PREDICTING ACTIONS IN IMAGES USING DISTRIBUTED LEXICAL REPRESENTATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE RESEARCH SERVICES
OF THE UNIVERSITY OF SHEFFIELD
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Abdulsalam Alsunaidi

August 2023

Dedicated to my parents, sisters, and brothers

Abstract

Artificial intelligence has long sought to develop agents capable of perceiving the complex visual environment around us and communicating about it using natural language. In recent years, significant strides have been made towards this objective, particularly in the field of image content description. For instance, current artificial systems are able to classify images of a single object with a high level of accuracy that is sometimes comparable to that of humans.

Although there has been remarkable progress in recognising objects, there has been less headway in action recognition due to a significant limitation in the current approach. Most of the advances in visual recognition rely on classifying images into distinct and non-overlapping categories. While this approach may work well in many contexts, it is inadequate for understanding actions. It constrains the categorisation of an action to a single interpretation, thereby preventing an agent from proposing multiple possible interpretations.

To tackle this fundamental limitation, this thesis proposes a framework that seeks to describe action-depicting images using multiple verbs, and expands the vocabulary used to describe such images beyond the limitations of the training dataset. In particular, the framework leverages lexical embeddings as a supplementary tool to go beyond the verbs that are supplied as explicit labels for images in datasets used for supervised training of action classifiers. More specifically, these embeddings are used for representing the target labels (i.e., verbs). By exploiting a richer representations of human actions, this framework has the potential to improve the capability of artificial agents to accurately recognise and describe human actions in images.

In this thesis, we focus on the representation of input images and target labels. We examine various components for both elements, ranging from commonly used off-the-shelf options to custom-designed ones tailored to the task at hand. By carefully selecting and evaluating these components, we aim not only to improve the accuracy and effectiveness of the proposed framework but also to gain deeper insight into the potential of distributed lexical representations for action prediction in images.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors, Professor Rob Gaizauskas and Dr Josiah Wang, for their unwavering guidance, support, and encouragement throughout my PhD journey. Their insightful feedback and constructive criticism have been invaluable in shaping my research and academic growth. I am particularly grateful of Professor Gaizauskas' exceptional understanding and personal support, which extends beyond the academic realm and has been instrumental in helping me overcome numerous challenges throughout my academic journey. Additionally, I am grateful for the support and guidance provided by Dr Wang, particularly during the initial phase of my research, and for his continued supervision despite his departure from the University of Sheffield during my first years.

I also would like to express my gratitude to my examiners, Professor Frank Keller and Dr Mark Stevenson, for their dedicated time, insightful feedback, and valuable suggestions during the examination process.

Furthermore, I wish to extend my gratitude for the support I have been fortunate to receive from individuals in Sheffield. Their expert guidance and mentorship have played an indispensable role in helping me reach this significant milestone.

I would like to extend my gratitude to the Saudi Arabia Cultural Bureau in London and King Saud University for awarding me the scholarship that made it possible for me to pursue my research and academic studies. Their generous support has been a significant contributor to my success, and I am truly grateful for their assistance. In particular, I am especially thankful to the Saudi Arabia Cultural Bureau for their financial support in obtaining the necessary medical treatment during my studies. Their compassionate approach and support helped me to navigate the challenges and stay focused on my research.

Finally, I am grateful to Professors Hmood Al-Dossari and Sultan Alyahya from King Saud University for their guidance, assistance, and encouragement during my application process for the doctoral scholarship.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Context. The field of computer vision has experienced significant advances in recent years, with remarkable improvements in various benchmark tasks. For example, the top1 accuracy on the ImageNet classification task has increased from 54.3% (Sánchez and Perronnin, 2011) to 88.5% (Zhang et al., 2023). Additionally, the box average precision (box AP) score on the MSCOCO object detection task has also risen from 19.7% (Girshick, 2015) to 60.9% (Chen et al., 2023).

Despite this rapid progress, it is evident that significant challenges still persist, especially in more abstract tasks that involve understanding social constructs and context. One such challenging task is the ability to describe actions in images from various perspectives without being limited by a small set of verbs. Verbs are naturally more abstract, making them harder to learn even for humans, which is why children acquire them later in their development (Hirsh-Pasek and Golinkoff, 2010). Furthermore, verbs are processed in different regions in the brain due to their abstraction level, suggesting that they require a different processing mechanism (Moseley and Pulvermüller, 2014).

Problem statement. Most studies in computer vision assume that each image action can be grounded to one particular verb. However, this thesis argues that assigning one verb to an action-depicting image is insufficient. The meaning of an image action is multifaceted and although one verb might be sufficient for describing the meaning of one aspect, it cannot do so for all aspects. Figure 1.1 illustrates that even with a simple action, one verb is not enough to capture all its aspects.

There are at least three situations where multiple verbs are required for describing a single action in the image. First, actions can possess multiple layers of significance, with one layer focusing on the practical mechanics of executing the action, and the other emphasising the motive or intention behind it. As an example, consider a hiker in a mountain range. The action of *hiking* refers to the physical aspect, while the underlying purpose is to *explore* nature. To fully capture both the physical aspect and the purpose of the action, different verbs may be necessary. However, when describing the purposive aspect of the action it is vital to comprehend the context in which the action takes place. For instance, the act of holding a sword can have different connotations; it could mean to *battle*, to *kill*, to *execute*, or even to *dance* and *celebrate*.

The second situation is when describing multiple actions carried out by the same agent. People can perform different, yet highly correlated, actions simultaneously (e.g., standing while talking, and sitting while reading). One might say this can also be formulated as one action specifying the physical state of the agent (e.g., *sitting*) while doing the other action (e.g., *reading*). The third situation is when the choice of action description depends on the type of the target audience. If the target audience prefers a description in formal language, the system in this case should output converse instead of chat. The same goes for those educated in a specific field (e.g., using the verb *macerate* or should be appreciated by people who are into cooking but not by others).

Our observations suggest that relying exclusively on supervised methods that use visual data is insufficient for effectively addressing the task. This is because interpreting image actions is a complex process that requires more information than that provided by the images used for training. To address the verb prediction task effectively, prior social and common knowledge is required.

Additionally, using conventional supervised techniques to approach this problem leads to an uncontrollable increase in the number of classes. For example, a simple classifier for an action like *hold* would not suffice; rather, the model would have to distinguish between various actions like "*hold sword to fight*" and "*hold sword to dance*" considering them as separate classes. As a result, gathering and annotating adequate training images for an extensive set of verb combinations is practically unfeasible.

Proposed solution. To address these challenges, we propose the use of lexical embeddings to improve the capability of systems automatically generating descriptions of actions in images. These embeddings act as a means to exploit knowledge present in the textual modality. The main objectives of this thesis are to assess the effectiveness of incorporating these embeddings and evaluate the feasibility of this solution. Our hypothesis is that a blended approach, combining both visual perception and knowledge extracted from text, will offer advantages over those

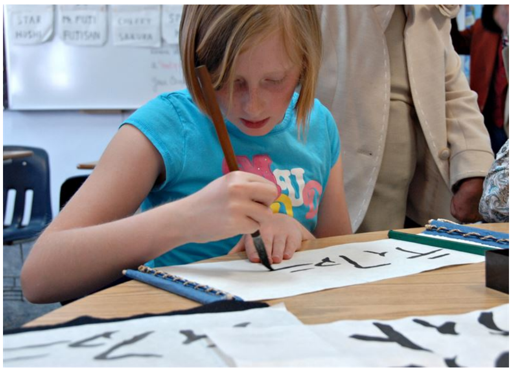


Figure 1.1: This example [1] shows the impracticality of describing (labelling) the action in the image with a single label, as the following verbs can be equally accurately used to describe the action: *write*, *learn*, *practise*, *sit*, and *hold*.

that rely solely on visual perception.

Specifically, our thesis proposes a framework that should generate more detailed and precise descriptions of actions by leveraging both visual cues and a wider range of verb labels. In this thesis, the framework is designed to generate a single label for each image; however, the framework is intentionally designed to be capable of generating multiple labels for each image.

Scope. The primary focus of this thesis is on human actions as they possess multi-layered significances. Studying human actions presents an excellent opportunity to examine the extent to which images require the linking of multiple verbs. The scope of this thesis is limited to images displaying a central or primary action performed by an individual or a group. Hence, the aim of this thesis is not to recognise multiple distinct actions within a single image.

This thesis focuses on analysing static images despite their limitations for action prediction compared to videos. Actions involving motion, such as rubbing, are harder to predict using images. However, this choice is justified by the technical complexities and extensive computational resources required for video processing—a hurdle that proved difficult to overcome during the initial stages of this research. Additionally, the scarcity of video datasets annotated with verbs added an additional barrier for working on videos.

[1] Image source: `https://media.defense.gov/2008/Apr/10/2000634626/-1/-1/0/080405-F-9433M-087.JPG`

1.2 Challenges in Utilising Distributed Lexical Representations

Leveraging distributed representations for verb prediction offers exciting possibilities, but it also presents several challenges that require thoughtful attention. One critical challenge is selecting a representative subset of verbs for model training. This entails exploring various strategies for identifying the appropriate set of training verbs and understanding how their characteristics may influence the model's generalisation ability. In addition to selecting the essential verbs for model training, it is equally crucial to identify verbs that can be predicted with a reasonable accuracy without prior training. These unsupervised verbs, which are not included in the training dataset, can benefit from the model's generalisation capacity during testing. Therefore, careful selection of such unsupervised verbs is crucial to harness the full penitential of our proposal.

Another challenge is the positioning of verbs in the lexical space. The lexical embedding space is not specifically designed for verbs or image-based tasks, making accurate navigation and similarity measurement between verbs difficult. A significant issue is that verb embeddings tend to be more proximal to one another than to other word embeddings, leading to potential confusion and imprecise distinctions.

Finally, the limited descriptive power of the training text presents a challenge. Although the lexical embeddings under investigation are trained on a vast amount of text, much of it derives from genres that are not well-suited to depicting visual content, such as Wikipedia and news articles. This limitation implies that the model may not possess adequate descriptive power to predict verbs accurately.

1.3 Research Questions and Thesis Contribution

This thesis aims to investigate the effectiveness of distributed lexical embeddings for predicting verbs that comprehensively describe actions in images, with a focus on challenging verbs that are difficult to learn using supervised methods. To achieve this objective, the following research questions will be addressed:

- To what extent can distributed lexical embeddings capture the essential information required to describe actions, especially for abstract verbs that lack direct or concrete visual cues?

- How does the use of distributed lexical embeddings improve the generalisation of verb prediction models to unobserved verbs?

In addressing these questions, the thesis makes the following contributions:

(i) It introduces the I2A framework as a tool for describing various perspectives on actions within images. Previous research has incorporated distributed representations for predicting image content, but this approach has not been applied to the task of describing different interpretations of image actions. Additionally, to our best knowledge, no research has investigated modifying the framework components to optimise them for this purpose.

(ii) It investigates the performance of verb prediction using off-the-shelf verb embeddings in both supervised and zero-shot settings. A variety of zero-shot experiments are conducted to identify which verbs are more suitable for training and which are more appropriate for testing purposes.

(iii) It examines how varying verb embeddings impacts the task of predicting verbs, with the goal of gaining a deeper understanding of the specific characteristics of embeddings that influence verb prediction.

(iv) It explores how different image representations affect the task of predicting verbs. The goal is to gain a deeper understanding of which input features are crucial for accurate verb prediction and to assess the upper-bound performance by incorporating an oracle in the input feature extraction process. This upper-bound re-evaluation is intended to determine if the framework's performance could be further enhanced, especially in the zero-shot setting, by using more advanced image feature extraction techniques.

1.4 Thesis Outline

This thesis presents an approach for addressing the verb prediction task that is capable of generating multiple verbs for the purpose of describing an action depicted in an image, without the need for training each individual verb in a supervised fashion. This introductory chapter has provided an overview of the motivations and challenges that the work in this thesis aims to address. The rest of the thesis is structured as follows:

Chapter 2 provides an overview of theories and studies from various disciplines that demonstrate the importance of describing actions from different perspectives. This chapter also reviews various tasks that involve the direct or indirect prediction of verbs. Additionally, it reviews different methods for representing target labels as lexical embeddings and presents the primary methods for addressing zero-shot learning.

Chapter 3 provides a formal definition of the task, introduces the proposed I2A framework, and discusses its main components. It also introduces the dataset used in this thesis.

Chapter 4 presents initial experiments with the I2A framework using its default components. **Section 4.1** evaluates the feasibility of the framework through experiments in a supervised setting, comparing its performance to established baselines. **Section 4.2** investigates the performance of the framework in the zero-shot setting, showcasing its use of distributed lexical embeddings, and evaluating its performance under different conditions.

Chapter 5 introduces techniques for enhancing the representation of target embeddings, making them particularly suitable for the verb prediction task.

Chapter 6 presents various experiments to assess the effect of varying the input representation. **Section 6.1** evaluates the characteristics of effective visual feature extractors through experiments with multiple extractors. **Section 6.2** introduces a second approach for representing input images by encoding semantic concepts depicted in the images. This section also conducts an extensive investigation of the original annotation of image content in the imSitu dataset.

Chapter 7 summaries the research conducted in the previous chapters and explores potential future research directions that can build upon the findings and contributions of this work.

Chapter 2

Literature Review

With this literature review, we aim to provide a comprehensive overview of the theories, approaches, and techniques that inform our understanding of the multifaceted nature of actions and their application in the verb prediction task.

Verbs play a central role in describing actions in images, as they capture the essence of an action or event. However, the interpretation of verbs can be complex and multifaceted, shaped by the context in which they are used, and cultural factors. To fully grasp the rich and nuanced meanings of actions, Section 2.1 will provide theoretical justifications for our verb prediction approach and introduce the theories and perspectives from multiple disciplines that support the claim that actions can have multifaceted interpretations. Section 2.2 will examine various approaches that have been developed for action-related tasks, which also represent verbs as targets or parts of the targets.

Representing verbs as targets is a challenging task that requires taking into account their complex and nuanced meanings. Section 2.3 will review different approaches for representing verbs as targets, focusing mainly on distributed-based representation, where the meaning of the verbs is expressed through multiple variables. This section will also explore how these approaches can be utilised to capture the various interpretations of actions.

Finally, Section 2.4, will provide an overview of the zero-shot learning setup and discuss its main techniques. Zero-shot learning aims to predict targets of classes that have not been encountered during training. The section will discuss how this setup can be utilised to extend action interpretations to the ones the are not supported in existing action datasets.

With this preamble, we aim to provide a comprehensive overview of the theories, approaches, and techniques that inform our understanding of the multifaceted nature of actions and how this can be applied in the verb prediction task.

2.1 Understanding Actions in Images

This section aims to justify the need to label actions in images with more than one verb, drawing on supporting evidence from various scientific disciplines such as linguistics, philosophy, psychology, etc. The section does not intend to provide an exhaustive overview of this complex topic, but rather provides a justification for the argument that using one verb as an action label may not be sufficient.

2.1.1 Action Identification

Vallacher and Wegner (1987) discuss how people identify their own actions and the factors that impact action identification, a theory known as *action identification theory*. They suggest that people can recognise and view an action either with a low-level identity, specifying how the action is performed, or a high-level identity, signifying why or with what effect the action is performed. They point out that identity is a relative concept (e.g., “taking a test” is high level compared to “answering questions” but also low level compared to “showing one’s knowledge”). They also state that people tend to choose the identity that provides the most comprehensive understanding of what they are doing, indicating a preference for higher-level identity.

Mange et al. (2015) hypothesise that low- and high-level identities of actions have different social values, with higher-level identities having greater value due to their social utilities. While the two studies aim to understand how humans identify their actions and what factors affect their perception, we can apply this theory to understand how humans identify actions in images. This is supported by studies showing that humans can accurately infer high-level identities of actions performed by others (Baker et al., 2009; Saxe and Kanwisher, 2003). Figure 2.1 displays sample images depicting actions that can be identified at varying levels.

2.1.2 Highlighting Different Meanings in Images

In his study of Renaissance art, Panofsky (1955) identified three levels of meaning: (1) primary meaning, which requires practical experience of objects; (2) secondary meaning, which requires requires prior familiarity with specific themes and concepts; and (3) intrinsic meaning, which requires knowledge about the artist and their social background. To explain his classification system, Panofsky uses the example of a person lifting their hat as a greeting gesture. The primary meaning only addresses the literal meaning (the fact of lifting a hat), whereas the secondary meaning addresses the symbolic meaning (lifting a hat stands for a greeting). The intrinsic meaning, on the other hand, focuses on why the artist depicted the scene in this way and what can be learned about their personality and background.

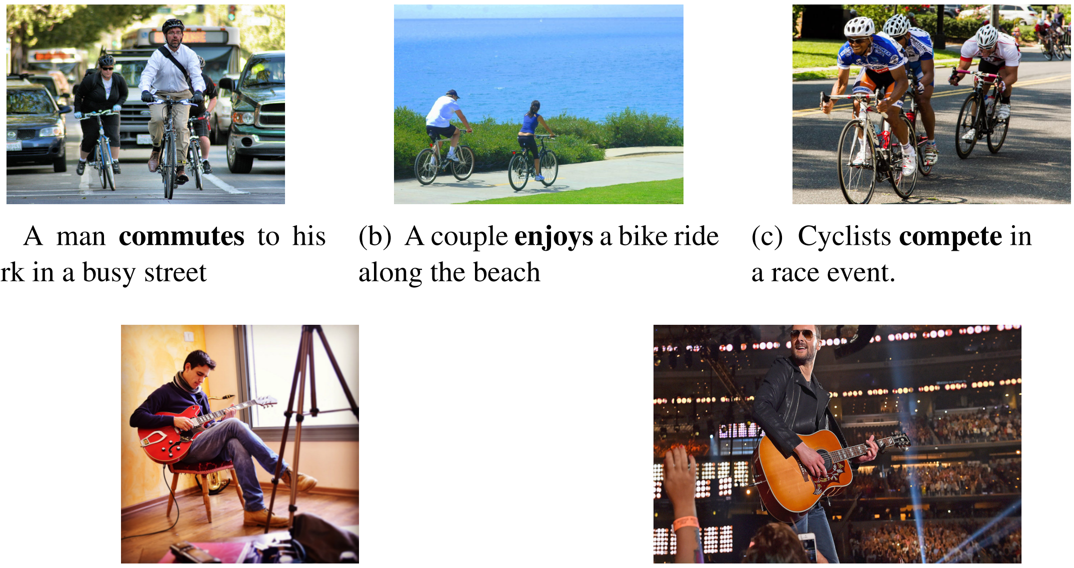
Current vision systems have achieved some success in recognising the literal meanings of images by learning from large amounts of data. However, they fall short in recognising the emotional and cultural associations of images, which require a deeper understanding beyond just visual information. This level of understanding requires access to cultural and literary context, which these systems lack as they are unable to incorporate knowledge from other sources, such as text.

Panofsky's theory has become widely accepted not only in art but also in other fields that require a thorough understanding of images. Shatford (1986) extended the theory with the goal of providing a framework for indexing and retrieving pictorial materials. She identifies four facets, namely *who*, *what*, *where*, and *when*, through which the three levels of Panofsky's meanings could be viewed. The *what* facet of her framework included terms that described events, actions, conditions, and emotions, answering questions such as: "What are the creatures or objects in this picture doing?", "What is their condition or state of being?", "What emotions are conveyed by these actions or conditions?", "What abstract ideas do these actions or conditions symbolise?" (Shatford, 1986, p. 52). To illustrate her point, Shatford provided an example using the Pieta picture, where *death* could be considered a generic aspect, and *sorrow* an emotional aspect.

The key takeaway from this section is to appreciate the fact that there can be several valid ways to label an image during the indexing phase, as we cannot anticipate how someone might characterise the image during the retrieval process. Therefore, it is essential to remain flexible in our approach to image labelling, acknowledging that multiple characterisations may be appropriate.

2.1.3 Expressing Thematic Roles

The image could either depict the person performing the action or the person receiving it. Therefore, multiple verbs may be required to convey both the action and the participants' roles. For instance, consider two images that can be accurately labelled with the verb *interrogate*. However, one image shows a person conducting the interrogation, while the other depicts a person undergoing it. In this case, it may be necessary to use more than just the verb *interrogate* to effectively emphasise the participant's role.



(a) A man **commutes** to his work in a busy street

(b) A couple **enjoys** a bike ride along the beach

(c) Cyclists **compete** in a race event.

(d) Practices

(e) Performs on stage.

Figure 2.1: Actions with multifaceted interpretations. Two groups of images (top and bottom) are shown. Both groups have a shared primary meaning but are distinct in secondary meaning. The top group can be labelled with verbs such as *riding* and *biking*, but the context of the images is not fully captured. Similarly, the bottom group shares the common activity of playing guitar, but *playing* fails to express the nuanced context portrayed in the images.

