12665-0-3-4-6). Here, the context is comprised of step descriptions and images where the questions are generated using the step titles in the recipe. Correct answer for the question is highlighted in green. Answers selected by neural models are correct, marked as green whereas Hasty Student's prediction is wrong and marked as red.



Context Modalities: Titles and Descriptions of Steps

Recipe: Peppermint-Patty-Pudding-Shot

Step 1: **Gather Ingredients** To make peppermint patty pudding shots you will need: 1 small box of chocolate pudding3/4 cup of milk3/4 cup of peppermint schnapps1 tub of cool . . .

Step 2: **Mixing of Ingredients** First wisk together milk and pudding. Once that is combined add in the peppermint schnapps. Then fold in the cool whip.. . .

Step 3: **Prep for Serving** I then scoop the pudding into small plastic cups with lids. I buy them from a local Chinese restaurant, they are the perfect size. Throw these in the freezer until . . .

Step 4: **Serve** Pull them out of the freezer and sprinkle with the crushed peppermint. You can either lick them out of the cup or eat with a spoon :) I hope you enjoy them as much as we did . . .

Visual Cloze Style Question

Question Choose the best image for the missing blank to correctly complete the recipe

Answer A. B. C. D.

Hasty Student: C
Neural Baseline (Multimodal): C

Visual Coherence Style Question

Question Select the incoherent image in the following sequence of images

Answer A. B. C. D.

Hasty Student: C
Neural Baseline (Multimodal): B

Visual Ordering Style Question

Question Choose the correct order of the images to make a complete recipe

(i) (ii) (iii) (iv)

Answer A. (i)-(ii)-(iii)-(iv) B. (iii)-(i)-(iv)-(ii) C. (ii)-(iv)-(iii)-(i) D. (i)-(iii)-(iv)-(ii)

Hasty Student: B
Neural Baseline (Multimodal): A

Figure 4.8 Sample question types taken from the RecipeQA test set. The correct answers are indicated by green frames or text highlighted in green. Wrong answers are marked as red.

more realistic, new datasets such as SQuAD [46], NewsQA [29], MSMARCO [37], CLEVR [51], COMICS [38] and FigureQA [127] have been proposed.

In the following, we briefly discuss the publicly available datasets that are closely related to our problem and provide an overview in Table 4.3.

Table 4.3 Comparison of the RecipeQA dataset to other multimodal machine comprehension datasets.

Dataset	#Images	#Questions	Modality
COMICS	1.2M	750K	Image/Text
MovieQA	408	14,944	Image/Video/Text
TQA	3,455	26,260	Image/Text
RecipeQA	250,730	36,786	Image/Text

The closest works to ours are [38], [50] and [52] where data multi-modality is the key aspect. COMICS dataset [38] focuses on comic book narratives and explores visual cloze style questions, introducing a dataset consisting of drawings from comic books. The dataset is constructed from 4K Golden Age (1938-1954) comic books from the Digital Comics Museum and contains 1.2M panels with 2.5M textboxes. Three tasks are evaluated in this context, namely text cloze, visual cloze, and character coherence. MovieQA dataset [50], comprises 15K crowdsourced questions about 408 movies. It consists of movie clips, subtitles, and snapshots and is about comprehending stories about movies. TQA dataset [52], has 26K questions about 1K middle school science lessons with 3.5K images, mostly of diagrams, and aims at addressing middle school knowledge acquisition using both images and text. Since the audience is middle school children, it requires limited reasoning.

RecipeQA substantially differentiates from the previous work in the following way. Our dataset consists of natural images that are taken by anonymous users in unconstrained environments, which is a major diversion from COMICS and TQA datasets.

It should also be noted that there has been a long history of research involving cooking recipes. Recent examples include parsing of recipes [140, 141], aligning instructional text to videos [142, 143], recipe text generation [144], learning cross-modal embeddings [136], tracking entities and action transformations in recipes [145].

Finally, to our best knowledge, there is no dataset focusing on “how-to” instructions or recipes; hence, this work will be the first to serve multimodal comprehension of recipes having an arbitrary number of steps aligned with multiple images and multiple sentences.

4.5. Discussions

In this chapter, we presented RecipeQA dataset, which consists of roughly 20K cooking recipes with over 36K context-question-answer triplets. To our best knowledge, RecipeQA is the first machine comprehension dataset that deals with understanding procedural knowledge in a multimodal setting. Each one of the four question styles in our dataset is specifically tailored to evaluate a particular skill and requires connecting the dots between different modalities. Results of our baseline models demonstrate that RecipeQA is a challenging dataset and hopefully will be useful for other researchers to promote the development of new methods for multimodal machine comprehension. We also hope that RecipeQA will serve other purposes for related research problems on cooking recipes as well.

5. PROCEDURAL UNDERSTANDING OF COOKING RECIPES

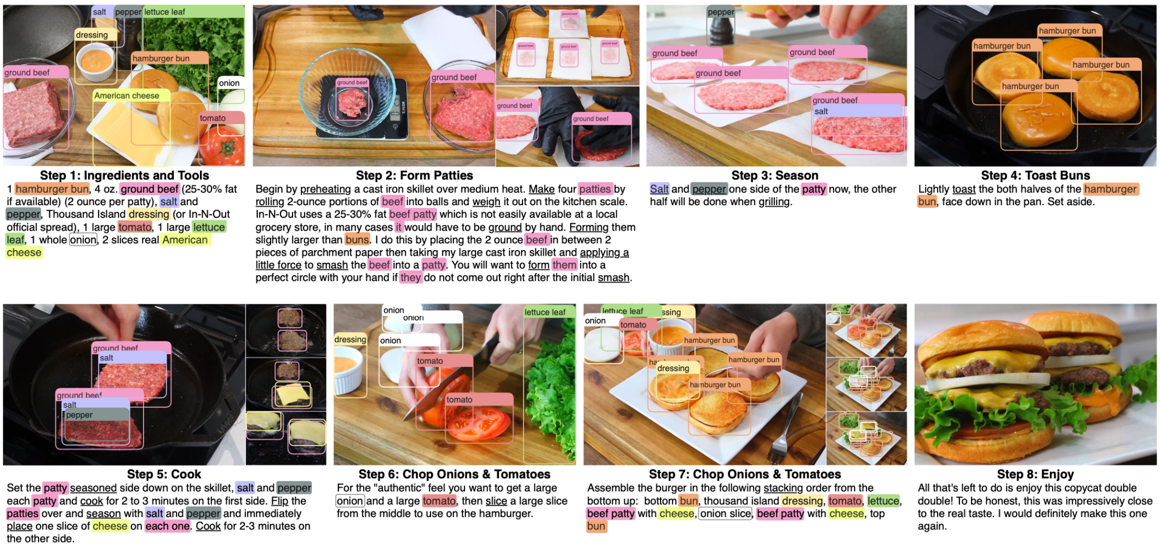
In this chapter, we explore understanding multimodal procedural knowledge and describe a novel architecture for it. In particular, we propose Procedural Reasoning Networks (PRN) for understanding procedural commonsense knowledge, based on our work,

- Mustafa Sercan Amac, Semih Yagcioglu, Aykut Erdem, and Erkut Erdem. Procedural reasoning networks for understanding multimodal procedures. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 441–451. 2019.

We only made minor changes to fit the text in the narrative of the thesis and made small corrections and changes in the text and figures and added further discussion regarding related work that reports results with the RecipeQA dataset.

Procedural understanding of cooking recipes is a challenging problem as it involves identifying various entities, establishing their temporal and causal relationships, and tracking changes in their states. In this study, differing from most of the previous work, we investigate how can multimodality be used for providing a complementary semantic signal without relying on strong inductive biases. In order to accomplish this, we introduce an entity-aware novel neural model which is equipped with an external relational memory unit that helps to monitor the changes in the state of entities. The proposed model learns to update entity states while reading the textual instructions considering each entity's relation to the other. The experimental analysis on RecipeQA [130] visual reasoning tasks shows that the proposed approach significantly outperforms the previous baselines. Additionally, through further analysis, we discover that the proposed model can learn effective dynamic entity representations without relying on any entity-level supervision.

A tremendous amount of commonsense knowledge about our world is procedural in nature and involves steps that show ways to achieve specific goals. Understanding and reasoning



Step 1: Ingredients and Tools
1 hamburger bun, 4 oz. ground beef (25-30% fat if available) (2 ounce per patty), salt and pepper, Thousand Island dressing (or In-N-Out official spread), 1 large tomato, 1 large lettuce leaf, 1 whole onion, 2 slices real American cheese

Step 2: Form Patties
Begin by preheating a cast iron skillet over medium heat. Make four patties by rolling 2-ounce portions of beef into balls and weigh it out on the kitchen scale. In-N-Out uses a 25-30% fat beef patty which is not easily available at a local grocery store, in many cases it would have to be ground by hand. Forming them slightly larger than buns. I do this by placing the 2 ounce beef in between 2 pieces of parchment paper then taking my large cast iron skillet and applying a little force to smash the beef into a patty. You will want to form them into a perfect circle with your hand if they do not come out right after the initial smash.

Step 3: Season
Salt and pepper one side of the patty now, the other half will be done when grilling.

Step 4: Toast Buns
Lightly toast the both halves of the hamburger bun, face down in the pan. Set aside.

Step 5: Cook
Set the patty seasoned side down on the skillet, salt and pepper each patty and cook for 2 to 3 minutes on the first side. Flip the patties over and season with salt and pepper and immediately place one slice of cheese on each one. Cook for 2-3 minutes on the other side.

Step 6: Chop Onions & Tomatoes
For the "authentic" feel you want to get a large onion and a large tomato, then slice a large slice from the middle to use on the hamburger.

Step 7: Chop Onions & Tomatoes
Assemble the burger in the following stacking order from the bottom up: bottom bun, thousand island dressing, tomato, lettuce, beef patty with cheese, onion slice, beef patty with cheese, top bun

Step 8: Enjoy
All that's left to do is enjoy this copycat double double! To be honest, this was impressively close to the real taste. I would definitely make this one again.

Figure 5.1 A recipe for preparing a cheeseburger (adapted from the cooking instructions available at `https://instructables.com/id/In-N-Out-Double-Double-Cheeseburger-Copycat`). Each basic ingredient (entity) is highlighted by a different color in the text and with bounding boxes on the accompanying images. Over the course of the recipe instructions, ingredients interact with each other and change their states with each cooking action (underlined in the text), which in turn alter the visual and physical properties of entities. For instance, the *tomato* changes its form by being *sliced up* and then *stacked* on a *hamburger bun*.

about procedural texts (*e.g.* cooking recipes, how-to guides, scientific processes) are very hard for machines as it demands modeling the intrinsic dynamics of the procedures [130, 146, 147]. That is, one must be aware of the entities present in the text, infer relations among them, and even anticipate changes in the states of the entities after each action. For example, consider the cheeseburger recipe presented in Fig. 5.1. The instruction "*salt and pepper each patty and cook for 2 to 3 minutes on the first side*" in Step 5 entails mixing three basic ingredients, the *ground beef*, *salt* and *pepper*, together and then applying heat to the mix, which in turn causes chemical changes that alter both the appearance and the taste. From a natural language understanding perspective, the main difficulty arises when a model sees the word *patty* again at a later stage of the recipe. It still corresponds to the same entity, but its form is totally different.

Over the past few years, many new datasets and approaches have been proposed that address this inherently hard problem [146–149]. To mitigate the aforementioned challenges, the existing works rely mostly on heavy supervision and focus on predicting the individual state changes of entities at each step. Although these models can accurately learn to make local predictions, they may lack global consistency [148, 149], not to mention that building such annotated corpora is very labor-intensive. In this work, we take a different direction and explore the problem from a multimodal standpoint. Our basic motivation, as illustrated in Fig. 5.1, is that accompanying images provide complementary cues about causal effects and state changes. For instance, it is quite easy to distinguish raw meat from cooked one in the visual domain.

In recent years, tracking entities and their state changes have been explored in the literature from a variety of perspectives. In an early work, [150] proposed a dynamic memory-based network that updates entity states using a gating mechanism while reading the text. [151] presented a more structured memory-augmented model which employs memory slots for representing both entities and their relations. Pavez *et al.* [152] suggested a conceptually similar model in which the pairwise relations between attended memories are utilized to encode the world state. The primary distinction between these works and our method is that by utilizing relational memory core units, we also allow memories to interact with each other during each update.

Perez *et al.* [153] showed that similar ideas can be used to compile supporting memories in tracking dialogue state. Wang *et al.* [154] has shown the importance of coreference signals for reading comprehension tasks. More recently, [155] introduced a specialized recurrent layer that uses coreference annotations for improving reading comprehension tasks. On the language modeling task, [156] proposed a language model which can explicitly incorporate entities while dynamically updating their representations for a range of tasks *e.g.* language modeling, entity prediction, and coreference resolution.

Our work builds upon and contributes to the growing literature on tracking state changes in procedural text. Bosselut *et al.* [146] presented a neural model that can learn to explicitly predict state changes of ingredients at different points in a cooking recipe. Mishra *et al.* [147] proposed another entity-aware model to track entity states in scientific processes. Tandon *et al.* [148] demonstrated that the prediction quality can be boosted by including hard and soft constraints to eliminate unlikely or favor probable state changes. In follow-up work, [149] exploited the notion of label consistency in training to enforce similar predictions in similar procedural contexts. Das *et al.* [157] proposed a model that dynamically constructs a knowledge graph while reading the procedural text to track the ever-changing entities states. As discussed in the introduction, however, these previous methods use a strong inductive bias and assume that state labels are present during training. In our study, we deliberately focus on unlabeled procedural data and ask the question: Can multimodality help to identify and provide insights into understanding state changes?

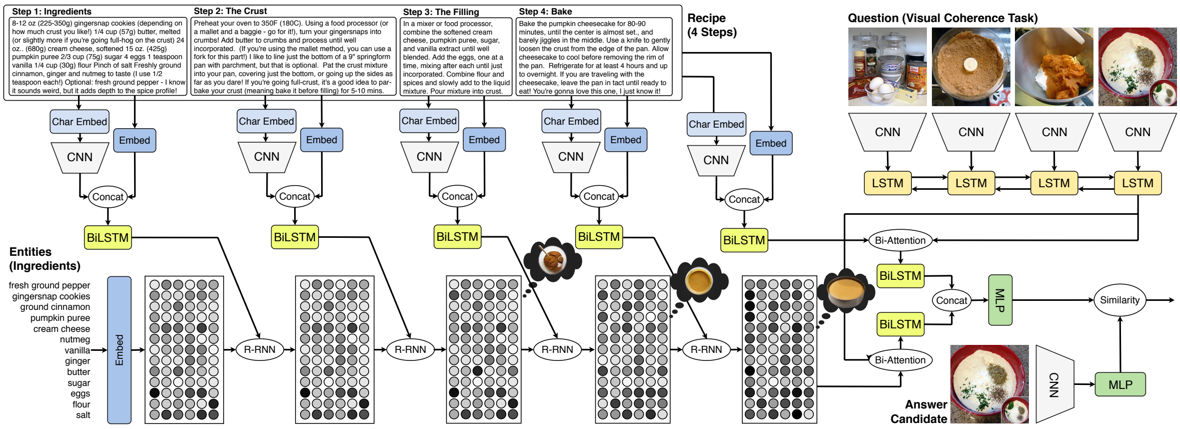
In particular, we take advantage of the proposed RecipeQA dataset [130] and explore whether it is possible to have a model which employs dynamic representations of entities in answering questions that requires a multimodal understanding of procedures. To this end, inspired from [158], we propose Procedural Reasoning Networks (PRN) [159] that incorporate entities into the comprehension process and allow us to keep track of entities, understand their interactions, and accordingly update their states across time. We report that our proposed approach significantly improves upon previously published results on visual reasoning tasks in RecipeQA, which test understanding causal and temporal relations from images and text. We further show that the dynamic entity representations can capture the semantics of the state information in the corresponding steps.

5.1. Visual Reasoning in RecipeQA

In our study, we particularly focus on 3 different visual reasoning tasks proposed in RecipeQA, namely *cloze*, *coherence*, and *ordering* tasks, each of which examines a different reasoning skill [7]. We briefly describe these tasks below.

Visual Cloze. In this task, the question is formed by a sequence of four images from consecutive steps of a recipe, where one of them is replaced by a placeholder. A model

[7] We intentionally leave the textual cloze task out from our experiments as the questions in this task does not necessarily need multimodality.



Step 1: Ingredients

8-12 oz (225-350g) gingersnap cookies (depending on how much crust you like!) 1/4 cup (57g) butter, melted (or slightly more if you're going full-hog on the crust) 24 oz.. (680g) cream cheese, softened 15 oz. (425g) pumpkin puree 2/3 cup (75g) sugar 4 eggs 1 teaspoon vanilla 1/4 cup (30g) flour Pinch of salt Freshly ground cinnamon, ginger and nutmeg to taste (I use 1/2 teaspoon each!) Optional: fresh ground pepper - I know it sounds weird, but it adds depth to the spice profile!

Step 2: The Crust

Preheat your oven to 350F (180C). Using a food processor (or a mallet and a baggie - go for it!), turn your gingersnaps into crumbs! Add butter to crumbs and process until well incorporated. (If you're using the mallet method, you can use a fork for this part!) I like to line just the bottom of a 9" springform pan with parchment, but that is optional. Pat the crust mixture into your pan, covering just the bottom, or going up the sides as far as you dare! If you're going full-crust, it's a good idea to par-bake your crust (meaning bake it before filling) for 5-10 mins.

Step 3: The Filling

In a mixer or food processor, combine the softened cream cheese, pumpkin puree, sugar, and vanilla extract until well blended. Add the eggs, one at a time, mixing after each until just incorporated. Combine flour and spices and slowly add to the liquid mixture. Pour mixture into crust.

Step 4: Bake

Bake the pumpkin cheesecake for 80-90 minutes, until the center is almost set., and barely jiggles in the middle. Use a knife to gently loosen the crust from the edge of the pan. Allow cheesecake to cool before removing the rim of the pan. Refrigerate for at least 4 hours and up to overnight. If you are traveling with the cheesecake, leave the pan in tact until ready to eat! You're gonna love this one, I just know it!

Figure 5.2 An illustration of our Procedural Reasoning Networks (PRN). For a sample question from the visual coherence task in RecipeQA, while reading the cooking recipe, the model constantly performs updates on the representations of the entities (ingredients) after each step and makes use of their representations along with the whole recipe when it scores a candidate answer.

should select the correct one from a multiple-choice list of four answer candidates to fill in the missing piece. In that regard, the task inherently requires aligning visual and textual information and understanding temporal relationships between the cooking actions and the entities.

Visual Coherence. The goal of this task is to identify the inconsistent image within a sequence of four images considering the textual instructions of a cooking recipe. To succeed in this task, a model should have a clear understanding of the procedure described in the recipe and at the same time connect language and vision.

Visual Ordering. The visual ordering task is about grasping the temporal flow of visual events with the help of the given recipe text. The questions show a set of four images from the recipe and the task is to sort jumbled images into the correct order. Here, a model needs to infer the temporal relations between the images and align them with the recipe steps.

5.2. Procedural Reasoning Networks

In the following, we explain our Procedural Reasoning Networks model [159]. Its architecture is based on a bi-directional attention flow (BiDAF) model [160][8] but also equipped with an explicit reasoning module that acts on entity-specific relational memory units. Fig. 5.2 presents the overview of the proposed neural architecture which comprises five main modules: An input module, an attention module, a reasoning module, a modeling module, and an output module. Note that the question-answering tasks we consider here are multimodal in that while the context is a procedural text, the question and the multiple-choice answers are composed of images.

1. **Input Module** extracts vector representations of inputs at different levels of granularity by using several different encoders.
2. **Reasoning Module** scans the procedural text and tracks the states of the entities and their relations through a recurrent relational memory core unit [158].
3. **Attention Module** computes context-aware query vectors and query-aware context vectors as well as query-aware memory vectors.
4. **Modeling Module** employs two multi-layered RNNs to encode previous layers' outputs.
5. **Output Module** scores a candidate answer from the given multiple-choice list.

At a high level, as the model is reading the cooking recipe, it continually updates the internal memory representations of the entities (ingredients) based on the content of each step – it keeps track of changes in the states of the entities, providing an entity-centric summary of the recipe. The response to a question and a possible answer depends on the representation of the recipe text as well as the last states of the entities. All this happens in a series of implicit relational reasoning steps and there is no need for explicitly encoding the state in terms of a predefined vocabulary.

[8] Our implementation is based on the implementation publicly available in AllenNLP [160].

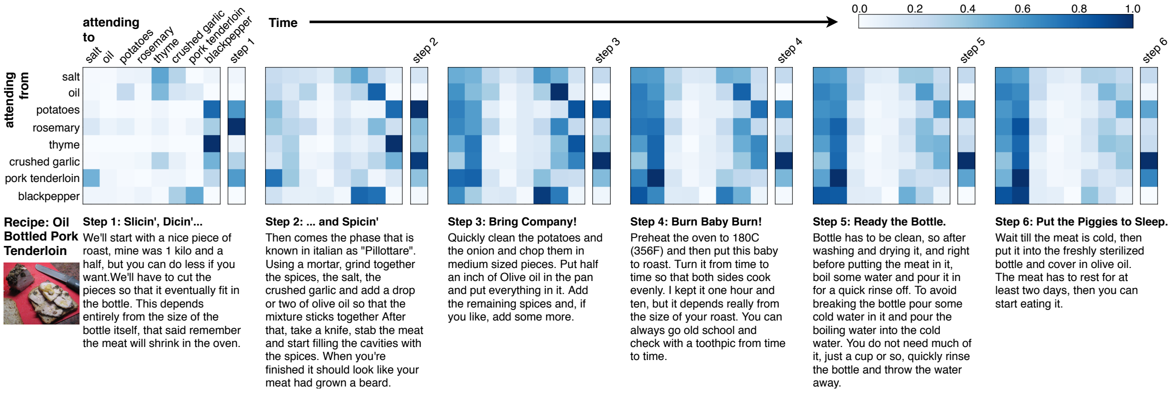
5.2.1. Input Module

Let the triple $(\mathbf{R}, \mathbf{Q}, \mathbf{A})$ be a sample input. Here, $\mathbf{R}$ denotes the input recipe which contains textual instructions composed of N words in total. $\mathbf{Q}$ represents the question that consists of a sequence of M images. $\mathbf{A}$ denotes an answer that is either a single image or a series of L images depending on the reasoning task. In particular, for the visual cloze and the visual coherence type questions, the answer contains a single image ($L = 1$) and for the visual ordering task, it includes a sequence.

We encode the input recipe $\mathbf{R}$ at character, word, and step levels. Character-level embedding layer uses a convolutional neural network, namely the CharCNN model by [161], which outputs character-level embeddings for each word to address the representational challenges in the existence of out of vocabulary words. In the word embedding layer, we use the GloVe model as a pretrained model [162], to extract word embeddings (We also consider pre-trained ELMo embeddings [163] in our experiments but found out that the performance gain does not justify the computational overhead.). We concatenate character-level and word-level embeddings and then feed this vector to a two-layered highway network [164] to obtain a contextual embedding for each word in the recipe. This results in the matrix $\mathbf{R}' \in \mathbb{R}^{2d \times N}$.

We utilize another layer leveraging the previous layers, to encode the steps of the recipe in an individual manner. Specifically, we obtain a step-level contextual embedding of the input recipe containing T steps as $\mathcal{S} = (\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_T)$ where s_i represents the final state of a BiLSTM encoding the i-th step of the recipe obtained from the character and word-level embeddings of the tokens exist in the corresponding step.

We represent both the question $\mathbf{Q}$ and the answer $\mathbf{A}$ in terms of visual embeddings. Here, we employ a pre-trained ResNet-50 model [165] trained on ImageNet dataset [166] and represent each image as a real-valued 2048-d vector using features from the penultimate average-pool layer. Then these embeddings are passed first to a multilayer perceptron (MLP) and then its outputs are fed to a BiLSTM. We then form a matrix $\mathbf{Q}' \in \mathbb{R}^{2d \times M}$ for the question by concatenating the cell states of the BiLSTM. For the visual ordering task, to represent the



Recipe: Oil Bottled Pork Tenderloin

Step 1: Slicin', Dicin'... We'll start with a nice piece of roast, mine was 1 kilo and a half, but you can do less if you want.We'll have to cut the pieces so that it eventually fit in the bottle. This depends entirely from the size of the bottle itself, that said remember the meat will shrink in the oven.

Step 2: ... and Spicin' Then comes the phase that is known in italian as "Pillottare". Using a mortar, grind together the spices, the salt, the crushed garlic and add a drop or two of olive oil so that the mixture sticks together After that, take a knife, stab the meat and start filling the cavities with the spices. When you're finished it should look like your meat had grown a beard.

Step 3: Bring Company! Quickly clean the potatoes and the onion and chop them in medium sized pieces. Put half an inch of Olive oil in the pan and put everything in it. Add the remaining spices and, if you like, add some more.

Step 4: Burn Baby Burn! Preheat the oven to 180C (356F) and then put this baby to roast. Turn it from time to time so that both sides cook evenly. I kept it one hour and ten, but it depends really from the size of your roast. You can always go old school and check with a toothpic from time to time.

Step 5: Ready the Bottle. Bottle has to be clean, so after washing and drying it, and right before putting the meat in it, boil some water and pour it in for a quick rinse off. To avoid breaking the bottle pour some cold water in it and pour the boiling water into the cold water. You do not need much of it, just a cup or so, quickly rinse the bottle and throw the water away.

Step 6: Put the Piggies to Sleep. Wait till the meat is cold, then put it into the freshly sterilized bottle and cover in olive oil. The meat has to rest for at least two days, then you can start eating it.

Figure 5.3 Sample visualizations of the self-attention weights demonstrating both the interactions among the ingredients and between the ingredients and the textual instructions throughout the steps of a sample cooking recipe from RecipeQA (darker colors imply higher attention weights). The attention maps do not change much after the third step as the steps after that mostly provide some redundant information about the completed recipe.

sequence of images in the answer with a single vector, we additionally use a BiLSTM and define the answering embedding by the summation of the cell states of the BiLSTM. Finally, for all tasks, these computations produce answer embeddings denoted by $\mathbf{a} \in \mathbb{R}^{2d \times 1}$.

5.2.2. Reasoning Module

As mentioned before, comprehending a cooking recipe is mostly about entities (basic ingredients) and actions (cooking activities) described in the recipe instructions. Each action leads to changes in the states of the entities, which usually affects their visual characteristics. A change rarely occurs in isolation; in most cases, the action affects multiple entities at once. Hence, in our reasoning module, we have an explicit memory component implemented with relational memory units [158]. This helps us to keep track of the entities, their state changes, and their relations in relation to each other over the course of the recipe (see Fig. 5.3). As we will examine in more detail in Section 5.3., it also greatly improves the interpretability of model outputs.

Specifically, we set up the memory with a memory matrix $\mathbf{E} \in \mathbb{R}^{d_E \times K}$ by extracting K entities (ingredients) from the first step of the recipe (The first steps of the recipes in

RecipeQA commonly contain a list of ingredients.). We initialize each memory cell $\mathbf{e}_i$ representing a specific entity by its CharCNN and pre-trained GloVe embeddings. For the entities that have multiple words in them such as *minced garlic*, we take the average of the word embeddings they contain, and OOV words are expressed with the average word vector of all the words. From now on, we will use the terms memory cells and entities interchangeably throughout the paper. Since the input recipe is given in the form of a procedural text decomposed into a number of steps, we update the memory cells after each step, reflecting the state changes that happened to the entities. This update procedure is modeled via a relational recurrent neural network (R-RNN), recently proposed by [158]. It is built on a 2-dimensional LSTM model whose matrix of cell states represents our memory matrix $\mathbf{E}$. Here, each row i of the matrix $\mathbf{E}$ refers to a specific entity $\mathbf{e}_i$ and is updated after each step of the recipe, referred to as t as follows:

$$\phi_{i,t} = \text{R-RNN}(\phi_{i,t-1}, \mathbf{s}_t) \tag{2}$$

where $\mathbf{s}_t$ denotes the embedding of recipe step t and $\phi_{i,t} = (\mathbf{h}_{i,t}, \mathbf{e}_{i,t})$ corresponds to the state of the R-RNN cells at step t with $\mathbf{h}_{i,t}$ and $\mathbf{e}_{i,t}$ denoting i-th row of R-RNN hidden state and the dynamic representation of entity $\mathbf{e}_i$ at the step t, respectively. The R-RNN model exploits a multi-headed self-attention mechanism [167] that allows memory cells to interact with each other and attend multiple locations simultaneously during the update phase.

In Fig. 5.3, we illustrate how this interaction takes place in our relational memory module by considering a sample cooking recipe and by presenting how the attention matrix changes throughout the recipe. In particular, the attention matrix at a specific time shows the attention flow from one entity (memory cell) to another along with the attention weights to the corresponding recipe step (offset column). The color intensity shows the magnitude of the attention weights. As can be seen from the figure, the internal representations of the entities are actively updated at each step. Moreover, as argued in [158], this can be interpreted as a form of relational reasoning as each update on a specific memory cell is operated in relation to others. Here, we should note that it is often difficult to make sense of these attention weights. However, we observe that the attention matrix changes very gradually near the completion of the recipe.

5.2.3. Attention Module

The attention module is in charge of linking the question with the recipe text and the entities present in the recipe. It takes the matrices $\mathbf{Q}'$ and $\mathbf{R}'$ from the input module, and $\mathbf{E}$ from the reasoning module and constructs the question-aware recipe representation $\mathbf{G}$ and the question-aware entity representation $\mathbf{Y}$. Following the attention flow mechanism described in [168], we specifically calculate attention in four different directions: (1) from question to recipe, (2) from recipe to question, (3) from question to entities, and (4) from entities to question.

The first two of these attentions require computing a shared affinity matrix $\mathbf{S}^R \in \mathbb{R}^{N \times M}$ with $\mathbf{S}^R_{i,j}$ indicating how similar the i-th recipe word and j-th image is in the question estimated by

$$\mathbf{S}^R_{i,j} = \mathbf{w}_R^\top [\mathbf{R}'_i; \mathbf{Q}'_j; \mathbf{R}'_i \circ \mathbf{Q}'_j] \tag{3}$$

where $\mathbf{w}_R^\top$ is a trainable weight vector, $\circ$ and $[;]$ denote elementwise multiplication and concatenation operations, respectively.

Recipe-to-question attention determines the images within the question that is most relevant to each word of the recipe. Let $\tilde{\mathbf{Q}} \in \mathbb{R}^{2d \times N}$ represent the recipe-to-question attention matrix with its i-th column being given by $\tilde{\mathbf{Q}}_i = \sum_j \mathbf{a}_{ij} \mathbf{Q}'_j$ where the attention weight is computed by $\mathbf{a}_i = \mathrm{softmax}(\mathbf{S}^R_i) \in \mathbb{R}^M$.

Question-to-recipe attention signifies the words within the recipe that have the closest similarity to each image in the question, and construct an attended recipe vector given by $\tilde{\mathbf{r}} = \sum_i \mathbf{b}_i \mathbf{R}'_i$ with the attention weight is calculated by $\mathbf{b} = \mathrm{softmax}(\max_{col}(\mathbf{S}^R)) \in \mathbb{R}^N$ where $\max_{col}$ denotes the maximum function across the column. The question-to-recipe matrix is then obtained by replicating $\tilde{\mathbf{r}}$ N times across the column, giving $\tilde{\mathbf{R}} \in \mathbb{R}^{2d \times N}$.