2.1.4 Summary

This section highlights the complexity of describing images, as the meaning of an image exists on multiple levels, each requiring different verbs to express them. To address this issue, two approaches can be considered. The first approach involves using a set of complementary verbs to label the image, which helps to provide a more complete and accurate description. As example of this approach would be using verbs like *interrogating* and *undergoing* to describe an image. The second approach involves composing a larger lexical unit by combining smaller units, which can be either verbs (including light verbs) or derived from verbs (including normalised verbs). An example of this approach would be “*undergoing interrogation*”. The thesis acknowledges the advantages of having a flexible system that can incorporate both approaches simultaneously, but primarily focuses on the first approach of using a set of complementary verbs to describe an image.

2.2 Verb Prediction

In this section, we will review the verb prediction task, emphasising the significance of the output produced and its impact on both task design and dataset creation. Our focus will be on labelled datasets for both images and videos, where the labels, specifically the verbs, serve as the primary target outputs for researchers in the field of verb prediction.

Initially, the verb prediction task was tackled as a standard classification problem, where the classes were mutually exclusive verbs that were easily distinguishable from each other, such as *eat*, *sit*, and *ride* (Ikizler et al., 2008). However, this approach faced the challenge of verbs lacking sufficient context, making them highly ambiguous. For example, the verb *play* could refer to playing a football game or playing an instrument, while the verb *paint* could refer to painting a wall or a piece of art. To address this issue, the candidate classes were limited to those that were specific enough on their own, such as *drink* or *run* (Sigurdsson et al., 2017), resulting in the elimination of ambiguous verbs such as polysemous verbs or light verbs like *take* and *put*, which require additional information (e.g. nouns and prepositions) to disambiguate them (e.g. *take medication*, *take shoes*, *take off shoes*).

This section further explores the verb prediction literature through the lens of target label specificity. The literature is categorised into four approaches based on how the output of the task is defined and how concreteness is achieved for it.

2.2.1 Verb-Object Representation

The first approach to creating concrete descriptions of actions involved pairing verbs with objects as in the form of verb-object pairs or subject-verb-object triplets. This approach emphasises the understanding of the relationship between verbs and their context, such as participating objects or location. Sadeghi and Farhadi (2011) introduced the visual phrases technique, which combined verbs and objects into single units (e.g. person-riding-bicycle). This technique aimed to simplify visual complexity by considering how objects' appearance can change based on the actions they participate in, leading to individual classes with distinctive visual features due to their high inter-similarities. This approach was influenced by Marszalek et al. (2009), who incorporated scene or location information (e.g. "*eating in kitchen*" or "*eating in cafe*"). These earlier datasets and others, such as the Sports Dataset (Li and Fei-Fei, 2007) and Stanford-40 (Yao et al., 2011), are restricted in both the quantity of images and number of classes. In the following we discuss a number of more recent and extensive datasets whose labels have served as targets for researchers working on verb predictions.

The "Humans Interacting with Common Objects" (HICO) dataset, as presented by Chao et al. (2015), consists of 47K images covering 600 distinct categories of human-object interactions. The focus of these interactions is exclusively on human subjects. One of the main objectives for creating HICO was to provide a diverse set of actions for each object, averaging 6.5 interactions per object. For example, the dataset covers various actions for bicycle, such as *ride*, *hold*, *sit on*, *walk*, and *repair*. This objective, therefore, made object categories (80 from MSCOCO (Lin et al., 2014)) the starting point in building HICO.

Yatskar et al. (2016) presented the situation recognition dataset (imSitu), which contains structured annotations of actions occurring in 126K images. Each image is annotated with a verb describing the action, the objects involved, and their respective roles in the action. This dataset provides insight into the relationships between objects and their roles in actions, and it enables machine learning models to generate comprehensive and meaningful descriptions of images.

However, it also increases output complexity and the scope of possible outputs, as imSitu has a large number of unique combinations, totalling 200K distinct situations, compared to just 600 in the case of HICO. The importance of verb categories in imSitu is emphasised through their role in both the image selection and annotation processes.

The authors initially considered almost 10K candidate verbs, but narrowed down the list to 1K verbs by eliminating non-visual verbs and those requiring technical background or advanced literacy. This list was further reduced to 504 due to the unavailability of suitable example images. The final set of verbs includes near-synonyms, making it difficult to differentiate between them and treat them as separate concepts, such as PHONING and TELEPHONING. Additionally, some verbs may also prove difficult to distinguish given the static nature of images, such as LAUGHING and GIGGLING.

On the other hand, some verb pairs exhibit nuanced but important differences, such as BIKING and RIDING (where biking is a type of riding), DIALLING and PHONING (where dialling is a sub-activity of phoning), and PACKAGING and PACKING. These subtle distinctions are of significance in the context of this thesis and make the imSitu dataset a promising resource for experimentation.

A key drawback of this approach is that the number of classes, i.e., the combinations of verbs and objects, increases at a rate higher than the increase in the number of objects. This can result in an inability to gather sufficient training data for each distinct class. For instance, if the task is to generate pairs of verbs and objects and the model is trained to detect 10 verbs and 50 objects, the number of possible combinations would be up to $10 \times 50 = 500$. However, research has shown that the number of classes that can occur in real-life scenarios is much smaller than the number of possible combinations (e.g., a person cannot sit on a pencil). This insight has led to various proposals to address this challenge. For example, Lu et al. (2016) suggests using written language as a prior, by utilising word vectors that are pretrained on large text data, which are then finetuned on the image dataset.

Ramanathan et al. (2015) proposed an extensive set of action descriptions, comprising subject-verb-object combinations, to tackle the issue of data scarcity. Their work aimed to understand the inter-relationships among actions as a means of generalising to unseen classes. For example, if there were few or no training images for the action "*person interacting with panda*", the model could still learn it by leveraging its relationship with other actions with more training data such as "*person feeding panda*" and "*person holding animal*". The study explored three types of relationships: implied-by, type-of, and mutual-exclusive. The model is expected to learn these relationships through supervised training on a subset of "action descriptions" and output the relationship type between two or three images. The study also incorporated a language prior module that utilised the WordNet hierarchy, which exhibits the three relationships. The authors constructed an extensive set of 27K unique action descriptions, but did not provide a full list or in-depth discussion on how the they were developed. The action descriptions were then used to construct a weakly-labelled image dataset through searching Google Images and collecting the top results, followed by a filtering step to remove noisy images.

ActivityNet, introduced by Heilbron et al. (2015), is a video dataset that features 203 human activities chosen from a four-level taxonomy designed by the Department of Labour for the American Time Use Survey (ATUS) [2]. The taxonomy covers over 2K activities, organised based on social interactions and activity location (Figure 2.2). Within the ATUS taxonomy, intermediate nodes are categories or domains, while leaf nodes are the specific actions. It is important to note that the taxonomy does not offer a hierarchical representation of verbs and their connections, but rather groups activities into distinct domains or areas of application. To ensure the inclusion of a diverse range of actions, the ActivityNet activities were manually selected from 7 out of the 18 top-level categories in the original taxonomy.

2.2.2 Free-Form Representation

This section reviews datasets that annotate actions through indirect methods, where verbs are generated as a byproduct. The focus is on two captioning datasets, MSCOCO (Lin et al., 2014) and Flickr30K (Young et al., 2014). The purpose of this review is to observe the way humans describe actions when limited restrictions are placed on the nature of the description and to study the relationship between verbs that describe a particular image.

Each image in MSCOCO and Flickr30K is annotated with five captions written by five human annotators. The captions are sentences that describe the visual content, with a focus on salient aspects of the image (i.e., describing people and their interactions with each other or the environment). It's important to emphasise that the captions are not meant to supplement images with additional information but rather to provide descriptions from individuals who are

[2] Source: `https://www.bls.gov/tus/lexicons/lexiconwex2021.pdf`

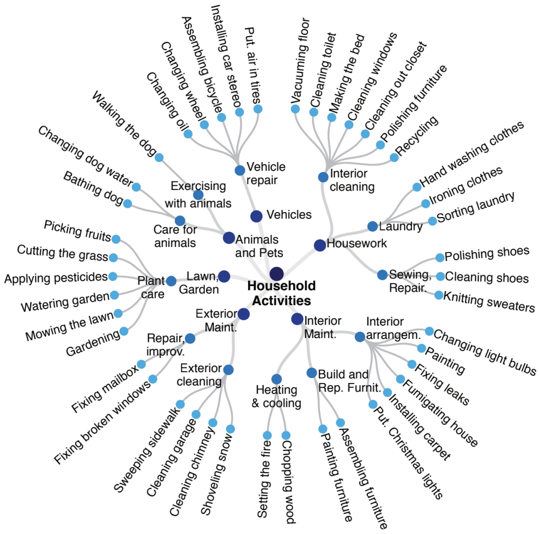


Figure 2.2: Visualisation of the sub-tree of the top level category *household activities* by Heilbron et al. (2015).

unfamiliar with the specific context of the images (Karpathy, 2016). Examples of captions from these datasets include:

- *Ballet dancers in a studio practice jumping with wonderful form.*
- *Four men cutting into a cake with Malaysia written on it.*
- *A deep-sea diver is in full equipment studying a sea turtle.*

These datasets offer a valuable insight into how individuals describe actions when given more creative freedom, without having to select verbs from a pre-defined list. This can result in a wider range of verbs being used, compared to action datasets that exclude abstract, specific, technical, or high-literacy verbs. We contend that a comprehensive description of actions requires not only a larger but also a more diverse set of verbs. A balanced mix of verbs that encompass different aspects of action description, such as concreteness, abstractness, reason, manner, etc., is crucial.

We conducted an analysis on captions of both datasets and found that our observations were relatively consistent across both; therefore, this section will specifically focus on one dataset: Flickr30K. To identify all verb instances in the captions, we employed the part-of-speech (POS) tagger from the spaCy (Honnibal and Montani, 2017) large language model (en_core_web_lg). The Flickr30K dataset was found to have a total of $2,239$ verbs, with an average of 5 unique verbs per image and 1.5 verbs per caption.

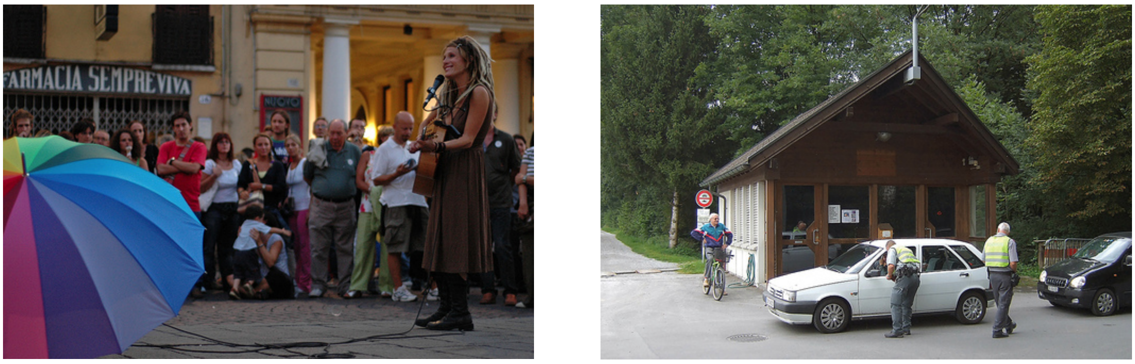
Our analysis revealed that the verb (lemma) distribution is highly skewed, with certain verbs (e.g., *wear*, *sit*, *stand*, *play*, and *walk*) occurring over 10K times, while $1,656$ occur fewer than 20 times. This imbalance raises doubts about the usefulness of most of these verbs, as the limited number of examples may lead to inadequate representation. Despite this, the $1,656$ verbs could be useful for evaluating zero-shot models, for instance.

This observation is also supported by the work of Alikhani and Stone (2019), reflecting the constraints imposed during annotation and dataset creation (i.e, how the content of images hugely influences how captions are written). Their study show that verbs are significantly underrepresented in image captioning corpora compared to the American National Corpus (ANC), a balanced corpus of spoken and written English (Ide and Suderman, 2004). In the ANC, 18.4% of the tokens have verb tags compared to only 2.6% in MSCOCO and 1.2% in Flickr30K. Furthermore, the frequency of verbs decreases sharply in image captioning corpora, especially across the most frequent 100 verbs, indicating less diverse use of verbs. For instance, the top 10 most frequent verbs in MSCOCO already account for $\%60$ of all verbal instances, and the top 30 and top 100 account for 72% and 86% respectively. In the ANC, top 10, top 30, and top 100 account for 44%, 56% and 71%. The study also shows that verbs in image captioning corpora are either stative (describing conditions rather than activities) or atelic (describing an ongoing process and in a general way without invoking the possible goal). Based on a sample of 500 captions from each dataset, only $\approx 5.6\%$ of MSCOCO and Flickr30K contain telic descriptions, compared to 58.8% for the ANC.

In our examination, we observed that the process of extracting verbs from captions is generally simple, except for verbs labelled as past participles. Past participles can be used to describe either object states (e.g., "... *a snow-covered mountain*") or actions "*A couple getting married*"). However, after consideration, we chose not to address this issue as instances of past participles only represent 7.5% of all verbal occurrences. The following provides more examples of past participle instances describing actions:

- *A young female student performing a downward kick to break a board held by her Karate instructor.*
- *An African American child is supported by his three peers while riding his bicycle.*
- *A little baby cradled in someones arms.*

Additionally, we carried out a manual examination on a subgroup of images with a higher verb count, aiming to analyse the features of these images and assess the impact of verbs in enhancing the descriptive nature of captions. Out of the 30K images, only 8611 had a minimum of six unique verbs across the set of five associated captions, while only 1945 had a minimum of eight unique verbs. Our analysis revealed that the abundance of verbs in the captions was a result of two primary factors: (1) the captions depicted various actions (as illustrated in Figure 2.3), and (2) the captions provided various perspectives or interpretations of the actions (as illustrated in Figure 2.4). It was noted that the majority of cases were primarily due to



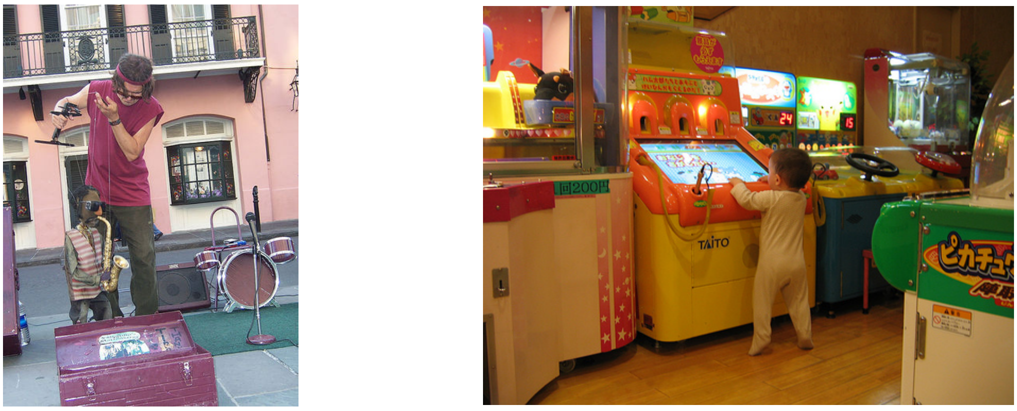
1. A group of people, of all ages, listening to a concert being performed by a solo practitioner.	1. Two men in florescent vests are standing next to parked cars in front of a small building while one of them converses with a driver and a woman on a bike is seen riding by.
2. A woman playing a guitar in front of a group of people standing outside.	2. A man leans into a car to talk to the driver, as a man on a bicycle looks on.
3. A crowd gathers around a lady with an acoustic guitar who is performing.	3. A man on a bicycle watching a row of cars waiting to go through a checkpoint.
4. A woman , playing music , smiles as a crowd surrounds her.	4. A park ranger talking to a tourist.
5. A woman performing with a guitar on a crowded street.	5. Two cars are parked outside.

Figure 2.3: Two images from Flickr30K annotated with a large set of verbs, each showing multiple actions.

the first factor, as it proved challenging to locate sufficient examples that illustrate the second factor. This finding is key in informing the decision to not pursue experimentation on captioning datasets. Section 3.3 will delve deeper into the discussion of the dataset chosen for experimentation in this thesis.

2.2.3 Sense-Based Verb Representation

This approach focuses on predicting the correct sense of a verb, instead of just the base form of the correct verb. Gella et al. (2016, 2019) proposed the task of Visual Sense Disambiguation (VSD) for verbs, which is inspired by the Word Sense Disambiguation (WSD) task. In VSD, the input is a pair of image and verb, and the output is the correct sense of the verb. The authors provided an example to demonstrate the importance of this task: three images labelled with the word *play* can have different meanings such as an athlete playing in a match, a guitarist playing music, and children playing in a sandbox.



1. Puppeteer wearing a purple sleeveless shirt showing off a puppet that looks like a black man with shades playing a saxophone.	1. A toddler is staring into the screen of a brightly colored orange and yellow video game in what appears to be a game center.
2. A man dressed in a burgundy shirt and black pants maneuvering a puppet which is holding a musical instrument.	2. A toddler wearing a onesie is stretching to see a video game in an arcade.
3. Puppeteer entertaining with a puppet playing the saxophone on the side of the street.	3. A toddler in a white onesie is playing a video game at an arcade.
4. A man wearing a red shirt has a puppet band, on the street.	4. A toddler takes his first glimpse of the wonders of video games.
5. A man manipulates a saxophone playing marionette.	5. A baby in an arcade tiptoeing to see the screen of a video game.

Figure 2.4: Two images from Flickr30K annotated with a large set of verbs, each showing a central action with multiple interpretations.

The authors proposed unsupervised techniques to mitigate the lack of datasets that label image actions with verb senses, and developed a small-scale dataset (3.5K images) for evaluation purposes. Creating a large-scale image dataset for verb senses presents a challenge as there are limited linguistic resources available to organise verb senses in a hierarchical manner that would be useful for building the image dataset (Barnard and Johnson, 2005; Chen et al., 2015). This task formulation influenced the selection of verb senses, leading to a focus on polysemous verbs.

HICO, previously discussed in Section 2.2.1, allocates some attention to verb senses for the purpose of simplifying the set of verbs considered when building the dataset. To define verb categories, HICO extracts a candidate set from MSCOCO captions using linguistic tools such as a dependency parser and the Google N-Gram dataset. Verbs with similar meanings are manually grouped into categories (e.g. *repair* and *fix* in the context of bicycles) and then linked to WordNet senses, i.e. synsets. For instance, out of many verbs used to describe bicycle-related actions, the following are grouped as one verb: *repair*, *mend*, *fix*, *bushel*, *doctor*, *furbish up*, *restore*, and *touch on*. The purpose of grouping verbs is to simplify the action recognition task by separating language understanding from vision recognition. Therefore, WordNet synsets are merely used to facilitate the grouping of verbs, meaning that the verb sense labelling in HICO is not designed to address VSD or handle overlapping senses. The definition of verb and object categories is influenced by the MSCOCO dataset, leading to a similar dataset to MSCOCO since verb and object categories determine the selection of images.

2.2.4 Multiple-Verb Representation

Wray and Damen (2019) propose a multiple verb-only representation for the target output of the verb prediction task, which deals with the overlap of context among verbs by using an object-agnostic approach. They use two techniques for representing actions in videos, hard assignment and soft assignment. To motivate their approach, they provide an example that highlights the ambiguities that can arise from verb-object representations, such as the ability to open a door by pushing or pulling, or a bottle by turning and a package by cutting.

In the hard assignment method, the target is represented by a binary vector over the set of target verb classes. The indices of verbs that describe the action are activated (assigned a value of 1), while the rest of the indices are assigned a value of 0. In the soft assignment, each verb in the target vector is given a relevance value between 0 and 1, which is calculated from scores provided by 30 to 50 annotators for each example. These scores are then normalised. The motivation behind this approach is that even if multiple verbs can be used to label a video, they may not have equal validity or relatedness.

An examination of the annotators' agreement in the annotation data showed a tendency to agree on verbs that are either highly relevant or certainly irrelevant, but disagreement was found on a group referred to as secondary verbs. Although these verbs are not critical to describing the action, some annotators did not deem them irrelevant.

Their experiments showed that both the hard and soft assignment techniques outperform the verb-object representation as in Section 2.2.1. Hard assignment is more appropriate for verb prediction, while soft assignment is better suited for retrieval. However, the main drawback of the approach is its labour-intensive nature, as annotators need to evaluate the relevance of each verb instead of simply selecting the most applicable one as in the single-verb approach.

The HICO dataset, previously discussed in Sections 2.2.1, and 2.2.3, assigns multiple verbs to each image, treating all labels as equally important. The major challenge with HICO is the imbalanced distribution of verbs, where some verbs such as *sit* and *stand* occur much more frequently, whereas the majority of verbs have only a few occurrences. Despite attempts to rectify this through increased query iterations during image collection, the source of this imbalance may be traced back to HICO's reliance on MSCOCO for creating its categories of verbs and objects.

2.2.5 Summary

In this section, we discussed four main types of tasks that involve predicting verbs, both directly and indirectly. We focused on how these verbs are presented in the tasks and the types of resources that have been constructed for these tasks. Additionally, we explored the role of verbs in dataset construction approaches.

In this section, we discussed the verb-object representation approach, depends on information about the objects and scenes within an image to convey the meaning of the image action. However, this approach has limitations in capturing the multi-faceted nature of action meanings. It places the onus on the end-user, i.e., the human, to reason and derive additional interpretations of the action from the available information.

In addition to the verb-object representation approach, we also examined the free-form representation approach, which allows for more detailed descriptions of image actions and often includes multiple verbs. This approach enables us to analyse the co-occurrence of verbs within image descriptions. However, one significant challenge with the free-form representation approach is the highly uneven distribution of verbs within captions, which can limit the utility of captioning datasets for verb prediction tasks like the one presented in this thesis.

We also discussed the sense-based approach, which is highly specific to the sense disambiguation task and necessitates meticulous annotations of images. This approach has the potential to yield more precise information about the actions depicted in images. However, it can be time-consuming and require substantial resources to execute, especially when applied to a large-scale dataset. Additionally, we explored the multiple-verb representation, with a focus on Wray and Damen's (2019) soft-assignment method, where labels are assigned varying degrees of relevance.

2.3 Target-Label Representation

Earlier in Chapter 1, we established the premise that each image is assigned a single verb or a set of verbs that characterise the primary action depicted in the image. In Section 2.1, we highlighted the need for a diverse set of verbs when describing the primary action in the image, and Section 2.2 presented an overview of various tasks that approached verb prediction, either directly or indirectly, and analysed the output format of these tasks.

As our task requires the output to be represented in the form of one verb or a set of verbs, this section explores multiple vector-based approaches to verb representation, categorised into four approaches. Each of these categories will be discussed in detail in separate sections.

2.3.1 One-Hot Representation

The approach represents target labels as discrete symbols, each assigned a unique integer number, which can be based on alphabetical order or the order of observation. This integer representation can be transformed into a vector representation, where each word in the set maps to a distinct, sparse vector of L dimensions (where L is the number of distinct labels or classes). This representation, known as one-hot vector, assigns a value of 1 to the i-th dimension for label i and 0s in all other dimensions. One-hot vectors were previously widely utilised in machine learning and still in classification tasks that require mutually exclusive classes. However, in recent years, the use of this representation has declined with the advent of deep learning techniques, which have demonstrated superior ability in learning complex representations.

In this thesis, we view one-hot embeddings as insufficient due to the fact that the target labels (verbs) are not mutually exclusive and cannot be arranged as disjoint sets of classes. Furthermore, these embeddings lack the ability to measure similarity or relatedness without the use of supplementary resources like WordNet as in Resnik (1995). Our proposed framework emphasises the importance of capturing the relationships between verbs in the representation, which is crucial for expanding vocabulary and handling unseen verbs during training (as discussed later in Chapter 3).

2.3.2 Feature-Based Representation

A second method for representing verbs is through distributed representations using a set of hand-selected features or attributes. These provide valuable insight into human concept representation and have been applied successfully in NLP tasks, particularly as a proxy for modelling perceptual information (Silberer and Lapata, 2014; Silberer et al., 2017). To develop a feature norm dataset, individuals are asked to identify the most crucial attributes of a concept. For example, in the widely used McRae dataset (McRae et al., 2005), the concept of shrimp is associated with features such as being small, edible, and living in water.

Feature norm datasets can be transformed into vector representations, but the resulting vectors tend to be sparse, with each concept only having a limited number of features (e.g. McRae has an average of 13.7 features per concept). One of the main advantages of feature norm datasets is that they cover a range of properties, including visual-form and surface properties, rather than just taxonomic or encyclopedic ones. Table 2.1 shows the featural representation for moose.

However, these datasets still only provide a partial view of human conceptual knowledge and creating them is both expensive and labour-intensive, as extensive human annotations are required to determine the features associated with a given concept (Fagarasan et al., 2015). As a result, coverage is limited to a specific set of concepts and features. For instance, the McRae dataset only has 541 concepts described using 2526 features, while the largest in terms of coverage, the Cambridge Centre for Speech, Language, and Brain (Devereux et al., 2013), includes only 638 concepts. Most of these resources are designed only for noun concepts, with a few exceptions, such as Vinson and Vigliocco (2008), that include verbs in their studies. Additionally, the subjectivity of this method raises concerns as it is based solely on human annotators, which may introduce biases.

In the field of computer vision, feature-based representations are distinct from other feature-based representations in two ways. Firstly, they consist of binary features and secondly, the feature space is tailored to a specific task or dataset. For instance, features such as *white*, *eat fish*, *stripes*, and *water* might be relevant for animal recognition (Lampert et al., 2014), while *vegetation*, *vacationing*, and *natural light* may be suitable for scene recognition (Patterson et al., 2014). Object recognition (Farhadi et al., 2009; Ferrari and Zisserman, 2007) could benefit from features such as *horn*, *wing*, *door*, and *wood*, and action recognition (Liu et al., 2011) might use features like *indoor related*, *translation motion*, and *torso twist.*

Table 2.1: The featural representation for moose in the McRae dataset (2005)

Feature	Production Frequency	Brain Region Classification
is large	27	visual-form and surface
has antlers	23	visual-form and surface
has legs	14	visual-form and surface
has four legs	12	visual-form and surface
has fur	7	visual-form and surface
has hair	5	visual-form and surface
has hooves	5	visual-form and surface
is brown	10	visual-colour
hunted by people	17	function
eaten as meat	5	function
lives in woods	14	encyclopedic
lives in wilderness	8	encyclopedic
an animal	17	taxonomic
a mammal	9	taxonomic
an herbivore	8	taxonomic

Feature-based representations are inefficient due to their task-specific and labour-intensive nature. Also, research has shown that feature-based techniques are inadequate for object classification (Yu et al., 2013), which raises concerns about their performance for other tasks.

2.3.3 Distributional Representation

In the preceding section, we explored a method of obtaining a distributed representation through representing verbs as vectors of attribute or feature values. This section introduces a second method, which is founded on the distributional hypothesis and employs patterns of verb usage in large text corpora to construct the representation. The distributional hypothesis postulates that words that appear in similar contexts tend to have similar meanings (Firth, 1957). This idea has greatly influenced the development of distributional vectors and offers practical approaches for constructing representations from vast text resources (Deerwester et al., 1990; Bengio et al., 2003; Collobert and Weston, 2008).

One key aspect of these algorithms is that they do not require any kind of feature engineering as in the featural representation.

The earliest approach for creating distributional representations is the count-based approach, which entails constructing a frequency matrix and subsequently undergoing linguistic and mathematical processing. Although there are numerous variations and applications of this frequency matrix, the one that has had the most direct impact on current distributional models is the Hyperspace Analogue to Language (HAL) proposed by Lund and Burgess (1996). HAL innovated the use of a word-word frequency matrix instead of a word-document matrix and the notion of a fixed-sized context window, rather than the entire document.

The second method for constructing distributional representations involves using a shallow neural network-based approach. This method is efficient and enables simple updates to the embeddings. It gained widespread recognition with the work of Mikolov et al. (2013), who introduced the word2vec library, which learns representations by predicting context words given a target word (SkipGram) or by predicting the target word based on its surrounding context words (CBOW). Another influential contribution is the Global Vectors (GloVe) proposed by Pennington et al. (2014), which combines the global statistics of count-based techniques like LSA with the local context-based learning present in word2vec.

Measuring similarity. The main advantage of using distributional embeddings for verb representation is the ability to quantify the similarity or relationship between verbs. Attributional similarity between pairs, such as (*drive*, *commute*) and (*yell*, *argue*), can be directly measured using cosine similarity. However, it is important to note that distributional similarity yields only one metric, making it challenging to differentiate between various forms of similarity, including hypernyms, antonyms, and synonyms, as well as other semantic associations, such as those between *dine* and *table cloth*.

One solution to overcome this vagueness is to measure relational similarity between pairs of verbs, which adds context to the type of similarity being sought. For instance, the troponomy relationship can be measured through the analogy *nibble* to *eat* is like *sip* to *drink*, which can be determined by vector arithmetic ($\overrightarrow{sip} - \overrightarrow{drink} + \overrightarrow{eat}$). The next step is to calculate the cosine similarity between the resulting vector and each embedding in the vocabulary, and select the closest match, which would be $\overrightarrow{nibble}$.

Another solution to the vagueness of distributional similarity is to compare similarity based on specific semantic features such as morality, social class, pleasantness, gender, wealth, and sentiment (Grand et al., 2022). These semantic features are extracted from seed embeddings and can be used to analyse the expressiveness of a wider range of embeddings. For example, (Bolukbasi et al., 2016) conducted a principal component analysis (PCA) on a matrix composed of very feminine and very masculine embeddings and used the resulting first component, which explains the most variation among the embeddings, as the gender attribute.

Limitations. Since distributional embeddings are trained on text resources, they inherit the limitations and biases that exist in text. Bias in text could stem form human biases, e.g., *gender bias* as demonstrated by Bolukbasi et al. (2016). The nature of human communication and the medium used for communication, text, can also introduce *reporting bias* in distributional embeddings as discussed by Gordon and Durme (2013), which can negatively impact the capability of describing human actions.

Another limitation of distributional embeddings is their inability to capture the full meaning of a concept due to their reliance on linguistic resources alone. Representing meaning solely through linguistic resources fails to account for the grounding of meaning in perception (Harnad, 1990). Humans acquire knowledge not just through text but also through physical experiences and sensori-motor experiences (Louwerse, 2011), and as a result, distributional embeddings may lack the ability to effectively encode perceptual meaning, particularly in comparison to taxonomic and functional features as discussed by Lucy and Gauthier (2017).

Despite these limitations, distributional embeddings have been widely adopted and utilised in a range of visual-based applications due to their great potential. It is important, however, to acknowledge these limitations and understand the potential biases and drawbacks inherent in distributional embeddings.

2.3.4 Contextual Representation

Contextual representation is a subcategory of distributional representation, but it is discussed separately in this section to emphasise that it is not the focus of the experiments in this thesis, which concentrate on static distributional embedding (discussed in the previous section).

Since the advent of contextual models, such as those introduced by (Peters et al., 2018; Devlin et al., 2019; Radford and Narasimhan, 2018), they have become an essential component in achieving state-of-the-art results in various NLP tasks. The key distinction between contextual and distributional representation lies in the former's ability to learn a robust language model, allowing for an end-to-end architecture that seamlessly integrates the input representation (e.g., words) and the prediction module.

Aside from their primary applications in downstream tasks, contextual models also hold significant value in terms of extracting specific representations, such as variations in the meanings of words based on word sense, genre, or historical context. These models produce an instance-based representation of words, where the representation of a word changes dynamically based on the context in which it appears. For instance, the word *play* has different representations in the sentences "*A kid plays with his toys in the backyard*" and "*A person plays the guitar in a room*".

Studies by Loureiro and Jorge (2019) and Loureiro et al. (2022) utilised contextual models to extract specific representations from a small sense-annotated corpus such as SemCor. They proposed a technique that extracts contextualised embeddings of each unique sense instance, then computes a prototype representation for the sense through average pooling or cluster centroid calculation.

Contextual models are also potentially valuable for extracting representations in the visual domain. We believe that, even with limited training data, contextual models can produce representations that are more relevant to visually-oriented applications, compared to the static embeddings of the previous section. The representations generated by these models are expected to encode key information about the visual scene, including properties of the verbs, information about the agents, objects, and actions related to the verb. One major challenge we faced when considering the use of contextual models is the limited availability of publicly accessible datasets that are suitable for the verb prediction task, such as audio descriptions (AD) for movies and television shows that are created for the visually impaired.

Despite the potential of contextual embeddings, this thesis exclusively concentrates on static distributional embeddings due to their suitability for the research objective, which aims to establish a global representation of verbs, as opposed to context-specific ones (i.e., contextual representations). Furthermore, exploring static embeddings offers valuable insights, as elaborated in Chapter 5. Static embeddings also demonstrate superior performance compared to contextual embeddings across various out-of-context semantic tasks and datasets (Lenci et al., 2022).

2.4 Zero-Shot Learning

Zero-shot learning is a learning setup in machine learning where the model is tested on classes it has never seen during training. The goal is to predict the class of the new instances (or samples) based on auxiliary information that describes the distinguishable features of classes. That is, the auxiliary links the seen and unseen classes by describing the distinguishable features of both types of classes. For instance, in image classification, a model that was trained on grizzly bears but has never seen a panda can still identify a panda if it has access to the information that pandas are black-and-white bears. The problem of zero-shot learning is well-researched across various disciplines, particularly in computer vision, where object classification is a key area of study.

The use of zero-shot techniques is motivated by the need for a mechanism to handle highly descriptive verbs, which contain a rich amount of information about actions. These verbs present a considerable challenge to conventional supervised learning algorithms due to their limited range of application and rare occurrence in corpora and image datasets. This is due to the phenomenon that the greater the degree of descriptiveness, the narrower the scope of application for a verb (Snell-Hornby, 1983). This is exemplified by the following ranking of verbs based on the degree of descriptiveness provided by Boas (2008):

$$walk < jog < parade < stagger$$

There are two broad categories of zero-shot learning techniques that are relevant to vision-based tasks, namely attribute-based methods and embedding-based methods. Attribute-based methods (Lampert et al., 2014; Jayaraman and Grauman, 2014; Al-Halah et al., 2016; Farhadi et al., 2009; Rohrbach, 2017) typically employ a two-step procedure, in which the attributes of an image are first predicted and then, in step 2, matched to the class with the most similar attributes. The component that predicts image attributes is trained on a subset of classes, with the aim of generalising to unseen classes. However, this approach necessitates expensive manual annotation and is still susceptible to domain shift between the intermediate task (i.e., attribute prediction) and the target task (i.e., label prediction) (Fu et al., 2015). The nature of these attributes was discussed previously in Section 2.3.2.

Mapping-based learning, on the other hand, directly learns a mapping from an image feature space to a semantic space (Akata et al., 2016; Palatucci et al., 2009; Socher et al., 2013; Frome et al., 2013). As in the other category, the model learns to perform the matching based on a subset of classes only. The semantic space can be designed either by utilising a set of nameable attributes (detailed in Section 2.3.2) or by employing distributional embeddings (discussed in Section 2.3.3). In our thesis we employ the mapping-based approach that maps the input into a distributional space. This configuration leverages the ongoing advancements in representation learning, enabling the effective use of pretrained massive models without incurring the cost of task-specific annotations.

This section provided a brief overview of zero-shot learning techniques relevant to our thesis. For a more detailed look at the subject, we suggest the reader refer to Fu et al. (2018); Xian et al. (2019).

Zero-shot in action classification. The application of zero-shot tasks in action classification poses a greater challenge compared to other domains. This is primarily because of the scarcity of appropriate lexical resources. These resources play a critical role in determining which classes should be used for training and testing based on their degree of dissimilarity, as defined by these resources.

In object classification, ImageNet has a broader role than being just an image dataset. As it is constructed using WordNet, it offers a structured approach for selecting training and test classes based on the similarity between the two sets. For example, when training a model on the ILSVRC (1K ImageNet) classes, we can select test classes from the remaining 20K ImageNet classes by evaluating their similarity to the training classes. This can be achieved by quantifying the links or edges between the classes using WordNet. By using this method, we have the flexibility to adjust the level of difficulty in evaluating the model. For example, we can achieve this by selecting test classes that are either one or three links away from the training classes in terms of their relationship within WordNet.

Despite the added challenge, researchers have begun to tackle zero-shot learning for verbs or actions in the visual domain. One notable study is the work of Zellers and Choi (2017), which serves as a point of comparison for the current thesis since it was evaluated on the same dataset. Their study proposes a method for verb prediction that combines predefined attributes with distributional embeddings. The authors use lexical resources to identify 24 attributes that are considered relevant to the task of verb prediction.

2.5 Summary

In this chapter, we have presented a comprehensive overview of the problem addressed in this thesis from various perspectives. In Section 2.1 we have established the necessity of employing diverse descriptions of image actions, considering their multifaceted nature and varying interpretations. For instance, while one description may emphasise the manner in which the action is performed, another may prioritise the intention or objective behind it. It is worth noting that different descriptions often necessitate the use of distinct verbs, which is where the contribution of this thesis lies.

In Section 2.2, we conducted a review of several tasks that involve predicting verbs, either directly or indirectly, and examined the potential benefits of the resources generated from these tasks. Our analysis revealed that these tasks have limitations, such as not demonstrating the interrelationships between verbs when describing image actions, as in the verb-object format. Alternatively, they may be too specific for their intended use, as in the sense-based format, or may be challenging to implement on a large scale set of images and verbs, as in the soft assignment of multiple-verb format. In addition, we demonstrated that resources that present the output in a free-form format, i.e. captions, can potentially capture the interrelationships between verbs and usually covers a wider range of verbs. However, we also showed that they are challenging to use for the task presented in this thesis due to the extreme imbalance of verb counts in these resources. In Section 2.2, our objective was to review available resources for predicting verbs based on their output formulation. Our aim was to assess the limitations and potential benefits of these resources and determine the most suitable one for the task presented in this thesis. Section 3.3 will thoroughly discuss the dataset selected for experimentation in this thesis.

In Section 2.3 of our literature review, we have explored different techniques for representing verbs using distributed vectors. Two major approaches are feature-based representations, which involve identifying a set of semantic features for the verb (e.g., "*has legs*"), and data-driven representations, such as distributional embeddings. Distributional embeddings are created by analysing large amounts of text data and identifying patterns of word co-occurrence. These patterns are then used to create a vector representation for each word, including verbs, based on its distributional context in the text. While feature-based representations allow for greater interpretability, distributional embeddings have been shown to be highly effective in capturing subtle semantic relationships between verbs, such as similarity and relatedness. Given the objective of utilising existing large-scale resources in this thesis, distributional embeddings emerge as a highly compatible choice. Not only do they offer access to a wide range of pretrained models trained on vast corpora, but their ability to operate independently of manual curation or supervised training further reinforces their suitability for this task.

Finally, in Section 2.4, we provided a brief overview of zero-shot techniques and explained why they are relevant to the task presented in this thesis. We also identified a specific challenge that arises when applying these techniques to a verb prediction task.

Chapter 3

Methodology

Chapter 1 introduced the idea that information in addition to that derivable from visual inputs may be necessary to address the verb prediction task and that a potential source of this is text. The central hypothesis is that information distilled from the visual modality cannot entirely solve verb prediction, and incorporating other modalities can improve the model's predictive capability. In our work, the textual modality is used as an auxiliary source to extract lexical semantic information. Although text is not a perfect representation of knowledge, it can still provide valuable insights for the task, especially for the zero-shot setting, a setting where the textual modality should shine. Overall, we posit that a framework that incorporates semantics and facilitates zero-shot learning should yield a model better at predicting less-visual actions and capable of predicting multiple meanings of a given action.

This chapter discusses the verb prediction task in more detail and presents our proposed framework for building the prediction model. We first present a formal definition of the task (Section 3.1), distinguishing between the conventional supervised and zero-shot learning settings. Section 3.3 reviews the dataset selected for experimentation and discusses the dataset preparation process. Section 3.2 presents the proposed framework for building/training the model and explains how lexical semantic information is incorporated.

3.1 Task Definition

Formally, let x be an input image. Let $C = C_S \cup C_T$ be the set of all the possible action classes, where C_S is the set of classes observed during training, and C_T is the set of target classes that we want to predict at test time. The task is to learn a function $f(x)$ that assigns a target label $c \in C_T$ to an input image x given training instances from class C_S. We assume that an action class can be represented as a verb, which allows us to treat the problem of action prediction as a verb prediction task. In this regard, we consider three different settings for the verb prediction task:

Supervised learning. A standard supervised learning setting, where all classes are observed during training. That is, the set of classes C_S at training time is identical to the set of target classes C_T at test time; therefore, $C_S \equiv C_T$. Carrying out experiments in this setting allows us to operate in a controlled manner. It enables direct comparisons against many existing baselines trained only in this setting (i.e., standard vision classifier), usually comprising a softmax as a final layer. Furthermore, the supervised setting is ideal for studying the effect of lexical semantic information on less-visual actions. For example, we would like to compare the framework's performance on abstract classes (such as *comforting*) to standard label classifiers.

Zero-shot learning (ZSL). In a zero-shot learning setting (Lampert et al., 2014), the set of classes (or verbs) C_S observed at training time is disjoint from the target classes C_T at test time; i.e., $C_S \cap C_T = \emptyset$. Intuitively, a model under this setting is forced to predict a class label from C_T without having any knowledge about these classes at training time, for example by using associations and properties learned only from classes C_S. Such a setting can be useful for evaluating models that can predict novel target classes, for example abstract verbs (e.g., *greeting* or *comforting*) or classes that do not have many training examples (if any). We use this setting to demonstrate the ability of our proposed approach to predict such novel target classes.