Then, we construct the question-aware representation of the input recipe, $\mathbf{G}$, with its i-th column $\mathbf{G}_i \in \mathbb{R}^{8d\times N}$ denoting the final embedding of i-th word given by

$$\mathbf{G}_i = [\mathbf{R}'_i; \tilde{\mathbf{Q}}_i; \mathbf{R}'_i \circ \tilde{\mathbf{Q}}_i; \mathbf{R}'_i \circ \tilde{\mathbf{R}}_i;] \ . \tag{4}$$

Attention from the question to entities and from entities to the question is computed in a way similar to the ones described above. The only difference is that it uses a different shared affinity matrix to be computed between the memory encoding entities $\mathbf{E}$ and the question $\mathbf{Q}'$. These attentions are then used to construct the question-aware representation of entities, denoted by $\mathbf{Y}$, that links and integrate the images in the question and the entities in the input recipe.

5.2.4. Modeling Module

The modeling module takes the question-aware representations of the recipe $\mathbf{G}$ and the entities $\mathbf{Y}$ and forms their combined vector representation. For this purpose, we first use a two-layer BiLSTM to read the question-aware recipe $\mathbf{G}$ and to encode the interactions among the words conditioned on the question. For each direction of BiLSTM, we use its hidden state after reading the last token as its output. In the end, we obtain a vector embedding $\mathbf{c} \in \mathbb{R}^{2d\times 1}$. Similarly, we employ a second BiLSTM, this time, over the entities $\mathbf{Y}$, which results in another vector embedding $\mathbf{f} \in \mathbb{R}^{2d_E\times 1}$. Finally, these vector representations are concatenated and then projected to a fixed size representation using $\mathbf{o} = \varphi_o([\mathbf{c}; \mathbf{f}]) \in \mathbb{R}^{2d\times 1}$ where φ_o is a multilayer perceptron with $\tanh$ activation function.

5.2.5. Output Module

The output module takes the output of the modeling module, encoding vector embeddings of the question-aware recipe and the entities $\mathbf{Y}$, and the embedding of the answer $\mathbf{A}$, and returns a similarity score which is used while determining the correct answer. The answer candidate with the highest similarity score among all candidates is selected as the correct one. To train our proposed procedural reasoning network, we employ a hinge ranking loss [139], similar to the one used in [130], given below.

$$L = \max\{0, \gamma - \cos(\mathbf{o}, \mathbf{a}_+) + \cos(\mathbf{o}, \mathbf{a}_-)\} \tag{5}$$

where γ is the margin parameter, $\mathbf{a}_+$ and $\mathbf{a}_-$ are the correct and the incorrect answers, respectively.

5.3. Experiments

Throughout the section, we explain how we conducted the experiments, and then we present a detailed analysis of the outcomes of the Procedural Reasoning Networks (PRN) model.

5.3.1. Entity Extraction

Given a recipe, we automatically extract the entities from the initial step of a recipe by using a dictionary of ingredients. While determining the ingredients, we exploit Recipe1M [169] and Kaggle What's Cooking Recipes [170] datasets and form our dictionary using the most commonly used ingredients in the training set of RecipeQA. For the cases when no entity can be extracted from the recipe automatically (20 recipes in total), we manually annotate those recipes with the related entities.

5.3.2. Training Details

In our experiments, we separately trained models on each task, as well as we investigated multi-task learning where one model was trained to solve all these tasks at once. In total, PRN architecture consists of ∼12M trainable parameters. We implemented our models in PyTorch [171] using AllenNLP library [160]. We used ADAM for optimization and set learning rate to $1e-4$ using an early stopping condition where the patience is set to 10 indicating that the training procedure ends after 10 iterations if the performance would not improve. We considered a batch size of 32 due to our hardware constraints. In the multi-task setting, batches are sampled round-robin from all tasks, where each batch is solely composed of examples from one task. We performed our experiments on a system containing four NVIDIA GTX-1080Ti GPUs, and training a single model took around 2 hours. We employed the same hyperparameters for all the baseline systems. We plan to share our code and model implementation after the review process.

5.3.3. Baselines

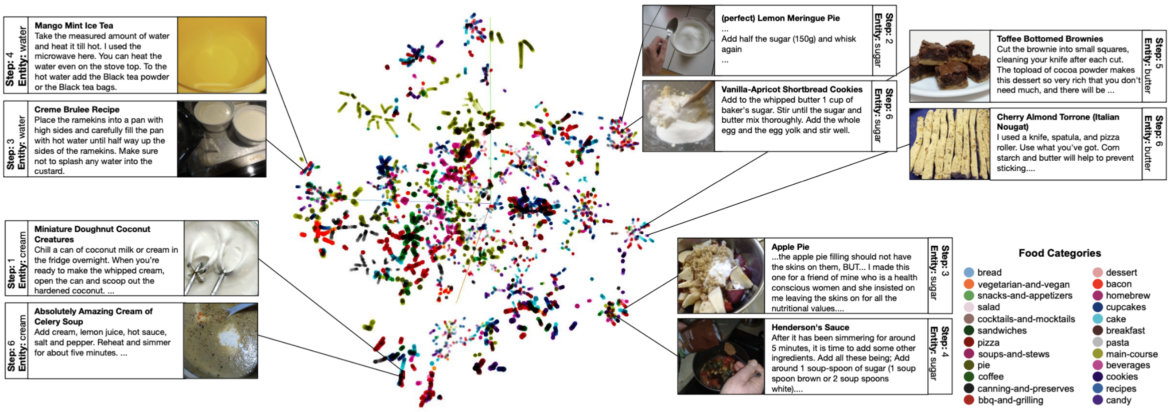
We benchmark our proposed model against several baselines and note that the results of the first two are previously reported in [130].

Hasty Student [130] is a heuristics-based simple model which ignores the recipe and gives an answer by examining only the question and the answer set using distances in the visual feature space.

Impatient Reader [30] is a simple neural model where its name comes from its behavior of repeatedly computing attention over the recipe after observing each image in the query.

BiDAF [168] is a strong neural reading comprehension model in which a bi-directional attention flow is employed to obtain question-aware embeddings and bases its predictions on this representation. Originally, it is a span-selection model from the input context. Here, we adapt it to work in a multimodal setting and answer multiple-choice questions instead.

BiDAF w/ static memory is an extended version of the BiDAF model which resembles our proposed PRN model in that it includes a memory unit for the entities. However, it does not make any updates on the memory cells. That is, it uses the static entity embeddings initialized with GloVe word vectors. We propose this baseline to test the significance of the use of relational memory updates.



Step: 4 Entity: water
Mango Mint Ice Tea
Take the measured amount of water and heat it till hot. I used the microwave here. You can heat the water even on the stove top. To the hot water add the Black tea powder or the Black tea bags.

Step: 3 Entity: water
Creme Brulee Recipe
Place the ramekins into a pan with high sides and carefully fill the pan with hot water until half way up the sides of the ramekins. Make sure not to splash any water into the custard.

Step: 1 Entity: cream
Miniature Doughnut Coconut Creatures
Chill a can of coconut milk or cream in the fridge overnight. When you're ready to make the whipped cream, open the can and scoop out the hardened coconut. ...

Step: 6 Entity: cream
Absolutely Amazing Cream of Celery Soup
Add cream, lemon juice, hot sauce, salt and pepper. Reheat and simmer for about five minutes. ...

Step: 2 Entity: sugar
(perfect) Lemon Meringue Pie
...
Add half the sugar (150g) and whisk again
...

Step: 6 Entity: sugar
Vanilla-Apricot Shortbread Cookies
Add to the whipped butter 1 cup of baker's sugar. Stir until the sugar and butter mix thoroughly. Add the whole egg and the egg yolk and stir well.

Step: 5 Entity: butter
Toffee Bottomed Brownies
Cut the brownie into small squares, cleaning your knife after each cut. The topload of cocoa powder makes this dessert so very rich that you don't need much, and there will be ...

Step: 6 Entity: butter
Cherry Almond Torrone (Italian Nougat)
I used a knife, spatula, and pizza roller. Use what you've got. Corn starch and butter will help to prevent sticking....

Step: 3 Entity: sugar
Apple Pie
...the apple pie filling should not have the skins on them, BUT... I made this one for a friend of mine who is a health conscious women and she insisted on me leaving the skins on for all the nutritional values....

Step: 4 Entity: sugar
Henderson's Sauce
After it has been simmering for around 5 minutes, it is time to add some other ingredients. Add all these being; Add around 1 soup-spoon of sugar (1 soup spoon brown or 2 soup spoons white)....

Figure 5.4 t-SNE visualizations of learned embeddings from each memory snapshot mapping to each entity and their corresponding states from each step for visual cloze task.

Table 5.1 Quantitative comparison for the PRN model results against the baselines.

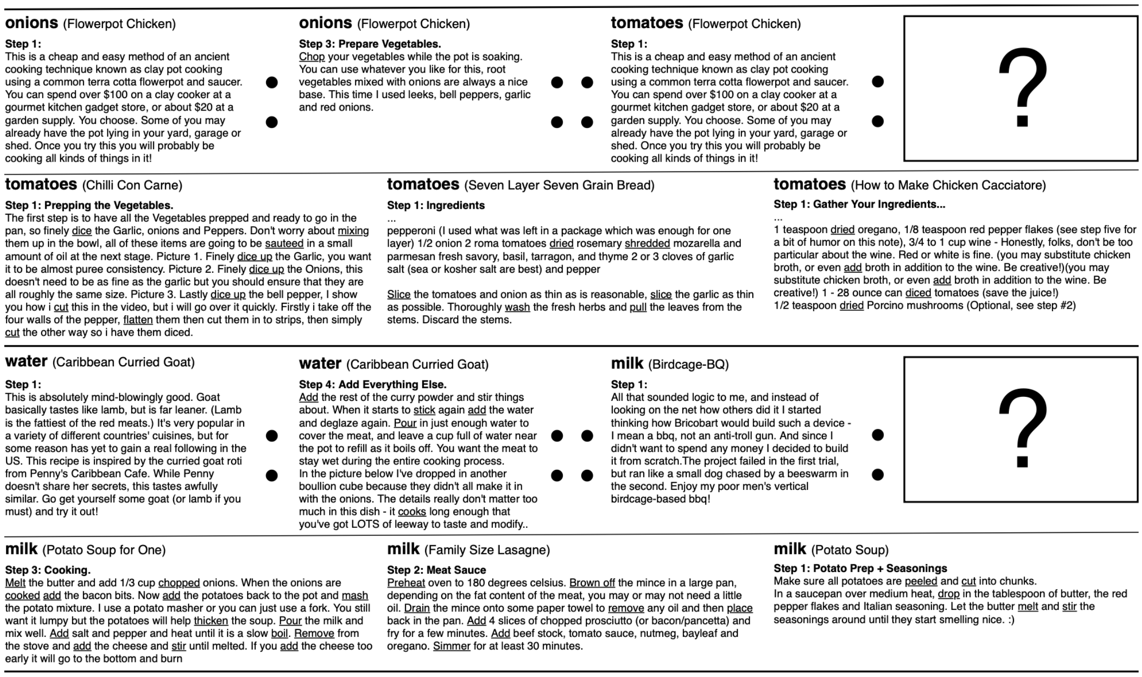
	Single-task Training				Multi-task Training			
Model	Cloze	Coherence	Ordering	Average	Cloze	Coherence	Ordering	All
Human*	77.60	81.60	64.00	74.40	–	–	–	–
Hasty Student	27.35	**65.80**	40.88	44.68	–	–	–	–
Impatient Reader	27.36	28.08	26.74	27.39	–	–	–	–
BIDAF	53.95	48.82	62.42	55.06	44.62	36.00	**63.93**	48.67
BIDAF w/ static memory	51.82	45.88	60.90	52.87	**47.81**	40.23	62.94	**50.59**
PRN	**56.31**	53.64	**62.77**	**57.57**	46.45	**40.58**	62.67	50.17

* Taken from the RecipeQA project website, based on 100 questions sampled randomly from the validation set.

5.3.4. Results

Table 5.1 presents the quantitative results for the visual reasoning tasks in RecipeQA. We present the results of our baseline models, which show the percentage of questions that were answered correctly out of the total number of questions in the test. In other words, we use the accuracy metric for calculating the model performance on these tasks, considering these tasks are multiple choice questions each consisting of 4 candidate answers per question. In the single-task training setting, PRN achieves state-of-the-art against other neural models. Moreover, it also performs as the best baseline on average. These findings demonstrate the importance of having a dynamic memory and keeping track of entities extracted from the recipe. In the multi-task training setting where a single model is trained to solve all the tasks at once, PRN and BIDAF w/ static memory perform comparably and give much better results than BIDAF. Note that the model performances in the multi-task training setting are worse than single-task performances. We believe that this is due to the nature of the tasks that some are more difficult than others. We think that the performance could be improved by employing a carefully selected curriculum strategy [172].

In Fig. 5.4, we illustrate the entity embeddings space by projecting the learned embeddings from the step-by-step memory snapshots through time with t-SNE to 3-d space from 200-d vector space. Color codes denote the categories of the cooking recipes. As can be seen, these step-aware embeddings show clear clustering of these categories. Moreover, within each cluster, the entities are grouped together in terms of their state characteristics. For instance, in the zoomed parts of the figure, chopped and sliced or stirred and whisked entities are placed close to each other.



onions (Flowerpot Chicken)

Step 1:
This is a cheap and easy method of an ancient cooking technique known as clay pot cooking using a common terra cotta flowerpot and saucer. You can spend over $100 on a clay cooker at a gourmet kitchen gadget store, or about $20 at a garden supply. You choose. Some of you may already have the pot lying in your yard, garage or shed. Once you try this you will probably be cooking all kinds of things in it!

:

onions (Flowerpot Chicken)

Step 3: Prepare Vegetables.
Chop your vegetables while the pot is soaking. You can use whatever you like for this, root vegetables mixed with onions are always a nice base. This time I used leeks, bell peppers, garlic and red onions.

::

tomatoes (Flowerpot Chicken)

Step 1:
This is a cheap and easy method of an ancient cooking technique known as clay pot cooking using a common terra cotta flowerpot and saucer. You can spend over $100 on a clay cooker at a gourmet kitchen gadget store, or about $20 at a garden supply. You choose. Some of you may already have the pot lying in your yard, garage or shed. Once you try this you will probably be cooking all kinds of things in it!

:

?

tomatoes (Chilli Con Carne)

Step 1: Prepping the Vegetables.
The first step is to have all the Vegetables prepped and ready to go in the pan, so finely dice the Garlic, onions and Peppers. Don't worry about mixing them up in the bowl, all of these items are going to be sauteed in a small amount of oil at the next stage. Picture 1. Finely dice up the Garlic, you want it to be almost puree consistency. Picture 2. Finely dice up the Onions, this doesn't need to be as fine as the garlic but you should ensure that they are all roughly the same size. Picture 3. Lastly dice up the bell pepper, I show you how i cut this in the video, but i will go over it quickly. Firstly i take off the four walls of the pepper, flatten them then cut them in to strips, then simply cut the other way so i have them diced.

tomatoes (Seven Layer Seven Grain Bread)

Step 1: Ingredients
...
pepperoni (I used what was left in a package which was enough for one layer) 1/2 onion 2 roma tomatoes dried rosemary shredded mozarella and parmesan fresh savory, basil, tarragon, and thyme 2 or 3 cloves of garlic salt (sea or kosher salt are best) and pepper

Slice the tomatoes and onion as thin as is reasonable, slice the garlic as thin as possible. Thoroughly wash the fresh herbs and pull the leaves from the stems. Discard the stems.

tomatoes (How to Make Chicken Cacciatore)

Step 1: Gather Your Ingredients...
...
1 teaspoon dried oregano, 1/8 teaspoon red pepper flakes (see step five for a bit of humor on this note), 3/4 to 1 cup wine - Honestly, folks, don't be too particular about the wine. Red or white is fine. (you may substitute chicken broth, or even add broth in addition to the wine. Be creative!)(you may substitute chicken broth, or even add broth in addition to the wine. Be creative!) 1 - 28 ounce can diced tomatoes (save the juice!)
1/2 teaspoon dried Porcino mushrooms (Optional, see step #2)

water (Caribbean Curried Goat)

Step 1:
This is absolutely mind-blowingly good. Goat basically tastes like lamb, but is far leaner. (Lamb is the fattiest of the red meats.) It's very popular in a variety of different countries' cuisines, but for some reason has yet to gain a real following in the US. This recipe is inspired by the curried goat roti from Penny's Caribbean Cafe. While Penny doesn't share her secrets, this tastes awfully similar. Go get yourself some goat (or lamb if you must) and try it out!

:

water (Caribbean Curried Goat)

Step 4: Add Everything Else.
Add the rest of the curry powder and stir things about. When it starts to stick again add the water and deglaze again. Pour in just enough water to cover the meat, and leave a cup full of water near the pot to refill as it boils off. You want the meat to stay wet during the entire cooking process.
In the picture below I've dropped in another boullion cube because they didn't all make it in with the onions. The details really don't matter too much in this dish - it cooks long enough that you've got LOTS of leeway to taste and modify..

::

milk (Birdcage-BQ)

Step 1:
All that sounded logic to me, and instead of looking on the net how others did it I started thinking how Bricobart would build such a device - I mean a bbq, not an anti-troll gun. And since I didn't want to spend any money I decided to build it from scratch.The project failed in the first trial, but ran like a small dog chased by a beeswarm in the second. Enjoy my poor men's vertical birdcage-based bbq!

:

?

milk (Potato Soup for One)

Step 3: Cooking.
Melt the butter and add 1/3 cup chopped onions. When the onions are cooked add the bacon bits. Now add the potatoes back to the pot and mash the potato mixture. I use a potato masher or you can just use a fork. You still want it lumpy but the potatoes will help thicken the soup. Pour the milk and mix well. Add salt and pepper and heat until it is a slow boil. Remove from the stove and add the cheese and stir until melted. If you add the cheese too early it will go to the bottom and burn

milk (Family Size Lasagne)

Step 2: Meat Sauce
Preheat oven to 180 degrees celsius. Brown off the mince in a large pan, depending on the fat content of the meat, you may or may not need a little oil. Drain the mince onto some paper towel to remove any oil and then place back in the pan. Add 4 slices of chopped prosciutto (or bacon/pancetta) and fry for a few minutes. Add beef stock, tomato sauce, nutmeg, bayleaf and oregano. Simmer for at least 30 minutes.

milk (Potato Soup)

Step 1: Potato Prep + Seasonings
Make sure all potatoes are peeled and cut into chunks.
In a saucepan over medium heat, drop in the tablespoon of butter, the red pepper flakes and Italian seasoning. Let the butter melt and stir the seasonings around until they start smelling nice. :)

Figure 5.5 Step-aware entity representations can be used to discover the changes that occurred in the states of the ingredients between two different recipe steps. The difference vector between two entities can then be added to other entities to find their next states. For instance, in the first example, the difference vector encodes the chopping action done on onions. In the second example, it encodes the pouring action done on the water. When these vectors are added to the representations of raw tomatoes and milk, the three most likely next states capture the semantics of state changes in an accurate manner.

Fig. 5.5 demonstrates the entity arithmetics using the learned embeddings from each entity step. Here, we show that the learned embedding from the memory snapshots can effectively capture the contextual information about the entities at each time point in the corresponding step while taking into account of the recipe data. This basic arithmetic operation suggests that the proposed model can successfully capture the semantics of each entity's state in the corresponding step[9].

5.4. Further Analysis

In the following, we provide implementation details and further analysis we did for the proposed Procedural Reasoning Networks model.

Entity Extraction. While initializing memory cells we used ingredients as entities and during each read we updated the memory cells, resulting in an implicit update for entity states. In that regard, detecting ingredients has been an important part of our experiments.

We followed a simple scheme for extracting entities from each recipe. We used Recipe1M [169] and Kaggle What's Cooking Recipes [170] datasets and formed a dictionary using the ingredients provided in those datasets from the food domain. Next, we scanned each recipe for existing ingredients using this dictionary.

In RecipeQA often the first steps contain ingredients of that recipe, but in some cases, ingredients might not be explicitly provided the first step. In order to address this we calculated the maximum number of unique ingredients in each step of a recipe and selected that step to extract ingredients. In cases ingredients can not be automatically extracted we manually annotated those recipes. In particular, we annotated 20 recipes manually.

Training Details. For training our model architecture in multitask mode, we used random shuffling for each batch. This is as well true for training in single mode, but by doing so we tried to reduce the task bias that sequential training might lead to.

[9] We used Gensim for calculating entity arithmetics using cosine distances between entity embeddings.

Analyses. For t-SNE embedding analyses, we used Embedding Projector[10] project. We used the embeddings learned by the model extracted from memory cells of each time step during each read. Each embedding corresponds to an entity state. In particular, we used entity embeddings and trained a t-SNE model with the following parameters perplexity=30, learning rate=10 for 1000 iterations. Due to a limitation in the embedding projector, the project downsamples data points to 10K, hence we used only 10K entity embeddings in our t-SNE analyses.

Hyperparameters. In Table 5.2 we demonstrate the hyperparameters we used to train our model.

Table 5.2 Training hyperparameter selection used in our experiments.

Hyperparameter	Value
Optimizer	Adam
Batch Size	32
Early Stopping	Yes
Learning Rate	1-e4
Patience	10
Memory Cell Size	61

5.5. Discussions

In this chapter, we presented a new neural architecture called Procedural Reasoning Networks (PRN) [159] for the multimodal understanding of step-by-step instructions. Our proposed model is based on the successful BiDAF framework but is also equipped with an explicit memory unit that provides an implicit mechanism to keep track of the changes in the states of the entities over the course of the procedure. Our experiments on the RecipeQA dataset's visual reasoning tasks reveal that the proposed PRN model significantly outperforms the previous baselines, indicating that it better understands the procedural text and the accompanying images. Additionally, we carefully analyze our results and highlight that the proposed approach can learn meaningful entity representations without any

[10] https://projector.tensorflow.org

entity-level supervision. Although we achieve state-of-the-art on RecipeQA, clearly there is still room for improvement compared to human performance. We also believe that the PRN architecture will be of value to other visual and textual sequential reasoning tasks.

Much recently, the RecipeQA dataset received a great deal of interest from both computer vision and natural language processing communities. There has been a body of research inspired by the RecipeQA dataset in various setups such as multimodal comprehension, procedural understanding, commonsense reasoning, information retrieval, image-text coherence, and cross-modal representation learning. In the following, we discuss the related follow-up studies that report results on RecipeQA as well as utilize the RecipeQA dataset for solving various research problems.

[173] proposed CITE, based on the RecipeQA dataset, and investigated multimodal discourse relations between image and text pairs. Authors leveraged recipes that have a one-to-one correspondence between each instruction and image to study the relationship between instructions and images with a broader goal of better understanding natural communication and commonsense reasoning. [174] proposed a method to identify image-sentence associations in documents with multiple images and multiple sentences without depending on explicit multimodal annotation during training leveraging the RecipeQA dataset. [175] explored the RecipeQA dataset and studied the inferential relations between images and text from the perspective of captioning by looking at the verb usages in text and the coherence of images and text co-occurring with images. [176] proposed a method to map activities in a given sequence with the matching instruction from the provided instruction sequences. The proposed model follows a regularization approach to identify and match with the instruction orders utilizing their partial orders. The authors then evaluate their proposed model performance on the RecipeQA textual cloze task and propose two other ordering tasks using RecipeQA. [177] proposed a transformer-based framework for reasoning tasks on RecipeQA to simulate interactions between various modalities at numerous steps. [178] proposed a dataset for sequence-to-sequence retrieval task. The authors used the RecipeQA dataset to augment the proposed dataset with the main objective to choose the image sequence that most accurately depicts the given textual input. [179] proposed an

alignment mechanism for procedural reasoning on RecipeQA. The authors investigate latent alignment space and the positional encoding of questions and answers to constrain the output space of the latent alignment. They leverage cross-modal representation based on cross-attention and analyze the benefits of information flow between images and instructions. [180] explore uncertainty measures for image-caption embedding-and-retrieval task on the RecipeQA dataset to assess sample reliability. The authors proposed a Bayesian neural network to quantify feature uncertainty and posterior uncertainty with a broader goal to improve retrieval performance by rejecting uncertain queries. [181] proposed a heuristic to reduce the inherent bias present in the RecipeQA dataset. The authors discuss that the question generation scheme in the RecipeQA dataset introduces a distributional bias, such that the beginning and final step titles often contain ‘Ingredients’ and ‘Enjoy’ therefore might lead models to only choose answers that contain these words without looking at the context, hence proposed a method to overcome this shortcoming. [182] proposed a method to model temporal structures for recipes in RecipeQA. In particular, the authors propose a knowledge-based deep heterogeneous graph matching model to reduce the divergence between recipe representations of entities across temporal changes and guarantee neighbourhood consensus through a graph-based approach. [183] proposed a cross-modal coherence model for text-to-image retrieval task for a joint understanding of co-occurring images and text and proposed CITE++ by extending the CITE dataset which is based on the RecipeQA dataset. [184] proposed Meta-RecipeQA to reduce the inherent biases in the RecipeQA dataset and proposed a model to solve comprehension and reasoning tasks on the RecipeQA dataset. The authors analyzed distance distributions between correct answers and incorrect answers and proposed methods to effectively reduce the bias in RecipeQA tasks. [185] demonstrated a heuristic to construct rich recipe representations in the form of plans leveraging the recipes in the RecipeQA dataset. The authors enriched the recipes in RecipeQA by adding allergens, activities, objects, and background knowledge such as tools and possible failures for each step of the recipes with a broader goal to improve retrieval efficiency. [186] proposed a method for sequencing unordered multimodal procedural instructions and utilized RecipeQA for multimodal event sequencing problem. In particular, the authors proposed different techniques to align images and text to improve the sequencing of unordered steps in multimodal instructions where images and text complement each other. [187] proposed a method to model entities in both their temporal and cross-modal relations and leveraged the temporal nature of the RecipeQA dataset as well as its multimodal nature. The authors examine entities both in textual modality and visual modality and encode entity relations through temporal relations as well as model their cross-modal relations in the recipes.

In summary, the RecipeQA dataset has been inspiring several researchers to investigate various aspects of the dataset and has become a traditional multimodal dataset in the literature. We hope that our work continues to attract more attention from different fields and so that more researchers can exploit RecipeQA for the procedural understanding of multimodal how-to instructions.

6. COMPOSITIONAL GROUNDED UNDERSTANDING OF ACTIONS

In this chapter, we explore the multimodal understanding of actions in a challenging compositional generalization setup which we surveyed thoroughly in Ch. 3.. In particular, we review the problem of compositional understanding of actions and propose models to show the contribution of multimodality towards models' compositional generalization abilities, based on our work[11]. We only made minor changes to fit the text in the narrative of the thesis and made small corrections in the text. In the following, we introduce the problem of compositional grounded understanding of actions and propose methods that can leverage multimodal signals to significantly improve models' compositional generalization abilities which we believe will be a step towards solving the compositional generalization problem which is an open research problem.

Humans can rapidly understand new concepts, relying upon and combining context information with basic concepts from their existing knowledge. Compared to humans, neural network models trained over increasingly large datasets perform impressively well on a wide range of tasks, but they often fail to compositionally generalize to unseen concepts. In this study, we investigate compositionality and systematic generalization in a perceptually grounded setting by using a dataset of everyday household activities. This dataset depicts sequences of activities in pursuit of a wide variety of goals, *e.g. preparing celery*, *washing plates*. Each activity is represented with crowd-sourced utterances that describe different steps of the activity alongside the egocentric video frames and audio features. We evaluate several unimodal and multimodal baselines on future utterance prediction and action anticipation tasks, that respectively aim at describing and predicting an activity involving novel compositions of seen concepts. The models that exploit visual and audio signals do indeed improve over text-only model when they are evaluated on the long tail of rare complex concepts.

[11] In submission

As a long-standing problem in language, compositionality has been widely studied for many years. It deals with describing the relationship between an unbounded number of sentences and a vast set of meanings from a finite set of rules [63]. Therefore, compositionality aims to address the problem of finding ways to define the meaning of an entire sentence as a function of the meaning of its constituents and the rules that are used to put those constituents together. In that regard, compositionality and systematic generalization have been used to characterize symbolic computation and human cognition [55, 65]. Humans exhibit compositional skills in a variety of areas, including visual scene comprehension and language understanding. As put by [76], *"Once a person learns the meaning of a new verb* 'dax'*, he or she can immediately understand the meaning of* 'dax twice' *and* 'sing and dax'*."* In a similar fashion, it has been shown that humans can learn a novel object's shape and leveraging prior color and concept information they can understand its compositions [61, 188].

Researchers have shown that neural models can perform well across different tasks that require effective generalization abilities [54]. The compositionality and systematicity of neural networks have been long debated whether – and to what extent – neural networks display compositional generalization [55, 66, 68, 71, 72]. Moreover, deep neural networks have been commonly criticized for requiring a very large number of training examples to succeed and argued to lack compositional abilities [189]. Discussions around neural networks' inability to generalize compositionally in order to capture the structure of the underlying problem, thus failing across various tasks have recently sparked a lot of interest in the machine learning community [53, 59, 61, 79, 81, 93, 190, 191].

Although recent studies towards understanding and improving the compositional generalization skills of neural models have sparked much interest in the research community, work around understanding the role of multimodal and grounded language processing has been limited. [76] investigated picking up new concepts and applying them in test time by coupling previously learned concepts with new concepts. [80] investigated how to acquire new words and how to predict novel compositions by learning textual representations from visual context for understanding instructional videos in a language modeling setup. Other existing works are centered around designing conceptual benchmark datasets specifically constructed for testing compositionality, *e.g.* [79, 93, 192]. These studies have demonstrated that many deep neural models fail to capture the compositional structures in the underlying tasks and cannot generalize well even on really simple textual data.

Our motivation in this study is to test linguistic compositionality and systematic generalization in a perceptually grounded setting and to understand whether leveraging visual and auditory cues can contribute to systematic generalization capabilities of deep models. Towards this goal, we turn our attention to multimodal how-to instructions as they provide a good test bed for our needs. More concretely, our contributions can be highlighted as:

- We curate a novel benchmark: Epic-Kitchens-100-Systematicity (`EK-100-SYS`) for future utterance prediction and action anticipation tasks, which can be used to analyze compositional generalization in a grounded setup.
- We implement several neural models, and through them, we analyze whether multimodality helps linguistic compositionality in the context of the proposed tasks.