Generalised zero-shot learning (GZSL). In a generalised zero-shot learning setting (Scheirer et al., 2013), the set of class labels C_S at training time is a strict subset of the target class labels C_T at test time; therefore, $C_S \subset C_T$. Under this setting, the model only learns from a subset of the possible set of action classes C_S at training time. At test time, the model generates a prediction from a pool of classes that contains both classes C_S and classes C_T. This setting enables us to test the framework under a relatively realistic and more challenging setup, given that the search space at test time contains a more extensive set of classes.

Our experimentation is limited to the first two settings only. The results from the GZSL setting proved challenging to analyse and interpret due to the mixed set of seen and unseen classes in the evaluation set.

3.2 The I2A Framework

Given an input image x, our Image2Action (I2A) model predicts one or more labels from a set of possible target class labels C_T. The key aspect of the model is an intermediate vector representation that connects the class labels C_S at training time to the target class labels C_T at test time. More specifically, we learn a regressor $h(x)$ that predicts a vector from training instances of class labels C_S. At test time, the function $f(x)$ will then predict a target class label $c \in C_T$ that has a vector representation v^c closest to the predicted vector $h(x)$:

$$f(x) = \underset{c \in C_T}{\operatorname{argmin}} D(h(x), v^c) \tag{3.1}$$

where $D(i, j)$ is some distance function between vectors i and j

We first introduce and motivate the use of vectors as an intermediate representation to verbs in Section 3.2.1. Section 3.2.2 describes the I2A framework in detail and explains the interactions between its components during training and testing processes. The subsequent sections describe the main components of I2A.

3.2.1 Distributed Lexical Representation of Verbs

In traditional classification tasks, each image is assigned a single label, which is represented by a one-hot encoded vector. The length of this vector is equal to the number of distinct image classes. This approach assumes that the labels are mutually exclusive, and that each image has only true label. However, this assumption cannot be valid for the verb prediction task, where multiple verbs can describe the same action from different perspectives (e.g., *lifting a hat* vs *greeting*). Instead of one-hot vectors, I2A encodes target labels using distributed lexical representations. The advantage of this approach is that it allows for a more nuanced representation of target classes (i.e., verbs) beyond simple word labels.

We hypothesise that distributed representations have the potential to encode valuable information from text that can complement the limitations of the visual modality in the verb prediction task. Our hypothesis is that a hybrid system combining multiple modalities holds promise as the optimal approach for addressing abstract tasks, as demonstrated in Figure 3.1. The following outlines the potential advantages of incorporating lexical embeddings in the verb prediction task.

(i) Since embeddings can be aggregated into a single vector, we anticipate that I2A will generate a vector that captures various interpretations of the image action. For example,

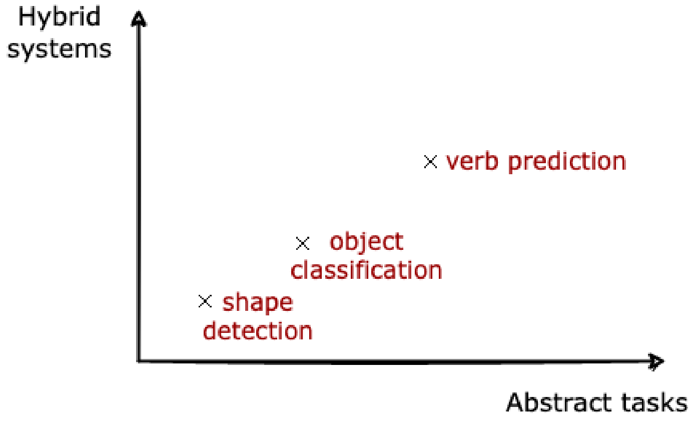


Figure 3.1: It is hypothesised that hybrid systems that use multiple modalities in addition to visual, can be advantageous for certain vision tasks whose goal is to comprehend concepts and ideas, rather than just detecting physical objects.

if an image depicts an athlete participating in a sprint race, instead of generating a one-hot vector solely for the verb *running*, I2A is expected to produce a vector that encompasses the concepts of *racing*, *running*, and *sprinting*.

(ii) The utilisation of word embeddings allows I2A to assign classes beyond those seen in the training data. This is particularly important for verbs with limited training examples but which have semantic connections to verbs with numerous examples. For example, if *eating* is included in the training set and we want I2A to recognise *dining* despite its absence in the training data, I2A in this case is expected to produce a vector that encompasses the concepts of *eating*, *formal event*, and *suits*, resulting in a vector that closely resembles the embedding of *dining*.

(iii) Lexical embeddings mitigate the reliance on visual similarity, thus enabling the handling of non-visual and less visually distinctive verbs. Unlike visual classifiers, which are limited by the intra-class similarity of visual features, lexical embeddings provide the capability to process verbs like *comforting* that can be performed in different visually distinctive manners.

(iv) Lexical embeddings are capable of encoding different lexical units and not being limited to single-word verbs. In English, two prominent examples highlight the significance of this versatility. The first is verb-particle pairs that convey a different meaning compared to the verb alone (e.g. *stick* vs *stick up*). The second is multi-word expressions comprised of verbs and conventionalised objects that express a meaning that cannot be derived from their individual parts, i.e. non-compositional meaning (e.g. *pay* vs *pay attention*).

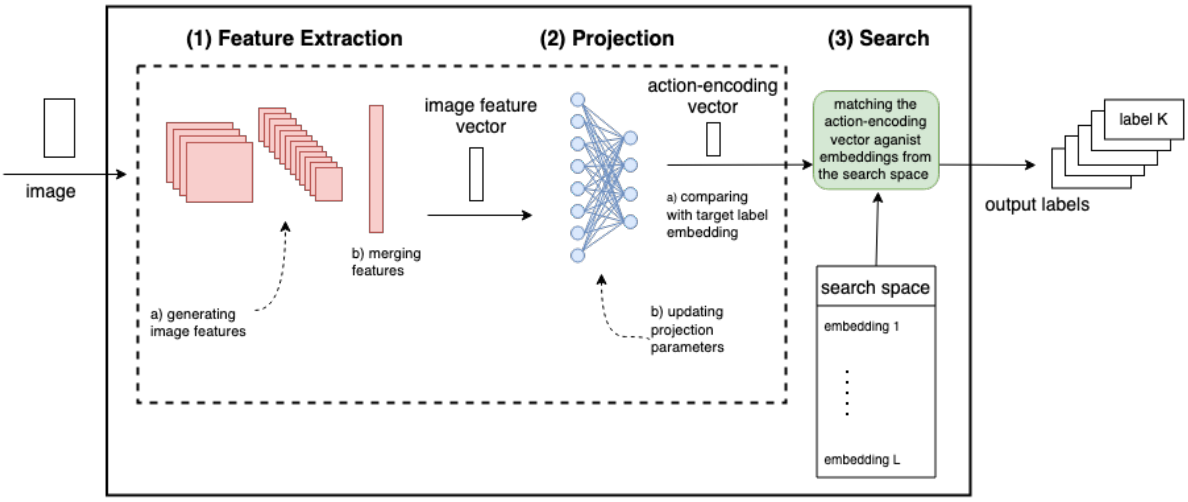

(1) Feature Extraction
(2) Projection
(3) Search
image
image feature vector
action-encoding vector
matching the action-encoding vector aganist embeddings from the search space
label K
output labels
a) generating image features
b) merging features
a) comparing with target label embedding
b) updating projection parameters
search space
embedding 1
embedding L

Figure 3.2: The I2A framework comprises three components, each represented with a distinct colour: (1) Image Feature Extraction (in red), (2) Projection into a Lexical Semantic Space (in blue), and (3) Search in the Lexical Semantic Space (in green). The image feature extractor shown here is a visual-based in the form of a convolutional neural network (CNN).The inner box illustrates the the steps involved in both training and testing processes, while the outer box shows the additional steps required specifically for the testing process.

3.2.2 Training and Optimisation

Our proposal takes a different approach to the task of action classification by casting it as a regression task that predicts a vector in the space of verb embeddings, rather than directly classifying the action as in traditional systems. This should allow us to explore the the space of all plausible verbs for describing the action instead of being committed to single verb. The flow of images through I2A's components is shown in Figure 3.2. Given an image as input, I2A predicts a vector that attempts to encode the semantics of the image action, referred to as the action-encoding vector (the process is discussed in details in Section 3.2.2).

This action-encoding vector is then compared with subset of the lexical semantic space embeddings specified at test time, and the k classes with the most similar embeddings are selected. The process can be seen as predicting the latent semantic properties of the action, which are expressed through the selection of different lexical embeddings.

I2A draws inspiration from prior work in the zero-shot recognition task for objects (Frome et al., 2013; Palatucci et al., 2009; Socher et al., 2013). However, unlike these studies, I2A does not have access to a taxonomic hierarchy such as WordNet for objects. These prior studies utilised the ImageNet dataset, which was constructed based on the WordNet hierarchy, allowing for varying levels of evaluation difficulty based on the similarity between the training and evaluation classes. It also could serve as a source of auxiliary information.

Our goal with I2A goes beyond just addressing the zero-shot action recognition task. We aim to develop a framework that can decode the diverse meanings of actions, including those that are not observed during training or less straightforwardly visual, such as *greeting* and *comforting*.

Training process. The training process of I2A begins by pairing each image with its corresponding vector, represented as distributed embedding, rather than the string label. In a general setting, the corresponding vector could be a verb embedding or a combination of embeddings for verbs related to the image. The images and their associated vectors are then fed into I2A as input ($x =$ image) and output ($y = n$–dimensional vector), respectively.

The training process of I2A involves two stages. First, the images pass through the image feature extraction component, which produces a compact and dense representation of the images in the form of either visual features or semantic feature (as discussed Section 3.2.3).

In the second stage, the projection component maps each image feature vector to an action-encoding vector in the lexical semantic space. The objective of this stage is to position the action-encoding vector close to the verb vectors that describe the image's action. The loss is calculated and backpropagated solely to the projection component, updating its parameters while keeping the weights of the image feature extractor fixed. The rationale for fixing the parameters of the visual feature extractor is discussed and justified in Section 3.2.3.

To summarise, the objective of the training step is to approximate the embedding of the true label or the combination of the true labels by utilising only two components of I2A: the image feature extractor and the projection component. The search component is not involved in the training process.

Training objective (Loss function). The choice of loss function is crucial for training deep learning models as performance on certain tasks will benefit more from particular loss functions than from others. In our work, the function is required to compare and evaluate vectors (or sets) of real numbers, which means standard loss functions such as the log loss and the mean squared error cannot be employed. In I2A, we choose a cosine-based function for measuring loss (Equation 3.2). The function aims to approximate the target vector (i.e., the embedding of the true class). Practically, I2A cannot achieve this goal but manages to produce a similar vector, the desired behaviour for the tasks in this thesis. Other loss functions (Yin and Shen, 2018; Kumar and Tsvetkov, 2019) are also considered for improving prediction accuracy. However, these functions focus primarily on improving training efficiency, and their results show only minor improvements in overall accuracy.

$$
\begin{aligned}
Loss &= 1 - \frac{1}{n}\sum_{i=1}^{n} cos(\mathbf{x}_{i_{pred}}, \mathbf{x}_{i_{target}}) \\
cos(x_1, x_2) &= \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\max(\|\mathbf{x}_1\|_2 \cdot \|\mathbf{x}_2\|_2)}
\end{aligned}
\tag{3.2}
$$

Test process. The evaluation process necessitates, in addition to the steps involved during training, the use of a search mechanism to identify the class embeddings that are most similar to the action-encoding vector. To achieve this, a search function is incorporated as the third component of I2A. This function takes the action-encoding vector as input, and performs a k-nearest neighbour search to determine the k closest class embeddings. The labels associated with these classes are then presented as the final prediction.

To summarise, I2A has three main components: (1) an image feature extractor (Section 3.2.3), which extracts visual features from images; (2) a projection function (Section 3.2.4), which maps the visual features into a lexical semantic embedding space; and (3) a search function (Section 3.2.5), which determines the nearest verb vectors to the vector associated to the input image.

3.2.3 Image Feature Extractor

This section provides a detailed discussion of the first component of I2A, presenting the two types of extractors considered in this thesis. The ***visual feature extractor*** is a neural network that extracts dense vector representations of the raw images (e.g., jpeg files), with the goal of capturing high-level visual features that are expected to be relevant to the verb prediction task. The salience of visual features is dictated by the dataset and task used for training the neural network. That is, the salient features relevant to an image classification task may differ from those relevant to an instance segmentation task. We detail our approach to devising the visual feature extractor in the following section, given that it is the main type used for experimentation throughout this thesis.

The ***semantic feature extractor*** on the other hand, extracts features of semantic concepts associated with the image to produce a dense and high-dimensional vector representation of its semantic features. It is assumed that the visual concepts have already been associated with the raw image, either through automatic or manual processes (i.e., training starts with these visual concepts). The extraction of semantic features can be achieved through various techniques, such as pooling operations over the embeddings of the visual concepts, or through the use of a deep neural network pretrained on an NLP task, such as sentence similarity. The experiments on semantic extractors can be found in Section 6.2.

3.2.3.1 Visual Feature Extractor

Convolutional Neural Networks (CNNs) are used as the visual feature extractor for I2A. CNNs are a type of deep learning neural network that are commonly used for image and video recognition tasks, and their architecture design is inspired by the structure of the human visual system and the process of visual perception (Goodfellow et al., 2016). CNNs are designed to learn spatial hierarchies of features from input images (as illustrated by the feature extraction component in Figure 3.2) in an automatic and adaptive manner, eliminating the need for manual feature engineering. The primary layer type in CNNs is convolutional layers, which extract features from images through the application of convolutional filters. Despite being introduced decades ago, it is only in recent years that a rich diversity of CNN architectures has emerged.

In our work, we chose to work with ResNet (He et al., 2016) as the default visual feature extractor. ResNet is still considered a strong baseline for many computer vision tasks and one of the best-performing CNN architectures on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015). ResNet comes in various block sizes, and in this work, ResNet50 was chosen as the default option due to its solid performance and compact size. More recently, transformers have shown promising results for computer vision tasks (Carion et al., 2020; Dosovitskiy et al., 2021). In this thesis, we have decided to concentrate solely on CNNs for visual feature extraction, as this area is not a central focus of the work.

Pretrained models. The default image feature extractor in I2A is a CNN model, as previously noted. In this section, we discuss the use of the ResNet50 model pretrained on the ILSVRC dataset. ILSVRC, which is a subset of the ImageNet dataset (Deng et al., 2009), consists of 1.2 million images depicting objects in a canonical view and annotated with a single label, encompassing 1000 distinct object classes. Pretrained CNNs are conventionally obtained by removing the task-specific softmax layer from a classification model. This layer generates a probability distribution of the output classes in classification tasks (such as the 1000 classes in the ILSVRC classification tasks). The prediction is ultimately made by selecting the class with the highest probability score.

CNNs pretrained on ILSVRC are proven to be a viable option for transfer learning. Kornblith et al. (2019) have demonstrated a positive correlation between performance on ILSVRC and the transferability to other datasets. That is, they find that CNN models that perform well on ILSVRC also do well as pretrained models. This finding has also been observed in other vision tasks such as object detection (He et al., 2016).

In the training process for new tasks, it is customary to further train or fine-tune the CNN component for task-specific weight adjustments. However, our approach deviates from this common practice and retains the original pretrained weights learned from the ILSVRC dataset, without undergoing any fine-tuning. This means that in our work the CNN component is utilised purely for feature extraction, as it remains unaffected and uninvolved in the training process. The CNN model employed in our work is obtained from the PyTorch repository (Paszke et al., 2019).

Feature vectors. The output of the visual feature extractor, such as a CNN, is a high-dimensional dense vector known as a visual feature vector or image feature vector. The composition of such feature vectors necessitates an aggregation method to derive and combine visual information from different parts of the image. In the context of a CNN, this aggregation is performed using pooling techniques that summarise signals from the feature maps obtained through convolution.

In typical CNNs, feature vectors are obtained through either an average or a max pooling operation. However, in our work, we adopt a different approach, where feature vectors are obtained by concatenating features extracted from both pooling operations. We hypothesise that the concatenation enables I2A to extract more comprehensive information from the images, which is critical given that the image features are not specifically engineered for the verb prediction task. The size of the feature vector is dependent on the CNN architecture and pooling method used, typically ranging from 512 to 4096 dimensions.

In I2A, feature vectors can be calculated in two modes, online and offline. This thesis uses the online mode, which computes feature vectors during the training phase by processing raw images for each iteration. Conversely, the offline mode calculates feature vectors by processing the entire dataset once and saving the results in a permanent storage. During training, these feature vectors are used as input, instead of the raw images. Although the online mode incurs a substantial increase in the duration of training when compared to the offline mode, it enables the application of image augmentation techniques, including geometric transformations (e.g. rotation, horizontal flipping, and zooming) and colour space transformations. These techniques help to regularise the model and reduce the risk of overfitting by providing variations to the training data (Shorten and Khoshgoftaar, 2019).

3.2.3.2 Semantic Feature Extractor

The previous section introduced the first method of extracting features from images using a visual feature extractor such as CNNs. Although CNNs can handle complex visual shapes and generate valid representations of visual concepts contained in images, we hypothesise that these visual features are insufficient for the task of verb prediction.

Therefore, an alternative, semantic-based, approach is considered for feature extraction,

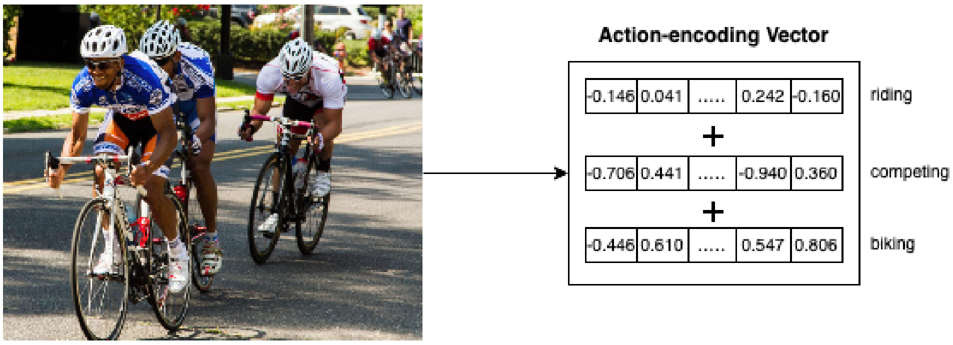


Figure 3.3: Example action-encoding vector that hypothetically encapsulates the representation of multiple verbs of which we can use to describe the image action.

which encodes the semantic content of images more explicitly, such as information about objects, spatial relations and scene types. However, this approach requires a reliable oracle, such as a human annotator or an accurate vision system, making it less practical for real-world applications. Examination of this approach will be presented in in Section 6.2.

3.2.4 Projection

The second component in I2A is the projection component, which serves as a regressor function that takes an image feature vector as input and outputs a vector in lexical semantic embedding space that we refer to as an "action-encoding embedding". In the simplest case, where images are assigned one verb only and all classes are observed during training, the action-encoding vector is expected to be a close estimate of the target embedding (i.e., the embedding of the true class). That is, the purpose of the training process is to optimise the projection component so that it accurately matches the image representation with its corresponding label representation. It's important to note, however, that the action-encoding vector is ultimately designed to serve as an embedding capable of capturing multiple meanings learned from various classes, as demonstrated in Figure 3.3.

The projection component is implemented as a two-layer fully connected feedforward neural network (FNN), with 2048 nodes in the first layer and 1024 nodes in the second layer. The number of nodes has been optimised for the size of input feature vector (4096 dimensions) and the action-encoding vector size (300 dimensions), but I2A should not be limited to these sizes. During the training process, only the projector's parameters are updated; they are initialised randomly.

The action-encoding vector is hypothesised to encode various meanings (or descriptions) of the action, as demonstrated in Figure 3.3. Word embeddings have been shown to encode distinct concepts related to a wordform in a single vector (Camacho-Collados and Pilehvar, 2018). For example, the embedding of *play* encodes the various senses of the word. While this presents a challenge for sense disambiguation, it is advantageous for I2A, as the action-encoding vector is designed to capture diverse action meanings.

3.2.5 Search

As discussed in Section 3.2.2, I2A, specifically the regressor component, is incapable of producing the precise set of floating-point numbers that form the distributed representation of the target. As a result, a search mechanism is needed, which serves to bridge this gap. The search component accepts the action-encoding vector generated by the projection component and outputs the labels of the k closest verbs similar verbs. The search component requires the specification of (1) a search space and (2) a distance measure in this space.