6.1. Problem Formulation

6.1.1. Future Utterance Prediction Task

Predicting what comes next plays a central role in cognition [193, 194] and also has been attributed as an interesting training scheme from a cognitive perspective [78]. We study this task on a multimodal dataset of people performing everyday household activities, *e.g. preparing celery*. Each video in our dataset consists of short clips (microsegments) that define sub-tasks of an activity, and each sub-task is described by a textual description, *e.g.* “pick up plate”, “put plate in sink”, “turn on water”, and “wash plate”, annotated by the actors after the recording. In the following, we formulate the future utterance prediction task as a language generation problem. Let $\mathcal{S} = (\mathbf{X}, \mathbf{V}, \mathbf{A})$ denote a triplet representing a short video clip with $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{K}$ being a sequence of K utterances, which describe a household activity and grounded with visual and audio signals, denoted by $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{K}$ and $\mathbf{A} = \{\mathbf{a}_i\}_{i=1}^{K}$, respectively. Our proposed future utterance prediction task involves generating

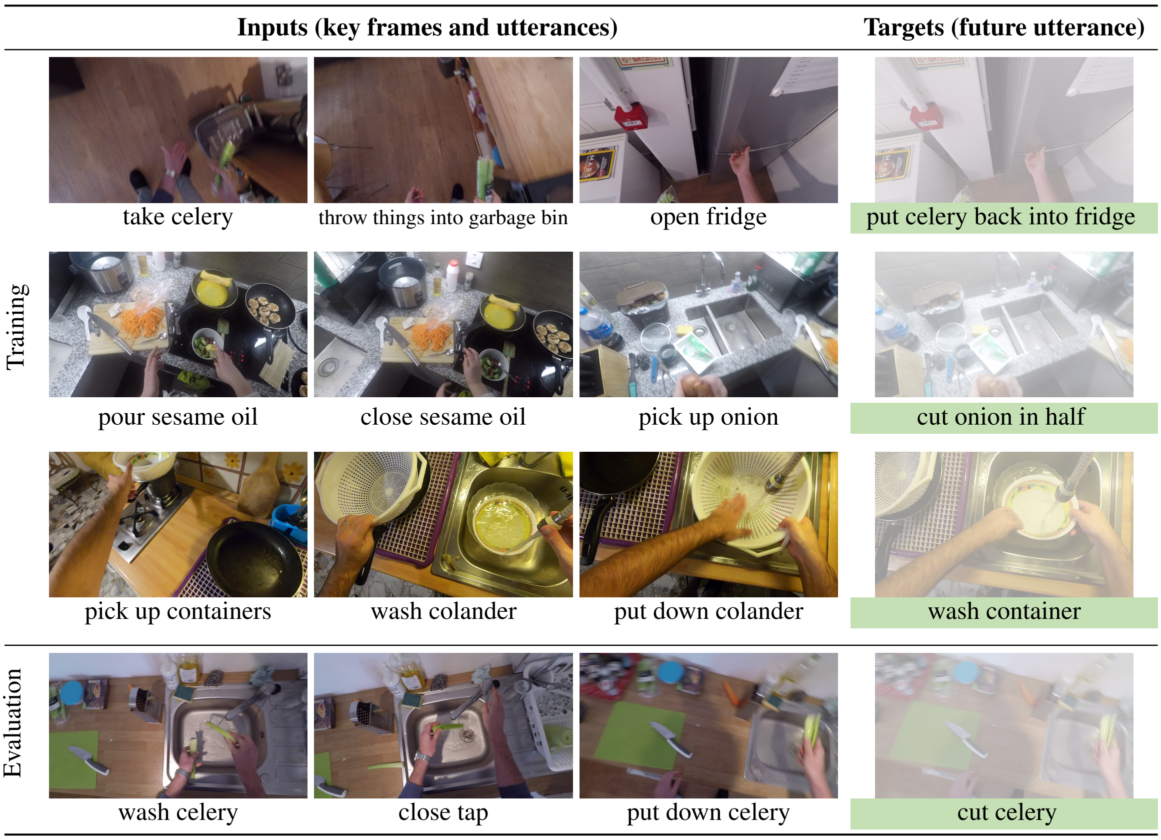


Figure 6.1 Overview of our systematicity setup for future utterance prediction task using microsegments from our `EK-100-SYS` dataset. During training time, the model has already been exposed to the primitives 'wash', 'close', 'put down', and 'celery' but not the "cut celery" composition and the goal is to be able to generalize to novel compositions of primitive elements in test time.

the $(K+1)^{th}$ utterance, $\mathbf{y} = \mathbf{x}_{K+1}$, following the preceding K utterances and multimodal cues. The training data comprised of multiple microsegments, $\{(\mathcal{S}, \mathbf{y})\}$.

During training, our objective is the minimization of the negative log-likelihood for generating the next utterance, where the multimodal models are conditioned on additional modalities such as image, or audio. Given the microsegment $\mathcal{S}$ and the model parameters θ, our objective is to minimize the negative log-likelihood of all next utterance tokens $\mathbf{y} = \{y_i\}_{i=1}^{m}$:

$$\log p(\mathbf{y}|\mathcal{S};\theta) = -\sum_{i=1}^{m} \log p(y_i|\mathcal{S};\theta) \tag{6}$$

6.1.2. Action Anticipation Task

We formulate action anticipation as a classification task, where we study the problem of predicting the next action with the target verbs and nouns. The main difference between this task and the future utterance prediction task is that utterance prediction is a natural language generation task. In particular, in action anticipation, the main objective is predicting the next action by leveraging the previously observed actions. Differing from other action anticipation tasks [9, 195, 196], our setup allows us to formulate action anticipation in a compositional manner by predicting the verb and noun separately.

More formally, let $\mathcal{S} = (\mathbf{X}, \mathbf{V}, \mathbf{A})$ denote a triplet representing a video clip with $\mathbf{X} = \{\mathrm{x}_i\}_{i=1}^{K}$ representing a sequence of K utterances, which describe a household activity and grounded with visual and audio signals, denoted by $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{K}$ and $\mathbf{A} = \{\mathbf{a}_i\}_{i=1}^{K}$, respectively. Our action anticipation task involves predicting the verb/noun in the $(K+1)^{th}$ utterance, $\mathbf{y} = \mathbf{x^C}_{K+1}$, following the preceding K utterances and multimodal cues where C denotes the verb or noun class.

6.2. EK-100-SYS Dataset

We use the EPIC-Kitchens-100 dataset (EK-100) as the starting point for our experiments [9]. EK-100 contains first-person videos of unscripted daily kitchen activities in natural household environments. Each video $\mathbf{V}$ is split into a sequence of shorter clips $\mathbf{v}_1, \ldots, \mathbf{v}_k$, which have manually annotated English narrations of the activities within clips denoted by $\mathrm{x}_1, \ldots, \mathrm{x}_k$. The clips also have audio tracks $\mathbf{a}_1, \ldots, \mathbf{a}_k$, which only contain sounds, *e.g.* a knife cutting an onion or a person opening the fridge. We denote a sequence of video clips – audio tracks – narrations as an *instance*.

Recall from Section 6.1.1. that our aim is to model sequences of video clips. Therefore, we select instances from the EK-100 dataset with a window of $K = 4$ clips: the first 3 clips are used for context, while the final clip is used for prediction. This results in 22,136 instances that can be used for our experiments. In Fig. 6.1, we provide some examples, along with representative keyframes and their corresponding narrations.

6.2.1. Systematicity Splits (`EK-100-SYS`)

Given a dataset of video sequences, our main focus is to study how well models compositionally generalize to unseen combinations of concepts. In the same vein of compositional captioning [77] or novel compositions of object properties [62], we create a dataset where the distribution of the individual concepts is similar across the dataset, but the compositions of those concepts is different. Consider the example given in Fig. 6.1. The model has already seen the nouns CELERY, TAP and verbs WASH, CLOSE, PUT DOWN, CUT but it has not seen the combination of CUT CELERY during training. The model has to compositionally generalize to this new instance.

To obtain such splits of the dataset, we followed the *Maximum Compound Divergence* heuristic to create similar distributions of individual concepts (atoms) but different distributions of combinations of concepts [79]. We use the 97 verb classes and 300 noun classes in the `EK-100` dataset as the atoms. In particular, we assign each sample to a split based on the atomic and compound divergence (similarity) leveraging Chernoff coefficient $C_\alpha(P\|Q) = \sum_k p_k^\alpha \, q_k^{1-\alpha} \in [0, 1]$ [197] while using weighted distributions. In order to make the distribution of atoms similar between the train and the test splits, we use $\alpha = 0.5$ for atomic divergence. Here, we set $\alpha = 0.1$ to reflect the importance of having certain compounds to exist in P (train) instead of the probabilities them matching exactly for the P (train) and Q (test). Following this logic, we define compound divergence and atom divergence for U which corresponds to train split, and W which corresponds to test split, using the following equation.

$$\mathcal{D}_C(U\|W) = 1 - C_{0.1}(\mathcal{F}_C(U) \,\|\, \mathcal{F}_C(W))$$
$$\mathcal{D}_A(U\|W) = 1 - C_{0.5}(\mathcal{F}_A(U) \,\|\, \mathcal{F}_A(W))$$

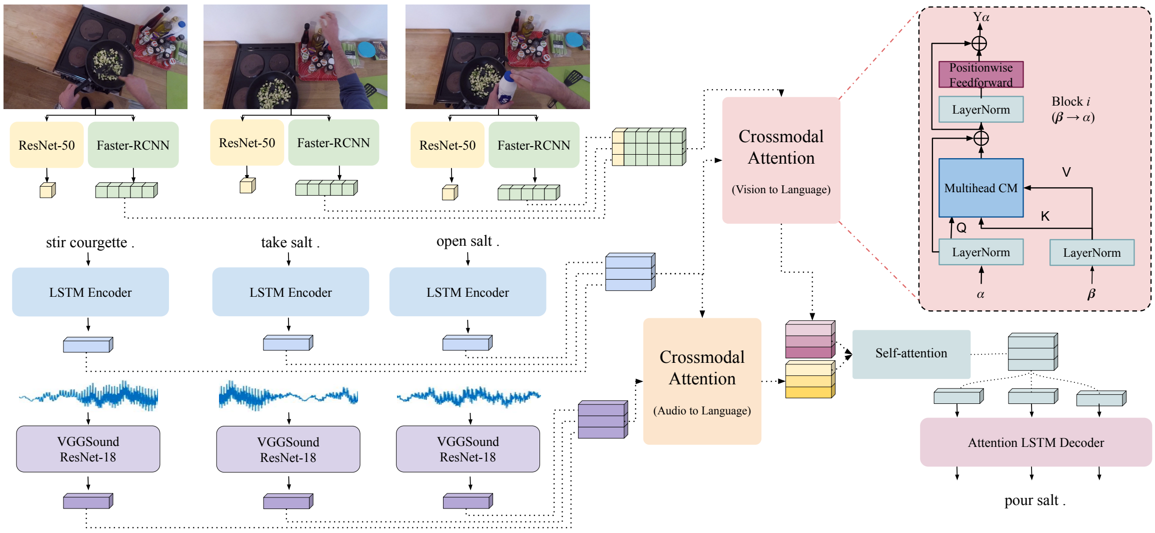


Figure 6.2 Overview of multimodal Audio, Vision, and Language baseline model which incorporates global image features, object level image features, audio features as well as textual features using two crossmodal self-attention blocks with an LSTM decoder to predict next utterances.

where $\mathcal{F}_A(T)$ denotes frequency distribution of atoms, and the $\mathcal{F}_C(T)$ describes compound distribution for a given set T and D_A and D_C denote atom and compound divergences, respectively. We calculated divergence scores for each data sample until we achieve the atomic divergence between the train and the test sets $D_A < 0.02$ and compound divergence between the train and the test sets $D_C < 0.6$, which represents a sweet spot in terms of target distributions as described before (see Fig. 6.4). Finally, we randomly divide this test set into two sets with similar distributions, one for validation and the other for testing.

The resulting `EK-100-SYS` dataset has 8,766 instances, which are split into 4,407 training, 2,184 validation, and 2,175 test instances. We use these splits to train and evaluate our models on future utterance prediction and action anticipation tasks.

6.3. Models for Future Utterance Prediction

We benchmark a text-only model, along with several multimodal models to assess the importance of different modalities in systematic generalization.

6.3.1. Text-only Unimodal Baseline (L)

Our first baseline is a text-only model to assess potential biases in the dataset [198]. This model is a 1-layer attention-based encoder-decoder neural model based on LSTM network [199] with a hidden size of 256 units. The decoder learns a time-step dependent context over the encoder hidden states [200]. The model is trained using only the textual utterances $\mathbf{x}_{1:K}$ from the microsegment as the input, and the next utterance $\mathbf{x}_{K+1}$ as the target, *i.e.* to predict $p\left(\mathbf{x}_{K+1}|\mathbf{x}_{1:K}\right)$. The model uses 200D word embeddings over a vocabulary of unique words in training samples where vocabularies are not shared between the encoder and the decoder. We use the same textual encoder and decoder in all multimodal baselines (See Appendix A for the number of trainable parameters and the vocabulary sizes).

6.3.2. Multimodal Baselines

We also evaluate several multimodal baselines that operate over combinations of the textual, visual, and audio modalities, as illustrated in Fig. 6.2.

6.3.3. Vision and Language (VL)

Our Vision and Language baseline encodes both textual and visual context for the future utterance prediction task. In particular, the model encodes the textual utterances $\mathbf{x}_{1:K}$ of each action from microsegments and the keyframe images $\mathbf{v}_{1:K}$ to predict the next utterance $\mathbf{x}_{K+1}$, *i.e.* $p\left(\mathbf{y} = \mathbf{x}_{K+1}|\mathbf{x}_{1:K}, \mathbf{v}_{1:K}\right)$. This model is adapted from a model that parses a visual scene and learns cross-modal self-attention [201] over textual inputs and visual data.

The visual inputs are encoded using pre-trained CNN, and the textual inputs are encoded using an LSTM. More specifically, for the visual modality, we extracted two types of features: one type represents global visual features, and the other represents object-level features. For the global features, we used a pre-trained ResNet50 model [165] with ImageNet weights [138]. Object-level features were extracted using a pre-trained Faster-RCNN object detector [202] with a ResNet-101 backbone [165] pre-trained on MSCOCO [89] and then finetuned leveraging the `EK-100` dataset. We extract visual features from 5 objects for each keyframe. The resulting representation of a visual keyframe is the concatenation of the global and the object-level features. The vector concatenation is then, using a 1D convolution, projected into a lower-dimensional space. The textual inputs are encoded using an LSTM with 200D word vectors through a 256D hidden layer. We then encode the visual and textual modalities by a cross-modal self-attention mechanism, CM. In this model, we consider two modalities α and β, sequences of each modalities are denoted as $X_\alpha \in \mathbb{R}^{T_\alpha \times d_\alpha}$ and $X_\beta \in \mathbb{R}^{T_\beta \times d_\beta}$, respectively and $T_{(\cdot)}$ corresponds to the length of the sequence and $d_{(\cdot)}$ represents the dimension of the feature. In this model, α is the language modality, and β is the visual modality. In the cross-modal attention, the textual features are the *keys*, and the visual features are the *queries* and *values*, for aligning visual features to textual features. Let the Query be denoted as $Q_\alpha = X_\alpha W_{Q_\alpha}$, the Keys are represented as $K_\beta = X_\beta W_{K_\beta}$, and the Values are denoted as $V_\beta = X_\beta W_{V_\beta}$, where $W_{Q_\alpha} \in \mathbb{R}^{d_\alpha \times d_k}, W_{K_\beta} \in \mathbb{R}^{d_\beta \times d_k}$ and $W_{V_\beta} \in \mathbb{R}^{d_\beta \times d_v}$ are learnable weights. The cross-modal self-attention from β to α is formulated as a latent adaptation $Y_\alpha \in \mathbb{R}^{T_\alpha \times d_v}$ as described in Eq. 7.

$$\begin{aligned} Y_\alpha &= \mathrm{CM}_{\beta \rightarrow \alpha}(X_\alpha, X_\beta) \\ &= \mathrm{softmax}\left(\frac{Q_\alpha K_\beta^\top}{\sqrt{d_k}}\right) V_\beta \end{aligned} \tag{7}$$

The output Y_α has the same length as Q_α, but it is represented in the feature space of V_β. This enables the model to fuse different modalities, learning an alignment between the visual and textual features. Finally, there is a self-attention layer [105] over the aligned vision and language features, which are the input to an attention-based LSTM decoder that generates the next utterance.

6.3.4. Audio and Language (AL)

The Audio and Language baseline has the same structure as the Vision and Language baseline. The key difference is that we represent the additional context using audio features instead of visual features. In particular, the model encodes both the textual utterances $\mathbf{x}_{1:K}$ and the accompanying audio data $\mathbf{a}_{1:K}$ to predict the next utterance $\mathbf{x}_{K+1}$, *i.e.* $p\left(\mathbf{x}_{K+1}|\mathbf{x}_{1:K}, \mathbf{a}_{1:K}\right)$. The audio features are 512D vectors extracted using VGGSound [203], which is pre-trained on 200K videos from YouTube videos totaling up to 550 hours of audio data. Here, the model learns a cross-modal attention block over the audio and the textual features, analogously to using the visual and textual features as inputs to an LSTM-based decoder.

6.3.5. Object and Language (OL)

The Object and Language baseline uses the same architecture as Vision and Language baseline, but we represent the visual context using the tags of the detected objects instead of the CNN visual features to explicitly encode the visual content. In this model, we embed object tags as a secondary set of textual features to our model along with the input utterances. Here, the object tags are represented as 292-dimensional one-hot encoded vectors (based on the number of unique tags) and projected to 256D with a simple linear layer. In this case, the cross-modal attention mechanism aligns object tag features with language features.