Search space. The search space is the set of classes made available for the search component, which is not restricted to the set of training classes. The search space is also flexible as it can be defined after training the verb prediction model. The space can be as big as the entire vocabulary of the embedding model or as small as a subset of training (or seen) classes. The search space can be viewed as a reflection to the three learning settings presented in Section 3.1: it can be either the same set observed in training (the supervised setting), a disjoint from the one observed in training (the ZSL setting), or a superset that includes all training classes (i.e., the GZSL setting).

In our work, a proper definition of the search space is particularly crucial for the ZSL setting. Hence, it may be necessary to determine what actions are required or advantageous for training and what actions can benefit from the trained model. For instance, training on classes with sufficient resources and testing on classes with limited resources can be a reasonable approach to use semantic embeddings in ZSL, but other approaches such as categorising classes based on abstraction, ambiguity, or imageability may also be effective (Coltheart, 1981; Wilson, 1988).

Similarity metric. In I2A, a metric is required to compare the action-encoding vector to each embedding in the search space. The chosen metric must be efficient in terms of computational speed and memory consumption, as the vectors are often large (up to 1000 dimensions) and the search space contains thousands of embeddings. Furthermore, the metric must be precise, as the embeddings in the space are expected to be very close, as they correspond to a subgroup of verbs that share many common characteristics. Cosine similarity is a commonly used metric for measuring the similarity between two vectors. It calculates the cosine of the angle between two vectors, which is equivalent to the normalised dot product.

I2A utilises a variant of the cosine metric family, known as the angular distance (Equation 3.3). This metric has been shown to be more effective in distinguishing highly similar vectors with small angles between them (Cer et al., 2018). Given that verb embeddings are likely to cluster in sub-regions of the embedding space, the use of angular similarity is particularly well-suited for our task. We have used the software of Boytsov and Naidan (2013) for both the computation of similarity and the performance of nearest neighbour matching.

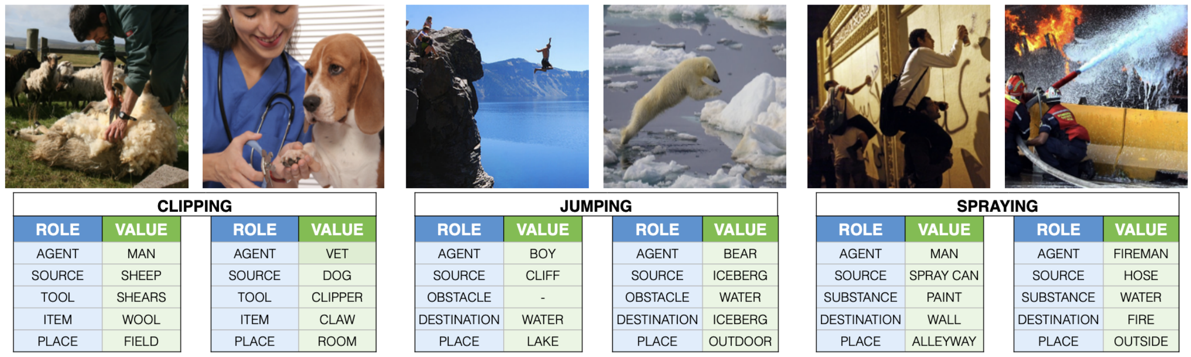
$$\begin{aligned} \textit{angular similarity} &= 1 - \textit{angular distance} \\ \textit{angular distance} &= \frac{arccos(cos(\mathbf{x}_{i_{pred}}, \mathbf{x}_{i_{target}}))}{\pi} \end{aligned} \tag{3.3}$$

3.3 Dataset

Dataset selection. In selecting a suitable dataset for our experimental work, we considered three options, namely imSitu, HICO, and MSCOCO, which are previously described in Section 2.2. After careful consideration, we chose the imSitu dataset as the primary focus of our research. The imSitu dataset serves as a starting point for establishing the validity of the I2A framework, as it offers the lowest level of complexity among the three datasets considered. This dataset is well-balanced in terms of the distribution of images per class, and its use of only one verb label per image makes it suitable for early experimentation of I2A framework. Moreover, imSitu comes with a comprehensive annotation of image content, which further supports its suitability for our research.

The situation recognition dataset (imSitu). The dataset contains structured and comprehensive annotations of actions occurring in $126,102$ images. Each image is annotated with a single verb describing the action, the objects involved, and their respective roles in the action. Images are distributed across 504 classes (or verbs), and the number of images per verbs ranges between 200 and 400 images, indicating that the dataset is relatively balanced, compared to other datasets considered such as HICO and MSCOCO. Figure 3.4 shows a set of examples taken from the original paper (Yatskar et al., 2016) along with their structured annotations (or situations as refereed to in the dataset). The image on the right shows a fireman spraying water on fire and is labelled with the following situation: $S = \big($ `spraying`, $\{$(`agent`, `fireman`), (`source`, `hose`), (`substance`, `water`), (`destination`, `fire`), (`place`, `outside`)$\}\big)$.

In this dataset, each verb (or class) is associated with a set of fixed roles derived from FrameNet, with an average of 3.5 roles per verb. The values corresponding to these roles are



CLIPPING

ROLE	VALUE	ROLE	VALUE
AGENT	MAN	AGENT	VET
SOURCE	SHEEP	SOURCE	DOG
TOOL	SHEARS	TOOL	CLIPPER
ITEM	WOOL	ITEM	CLAW
PLACE	FIELD	PLACE	ROOM

JUMPING

ROLE	VALUE	ROLE	VALUE
AGENT	BOY	AGENT	BEAR
SOURCE	CLIFF	SOURCE	ICEBERG
OBSTACLE	-	OBSTACLE	WATER
DESTINATION	WATER	DESTINATION	ICEBERG
PLACE	LAKE	PLACE	OUTDOOR

SPRAYING

ROLE	VALUE	ROLE	VALUE
AGENT	MAN	AGENT	FIREMAN
SOURCE	SPRAY CAN	SOURCE	HOSE
SUBSTANCE	PAINT	SUBSTANCE	WATER
DESTINATION	WALL	DESTINATION	FIRE
PLACE	ALLEYWAY	PLACE	OUTSIDE

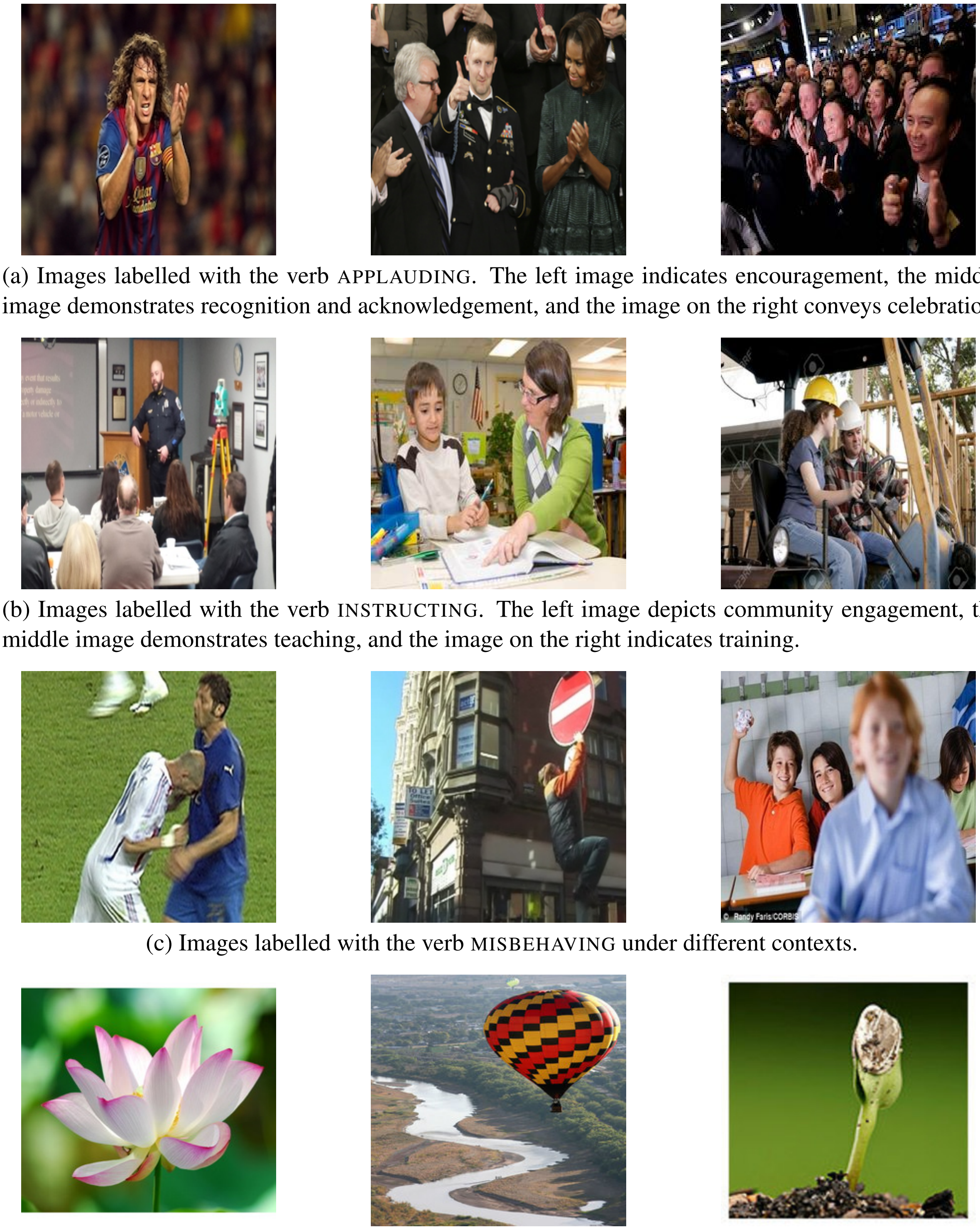
Figure 3.4: Examples from Yatskar et al. (2016) demonstrating six distinct situations, two of which share the same verb. The fixed roles for each verb are highlighted in blue, while the values of these roles (shown in green) depend on the specific actions depicted in the image and their corresponding interpretations.

linked to an extensive collection of 11.5K WordNet synsets, leading to a vast collection of distinct image annotations, encompassing over 200K situations.

The importance of verbs in imSitu is emphasised through their role in the image selection process. The authors initially considered almost 10K candidate verbs, but narrowed down the set to 1K verbs by eliminating non-visual verbs and those requiring technical background or advanced literacy. The effort to gather images for these 1K verbs led to further reduction to 504 due to the scarcity of suitable example images of some verbs. The final set of verbs intentionally excludes ambiguous verbs such as PLAYING; however, it also includes near-synonyms, making it difficult to differentiate between them and treat them as separate concepts, such as PHONING and TELEPHONING and PEEING and PISSING. Additionally, some verbs may also prove difficult to distinguish given the static nature of images, such as LAUGHING and GIGGLING.

Conversely, certain pairs of verbs exhibit nuanced but important differences, such as BIKING and RIDING (where biking is a type of riding), DIALLING and PHONING (where dialling is a sub-activity of phoning), and PACKAGING and PACKING. These subtle distinctions are of significance in the context of this thesis and make the imSitu dataset a promising resource for experimentation.

The significance of verbs in imSitu is further underscored by their contribution to the image annotation process where images are annotated according to the frame of their verbs (i.e., what objects to label and what are their roles in the action). Various human roles are encompassed within imSitu. For instance, “man”, the most frequent noun, appears in 44.6% of the roles. This should be a good characteristic for the task addressed in this thesis as we aim to understand actions with multiple different interpretations. Having images of humans engaged in various roles stands as a valuable resource for a such task.



(a) Images labelled with the verb APPLAUDING. The left image indicates encouragement, the middle image demonstrates recognition and acknowledgement, and the image on the right conveys celebration.

(b) Images labelled with the verb INSTRUCTING. The left image depicts community engagement, the middle image demonstrates teaching, and the image on the right indicates training.

(c) Images labelled with the verb MISBEHAVING under different contexts.

(d) Images from three different classes: SPROUTING, BALLOONING, and BLOSSOMING. These classes are unsuitable for this thesis due to their one-dimensional interpretability.

Figure 3.5: Sample images from the imSitu dataset. The upper three rows display distinct interpretations of actions within imSitu classes, highlighting the dataset's potential for the thesis task. The fourth row shows examples of unsuitable classes.

Our experimentation with the I2A framework on the imSitu dataset will play a crucial role in determining the feasibility of future experimentation with the remaining datasets, which are beyond the scope of this thesis. The imSitu dataset is available in two versions, one containing original images of varying quality, with most being high-quality, totalling a size of 34 gigabytes. The second version comprises resized images, with each image scaled down to 256×256 pixels, and has a smaller size of 3.7 gigabytes. For our research, we opted for the resized version of the dataset, primarily due to its compatibility with existing pretrained CNNs and its efficiency in training.

Dataset processing. All experiments are carried out on a filtered version of imSitu, containing only images of human actions. As previously discussed in Chapter 1 and 2, this thesis is particularly concerned with exploring the complex nuances of actions, and human actions were deemed most suitable for this purpose. Figure 3.5 shows examples of classes deemed appropriate for the task, contrasting with others (displayed in the bottom row) that are not applicable. The filtering process were performed by leveraging the comprehensive annotation of imSitu, which not only annotated the target (class) but also the content of the image. The process involves removing images that are not annotated with labels describing a person or a group of people. Utilising the WordNet hierarchy, all images in imSitu were removed if they did not contain in their annotations a synset which was either `PERSON.N.01`, `PEOPLE.N.01` or `SOCIAL_GROUP.N.01` or a hypernym of one of these two synsets. This process resulted in a reduction of the number of instances in most classes, particularly those related to non-human actions such as SPROUTING. Classes were eliminated if their instances were reduced to fewer than 90 examples. The filtering process reduced the number of classes from 504 to 463 and the total number of images from $126,102$ to $108,342$.

3.4 Summary

This chapter has provided a formal definition of the task of predicting verbs for actions in images and discussed three possible learning settings for it: supervised learning and two variants of zero-shot learning (Section 3.1). Section 3.2 introduced the I2A framework and explained how word embeddings are incorporated in the prediction process. The section also describes the three components of I2A: (1) the image feature extractor, (2) the projector, and (3) the search component. Two approaches to image feature extraction are distinguished: a visual-based and a semantic-based approach. Section 3.3 discussed the process of selecting and preparing the dataset.

Chapter 4

Initial Experiments with the I2A Framework

This chapter presents experimental results utilising the I2A framework proposed in the previous chapter, with default or standard components outlined in Section 3.2. The default visual feature extractor is ResNet50 pretrained on ILSVRC, and the projection component uses two layers. To represent target labels, we use the pretrained GoogleNews embeddings, which are trained using the Continuous Bag of Words (CBOW) method (discussed in Section 2.3.3) on a large set of text comprising news articles, Wikipedia articles, parliamentary text, non-news web pages, and other sources, totalling around 100 billion words. The embedding size is set to 300 dimensions, and the vocabulary size is three million words and phrases. These embeddings have demonstrated good performance in various tasks, as discussed in Section 2.3.3. In this chapter, we construct models using the GoogleNews embeddings under two learning settings: supervised (Section 4.1) and zero-shot (ZS) (Section 4.2), with a view to establishing how well this approach to verb target class representation performs on verb prediction task.

In the remainder of this chapter, we will refer to the model created using this setup as the ***I2A model***. However, in the subsequent chapters, we will differentiate between model variants by naming them after their unique components. For example, in Chapter 5, we will refer to this model as the “google-news” variant to distinguish it from other variants that use different embeddings. Similarly, in Chapter 6, we will refer to this model as the “ILSVRC” variant to distinguish it from other variants that use different image feature extractors. The default settings are also employed in experiments in the subsequent two chapters, varying only the target label representation in Chapter 5 and the component that extracts features from input images in Chapter 6.

4.1 Supervised Setting

In this section, we describe the construction of the I2A model under the supervised setting. The model is trained on labelled images from *all classes*, with the objective of learning a mapping function that can predict an action-encoding embedding based on the input image. The action-encoding embedding is expected to represent one interpretation of the image action, which is the same as target label associated with the image. This differs from the zero-shot setting in Section 4.2 where the model is trained on a subset of verbs and then is expected to be able to generalise to new, unseen verbs by utilising information encoded in the distributed representation of related verbs.

4.1.1 Experimental Setup

As previously discussed in Section 3.3, we have selected the imSitu dataset for experimentation in this thesis. The dataset is split into a 75:25 ratio for training and for testing, respectively, using stratified sampling to maintain consistent sampling ratios across classes.

The 1cycle learning rate policy (Smith, 2018) has been applied, with 1 cycle and 20 iterations or epochs per cycle. The 1cycle policy cyclically varies the learning rates between two previously specified bounds (as shown in Figure 4.1), promoting super-convergence and leading to a greater boost in performance in less training time compared to standard training policies. All experiments in this thesis follow the same setup and use the same hyperparameters unless stated otherwise.

The model has approximately 34 million parameters, of which only 11 million are trainable, all located in the projection component. Figure 4.2 shows the loss and top1 accuracy over training epochs. Although the left plot shows some indications of overfitting, we decided to train the model for 20 epochs instead of 10. This is because the overfitting signal is based on an intermediate output (i.e., action-encoding vectors) which may not necessarily translate to a direct impact on the final output (i.e., verb labels). It is worth noting that the loss function not being directly linked to the evaluation criteria is a common issue in machine learning, as discussed in Qin et al. (2008). Therefore, while we should be mindful of this issue, it may not necessarily be a significant concern in this particular case.

4.1.2 Baselines

In this section, we introduce three baselines to compare with the I2A model. The purpose of these baselines is to establish a performance range against which we can measure the model's effectiveness. The first baseline (random prediction) serves as a lower-bound, and the second

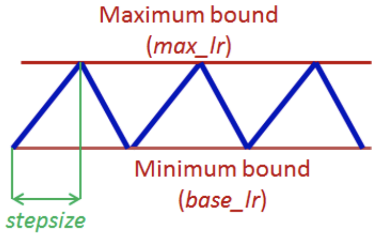


Figure 4.1: The diagram, reproduced from Smith (2018), depicts how learning rates changes cyclically over batches. The blue lines represent learning rate values changing between bounds.

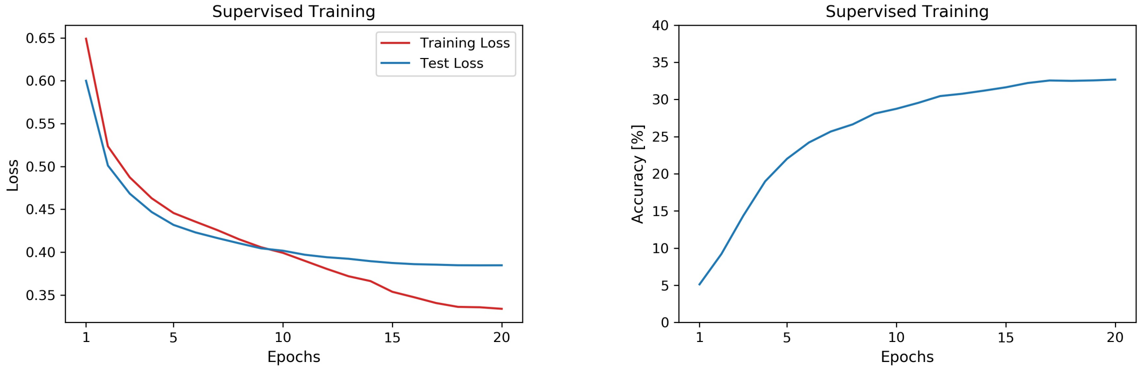


Figure 4.2: Performance over epochs. The left plot shows loss, and the right plot shows top1 accuracy.

(vis-proto) allows us to directly compare the use of visual representations of the target class with the use of lexical semantic embeddings. The third (label-cls) acts as an upper-bound. The performance of these baselines is discussed in the following section, along with the I2A model.

1. Random prediction acts as the lower bound on the level of performance that can be expected when assessing a model trained on a dataset of 463 classes. This baseline randomly selects five predictions for each test image. These random predictions are then fed to the evaluation code to calculate top1 and top5 accuracy. Alternatively, accuracy can be calculated directly via Equations 4.1 and 4.2, where C_T is the set of test (target) classes.

$$\textit{Top1 accuracy} = \frac{1}{|C_T|} \tag{4.1}$$

$$\textit{Top5 accuracy} = \sum_{i=1}^{5} \frac{1}{|C_T| - (i-1)} \tag{4.2}$$

2. Visual prototype prediction (vis-proto) follows the same process devised for I2A and uses all of its three components. The only difference is that it uses visual class prototypes or exemplars as the target vectors instead of the lexical semantic embeddings. The class prototypes are

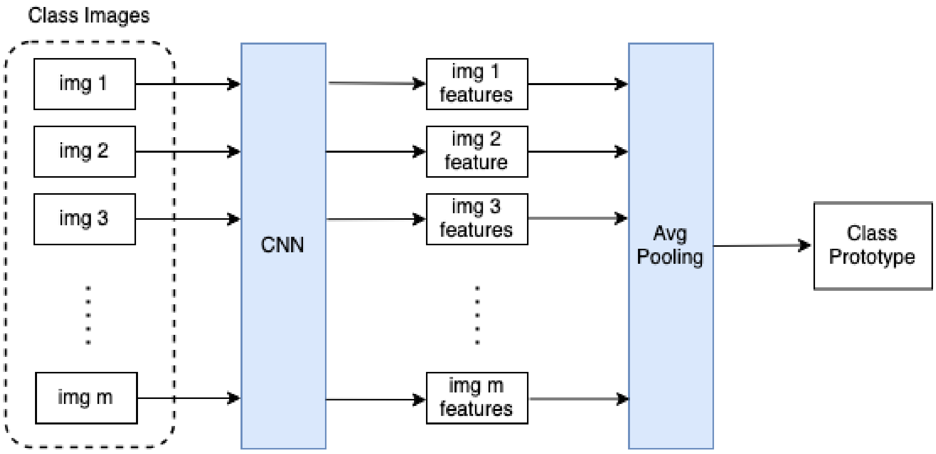


Figure 4.3: The process of creating the class prototype for a single class.

calculated using visual features extracted from images in the imSitu training set. The visual features are grouped by class and then aggregated using the average mean pooling operation, resulting in a single vector that is used as the target vector for class instances (Figure 4.3). By excluding lexical semantic information and having the same learning complexity as I2A, this baseline provides a controlled comparison against the results of I2A.

3. Direct label classification (label-cls) is a conventional image classifier that acts as the upper bound on the level of performance that can be obtained for a model trained on a dataset of 463 classes. The classifier consists of: (1) the CNN module, (2) the penultimate layers, and (3) the softmax layer. The first two components are similar to the I2A framework. The CNN module of the classifier is the same image feature extractor used to train the I2A model (i.e., ResNet50 and pretrained on the ILSVRC dataset with frozen weights). The penultimate component has the same architecture as the projection component of I2A, and the classifier has the same training hyperparameters as I2A, which were discussed in Section 4.1.1. The primary difference lies in the third module, where the classifier uses a softmax layer, whereas I2A uses a search component. That means that during training, only the weights of the penultimate and softmax layers are updatable.

4.1.3 Results

In this section, we analyse the performance of the I2A model and compare it to the baselines introduced earlier, highlighting the strengths and limitations of I2A. To evaluate the model's performance, we use the macro-averaging accuracy metric, which is more appropriate than the

Table 4.1: Comparison of I2A model performance with baseline models.

	Top1	**Top5**
I2A model	32.48	51.45
random prediction	00.21	01.08
visual prototype prediction (vis-proto)	21.49	41.77
direct label classification (label-cls)	37.04	63.18

micro-averaging accuracy given that the imSitu dataset is slightly imbalanced. Equation 4.3 shows the formula for this metric where Acc_i is the accuracy for class i.

$$Acc_{macro} = \frac{Acc_1 + Acc_2 + \cdots + Acc_k}{k} \tag{4.3}$$

The results presented in Table 4.1 clearly demonstrate the superior performance of the I2A model over the random prediction baseline (with a top1 accuracy of 32.48% compared to 00.21%) and the visual prototype prediction (vis-proto) baseline (with a top1 accuracy of 32.48% compared to 21.49%). The results highlight the contribution of textual information in I2A. However, the model falls short of the direct label classification baseline, which achieves a top1 accuracy of 37.04%.

Despite not matching the performance of the direct label classification baseline, the performance of I2A model is still promising given the challenge of producing dense vectors instead of one-hot vectors and the potential benefits of incorporating textual information for predicting previously unseen verbs. These benefits will be discussed further in Section 4.2.

In addition to using the macro-averaging accuracy metric, the F1 metric is also applied to measure the model performance. F1 yields similar results. We have opted to use the macro-averaging accuracy metric due to its wide adoption in image classification tasks and for measuring the performance of top k. Throughout the rest of this thesis, we will refer to macro-averaging accuracy as the accuracy metric. We will only use F1 sparingly, and solely to show the performance of individual classes when required.

4.1.4 Cross-Baseline Analysis

Qualitative analysis. To study the effect of textual information on I2A, we conducted a qualitative analysis of the I2A model and compared its performance on individual classes with that of the visual prototype prediction baseline (vis-proto). Table 4.2 categorises the classes into three groups based on I2A performance compared to vis-proto: (1) significantly better, (2) significantly worse, and (3) the same.

Table 4.2: Comparison of the I2A performance with the visual prototype (vis-proto) model on specific individual classes. The classes are divided into three groups: (1) those in which I2A performs significantly better, (2) those in which I2A performs significantly worse, and (3) those in which both models perform equally bad.

I2A >> vis-proto		**I2A << vis-proto**		**performing the same**	
LEADING	NAGGING	RAFTING	PROWLING	SURFING	PRACTICING
FRISKING	MOURNING	POTTING	MICROWAVING	READING	GRIMACING
PINCHING	KNEELING	FORDING	LAUNCHING	PAYING	STARING
HITTING	MEASURING	MOWING	FLOSSING	RIDING	CHOPPING
FLINGING	GRINNING	FLEXING	PANHANDLING	STRIKING	FLOATING

The first group includes classes where the I2A model performed significantly better compared to vis-proto; a group that greatly benefits from semantic information. One of the goals of the thesis is to investigate the characteristics of these classes and what makes them good candidates for I2A. At first look, these classes seem to be less concrete; some are ambiguous, such as HITTING, and others are abstract, such as LEADING, NAGGING, and MOURNING. This group may provide some insights on where I2A can be used to unlock abstract interpretations of classes.

Classes in the second group exhibited strong associations with specific objects or a small set of objects. For instance, images of LAUNCHING always contain rockets and images of MICROWAVING always contain microwaves. These objects are classes in the pretraining dataset (ILSVRC) of the image feature extractor, and therefore the corresponding classes in imSitu are easily predicted without an additional information from the auxiliary medium (i.e., the distributional embeddings). Section 4.1.7 explores the relationship between the classes of ILSVRC and imSitu.

The third group consists of classes on which both models performed poorly (i.e., near zero-accuracy performance). We are unsure why the two models underperformed on these classes. However, since the two models share the same framework design, we believe that these classes should help further investigate inherit limitations of the I2A framework.

Quantitative analysis. In previous chapters, we put forth the hypothesis that conventional image classifiers may not excel in identifying abstract interpretations of image actions, and we recommended integrating word embeddings to improve their performance. To test this claim, we conducted a cross-baseline analysis by comparing the performance of the I2A model to two baselines (label-cls and proto-vis) on two sets of classes: concrete and abstract. We will elaborate on this categorisation further in Section 4.2.5.

Table 4.3: I2A performance compared to the baselines on two subsets of imSitu classes: the 67 most concrete and the 67 least concrete. It also shows the performance margin difference between the two sets. Section 4.2.5 discusses the identification of these classes in detail.

	Least Concrete	**Most Concrete**	**% Difference**
I2A model	22.23	39.70	78.59
visual prototype prediction (vis-proto)	15.14	27.73	83.16
direct label classification (label-cls)	29.03	42.48	46.33

Based on the results presented in Table 4.3, our findings indicate that all three models displayed notably inferior performance on abstract classes when compared to their performance on concrete classes, which was in line with our expectations for the baselines. The I2A model, however, is initially assumed to outperform the others on abstract classes, but this did not occur. In fact, the performance gap between the two sets was more pronounced for the I2A model than for the label-cls baseline.

Section 4.2 will further examine the I2A model's performance on unobserved abstract and concrete classes. Additionally, the next chapter will propose methods for addressing the I2A model's underperformance on abstract classes by enhancing and customising the lexical semantic representation of actions.

4.1.5 Performance Variation

In this section, we examine the performance variation of the I2A model from different perspectives. Firstly, we assess the performance variance across multiple samples of training images. In particular, we conducted a 4-fold cross-validation experiment with individually stratified folds to assess I2A's robustness. The results, shown in Table 4.4, indicate that I2A has consistent performance across all four folds, demonstrating robustness against instance variations caused by sampling.

Next, we investigate the variation in I2A's performance across individual classes, which is observed to be substantial. For top1 accuracy, a quarter of the classes exhibit zero accuracy, resulting in a right-skewed distribution, as illustrated by the left histogram in Figure 4.4. Only a small fraction of classes perform above 75%, while many perform below 25%. On the other hand, top5 accuracy has a more symmetrical distribution, and the performance gap across classes is less pronounced, with the number of classes scoring above 75% being similar to the number of classes scoring below 25%. The right-skewed distribution of top1 accuracy is concerning and calls for further investigation. Sections 4.1.6 and 4.1.7 delve into the causes and seek to determine whether it is a fundamental limitation of I2A or a dataset-specific issue

Table 4.4: Performance of the I2A framework based on 4-fold cross-validation. The first fold (CV1) is the one used to build the model in Section 4.1.3.

	CV1	**CV2**	**CV3**	**CV4**
Top1	32.48	32.74	32.67	32.92
Top5	51.45	51.54	51.12	51.68

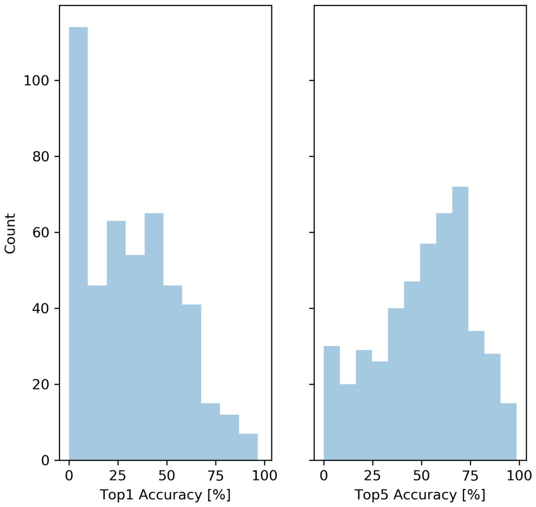


Figure 4.4: Histograms illustrating the distribution of class accuracy, where the x-axis represents accuracy ranging from 0% to 100% and the y-axis indicates the number of classes (frequency). **Left:** shows the distribution of top1 accuracy. **Right:** shows the distribution of top5 accuracy.

related to imSitu.

Additionally, we analyse the distribution of class predictions during evaluation to determine whether the I2A model shows a bias towards predicting certain classes. We observe that the model generates a skewed prediction distribution despite the test set having between 36 and 98 examples for each class. That is, a small number of classes receive a large portion of the predictions (e.g., four out of the 463 classes account for 12% of test predictions), while a large number of classes receive no predictions (38 classes, which constitute 8% of all classes). Table 4.5 shows the ten most frequently predicted classes. Importantly, the over- or under-prediction of certain classes is not attributed to high sensitivity to the number of training examples. This is evident from the training set, which does not exhibit the same skewness observed in the prediction distribution. Additionally, a weak correlation (r-value=-0.03) between prediction frequency and training size further supports this finding. Nonetheless, the prediction distribution remains a fundamental underlying cause of the performance variation between

Table 4.5: The I2A model's ten most predicted classes, including the number of predictions made for each class, as well as the corresponding number of examples in the evaluation set.

Class	Predictions	Eval-Set	Class	Predictions	Eval-Set
1. RUBBING	1308	68	6. SMILING	245	64
2. PUTTING	786	97	7. THROWING	241	93
3. TALKING	603	64	8. REPAIRING	239	87
4. FLINGING	397	83	9. JUMPING	222	58
5. GLUEING	286	71	10. ARRESTING	220	54

classes observed earlier, particularly given the evaluation metric we have adopted.

4.1.6 Understanding Model Failures

To gain insights into the performance of the I2A model, it is important to conduct a thorough analysis of its failures. This section presents the results of a confusion analysis, an investigation into the impact of polysemous classes, and a misprediction analysis, all aimed at identifying areas where the model can be improved.

Confusion analysis. With a dataset as large as imSitu, displaying a confusion matrix of size 463×463 can be overwhelming. Instead, in this section, we present a more focused approach to identify the most confusing classes. Specifically, we focus on the top 12 classes that are mistakenly predicted as true classes, along with the corresponding true classes, as presented in Table 4.6.

To determine which classes are the most confusing, we use a criterion where a class (p) is considered confusing if the model predicts it incorrectly as the true class (t) for more than 20% of the images belonging to class t. For instance, if the evaluation set has 60 images for class t, and 15 of them are incorrectly predicted as class p, then we consider class p to be a confusing class. By using this approach, we can provide more targeted insights into the most common classification errors made by the model.

The confusion table reveals that RUBBING is the most confusing for the model, with 26 other classes mistakenly predicted as RUBBING. Many of these classes lack any apparent semantic or visual correlation with RUBBING. Even when considering that some of these classes may in fact share similarities or contextual relevance with RUBBING, it is unclear why the model is fixated on predicting RUBBING instead of a more appropriate related class.

The second most confusing class for the model is TALKING, with 13 other classes mistakenly predicted as TALKING. Some of these classes, like SAYING, are near-synonyms, while

Table 4.6: The top 12 most confusing classes (p), where the I2A model frequently confuses class t with class p. The first row shows that the model mispredicts a significant proportion of images from 26 classes as RUBBING.

Predicted class (p)	**True classes (t)**
RUBBING (26)	GIGGLING, YAWNING, WINKING, WHISTLING, SNIFFING, POKING, PUCKERING, LICKING, BITING, NIPPING, TILTING, IGNORING, TICKLING, LATHERING, SCRATCHING, SMEARING, WIPING, COVERING,PINCHING, SLAPPING, STROKING, YANKING, PERSPIRING, PATTING, NUZZLING, CLINGING
TALKING (13)	SCOLDING, IGNORING, LECTURING, TRAINING, SAYING, SPEAKING, ENCOURAGING, ASKING, COMMUNICATING, DISCUSSING, DISCIPLINING, NAGGING, REASSURING
ARRESTING (6)	DETAINING, HANDCUFFING, FRISKING, RESTRAINING, APPREHENDING, SUBDUING
SMILING (5)	WRINKLING, CRYING, SQUINTING, WINKING, GRINNING
TILLING (5)	HOEING, FARMING, MOWING, PLOWING, SOWING
CHEWING (4)	GNAWING, EATING, PUCKERING, BITING
GIGGLING (2)	POUTING, DROOLING
REPAIRING (2)	INSTALLING, FIXING
THROWING (2)	PITCHING, FLINGING
PEDALLING (2)	RIDING, BIKING
PHONING (2)	CALLING, TELEPHONING
RECOVERING (2)	RECUPERATING, AILING

others convey variations in the action's interpretation, such as DISCUSSING, NAGGING, and ENCOURAGING. We argue that the model's predictions for many instances of these cases are correct, even though they contradict the ground truth labels. The same observation applies to other classes in the table, such as ARRESTING, TILLING, REPAIRING.

Eight out of the 12 classes shown in the table are partially attributed to either near-synonymy or different levels of related meaning. It is worth mentioning that imSitu contains many semantically similar classes, posing difficulties for both the textual modality (lexical embeddings of target labels) and the visual modality (image feature extractor), as synonymous classes tend to have visually similar images as well as similar lexical embeddings. The remaining three classes (SMILING, CHEWING, and GIGGLING) highlight a distinct challenge as they resemble distinct facial expressions that the image feature extractor may struggle to capture useful features for. Further discussion on these issues, particularly with respect to class RUBBING and the three facial expression classes, is provided in the following paragraph.

Visual aspects of the most confusing class. Class RUBBING by itself manifests many challenges. Firstly, it is a motion-based action, which makes it difficult to detect from still images. Secondly, it lacks any clear association with particular objects, which affects the image feature extractor's ability to extract visually discriminative features. Furthermore, the image feature extractor used in the I2A model is a CNN solely trained on the ILSVRC dataset, consisting of objects in canonical views. As a result, the feature extractor struggles to distinguish subtle differences among contact-based actions (e.g., LATHERING, TICKLING, and SCRATCHING) or facial expressions (e.g., GIGGLING and WINKING). The feature extractor's inability to produce distinctive features for these classes leads us to believe that it sees no difference between them, and consequently merges a significant portion of them into the class RUBBING.

Performance of polysemous classes. Lexical ambiguity is a significant factor that may contribute to the confusion observed earlier in this section. To identify potentially polysemous classes in imSitu, we can count their number of senses in WordNet and select those with more than the average number of senses for imSitu classes, which is 5.6. Among the top 12 confusing classes from Table 4.6, only three meet this criterion: TALKING,THROWING, and RECOVERING. We also explored the relationship between polysemy and class accuracy. On average, polysemous classes (i.e., those with more than six senses) scored 26.93% for top1 accuracy, compared to 34.31% for non-polysemous classes. This difference is more pronounced in extreme cases, where classes with more than ten senses averaged a score of 23.94%, while classes with less than three senses averaged 37.65%.

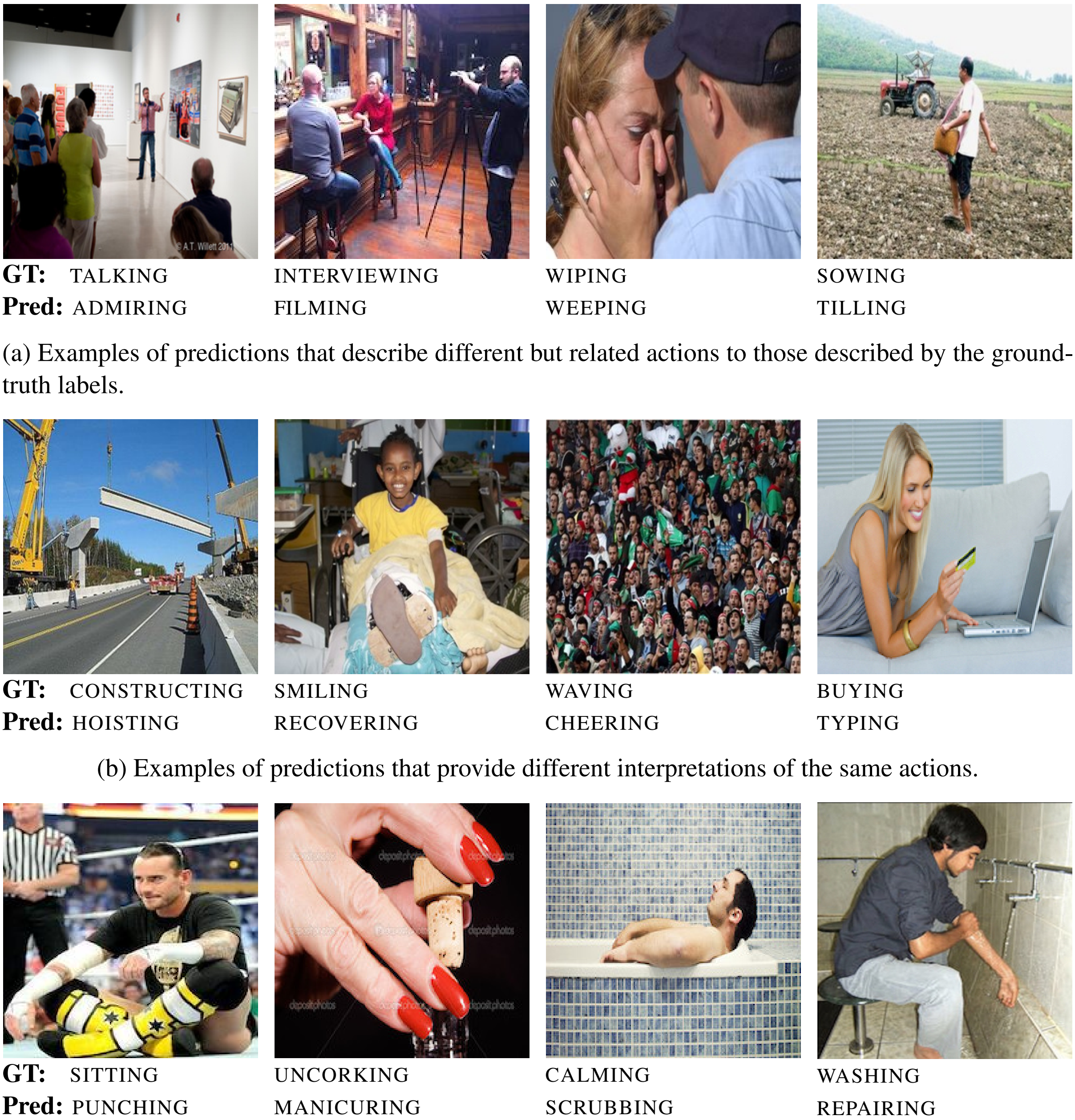
Top1 misprediction analysis. In the rest of this section, we will present an instance-based analysis of the model's incorrect predictions. In the first analysis, we randomly sampled 200 images from the incorrect top1 predictions to determine if any of these mispredictions can be considered as "plausibly correct". We consider a prediction to be plausibly correct if it (a) describes a related but different action in the image, such as an image showing a person CRYING while be comforted could also be labelled as COMFORTING, or (b) suggests a different meaning of the action, such as an image labelled as MOURNING being appropriately labelled as CRYING. Most of the observed instances fall under the second category, which is highly relevant to our thesis. We found that 27% of the incorrect top1 predictions could be considered "plausibly correct" despite not being ground truth according to the dataset annotation. This implies that if we were to apply a different evaluation approach, the I2A model's top1 accuracy could potentially increase to 50.7%. Figure 4.5 shows a sample of mispredictions where (a) and (b) provide examples of the two categories.