6.3.6. Audio, Vision, and Language (AVL)

In the Audio, Vision, and Language (AVL) baseline, we leverage the audio, visual, and textual data using two cross-modal self-attention blocks. We use textual utterances $\mathbf{x}_{1:K}$ of each action along with the visual features $\mathbf{v}_{1:K}$ from the keyframes, and the VGGSound audio features $\mathbf{a}_{1:K}$ to guess the future utterance $\mathbf{x}_{K+1}$, *i.e.* $p\left(\mathbf{y} = \mathbf{x}_{K+1}|\mathbf{x}_{1:K}, \mathbf{v}_{1:K}, \mathbf{a}_{1:K}\right)$. In the AVL baseline, input fed to the self-attention layer before the decoder is the concatenation of the audio-aligned textual features from the audio-textual cross-modal block with the visual-aligned textual features from the visual-textual cross-modal block.

6.3.7. Object, Audio, and Language (OAL)

We perform an extra experiment to determine whether adding an extra modality to the OL baseline model improves its performance by coupling the object tags with audio features. Here, we include the extracted audio features from each microsegment to the OL model and train accordingly.

6.3.8. Pretrained Vision and Language (PVL)

In order to understand the importance of pretraining on a large scale aligned audio, visual, and linguistic data, we also utilize the Merlot Reserve model [204]. This model learns multimodal video embeddings over video frames, text, and audio. We extract multimodal vision and language features through its pre-trained encoder while considering the same decoder network as the other baseline models. We report our experimental results with this pre-trained transformer baseline.

6.4. Models for Action Anticipation

Different than the future utterance prediction task which is formulated as a conditional language generation problem, the action anticipation task requires (grounded) language understanding. We modify the architectures of the models described in the previous section to adapt them to this task. In particular, we replace the final layer in these models with two new fully connected layers and finetune the pre-trained models by considering a classification objective that involves predicting either the VERB or the NOUN in the anticipated action.

6.5. Experimental Setup

In this section, we explain the methodology we used for our experiments and describe the measures we used to evaluate the performance of our baseline models, along with the training details.

6.5.1. Evaluation Metrics

For assessing the baseline performance, we use a set of metrics. In particular, we use BLEU [41], Exact Match (EM), and Categorical Accuracy (CA) metrics. For BLEU, we use the NLTK toolkit and report unigram BLEU scores. For EM, we calculate an accuracy score between the generated text sequence and the groundtruth. CA uses the verb and noun categories in `EK-100` and calculates the categorization accuracy based on noun category match between the predicted sequence and groundtruth, *e.g*. the verbs *slice*, *dice*, and *chop* fall into the same verb category *cut*, and the nouns *mozzarella*, *paneer* and *parmesan* are grouped into the same noun category *cheese*.

6.5.2. Training Details

In all baselines, we use the same set of parameters outlined as follows. In particular, we utilize SGD optimizer and the momentum parameter to 0.9 and optimizer is initialized with $1e-1$ as learning rate, and also 128 instances were used as batch size. Additionally, we use the ReduceOnPlateau scheduler to reduce the learning rate during training when validation loss metric plateaus. To train the models for future utterance prediction, we employed cross-entropy loss and initialized network weights via uniform distribution, and used a dropout rate of 0.3 for both the encoder and the decoder. We used an early stopping strategy and stopped the training if validation BLEU did not improve after a certain threshold (patience = 50). We clipped gradients and set the gradient threshold to 0.1, and used a 4-head attention mechanism in the crossmodal block in all our multimodal models. As a preprocessing step, we replace multiword tokens with a single word. For example, each occurrence of “olive oil” is replaced with “olive_oil”. While training the models for action anticipation, we again used early stopping but stopped the training if validation loss did not improve after a certain threshold (patience = 5).

6.6. Results

6.6.1. Future Utterance Prediction

Table 6.1 Future utterance prediction results. Using audio, visual, or object features always improves performance compared to the language-only unimodal baseline. The reported results are the mean and standard deviation using over three independent runs.

Inputs	BLEU val	BLEU test	EM val	EM test	CA val	CA test
L	10.88 ± 0.7	10.55 ± 1.0	0.91 ± 0.4	0.61 ± 0.4	2.18 ± 0.6	2.16 ± 0.9
VL	19.49 ± 1.2	19.43 ± 0.7	4.60 ± 0.8	4.62 ± 0.4	8.62 ± 0.6	8.38 ± 0.3
AL	24.38 ± 0.9	25.21 ± 1.3	6.54 ± 0.4	6.29 ± 0.5	12.46 ± 0.7	11.67 ± 1.1
AVL	19.53 ± 0.1	20.10 ± 0.1	4.70 ± 0.1	4.72 ± 0.2	8.77 ± 0.3	9.13 ± 0.5
OL	24.36 ± 0.8	25.60 ± 1.4	6.13 ± 0.4	6.17 ± 0.3	12.28 ± 0.7	12.52 ± 0.4
OAL	24.97 ± 0.4	25.55 ± 0.5	6.17 ± 0.4	5.72 ± 0.6	12.39 ± 0.7	12.06 ± 1.0
PVL	26.48 ± 0.7	**27.78** ± 0.5	6.73 ± 0.7	**7.23** ± 0.5	13.27 ± 0.5	**14.28** ± 0.7

Table 6.1 shows the results of the future utterance prediction experiments. As can be seen, all of the multimodal models outperform the language-only baseline. Models that use visual features (VL) improve by 9 BLEU, 4 EM, and 10 CA points compared to the language-only model. Using visual features with a pre-trained model brings the largest overall improvement in performance; BLEU, EM, and CA increase by approximately 17, 6, and 19 points, respectively. However, using the combination of audio, visual, and language features (AVL) or using audio features in addition to object tags (OAL) do not bring further improvements, highlighting the difficulty of fusing multiple modalities. The performance of PVL shows the benefit of using pre-trained multimodal representations, as opposed to fusing separate unimodal encodings.

Table 6.2 Object-level attention accuracy at different layers of the crossmodal attention block in the VL model.

Layer1	Layer2	Layer3	Layer4	Layer5
8.6	9.8	10.4	8.8	9.8

To further investigate the behavior of the VL baseline, we examine the role of different layers in the crossmodal attention block in predicting which object regions correspond to the nouns in the target utterance. A model succeeds in this prediction task if the attention weight to the expected object region is maximized. In Table 6.2, we report these object-level attention accuracies. We observe that the middle layers and final layer maximize attention to the expected object region. This indicates that to achieve better generalization, the VL model needs to use the semantic information encoded in the latter layers of the crossmodal attention block.

Inputs (utterances and auxiliary modalities)				Prediction (future utterance)
Image				GT : place bowl L : put fridge OL : close bin VL : wash bowl AL : close bowl AVL : place bowl OAL : place bowl PVL : close bowl
Text	clean bowl .	open dishwasher .	open drawer .	
Image				GT : rinse fork L : put_down bowl OL : rinse fork VL : put fork AL : put fork AVL : place bowl OAL : rinse fork PVL : rinse fork
Text	turn_on tap .	rinse chopsticks .	take fork .	

Figure 6.3 Future Utterance Prediction qualitative results. Each model considers different combinations of input modality, as described in Section 6.3.. The targets, *rinse fork* and *place bowl* have not been observed by during the training phase; the multimodal inputs from Object, Audio, and Language are needed to predict them.

We demonstrate a qualitative baseline comparison in Fig. 6.3. In training time, *place bowl*, or *rinse fork* compounds have never been observed by the models. In both of these examples, text-only unimodal model fails to generalize to novel compositions whereas OAL baseline predicts the target composition correctly for both examples.

6.6.2. Action Anticipation

We report the results in Table 6.3 on the action anticipation task. We present the results for the val and test sets to demonstrate that this task poses a more challenging case for the models. The trend in performance is in line with those for the future utterance prediction task results. Even though there is not one particular model that outperforms the remaining models, multimodality improves the overall compositional generalization in the action anticipation task. Once again, the Object and Language and the Pretrained Vision and Language models perform strongly in both measures.

Table 6.3 Quantitative comparison of baselines for action anticipation task for predicting compound action (nouns and verbs). We report the mean across three runs. Best results are highlighted in bold, while the second-best results are underlined.

	EM val	EM test	CA val	CA test
L	2.32 ± 0.6	1.99 ± 0.3	6.34 ± 0.7	6.02 ± 0.9
VL	2.16 ± 0.2	2.02 ± 0.0	7.45 ± 0.3	7.12 ± 0.6
AL	5.03 ± 0.5	4.45 ± 0.9	12.04 ± 1.4	11.36 ± 2.6
AVL	3.23 ± 0.3	2.84 ± 0.6	9.95 ± 0.5	8.76 ± 0.9
OL	4.74 ± 1.0	5.31 ± 0.5	13.30 ± 1.6	14.50 ± 1.2
OAL	6.09 ± 0.7	5.14 ± 1.1	13.26 ± 0.4	12.48 ± 1.3
PVL	6.39 ± 0.5	**6.68** ± 0.3	14.06 ± 0.1	**14.92** ± 0.3

6.7. Related Work

Compositionality. Much recently, the compositionality problem has been investigated in various settings. [78] analyzed the capacity of artificial neural networks in linguistic compositionality. [75] examined systematicity and compositionality with a human-like number of examples. [77] investigated compositional generalization, in terms of a model's performance to composing unseen combinations of concepts when describing images. [205] explored compositionality in sentence embeddings for understanding how words combine for generalizing to unencountered words and phrases. [206] examined compositionality in sentence vector representations by probing the compositional information prevalent in the embeddings using a set of composition methods. [207] analyzed measuring compositionality in the aspect of representation learning *e.g.* a learned embedding. [74] investigated systematic generalization in a VQA-like setting. [76] examined compositionality in terms of systematic generalization for a meta-learning setting. [208] considered learning entire rule systems from examples instead of learning to predict the correct output given a novel input. [209] studied the emergence of systematic generalization in a situated agent setting where the agent learns to perform tasks based on textual instructions and visual observations. [62] proposed gSCAN dataset by extending the SCAN dataset to a 2d grid world setting for situated language understanding using grounded instructions. [210]

suggested a transformer-based method for analogical reasoning in language acquisition setup coupled with visual data to pick up novel words.

Among the prior work, the closest work to ours is [80] in which the authors investigated compositionality and generalization in a novel word acquisition setting from narrated videos. They suggested training a model using a masking strategy with a reference set where the model learns to map the masked words in the target tokens using the reference set examples in the same episode which is an image-text pair sequence. Another work that is closely similar is [211], where the authors create a setup to examine the systematic generalization in unseen compound acquisition setting from paired image-caption streams. They concluded that continual learning methods show little systematic generalization when trained in a shifting compound distribution. In our study, we focus on systematic generalization to novel compositions where models need to generalize to unseen compositions and hence learn how to learn primitive elements and concepts in the training set to generalize to novel compositions (*e.g.* see Fig.6.1), whereas in [80], the models learn to map target words from the reference set examples consisting of image-text pairs, and [211] investigate how well continual learning models systematically generalize under shifting compound distributions in a visually grounded setting.

Visually Grounded Reasoning. Neural reasoning models have shown good generalization across biased data splits toward addressing this problem. [61] proposed the CLEVR-CoGenT dataset derived from the CLEVR dataset to test the systematic generalization of models on visual reasoning tasks. [212] investigated how models generalize to previously unseen scenes in real buildings utilizing natural language navigation instructions that are grounded with visual information. [213] studied how models translate grounded instructions to robot actions to accomplish household tasks in novel environments. Related to our work, [214] predicted future utterances from multimodal data using instructional videos and used the transcribed speech as text where the goal is to rank correct utterances among candidates. However, we focus on predicting future utterances by generating utterances to assess the systematic generalization abilities of models trained on different and multiple multimodalities.

6.8. Experimental Setup

Systematicity Split (`EK-100-SYS`). Fig. 6.4 illustrates the atomic and compound distributions over the constructed training, validation, and test splits of our proposed systematicity setup. As can be seen, while these splits have similar distributions over atoms, training, and val/test splits do differ in terms of compounds.

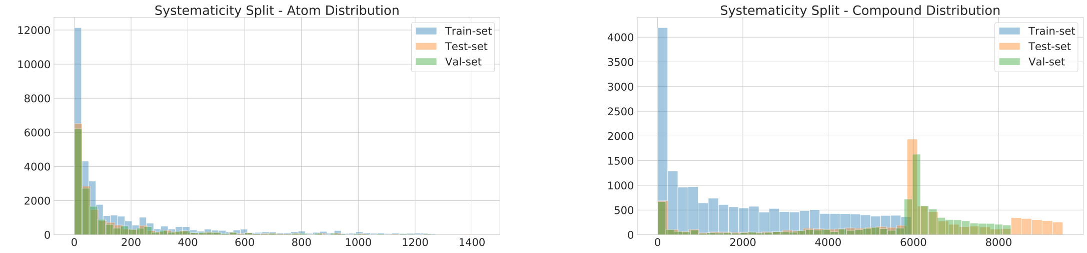


Figure 6.4 For train/val/test splits for systematicity split setup, the plot at the top demonstrates the distribution of atoms while the plot at the bottom shows the distribution of compounds.

Choosing Keyframes from Videos. In our experimental setup, we choose and use representative images from each microsegment. While choosing such a representative image for each video sequence, we follow a simple heuristics-based strategy. In particular, we run an object detector on the video frames and select the frames containing the maximum number of object proposals captured by the object detector as the representative frames.

Table 6.4 Model sizes and their training times along with the vocabulary sizes considered in our experiments.

Model	Parameters	Training Time
L	2,827,218	10 mins
OL	6,833,874	35 mins
VL	7,301,074	18 mins
AL	6,907,858	31 mins
AVL	15,331,026	37 mins
OAL	14,863,826	39 mins
PVL	6,973,394	31 mins

Model Sizes and Training Time. In Table 6.4, we illustrate baseline model sizes and training time for the future utterance prediction task. All of our models are implemented with PyTorch and trained with Nvidia 1080Ti GPUs.

Further Analysis. In Table 6.5 we provide how models generalize for isolated verbs and in Table 6.6 we provide generalization performance for isolated nouns for the action anticipation task.

Table 6.5 Quantitative comparison of baselines for action anticipation task for predicting target verbs. We report the mean across three runs. Best results are highlighted in bold, while the second-best results are underlined.

	EM val	EM test	CA val	CA test
L	19.05 ± 0.1	18.97 ± 0.2	38.55 ± 0.4	**40.21** ± 0.7
VL	21.15 ± 0.3	**19.69** ± 0.5	38.64 ± 1.0	37.67 ± 1.2
AL	19.45 ± 0.6	19.44 ± 0.9	37.02 ± 1.4	37.02 ± 2.2
AVL	20.52 ± 0.4	19.58 ± 0.5	38.75 ± 1.4	38.70 ± 1.1
OL	18.94 ± 0.6	18.79 ± 0.2	38.51 ± 1.1	38.98 ± 1.1
OAL	18.82 ± 1.2	19.59 ± 0.7	36.09 ± 1.1	37.17 ± 1.8
PVL	18.88 ± 0.6	19.36 ± 0.2	35.00 ± 0.9	36.23 ± 1.1

Table 6.6 Quantitative comparison of baselines in the action anticipation task for predicting target nouns. We report the mean across three runs. Best results are highlighted in bold, while the second-best results are underlined.