Furthermore, we conducted an additional experiment by selecting 200 images with incorrect top1 predictions, but correct top5 predictions based on the ground-truth labels. The goal of this experiment was to identify better-quality predictions. Our analysis revealed that 36% of the predictions in this sample had top1 predictions that were considered "plausibly correct". While this percentage represents an increase compared to the previous sample, where only 27% of the predictions were plausibly correct, we had hoped for a more substantial increase to strengthen our argument about generating multiple levels of meaning. Nonetheless, our analysis suggests that the I2A model has the ability to generate high-quality predictions beyond just the top1 prediction, indicating its potential to capture various levels of meaning.

4.1.7 Discussion

The purpose of this section is to investigate the possible reasons behind the findings presented in Sections 4.1.5 and 4.1.6. We will delve into four potential factors that could contribute to the observed results. Two of these factors are related to the distributed representation of the target labels, while the remaining two are associated with the visual feature extractor and the imSitu images. Through a detailed analysis of these factors, we aim to gain a better understanding of the underlying mechanisms that affect the outputs obtained in the previous sections.

The effect of lexical embeddings on accuracy. To explore the impact of lexical embeddings on accuracy, we observe the position of three groups of classes: easy, hard, and regular. Easy classes are those that the model predicts accurately with high performance, while hard classes



(a) Examples of predictions that describe different but related actions to those described by the ground-truth labels.

(b) Examples of predictions that provide different interpretations of the same actions.

(c) Examples of incorrect predictions that suggest spurious relationships between verbs and objects.

Figure 4.5: A sample of mispredictions. We argue that the ones in (a) and (b) are acceptable verbs. In (c), I2A seems to learn a false correlation between certain objects and classes.

are those that the model fails to predict correctly. Regular classes fall between these two extremes. We define a class as hard if it scores $<= 1\%$ on top1 accuracy and $<= 15\%$ on top5, and a class as easy if it scores $>= 70\%$ on top1 and $>= 85\%$ on top5.

Our analysis aims to determine if there are specific characteristics related to how hard classes are positioned in the embedding space that could be the possible cause of poor performance. To visually explore these relationships, we generate two visualisations of imSitu classes using UMAP (McInnes et al., 2018), with classes coloured according to their category. UMAP, or Uniform Manifold Approximation and Projection, is a technique for reducing the dimensionality of complex high-dimensional data and making it more understandable through visualisation. Compared to other methods such as t-SNE (van der Maaten and Hinton, 2008), UMAP is preferred for its superior ability to preserve the global structure of embeddings. This should enable us to more accurately capture and comprehend the underlying relationships between data points in high-dimensional space, providing valuable insights into the nature of the embeddings.

Figure 4.6 depicts these visualisations, with the left plot optimised to show local relationships and the right plot showing global relationships between classes. The interpretation of the transformed space can be challenging, as the shape of the layout varies based on UMAP's hyperparameters. However, the figure aims to demonstrate how classes are placed relative to their neighbours, with the perplexity parameter controlling this relationship. The left plot is set to 4, highlighting the relationship with the four nearest neighbours, while the right plot is set to 50, considering the 50 nearest neighbours.

The figure suggests that the positioning of classes in the embedding space may play a role in performance discrepancy between easy and hard classes. The global view in in the right plot suggests that hard classes occupy a different region of the space than easy classes, while the local view in the left plot shows that easy and hard classes can still be close neighbours. We illustrate our observation with the following example, class a could be the closest neighbour of class b, but the relationship is not mutual as class b has other 5 or 10 classes closer to it than a.

The goal of this analysis was to highlight the importance of considering the positioning of classes within the embedding space when evaluating model performance and suggests that the geometric properties of hard class embeddings may contribute to classification difficulties. The next analysis provides a further assessment of the issue and from a different perspective.

Hubness in the embedding space. This analysis studies the effect of *hubness* on class accuracy. Hubness refers to the phenomenon where certain embeddings, known as hubs, frequently appear in the top neighbour lists of many action-encoding vectors generated by the projection component of I2A. These hubs are believed to be located near many other classes in the space

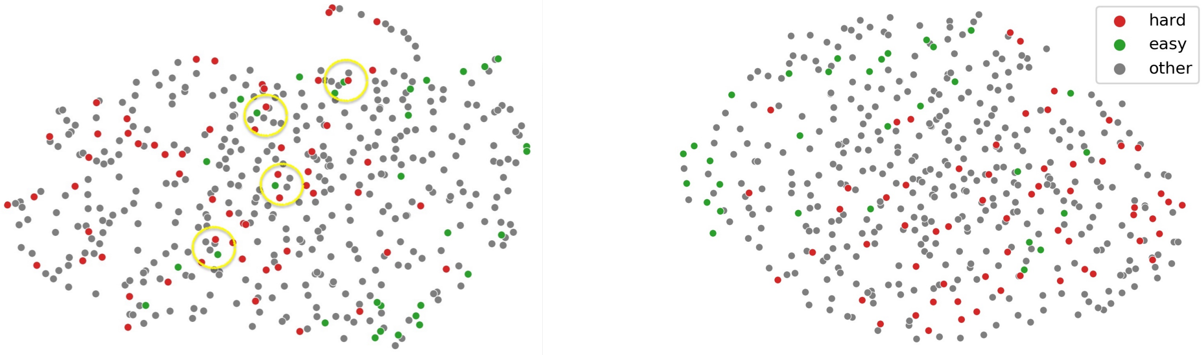


Figure 4.6: UMAP visualisations of imSitu classes. **Left:** shows a local view with a perplexity value of 4. **Right:** displays a global view with a perplexity value of 50. Noticeably, hard classes (in red) and easy classes (in green) often form distinct clusters, although a few exceptions are highlighted in yellow circles.

Table 4.7: Top most frequent classes appearing in the five nearest-neighbours (5-NN) of imSitu classes. The second column of this table indicates the number of times a class appears in the 5-NN list of other classes, while the third column shows the average cosine similarity score between the class and its neighbours from the second column.

Class	5-NN Appearance Frequency	Average cosine similarity
RUBBING	35	0.49
FLINGING	25	0.47
PUTTING	24	0.44
PULLING	23	0.47
SQUEEZING	20	0.45

of evaluation classes without having any meaningful similarity with them (Radovanović et al., 2010; Lazaridou et al., 2015). To assess the effect of hubness, we examine the relationship between hubs and confusing classes. We identify a class as a potential hub if it appears among the five nearest neighbours (5-NN) of a substantial number of other classes, and we use cosine similarity to measure the degree of proximity. Table 4.7 shows the most frequently appearing classes in 5-NNs, with RUBBING being the most frequent at 35 times, which is six times higher than if the embeddings were equidistant. This observation highlights the role of hubness in explaining why RUBBING is the most predicted class and the most confusing class, as shown in Tables 4.5 and 4.6, respectively.

Impact of the visual feature extractor's pretraining dataset. The impact of the dataset used for pretraining the visual feature extractor on the I2A model performance was investigated. Specifically, the feature extractor was solely trained on ILSVRC and no finetuning conducted, as discussed earlier in Section 3.2.3. As a result, the model may have relied on simplistic associations between actions and objects commonly observed in the imSitu dataset, such as tents being frequently associated with camping. To investigate this claim, an off-the-shelf ILSVRC classifier provided by PyTorch (Paszke et al., 2019) was employed to perform object classification on the entirety of the imSitu dataset. The classifier is trained to predict the most probable object among the 1000 ILSVRC classes. Subsequently, information theory techniques were employed to compare the classifier's predictions to the ground-truth of imSitu. Normalised entropy (Equation 4.4) was calculated for each class to measure the correlation between the imSitu ground-truth labels and the ILSVRC classifier predictions.

$$H_c = \frac{-\sum_{i=1}^{n} P(x_i \mid c) \times \log_2 P(x_i \mid c)}{\log_2 n} \tag{4.4}$$

where:
$p(x_i)$: is the probability of predicting object x_i for given the imSitu class c.
n: is the number of ILSVRC object classes.
c: is a given imSitu class.

Table 4.8 present classes with the lowest entropy values, indicating a strong correlation with a small number of ILSVRC classes, sometimes as low as one. For example, a closer look at the classifier's predictions for BALLOONING revealed that the majority of class images are classified as a *balloon* (an ILSVRC class). Specifically, 338 images (out of 352) were classified as a *balloon*, nine as a *parachute*, and one each for the following: *go-kart*, *maraca*, *slot*, *cliff*, *coral reef*, *geyser*. In contrast, SNIFFING exhibited the highest entropy, with its images being classified (or rather guessed) as 212 different ILSVRC objects, with each object being predicted only a few times.

Table 4.9 shows the top 10 performing classes according to F1 score achieved by the I2A model for verb prediction. The results of this table, along with Table 4.8, suggest a strong association between F1 scores and normalised entropy values, with classes that rely on specific objects demonstrating stronger performance according to the F1 metric. Nevertheless, while the observation holds for extreme cases, as shown in the tables, the correlation between entropy and F1 across all classes is weak (r-value = -0.06 and p-value = 0.23).

Image and annotation quality. The imSitu dataset is available in two versions, as discussed in Section 3.3. The first version contains the original high-quality images, while the second version consists of resized images with smaller dimensions and file size (approximately 10% of the original size). The I2A framework utilises the resized version of the imSitu dataset, where images are resized to 256×256 pixels. This resizing allows for more efficient training

Table 4.8: Lowest-entropy classes.

	Class	Entropy
1	BALLOONING	0.02
2	PITCHING	0.08
3	PARACHUTING	0.09
4	RAFTING	0.09
5	SKIING	0.09
6	LAUNCHING	0.14
7	MOPPING	0.14
8	CAMPING	0.15
9	BIKING	0.15
10	MOWING	0.15

Table 4.9: Top-performing classes according to F1.

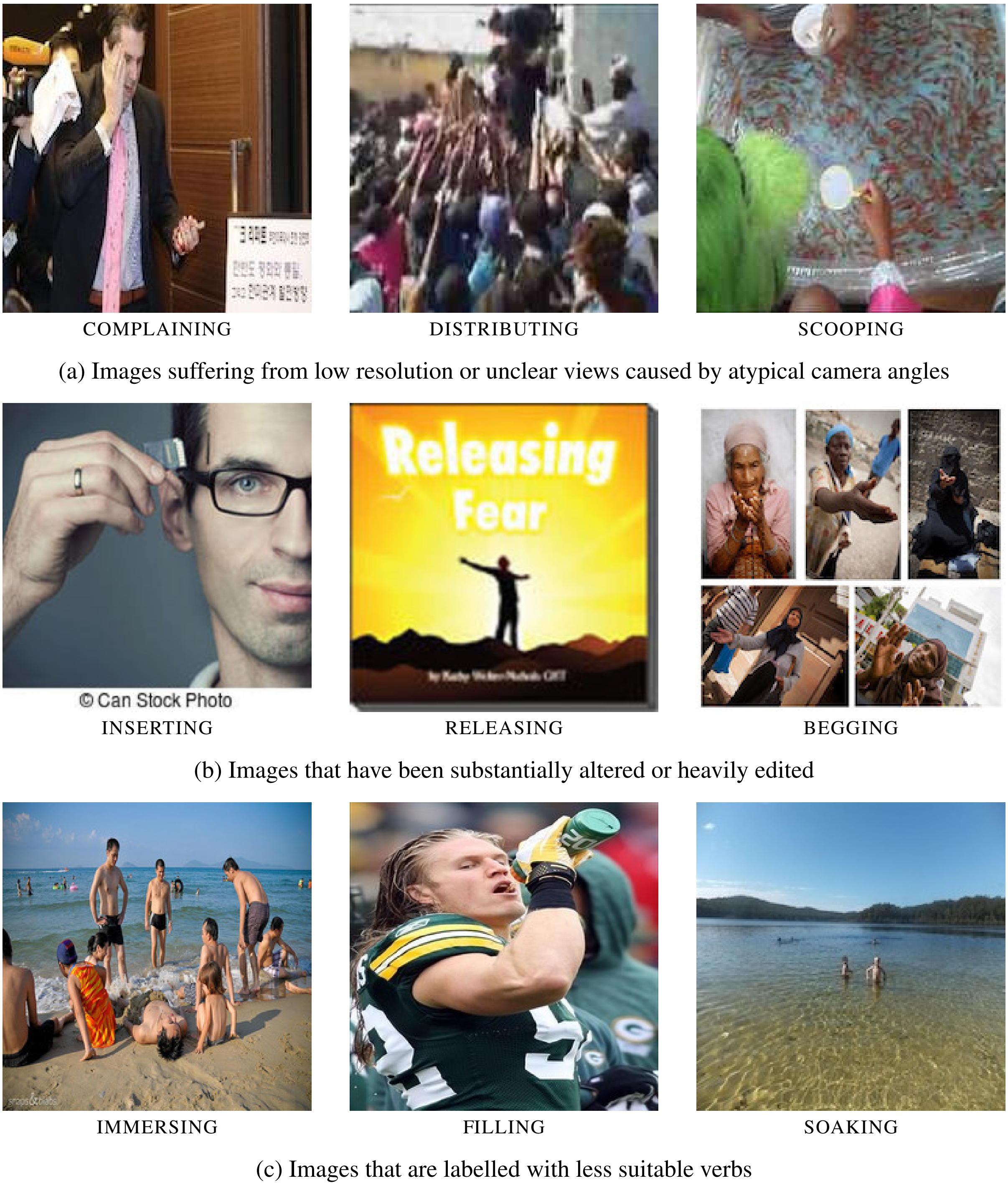
Class	F1
BALLOONING	94.97
STINGING	94.62
LAUNCHING	93.02
SHEARING	89.33
SURFING	88.61
RAFTING	88.11
CAMPING	83.20
PILOTING	79.21
DRUMMING	79.17
ROWING	78.26

and compatibility with existing CNNs. However, it has negatively affected the quality of some images, as seen in the DISTRIBUTING image in the top row of Figure 4.7.

Despite our attempts to train the I2A model on the original high-quality dataset, we faced technical difficulties with the image feature extractor of I2A. The CNN used in I2A is pretrained on ImageNet images of size 224×224, which limited its ability to process larger high-quality images in the dataset.

Another issue that we have observed in the dataset is the presence of watermarks in critical areas of some images, which can hinder the detection of certain elements. For instance, a copyright watermark in the shape of a tennis ball placed between two tennis players during a match, or a watermark on a person's face, which makes it difficult to detect facial expressions accurately. Another issue is the use of composite images, such as the BEGGING image in the second row of the figure, which is a collage of multiple images. Additionally, some images have undergone extensive editing, as seen in the RELEASING image.

A third issue we identified in the imSitu dataset is related to the selection of verbs used as target labels. As illustrated in the third row of the figure, some of the target verbs appear unnatural or indirectly related to the actions depicted in the image. For instance, describing the action of drinking as filling, as shown in the FILLING image, is not intuitive, regardless of how it is performed. This issue can result in a mismatch between the target label and the actual action in the image, which can significantly affect the performance of models trained on this dataset. Therefore, it is crucial to carefully choose target labels that accurately represent the actions taking place in the images. This emphasises the need for a different annotation process that prioritises the selection of appropriate target verbs to ensure they accurately describe the actions in the images. Nevertheless, these image quality issues should not be concerning as they account for approximately 6% of the first sample of mispredicted images.



(a) Images suffering from low resolution or unclear views caused by atypical camera angles

(b) Images that have been substantially altered or heavily edited

(c) Images that are labelled with less suitable verbs

Figure 4.7: A sample of mispredicted images that appear to suffer from quality issues. The identified issues are presented in three categories, with samples of each category shown in a separate row.

Table 4.10: a set of troponomy-based queries performed on the default word embedding model (GoogleNews model) to extract specific verbs. These queries were conducted using the "most_similar" function available in the Gensim library (Řehůřek and Sojka, 2010).

Query	Output	Similarity score
`nibble to eat is ____ to walk`	stroll	0.56
`nibble to eat is ____ to drink`	sip	0.56
`snore to sleep is ____ to talk`	prattle	0.50
`nibble to eat is ____ to talk`	chit chat	0.42
`weep to cry is ____ to laugh`	chuckle	0.56
`lisp to talk is ____ to walk`	awkward gait	0.45
`nibble to eat is ____ to yell`	shout	0.55

4.2 Zero-Shot Setting

In the previous section, we introduced a supervised learning approach that involved training a model to learn to describe actions using a finite set of verbs. However, this approach has a significant limitation as it restricts the ability to describe image actions to only the set of verbs the model has encountered during training. As discussed in Chapters 1 and 3, this poses a challenge as it is impractical to have training data for all possible interpretations of an image action.

Chapter 3 advocates for zero-shot (ZS) learning as a valuable technique for predicting various action meanings beyond what the model was explicitly trained for. I2A has the capability to explore a space that encompasses verbs not encountered during training, and it can in theory conduct algebraic operations to perform a guided search for extracting specific interpretations of the actions. Table 4.10 offers a glimpse of the queries that can be employed to extract verbs that describe the "how" using troponomy relations.

This section presents various ZS experiments, in which classes are divided into training classes (or seen classes) and test classes (or unseen classes). Four methods of class sampling are used: random, random with replacement, cluster-based, and sampling based on semantic properties of verbs.

4.2.1 Experimental Setup

The sampling method plays a crucial role to any machine learning task, particularly in the ZS learning setting (Xian et al., 2019). In this section, we organise our experimental setup based on the sampling method to emphasise its importance in the ZS setting. Section 4.2.2

discusses random sampling, a standard technique for any learning task, including the ZS learning. Section 4.2.3 discusses random sampling with replacement to investigate the performance on certain test classes when the the model is trained on slightly different training sets. Section 4.2.4 presents a cluster-based method to establish performance boundaries. Section 4.2.5 introduces a method that uses semantic properties of verbs to identify the classes on which I2A performs well and the characteristics of these classes.

For the hyperparameters, we use the same settings discussed earlier in Section 4.1.1. The 1cycle policy (Smith, 2018) is applied, with one cycle for 20 iterations or epochs. The model has approximately 34 million parameters, of which only 11 million are trainable. All of the trained parameters are in the projection component.

4.2.2 Random Sampling

Random sampling is a fundamental technique in machine learning used to randomly select test samples. However, relying solely on random sampling may not always provide reliable evaluations of a model's performance. To enhance the reliability of the evaluation, it is recommended (James et al., 2013) to use the k-fold cross-validation (CV) procedure. This involves dividing the dataset into k equal-sized folds, where each fold contains randomly selected instances from each class. For each of the k folds, a model is trained on instances of the remaining $k - 1$ folds and tested on the current fold. This process is repeated for each of the k folds, so that each instance in the dataset is used for testing exactly once. of the dataset. The final score is calculated by averaging the scores of the k models.

In this section, we use CV to split classes (not instances) into k folds, where each fold contains a unique set of classes. The choice of k is critical for obtaining reliable and statistically-sound evaluation scores. A commonly used value for k is 5 or 10, which have been shown to yield low error rate estimates (James et al., 2013). However, in the context of zero-shot learning, the choice of k poses additional challenges. Decreasing the number of folds increases the size of test classes, which can create difficulties for the search component as it needs to operate within a more expansive search space. For instance, in the extreme case of 2-fold CV, the search space contains 232 classes, whereas, in 10-fold CV, the space contains only 46 classes. Conversely, increasing the number of folds can be computationally demanding since the number of models grows with the number of folds.

To balance efficiency and complexity, we have chosen to set k to 7. This value is believed to maintain a healthy balance between the number of models (which affects efficiency) and the number of test classes (which affects complexity). Additionally, 7 sits between the two commonly used k values for cross-validation.