	EM val	EM test	CA val	CA test
L	10.15 ± 2.3	10.04 ± 2.4	15.42 ± 2.0	15.86 ± 2.2
VL	11.87 ± 0.5	12.2 ± 1.1	19.94 ± 0.2	21.17 ± 0.8
AL	23.13 ± 0.6	23.48 ± 0.8	30.65 ± 1.0	31.52 ± 1.6
AVL	16.34 ± 0.1	16.16 ± 0.6	24.98 ± 0.3	25.34 ± 0.4
OL	27.08 ± 0.8	28.12 ± 0.6	35.49 ± 0.5	36.49 ± 0.1
OAL	28.81 ± 0.4	31.28 ± 1.1	35.34 ± 0.7	38.03 ± 1.0
PVL	31.64 ± 2.8	**33.33** ± 2.0	37.45 ± 3.4	**39.06** ± 1.7

6.9. Discussions

In this chapter, we presented an investigation of linguistic compositionality and systematic generalization in a perceptually grounded setting. We showed how a multimodal how-to instructions dataset can be utilized as a challenging test bed for this purpose. We designed the future utterance prediction and action anticipation tasks and followed a methodical approach to generate train, test, and validation sets in our systematicity split. Additionally, we experimented with several baseline models and investigated models' ability to generalize to novel compositions and showed how multimodal data can contribute towards solving systematic generalization problem. We found that a multimodal encoder pre-trained on video data gave the best generalization performance. Our findings indicate that the models that exploit visual and audio signals do indeed improve over the text-only model when they are evaluated on the long tail of rare complex concepts. We hope our work will stimulate further research along these directions. That being said, the textual utterances that we consider in our work are simplistic and do not capture all of the complexities of natural language. Hence, extending this work to a more natural source of language data will be quite interesting. Furthermore, we highlight a few limitations of our work, which are listed below.

In our work, to analyze how visual and audio signals affect linguistic compositionality, we proposed a new dataset called `EK-100-SYS` that is curated from the EPIC-Kitchens-100 dataset [9]. As mentioned, this dataset involves videos including daily kitchen activities in natural household environments. Hence, it could be interesting to conduct future studies in an open-domain setting to alleviate limitations of the domain-specific nature of the EPIC-Kitchens-100 dataset.

We investigate several different multimodal models for both future utterance prediction and action anticipation tasks. However, it is important to note that for multimodal learning how to integrate different modalities is considered an open research problem. In the literature, different strategies for multimodal data fusion have been proposed. Our experimental analysis could be further extended by considering some models that fuse the modalities in a way different than ours. More interestingly, from a systematic generalization point of view, an analysis could also be carried out to explore the most effective fusion scheme.

7. CONCLUSION

In the last few years, we have been observing tremendous progress in both natural language processing as well as computer vision fields which is paving the way toward a joint understanding of language and vision tasks.

In this thesis, we explore the problem of understanding multimodal how-to instructions with images and text and argue that joint understanding of multimodal how-to instructions is the next frontier in AI research. We broadly define "multimodal machine comprehension" and investigate the proposed research problem from the point of comprehension, reasoning, and systematic generalization.

Our emphasis is on understanding multimodal how-to instructions, such as cooking recipes with step-by-step instructions, and analyzing everyday household task videos consisting of multiple modalities, which are rich in context, objects, scenes, interactions, and temporal relations and procedures. We analyze the multimodal machine comprehension problem through three different lenses:

- We propose a multimodal machine comprehension dataset consisting of cooking recipes with images and text (Chapter 4.)
- We propose a new neural model to understand multimodal procedures and to keep track of the entity state changes and interactions between entities in recipes (Chapter 5.)
- We propose a systematic generalization setup leveraging a natural dataset consisting of household actions and show the contribution of grounding towards models' compositional generalization abilities (Chapter 6.).

In particular, throughout this thesis, we thoroughly analyzed the multimodal machine comprehension problem for how-to instructions with images and text. Toward understanding multimodal how-to instructions, we presented RecipeQA, which consists of cooking recipes with context-question-answer triplets. To our knowledge, RecipeQA is the first machine comprehension dataset that deals with understanding procedural knowledge in a multimodal setting. Each one of the four question styles in our dataset is specifically tailored to evaluate a particular skill and requires connecting the dots between different modalities. The results of our baseline models demonstrate that RecipeQA is a challenging dataset for multimodal machine comprehension. Furthermore, we proposed a new neural architecture called Procedural Reasoning Networks (PRN) for the multimodal understanding of step-by-step instructions. We presented a model which is equipped with an explicit memory unit that provides an implicit mechanism to keep track of the changes in the states of the entities over the course of the procedure. Using the proposed method and utilizing the RecipeQA dataset, we significantly improved upon the previous results for the visual reasoning tasks, indicating the proposed neural model understands the procedural text and the accompanying images better. Additionally, we carefully analyzed our results and found that our approach learns meaningful dynamic representations of entities without any entity-level supervision. We explored the compositional generalization problem which stands as a serious challenge to the real success of deep learning models and reviewed the current state of compositional generalization research from the point of datasets, tasks, and modeling methodologies and discussed the open research challenges and limitations in the existing literature to lay the groundwork for our research problem. Finally, we presented an investigation of linguistic compositionality and systematic generalization in a perceptually grounded setting. We showed how a multimodal how-to instructions dataset can be utilized as a challenging test bed for this purpose. We designed the future utterance prediction and action anticipation tasks and followed a methodical approach in generating the training, validation, and test sets for benchmarking models' compositional generalization abilities in a multimodal how-to instructions dataset. We experimented with several baseline models and investigated models' ability to generalize to novel compositions and showed how grounding linguistic data with auxiliary multimodal data can contribute towards solving systematic generalization problem. We showed that multimodality can indeed improve models' compositional generalization performance even if models are not exposed to certain compositions during training time and can better generalize to novel compositions thanks to grounding textual data with auxiliary modalities such as images and audio.

Throughout this thesis, we made thorough analyses on various tasks and investigated the contribution of multimodality and grounding toward understanding how-to instructions with images and text. Our empirical findings indicate that both in the existence of another modality or with grounding, we showed that the models seem to benefit from this auxiliary signal. Our world is multimodal in nature, and the richness and complexity of data, including sensory information, visual, auditory, spatial *etc.* is astounding. We believe humans leverage all these sources of rich modalities while solving various tasks without too much effort or even thinking about them. From this perspective, we believe our empirical findings also seem to be aligned with this phenomenon that the better we can exploit more and multiple modalities, the better machines will perform in solving various tasks.

In conclusion, throughout this thesis, we investigated the multimodal understanding of how-to instructions from various perspectives, reviewed the current state of the literature, proposed methods and models to show how multimodality helps models generalization performance in various reasoning and comprehension tasks on different datasets, and showed how linguistic components grounded with different modalities can help models to systematically generalize to never seen compositions even models were not exposed to these compositions during training time.

7.1. Future work and open directions

In this thesis, we proposed two new datasets, namely the `RecipeQA` dataset, curated from online cooking recipes, and `EK-100-SYS`, curated from the EPIC-Kitchens-100 dataset consisting of videos including daily kitchen activities. Even though we believe cooking recipes capture the overall characteristics and challenges of how-to instructions, hence a good representation of the problem space, an interesting extension to our work would be to explore different domains beyond cooking recipes, and daily household tasks, such as building a larger-scale how-to instructions benchmark dataset to include broader categories and different domains *e.g.* auto repair or do-it-yourself type of how-to guides and conduct future studies in an open-domain setting.

The datasets we curated, in particular, the RecipeQA and EK-100-SYS are user-generated sources. Cooking recipes are authored by several users from the web but might include certain biases specific to the data source. This is also true for household tasks as well. Even though both data sources are comprised of user-generated data coming from multiple participants and authors, these datasets might contain cultural nuances and domain limitations or could be subject to inherent biases due to data collection and curation approaches followed while curating these datasets. An interesting research direction could be to collect more diverse and larger datasets to reduce potential biases that might be present in these datasets.

The systematic generalization problem is an open research problem that remains to be solved. Most of the previous work focused on artificially generated data as we discussed in the previous sections. In particular, we carefully curated the EK-100-SYS dataset for the systematic generalization problem which is a multimodal dataset. One particular reason for choosing the Epic-Kitchens dataset to create such as benchmark was mainly because the RecipeQA dataset is noisier and a more complex dataset. One interesting research direction could be to explore the transferability of our empirical findings from the EK-100-SYS benchmark to a more noisy challenging dataset like the RecipeQA dataset to better understand models' compositional generalization abilities.

Throughout the thesis, we made thorough analyses with different reasoning and comprehension tasks on the proposed datasets and investigated the contribution of multimodality and grounding in different setups. An obvious future research direction would be exploring the contribution of multimodality in different downstream tasks such as navigation, entailment, word acquisition *etc*.

Visually grounded text processing has become an increasingly popular research field attracting researchers from both the vision and language communities. Nevertheless, in the existence of additional auxiliary modalities such as audio, object tags, or speech, joint modeling of such modalities remains an underexplored direction in the literature. Therefore building such models that can incorporate the contextual signals coming from different modalities in a joint fashion would be an interesting research direction to explore.

Furthermore, we explored how different modalities could improve models' generalization abilities through different setups and experiments. However, integrating different modalities remains an open research problem. An interesting research direction would be analyzing different fusion techniques, both early and late fusion strategies in the existence of additional modalities accompanying visual and textual data such as audio, *etc*.

In the last few years, we have been seeing rapid progress in the development of new neural models. Even though we did thorough empirical analyses throughout this thesis, extending the proposed analyses with larger novel neural model baselines with greater capacity could also be an interesting research direction.

An obvious extension of our work on compositional generalization is to design specific architectures for the joint modeling of multiple modalities to further improve the systematic generalization performance of neural models.

For procedural understanding, we proposed a novel neural model, namely the Procedural Reasoning Networks (PRN). After the introduction of PRNs, there has been a great deal of work focusing on Transformer based models, specifically for vision and language tasks. In our proposed PRN model we used a neural backbone for processing text and images. Nevertheless, we believe the proposed model could be improved by utilizing a transformer-based backbone such as utilizing a vision-and-language transformer, which could be a fruitful research direction.

A fruitful research direction would be to develop a large-scale comprehensive multimodal benchmark dataset and evaluation methods to test models' compositional generalization across different comprehension and reasoning tasks.

Another interesting research direction would be exploring the embodiment of multimodal how-to instructions setup towards developing real-world applications such as in robotics.

In particular, given the cooking recipe instructions, designing systems that can prepare the actual food by interacting with real-world objects and ingredients, which remains an open research problem and we hope our work could further stimulate research towards that direction.

Finally, the RecipeQA dataset introduces a challenging benchmark for understanding step-by-step how-to instructions. We hope our work would motivate further research towards understanding multimodal how-to instructions and facilitate research in this direction both in vision and language communities.

REFERENCES

[1] David Marr and A Vision. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*, 1(2), **1982**.

[2] Maggie Tallerman and Kathleen R Gibson. *The Oxford handbook of language evolution*. Oxford University Press, **2012**.

[3] Stephen R. Anderson. How many languages are there in the world? `http://www.linguisticsociety.org/content/how-many-languages-are-there-world`, **2010**. Accessed: 2017-05-10.

[4] Stephen R. Anderson. When nettles were dish of the day. `http://www.linguisticsociety.org/content/how-many-languages-are-there-world`, **2010**. Accessed: 2017-05-10.

[5] Peter RAULWING and H Meyer. The kikkuli text. *Hittite Training Instructions for Chariot Horses in the Second Half of the 2nd Millennium BC and Their Interdisciplinary Context*, **2009**.

[6] Cassie. Salad-in-a-jar 101: How to make mason jar salads + 4 fool-proof salad in a jar recipes. `https://wholefully.com/salad-in-a-jar-101/`, **2017**. Accessed: 2017-05-29.

[7] ale 8-1. How-to-de-seed-a-tomato. `http://www.instructables.com/id/How-to-De-Seed-a-Tomato`. Accessed: 2022-11-14.

[8] jessyratfink. Perfect-oven-sweet-potato-fries. `https://www.instructables.com/Perfect-Oven-Sweet-Potato-Fries/`. Accessed: 2022-12-11.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, **2020**.

[10] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, **2013**.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, **2014**.

[12] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433. **2015**.

[13] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137. **2015**.

[14] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer, **2011**.

[15] Rebecca Mason and Eugene Charniak. Nonparametric method for data-driven image captioning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598. **2014**.

[16] Semih Yagcioglu, Erkut Erdem, Aykut Erdem, and Ruket Cakici. A distributed representation based query expansion approach for image captioning. In *ACL (2)*, pages 106–111. **2015**.

[17] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690. **2014**.

[18] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2461–2469. **2015**.

[19] Lynette Hirschman, Marc Light, Eric Breck, and John D Burger. Deep read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 325–332. Association for Computational Linguistics, **1999**.

[20] Lucy Vanderwende. Answering and questioning for machine reading. In *AAAI Spring Symposium: Machine Reading*, page 91. **2007**.

[21] Ellen Riloff and Michael Thelen. A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 13–19. Association for Computational Linguistics, **2000**.

[22] Eugene Charniak, Yasemin Altun, Rodrigo de Salvo Braz, Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang, Changhee Pyo, Ye Sun, Wei Wy, et al. Reading comprehension programs in a statistical-language-processing class. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 1–5. Association for Computational Linguistics, **2000**.

[23] Hwee Tou Ng, Leong Hwee Teo, and Jennifer Lai Pheng Kwan. A machine learning approach to answering questions for reading comprehension tests. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical*

methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, pages 124–132. Association for Computational Linguistics, **2000**.

[24] W Wang, J Auer, R Parasuraman, I Zubarev, D Brandyberry, and MP Harper. A question answering system developed as a project in a natural language processing course. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 28–35. Association for Computational Linguistics, **2000**.

[25] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32. **2007**.

[26] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, volume 3, page 4. **2013**.

[27] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018. **2015**.

[28] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, **2015**.

[29] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, **2016**.

[30] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read

and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701. **2015**.

[31] Danqi Chen, Jason Bolton, and Christopher D Manning. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*, **2016**.

[32] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, **2015**.

[33] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448. **2015**.

[34] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics, **2010**.

[35] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, pages 858–867. Citeseer, **2013**.

[36] Aliaksei Severyn and Alessandro Moschitti. Automatic feature engineering for answer selection and extraction. In *EMNLP*, volume 13, pages 458–467. **2013**.

[37] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, **2016**.

[38] 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. *arXiv preprint arXiv:1611.05118*, **2016**.

[39] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004. **2016**.

[40] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–9. **2015**.

[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, **2002**.

[42] Michael Denkowski and Alon Lavie. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proc. of EACL Workshop on Statistical Machine Translation*. **2014**.

[43] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, **1994**.

[44] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in Neural Information Processing Systems*, pages 2296–2304. **2015**.

[45] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, **2015**.

[46] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, **2016**.

[47] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *Computer Vision–ECCV 2012*, pages 746–760, **2012**.

[48] Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. Zero-shot learning via visual abstraction. In *European Conference on Computer Vision*, pages 401–416. Springer, **2014**.

[49] C Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016. **2013**.

[50] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640. **2016**.

[51] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*, **2016**.

[52] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. **2017**.

[53] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the*

35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR, **2018**.

[54] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, **2015**.

[55] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, **1988**.

[56] Gary F Marcus. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282, **1998**.

[57] Jerry A Fodor and Ernest Lepore. *The compositionality papers*. Oxford University Press, **2002**.

[58] Paco Calvo and John Symons. *The architecture of cognition: Rethinking Fodor and Pylyshyn's systematicity challenge*. MIT Press, **2014**.

[59] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, **2020**.

[60] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, **2016**.

[61] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society, **2017**. doi:10.1109/CVPR.2017.215.

[62] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. **2020**.

[63] Gottlob Frege. Compound thoughts. *Mind*, 72(285):1–17, **1963**.

[64] Barbara Partee et al. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, **1995**.

[65] Paul Smolensky and Géraldine Legendre. *The harmonic mind: From neural computation to optimality-theoretic grammar (Cognitive architecture), Vol. 1*. MIT press, **2006**.

[66] M Christiansen, Nick Chater, et al. Generalization and connectionist language learning. *Mind and Language*, 9(3), **1994**.

[67] Franklin Chang. Symbolically speaking: A connectionist model of sentence production. *Cognitive science*, 26(5):609–651, **2002**.

[68] Frank van der Velde, Gwendid T van der Voort van der Kleij, and Marc de Kamps. Lack of combinatorial productivity in language processing with simple recurrent networks. *Connection Science*, 16(1):21–46, **2004**.

[69] Matthew M Botvinick and David C Plaut. Short-term memory for serial order: a recurrent neural network model. *Psychological review*, 113(2):201, **2006**.

[70] Francis CK Wong and William SY Wang. Generalisation towards combinatorial productivity in language acquisition by simple recurrent networks. In *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 139–144. IEEE, **2007**.

[71] Philémon Brakel and Stefan Frank. Strong systematicity in sentence processing by simple recurrent networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 31. **2009**.

[72] Stefan L Frank, P Calvo, and J Symons. Getting real about systematicity. *The architecture of cognition: Rethinking Fodor and Pylyshyn's systematicity challenge*, pages 147–164, **2014**.

[73] Samuel R Bowman, Christopher D Manning, and Christopher Potts. Tree-structured composition in neural networks without tree-structured architectures. *arXiv preprint arXiv:1506.04834*, **2015**.

[74] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron C. Courville. Systematic generalization: What is required and can it be learned? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, **2019**.

[75] Brenden M. Lake, Tal Linzen, and Marco Baroni. Human few-shot learning of compositional instructions. In Ashok K. Goel, Colleen M. Seifert, and Christian Freksa, editors, *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 611–617. cognitivesciencesociety.org, **2019**.

[76] Brenden M. Lake. Compositional generalization through meta sequence-to-sequence learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9788–9798. **2019**.

[77] Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikatte, and Desmond Elliott. Compositional generalization in image captioning. In Mohit Bansal and Aline Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 87–98. Association for Computational Linguistics, **2019**. doi:10.18653/v1/K19-1009.

[78] Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. *Phil. Trans. R. Soc. B*, 375(1791):20190307, **2020**.

[79] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*, **2019**.

[80] Dídac Surís, Dave Epstein, Heng Ji, Shih-Fu Chang, and Carl Vondrick. Learning to learn words from visual scenes. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pages 434–452. Springer, **2020**. doi:10.1007/978-3-030-58526-6_26.

[81] Jasmijn Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. Jump to better conclusions: SCAN both left and right. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi, editors, *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 47–55. Association for Computational Linguistics, **2018**. doi:10.18653/v1/w18-5407.

[82] Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4767–4780. Association for Computational Linguistics, Online, **2021**. doi:10.18653/v1/2021.acl-long.368.

[83] Verna Dankers, Elia Bruni, and Dieuwke Hupkes. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175. Association for Computational Linguistics, Dublin, Ireland, **2022**. doi:10.18653/v1/2022.acl-long.286.

[84] Adam Liška, Germán Kruszewski, and Marco Baroni. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*, **2018**.

[85] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2015**.

[86] Yuval Atzmon, Jonathan Berant, Vahid Kezami, Amir Globerson, and Gal Chechik. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*, **2016**.

[87] Emanuele Bugliarello and Desmond Elliott. The role of syntactic planning in compositional image captioning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 593–607. Association for Computational Linguistics, Online, **2021**. doi:10.18653/v1/2021.eacl-main.48.

[88] George Pantazopoulos, Alessandro Suglia, and Arash Eshghi. Combine to describe: Evaluating compositional generalization in image captioning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 115–131. Association for

Computational Linguistics, Dublin, Ireland, **2022**. doi:10.18653/v1/2022.acl-srw.11.

[89] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, **2014**.

[90] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, **2015**.

[91] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *ArXiv*, abs/1504.00325, **2015**.

[92] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086. **2018**.

[93] Harm de Vries, Dzmitry Bahdanau, Shikhar Murty, Aaron C. Courville, and Philippe Beaudoin. CLOSURE: assessing systematic generalization of CLEVR models. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*. **2019**.

[94] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980. **2018**.

[95] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, **2018**.

[96] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, **2019**.

[97] Zhengxuan Wu, Elisa Kreiss, Desmond C. Ong, and Christopher Potts. ReaSCAN: Compositional reasoning in language grounding. *NeurIPS 2021 Datasets and Benchmarks Track*, **2021**.

[98] Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105. Association for Computational Linguistics, Online, **2020**. doi:10.18653/v1/2020.emnlp-main.731.

[99] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*. **2018**.

[100] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, **2021**.

[101] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, **1990**.

[102] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, Doha, Qatar, **2014**. doi:10.3115/v1/D14-1181.

[103] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, **2017**.

[104] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 804–813. **2017**.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008. **2017**.

[106] Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607. **2022**.

[107] Arkil Patel, Satwik Bhattamishra, Phil Blunsom, and Navin Goyal. Revisiting the compositional generalization abilities of neural sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 424–434. **2022**.

[108] Ankur Sikarwar, Arkil Patel, and Navin Goyal. When can transformers ground and compose: Insights from compositional generalization benchmarks. *arXiv preprint arXiv:2210.12786*, **2022**.

[109] Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634. **2021**.

[110] 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**.

[111] Juyong Kim, Pradeep Ravikumar, Joshua Ainslie, and Santiago Ontanon. Improving compositional generalization in classification tasks via structure annotations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 637–645. **2021**.

[112] Jacob Andreas. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566. **2020**.

[113] Demi Guo, Yoon Kim, and Alexander M Rush. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552. **2020**.

[114] Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. Learning to recombine and resample data for compositional generalization. In *International Conference on Learning Representations*. **2020**.

[115] Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. Compositional

semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*, **2022**.

[116] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics, Brussels, Belgium, **2018**. doi:10.18653/v1/W18-5446.

[117] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., **2019**.

[118] Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. The GEM

benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672, **2021**.

[119] Lynette Hirschman and Robert Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300, **2001**.

[120] Christopher JC Burges. Towards the machine comprehension of text: An essay. Technical report, Technical report, Microsoft Research Technical Report MSR-TR-2013-125, **2013**.

[121] Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 3089–3096. **2017**.

[122] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, **2018**.

[123] Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Association for Computational Linguistics (ACL)*. **2016**.

[124] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*, pages 1601–1611. Association for Computational Linguistics, Vancouver, Canada, **2017**.

[125] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in

visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2017**.

[126] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251. Springer, **2016**.

[127] Samira Ebrahimi Kahou, Adam Atkinson, Vincent Michalski, Akos Kadar, Adam Trischler, and Yoshua Bengio. FigureQA: An annotated figure dataset for visual reasoning. In *International Conference on Learning Representations (ICLR) Workshops*. **2018**.

[128] Peter Norvig. A unified theory of inference for text understanding. Technical report, University of California at Berkeley, Berkeley, CA, USA, **1987**.

[129] Léon Bottou. From machine learning to machine reasoning - an essay. *Machine Learning*, 94(2):133–149, **2014**.

[130] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. **2018**.

[131] TechMartian. Creamy-coconut-chickpea-curry-vegan-and-gluten-fre. `https://www.instructables.com/Creamy-Coconut-Chickpea-Curry-Vegan-anc` Accessed: 2022-11-14.

[132] Marco Lui and Timothy Baldwin. `langid.py`: An off-the-shelf language identification tool. In *Association for Computational Linguistics (ACL) Demo Session*, pages 25–30. **2012**.

[133] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning (ICML)*, pages 957–966. **2015**.

[134] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2016**.

[135] Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. Sort story: Sorting jumbled images and captions into stories. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 925–931. **2016**.

[136] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2017**.

[137] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)*, pages 1188–1196. **2014**.

[138] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet: Large scale visual recognition challenge. *International Journal of Computer Vision*, 113:211–252, **2015**.

[139] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, pages 2493—-2537, **2011**.

[140] Jonathan Malmaud, Earl Wagner, Nancy Chang, and Kevin Murphy. Cooking with semantics. In *ACL 2014 Workshop on Semantic Parsing*, pages 33–38. **2014**.

[141] Jermsak Jermsurawong and Nizar Habash. Predicting the structure of cooking recipes. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–786. **2015**.

[142] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, **2015**.

[143] Ozan Sener, Amir Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *IEEE International Conference on Computer Vision (ICCV)*. **2015**.

[144] Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 329—-339. **2016**.

[145] Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. Simulating action dynamics with neural process networks. In *International Conference on Learning Representations (ICLR)*. **2018**.

[146] Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. Simulating Action Dynamics with Neural Process Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. **2018**.

[147] Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. **2018**.

[148] Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. **2018**.

[149] Xinya Du, Bhavana Dalvi Mishra, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. Be consistent! improving procedural text

comprehension using label consistency. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. **2019**.

[150] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking The World State with Recurrent Entity Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. **2017**.

[151] Trapit Bansal, Arvind Neelakantan, and Andrew McCallum. RelNet: End-to-End Modeling of Entities & Relations. In *NeurIPS Workshop on Automated Knowledge Base Construction (AKBC)*. **2017**.

[152] Juan Pavez, Hector Allende, and Hector Allende-Cid. Working memory networks: Augmenting memory networks with a relational reasoning module. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1000–1009. **2018**.

[153] Julien Perez and Fei Liu. Dialog state tracking, a machine reading approach using memory network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 305–314. **2017**.

[154] Hai Wang, Takeshi Onishi, Kevin Gimpel, and David McAllester. Emergent predication structure in hidden state vectors of neural readers. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 26–36. Association for Computational Linguistics, Vancouver, Canada, **2017**.

[155] Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Neural Models for Reasoning over Multiple Mentions using Coreference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. **2018**.

[156] Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A Smith. Dynamic Entity Representations in Neural Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. **2017**.

[157] Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR)*. **2019**.

[158] Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational Recurrent Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7299–7310. **2018**.

[159] Mustafa Sercan Amac, Semih Yagcioglu, Aykut Erdem, and Erkut Erdem. Procedural reasoning networks for understanding multimodal procedures. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 441–451. **2019**.

[160] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6. Association for Computational Linguistics, Melbourne, Australia, **2018**.

[161] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. **2014**.

[162] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the Conference*

on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543. **2014**.

[163] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237. **2018**.

[164] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. **2015**.

[165] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Rearning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. **2016**.

[166] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. **2009**.

[167] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. **2017**.

[168] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR)*. **2017**.

[169] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1M: A Dataset for

Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *arXiv preprint arXiv:1810.06553*, **2018**.

[170] Yummly. Kaggle What's Cooking? `https://www.kaggle.com/c/whats-cooking/data`, **2015**. [Accessed: 2018-05-31].

[171] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic Differentiation in pytorch. In *NIPS-W*. **2017**.

[172] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, **2018**.

[173] Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. Cite: A corpus of image-text discourse relations. *arXiv preprint arXiv:1904.06286*, **2019**.

[174] Jack Hessel, Lillian Lee, and David Mimno. Unsupervised discovery of multimodal links in multi-image, multi-sentence documents. *arXiv preprint arXiv:1904.07826*, **2019**.

[175] Malihe Alikhani and Matthew Stone. "caption" as a coherence relation: Evidence and implications. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 58–67. **2019**.

[176] Ao Liu, Lizhen Qu, Junyu Lu, Chenbin Zhang, and Zenglin Xu. Machine reading comprehension: Matching and orders. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2057–2060. **2019**.

[177] Ao Liu, Shuai Yuan, Chenbin Zhang, Congjian Luo, Yaqing Liao, Kun Bai, and Zenglin Xu. Multi-level multimodal transformer network for multimodal recipe comprehension. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1781–1784. **2020**.

[178] Vishwash Batra, Aparajita Haldar, Yulan He, Hakan Ferhatosmanoglu, George Vogiatzis, and Tanaya Guha. Variational recurrent sequence-to-sequence retrieval for

stepwise illustration. In *European Conference on Information Retrieval*, pages 50–64. Springer, **2020**.

[179] Hossein Rajaby Faghihi, Roshanak Mirzaee, Sudarshan Paliwal, and Parisa Kordjamshidi. Latent alignment of procedural concepts in multimodal recipes. *arXiv preprint arXiv:2101.04727*, **2021**.

[180] Kenta Hama, Takashi Matsubara, Kuniaki Uehara, and Jianfei Cai. Exploring uncertainty measures for image-caption embedding-and-retrieval task. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(2):1–19, **2021**.

[181] Pritish Sahu, Karan Sikka, and Ajay Divakaran. Towards solving multimodal comprehension. *arXiv preprint arXiv:2104.10139*, **2021**.

[182] Yunjie Wu, Sai Zhang, Xiaowang Zhang, Zhiyong Feng, and Liang Wan. A knowledge-based deep heterogeneous graph matching model for multimodal recipeqa. In *ISWC (Posters/Demos/Industry)*. **2021**.

[183] Malihe Alikhani, Fangda Han, Hareesh Ravi, Mubbasir Kapadia, Vladimir Pavlovic, and Matthew Stone. Cross-modal coherence for text-to-image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10427–10435. **2022**.

[184] Pritish Sahu, Karan Sikka, and Ajay Divakaran. Challenges in procedural multimodal machine comprehension: A novel way to benchmark. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3654–3663. **2022**.

[185] Vishal Pallagani, Priyadharsini Ramamurthy, Vedant Khandelwal, Revathy Venkataramanan, Kausik Lakkaraju, Sathyanarayanan N Aakur, and Biplav Srivastava. A rich recipe representation as plan to support expressive multi modal queries on recipe content and preparation process. *arXiv preprint arXiv:2203.17109*, **2022**.

[186] Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. In *Proceedings of the 60th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4525–4542. **2022**.

[187] Huibin Zhang, Zhengkun Zhang, Yao Zhang, Jun Wang, Yufan Li, Zhenglu Yang, et al. Modeling temporal-modal entity graph for procedural multimodal machine comprehension. *arXiv preprint arXiv:2204.02566*, **2022**.

[188] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P. Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. SCAN: learning hierarchical compositional visual concepts. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, **2018**.

[189] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, **2017**.

[190] João Loula, Marco Baroni, and Brenden M. Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi, editors, *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 108–114. Association for Computational Linguistics, **2018**. doi:10.18653/v1/w18-5413.

[191] Jake Russin, Jason Jo, Randall C. O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *CoRR*, abs/1904.09708, **2019**.

[192] Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. Iterated learning for emergent systematicity in VQA. In *International Conference on Learning Representations*. **2021**.

[193] Moshe Bar. The proactive brain: using analogies and associations to generate predictions. *Trends in cognitive sciences*, 11(7):280–289, **2007**.

[194] Andy Clark. *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press, **2015**.

[195] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5562–5571. **2019**.

[196] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934. **2019**.

[197] JK Chung, PL Kannappan, CT Ng, and PK Sahoo. Measures of distance between probability distributions. *Journal of mathematical analysis and applications*, 138(1):280–292, **1989**.

[198] Jesse Thomason, Daniel Gordon, and Yonatan Bisk. Shifting the baseline: Single modality performance on visual navigation & QA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1977–1983. Association for Computational Linguistics, Minneapolis, Minnesota, **2019**. doi:10.18653/v1/N19-1197.

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

[200] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, **2014**.

[201] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Florence, Italy, **2019**.

[202] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, **2017**. doi:10.1109/TPAMI.2016.2577031.

[203] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, **2020**.

[204] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. *arXiv preprint arXiv:2201.02639*, **2022**.

[205] Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. Evaluating compositionality in sentence embeddings. *CoRR*, abs/1802.04302, **2018**.

[206] Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. Assessing composition in sentence vector representations. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1790–1801. Association for Computational Linguistics, **2018**.

[207] Jacob Andreas. Measuring compositionality in representation learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, **2019**.

[208] Maxwell Nye, Armando Solar-Lezama, Joshua Tenenbaum, and Brenden Lake. Learning compositional rules via neural program synthesis. *Proceedings of the NeurIPS*, **2019**.

[209] Felix Hill, Andrew K. Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Emergent systematic generalization in a situated agent. *CoRR*, abs/1910.00571, **2019**.

[210] Bo Wu, Haoyu Qin, Alireza Zareian, Carl Vondrick, and Shih-Fu Chang. Analogical reasoning for visually grounded language acquisition. *CoRR*, abs/2007.11668, **2020**.

[211] Xisen Jin, Junyi Du, and Xiang Ren. Visually grounded continual learning of compositional semantics. *CoRR*, abs/2005.00785, **2020**.

[212] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683. **2018**.

[213] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. IEEE, **2020**. doi:10.1109/CVPR42600.2020.01075.

[214] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. Look before you speak: Visually contextualized utterances. *CoRR*, abs/2012.05710, **2020**.