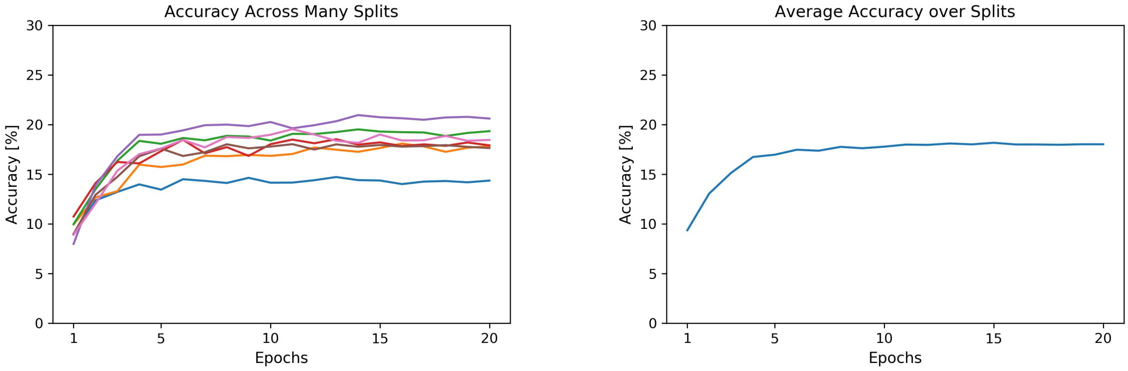


Figure 4.8: On the right, the plot shows the average accuracy of the seven splits over epochs. The graph indicates that the performance reaches a plateau early on, specifically at epoch 5. On the left, the plot illustrates the variation in performance across the seven folds. It is notable that most of the folds exhibit significant performance fluctuations over the epochs.

Results. The average performance for top1 and top5 accuracy over the seven folds is 17.92% and 41.39%, respectively. It is worth noting that the vis-proto and label-cls baselines presented in Section 4.1.2 cannot be directly compared to our results as they require supervised learning. However, we can compare the results to the random prediction baseline using the same equations from that section, keeping in mind that the size of target classes C_T decreases for the zero-shot experiment from 463 to 67 or 66. The top1 and top5 accuracy for the random prediction baseline are 1.5% and 7.7% respectively.

In Figure 4.8, the left plot demonstrates significant performance fluctuation across folds. The standard deviation for top1 and top5 is 2.10 and 3.10, respectively. The right plot indicates that the average performance plateaus at epoch 5, which is earlier than what we observed in the supervised setting discussed in Section $4.1.1$. The variance can be attributed to either the quality of the model, such as having an inadequate or appropriate set of classes for training, or the difficulty of the test classes in particular folds. Nevertheless, the experiment has prompted us to devise better sampling strategies, as we discuss in the following sections.

We also conducted a comparative analysis of our model's performance with the work of Zellers and Choi (2017), as presented in Chapter 2.4. To achieve this, we adopted their split of the dataset, with the test set consisting of 96 classes and all imSitu classes included without any filtering. Our model achieved a top1 accuracy of 16.85% and a top5 accuracy of 37.45% under this more challenging setup, considering the larger test set. Despite this, these results were still similar to the most comparable baseline in their paper, referred to as "DeVISE", which obtained a top1 accuracy of 16.50% and a top5 accuracy of 37.56%. This baseline employed a ResNet152 for image feature extraction that was fine-tuned on the imSitu training classes, along with GloVe embeddings trained on 840B tokens for target representation. By contrast, our model used comparatively less powerful components including an untuned ResNet50 and GoogleNews embeddings trained only on 100B tokens. In Section 5.4.1, we provide a comparison between our best performing model from Chapter 5 and the best performing model from their work.

4.2.3 Random Sampling with Replacement

In the previous section, we identified two potential causes for the variation in accuracy: the quality of the models across the seven folds and the varying difficulty levels of the test classes. In this section, we will focus on the first potential cause and investigate the performance of different models on specific test classes. To do so, we vary the models based on the training classes and observe their performance on these specific classes.

In contrast to the previous section, where we employed cross-validation to train and test our models, this section utilises a different methodology. Rather than dividing the dataset into seven folds, we conduct seven independent samplings of test classes. To illustrate, in the first sampling (iteration), we randomly select 67 out of the 463 imSitu classes as the test set and use the remaining classes to train the model. We repeat this process for the remaining iterations, allowing for the possibility of classes appearing in varying numbers of test sets.

Results. This section seeks to evaluate the consistency of I2A by constructing several models, each trained on a distinct set of classes and observing their performance on selected test classes. Table 4.11 provides examples of classes that exhibit extreme performance inconsistency (left) and those that demonstrate consistent performance (right). The standard deviation (SD) is used to measure the variability of individual class performance over several folds, and the calculation only includes classes that appear in multiple test sets, which is in this case amounts to 115 classes. The average SD of these classes is 4.67, calculated based on the absolute difference between their percentages.

The overall performance of this method is not reported, as it would be less reliable and based only on 310 classes, which are unevenly represented across iterations (i.e., some appear in two iterations, others in three, and a few in four). These results demonstrate the need for a deliberate selection of well-sampled and comprehensive training classes to achieve optimal performance in the zero-shot setting. Subsequent sections introduce systematic and semantically-driven techniques for partitioning classes into seen (training) and unseen (test) categories.

Table 4.11: A sample of classes divided into two groups based on performance consistency. **Right:** classes with the most consistent performance, either good or bad. **Left:** classes with the most inconsistent performance.

Inconsistent performance		**Consistent performance**	
Class	Top1 Accuracy [split1, split2, etc]	Class	Top1 Accuracy [split1, split2, etc]
BARBECUING	[0.79, 70.36, 67.59]	FILLING	[6.28, 6.28]
SOWING	[0.0, 71.05, 59.47]	LEANING	[1.81, 1.81]
WEEPING	[51.83, 23.39, 4.13]	FEEDING	[8.84, 8.84]
PEELING	[17.78, 10.67, 54.22]	TAPING	[0.44, 0.44]
GARDENING	[53.81, 17.26]	DIPPING	[0.00, 0.00, 0.27]
BOATING	[27.56, 13.78, 37.78, 52.00, 0.00]	JUMPING	[71.06, 70.64]
PATTING	[4.81, 40.06]	PERSPIRING	[2.79, 3.26]
POUTING	[25.26, 60.31]	IMMERSING	[0.00, 0.51, 0.00]
ASCENDING	[18.97, 52.82]	GRINDING	[5.90, 6.60]
SMILING	[55.86, 23.44]	DETAINING	[62.77, 63.64]

4.2.4 Cluster-Based Sampling

In the preceding section, we observed significant performance variation of I2A across different folds, and identified the generalisability of training classes as a contributing factor. In this section, we investigate the performance bounds of I2A and identify scenarios where upper and lower performance bounds can be achieved. Specifically, our analysis is based on the degree of generalisability of training classes. Training the model on a well-sampled and comprehensive set of classes is generally expected to lead to significantly higher performance. In contrast, inadequately sampled classes can present a challenging scenario for the model and result in substantially lower performance.

More concretely, this section utilises a cluster-based sampling technique that involves grouping classes that share similar features in the lexical semantic space and then randomly sampling test classes from each group. By using cluster-based sampling, we can select a well-structured and comprehensive set of training classes that accurately represents the entire domain. This approach aims to improve I2A's generalisability by ensuring that the model has learned from a diverse range of classes.

We employed spectral clustering (Ng et al., 2001) to group the classes into clusters. This choice was motivated by the solid performance of spectral clustering in extracting synonyms compared to other techniques, including k-means (Zhang et al., 2017). To ensure that the resulting clusters were coherent and balanced, we tested out several clustering techniques, and the spectral method produced the most satisfactory results. Specifically, the clusters generated

Table 4.12: Average performance of 10 models for each of the two clustered-based sampling techniques: sample-one and sample-whole.

Method of sampling test classes	Top1	Top5
sample-one: a class is sampled from each cluster	**20.36** (±2.79)	**44.78** (±2.58)
sample-whole: entire clusters are sampled	10.94 (±1.84)	32.64 (±3.01)
random sampling (Section 4.2.2)	17.92	41.39

by spectral clustering were well-populated, and the variance in cluster membership was smaller than in other techniques.

In our experiment, we grouped the classes into 67 clusters using spectral clustering. The number of clusters is meant to match the number of test classes used in previous sampling methods, which was determined by the choice of seven folds in the first type of sampling of this section. With the resulting clusters, we employed two different methods for sampling test classes: sample-one and sample-whole. The remaining classes were selected for training.

The ***sample-one*** method involves selecting one class from each cluster to be included in the test set. This method is intended to enhance the diversity of the training data, which in turn maximises the model's performance in the zero-shot setting. By using this method, the model can generalise better to unseen classes and achieve performance close to the upper limit of its capabilities. In contrast, the ***sample-whole*** method involves randomly selecting entire clusters of classes to be included in the test set. This method aims to evaluate I2A's ability to generalise to verbs from domains that are entirely distinct and unrelated to those encountered during training. It is regarded as a worst-case scenario and is designed to test the model's capacity to handle out-of-domain verbs. For example, this method may test the model's ability to predict cooking-related verbs none of which were included in the training set.

The results of our experiment, shown in Table 4.12, demonstrate that both models performed as expected. Sample-one outperformed random sampling significantly, while sample-whole performed significantly worse. We based our results on ten runs, each with its unique sample. These findings emphasise the importance of thoughtful sampling and the significance of acquiring a diverse range of training classes.

4.2.5 Sampling Based on Semantic Properties of Verbs

In the previous section, we proposed splitting classes into training and test sets based on clustering the lexical semantic embeddings of imSitu classes, where the goal is to train a generalisable model on a representative set of verbs. In this section, we suggest an alternative method of sampling that is based on the semantic properties of imSitu classes.

Throughout this thesis, we have argued that supervised learning cannot acquire all action meanings, as the visual modality can only learn a limited subset of actions through supervision. One way to approach this problem is to train a model on a representative set of classes, which we refer to as the core or basic classes. We hypothesise that a semantically core or basic set of verbs exists, which, if learned, would improve generalisation to unseen classes more effectively than randomly chosen verbs. However, to serve as a basis for predicting unseen classes, these core verbs must satisfy two conditions: (1) they must be core/basic/fundamental in some sense, and (2) they must be learnable and learned with enough accuracy to be deployed in the zero-shot setting.

Once the core verbs are identified and learned with sufficient accuracy, the model is tested on a secondary set of classes, which must also be composed carefully. This setup has the potential to yield the most significant benefit from I2A since the model is better equipped to generalise to new and unseen verbs. However, categorising each class as core or secondary requires a clear and robust definition of these categories, which may entail extensive research beyond the scope of our work.

In this section, we simplify the definition by considering certain properties from linguistic and psycholinguistic resources as estimators or proxies of the two categories. The section explores several methods based on established resources, where the core and secondary verbs are identified based on a single metric (i.e., a semantic property). These methods are not restricted to imSitu classes and can be applied to a broad spectrum of English verbs that are covered by these resources.

Lexical resources. Two resources are utilised for categorising classes into core and secondary: the MRC Psycholinguistic Databases (MRC) (Coltheart, 1981), and WordNet (Miller et al., 1990). MRC provides 21 linguistic and psycholinguistic attributes for 150k words, including 30k verbs. However, the coverage of some attributes is limited, e.g., meaningfulness covers only $1,504$ words. The most relevant attributes for our task are concreteness, imageability, and familiarity. We choose to focus on just one attribute – ***concreteness*** – since these three attributes are highly correlated with each other (Wilson, 1988), and working on all three may be redundant. The second lexical resource is WordNet (Miller et al., 1990), which also has a broad coverage of verbs (approximately 11K). WordNet is a lexical database that captures semantic relations between words, such as synonyms and hypernyms. In this section, Wordnet is utilised to rate the ***specificity*** and ***ambiguity*** of verbs.

We also considered other resources such as ACT-FASTaxonomy (Thornton and Tamir, 2022), which proposes six psychological dimensions for describing actions. However, we

Table 4.13: The 67 most concrete imSitu classes and the 67 least concrete classes according the the concreteness attribute in the MRC dataset.

Most Concrete	**Least Concrete**
chisel, coach, train, bite, sneeze, whip, cart, rock, pin, tie, tear, pump, shiver, paint, lecture, vault, weed, duck, prune, whistle, frown, tape, smile, pedal, shell, crown, write, board, sting, kiss, plant, farm, embrace, telephone, parade, pot, rake, wax, seal, water, milk, harvest, record, ski, fish, arch, bandage, drink, dock, swim, phone, button, chop, camp, eat, wheel, shop, nail, buckle, spray, fall, photograph, skate, plunge, drum, brush	sell, smear, mash, crush, build, lean, break, whirl, nag, help, interview, pray, read, leap, destroy, measure, call, sprint, pour, ignore, put, haul, count, discipline, type, trim, clean, rest, work, pat, turn, hang, imitate, attack, fill, empty, speak, protest, aim, smash, gasp, open, wait, bother, pull, bet, buy, vote, beg, carry, crawl, assemble, teach, hunt, weigh, repair, calm, throw, fall, stare, bury, give, plunge, drop, make, admire, ail, till

found these dimensions difficult to use for our purpose and the scores of these dimensions are somewhat unintuitive. Furthermore, our preliminary experiments with these dimensions failed to yield clear conclusions either supporting or negating our thesis statement. Consequently, we made the decision to exclude this resource from our experiments in this section.

In the rest of this section, we report on three experiments based on three properties of verbs (obtained from WordNet or MRC): *concreteness*, *specificity*, and *ambiguity*. The key point of these experiments to build two models where their test sets exhibit contrasting features determined by the selected attribute (e.g., one model tested on the most concrete and the second model is tested on the least concrete).

Concreteness. In this experiment, we use the concept of concreteness as it pertains to image-based tasks, which has been defined in multiple ways in literature. Two main interpretations of concreteness exist: one that relates to the ability to be experienced by the senses, and another that relates to specificity (as opposed to generality) (Spreen and Schulz, 1966).

In this experiment, we employ the concreteness attribute from MRC, which defines concreteness based on the first interpretation. Therefore, verbs denoting to actions that can be perceived through our senses (such as sight, sound, touch, taste, or smell) are rated as highly concrete, while verbs referring to actions that cannot be directly experienced are rated as less concrete. The concreteness rating scale ranges from 100 to 700. However, it should be noted that only 221 classes of imSitu have a value for this attribute in the MRC. In this experiment, classes are sorted based on the concreteness rating, and the 67 most concrete classes are selected to form the test set for the first model (referred to as most-concrete). For the second model, the 67 least concrete are chosen to be the test set (referred to as least-concrete). Table 4.13 shows the classes of the two samples.

Specificity. In the second experiment, we adopt the second interpretation of concreteness, namely, specificity. To operationalise this concept, we utilise WordNet (Miller et al., 1990) to determine the path length from each verb to its root, which serves as a measure of its specificity. WordNet organises concepts in a hierarchical structure, resembling a tree. Verbs located at the lower levels of the hierarchy and distant from the root are typically more specific in meaning, while verbs located closer to the root are more generic.

In this experiment, the classes are sorted according to their path length values, and the 67 imSitu classes with the longest path lengths are selected to form the test set for the first model (referred to as most-specific). Conversely, for the second model, the 67 classes with the shortest path lengths are chosen to be the test set (referred to as least-specific).

Ambiguity. The third experiment uses ambiguity, which is also related to specificity in the sense that more specific verbs are less ambiguous. To quantify ambiguity, we use WordNet, which provides multiple senses for words to capture fine-grained distinctions between meanings. However, this level of detail may not be necessary for a visual-centric task.

In this experiment, initially sort classes by their sense count in WordNet. Then, we identify the 67 classes with the highest sense count and designate them as the test set for the first model, (the most-ambiguous). Conversely, we choose the 67 classes with the lowest number of senses as the test set for the second model (the least-ambiguous).

Results. Our experimental results, presented in Table 4.14, demonstrate intriguing patterns in the relationship between concreteness and model performance across both supervised and zero-shot (ZS) settings. Specifically, we observed that the most concrete classes achieved significantly higher performance than the least concrete classes in the supervised setting. However, the opposite was true in the ZS setting, where the least concrete classes outperformed the most concrete classes.

These findings suggest that the role of concreteness in model performance is contingent on the level of supervision provided. Our results also underscore the importance of considering the semantic information when predicting abstract concepts in zero-shot settings. The results indicate that abstract verbs receive less benefit from the supervised learning, and benefit greatly from the textual information when they cannot be observed during training (i.e., zero-shot).

Our results on specificity and ambiguity reveal a consistent pattern in their relationship to performance across both supervised and zero-shot (ZS) settings. We found that specific classes consistently outperformed generic ones, irrespective of the learning setting. Additionally, the

Table 4.14: I2A performance based on three sampling techniques

Method of sampling	Supervised		Zero-shot	
	Top1	Top5	Top1	Top5
most-concrete	**39.70**	**57.69**	15.19	37.11
least-concrete	22.32	42.98	**18.64**	**39.23**
most-specific	**36.38**	**55.46**	**20.39**	**43.26**
least-specific	29.83	48.73	16.24	39.92
most-ambiguous	24.22	44.44	14.36	33.37
least-ambiguous	**39.60**	**57.44**	**20.63**	**45.23**

least ambiguous classes consistently outperformed the most ambiguous ones. This finding aligns with the view proposed by Spreen and Schulz (1966) that specificity is more closely related to ambiguity than to concreteness.

4.3 Summary

In this chapter, we conducted preliminary experiments using I2A's default components, consisting of a CNN pretrained on ILSVRC as the image feature extractor, and target labels represented as embeddings from the GoogleNews embedding model. These experiments represent an important first step in assessing the potential of the I2A model using off-the-shelf components.

Section 4.1 reported on the supervised performance of I2A, and provided a set of analyses of the model's performance, including an in-depth examination of the classes that performed well or poorly, and an exploration of the factors contributing to these outcomes. It also provided a closer look at the imSitu dataset. Although the I2A model does not achieve the same performance as the direct label classification baseline, it still shows promise given the potential advantages of incorporating textual information to predict unseen verbs.

Section 4.2 covered a series of zero-shot experiments that vary in the sampling technique used to split classes into training and test sets. The objective of these experiments is to explore the most effective approach for sampling classes and to gain a better understanding of the performance limits of the I2A model. These experiments highlight the importance of careful sampling strategies in zero-shot learning scenarios and provide valuable insights into the factors that influence the model's performance.

In future work, we plan to extend the scope of our experiments by training the I2A model on the entire set of imSitu verbs and testing it on verbs outside the imSitu vocabulary. This task requires the development of a suitable method for sampling a set of English verbs and the implementation of an appropriate human evaluation technique. Nevertheless, the successful completion of this task would represent a major step forward in our understanding of the I2A model's capabilities and limitations.

Chapter 5

Varying the Representation of Target Labels

The previous chapter used semantic information in two ways: to represent the target labels and to select test classes. That chapter demonstrated potential promise in certain experiments, such as the sampling based on semantic properties (section 4.2.5). However, the usage of a distributed semantic representation (i.e., GoogleNews embeddings) for target labels yielded modest results, which could be attributed to the design of I2A or certain properties of the word embeddings used in Chapter 4. In this chapter, we investigate the latter and experiment with different ideas of creating distributed semantic representations for target classes (i.e., verbs). We believe that there are ample opportunities to improve the representation of verbs and, in turn, enhance overall performance.

The experimental setup follows the one outlined in Chapter 4, with the only change being the type of semantic embeddings. Google-news (the verb prediction (VP) model from Chapter 4) is used as a reference point or an additional baseline for experiments reported here.

Section 5.1 justifies the use of custom-made embeddings, explains the process of creating them, and examine I2A performance when using them. Section 5.2 proposes an efficient method of extracting verb-specific representations from existing resources (i.e., GoogleNews embeddings). Section 5.3 uses "zero semantic" embeddings to evaluate the effect of semantic information on the two learning settings. Section 5.4 introduces methods of processing existing semantic embeddings to make them more suitable for the verb prediction (VP) task. Section 5.4.1 highlights issues of the embedding space layout and suggests the use of a well-established technique for processing embeddings. Section 5.4.2 uses a method for enriching embeddings with lexical resources. Section 5.5 evaluates the effect of certain aspects of embeddings: the size of embeddings (Section 5.5.1), the size of training corpus (Section 5.5.2), and the genre of training corpus (Section 5.5.3).

5.1 Bespoke Embeddings

This section aims to improve the semantic representation of target labels by training verb-specific embeddings from scratch, which should be less ambiguous as they only contain verbal senses of the word. This stands in contrast to conventional embedding models which define vocabulary at the word level, resulting in conflation of different semantic concepts and compromised representation, known as the triangular inequality (Neelakantan et al., 2014). For example, the embedding of *plant* contains information about refineries and pollen, a suboptimal representation for either sense.

The ideal solution would be to work at the sense level when defining vocabulary. However, imSitu does not consider sense disambiguation, resulting in many polysemous classes. We considered enhancing the dataset with sense-level annotation and expanding the taxonomy but found it to be cost-prohibitive and leading to a substantial expansion of the dataset vocabulary (each class on average has six senses, based on calculations made using WordNet). Additionally, many of these senses have an imbalanced number of images, which is problematic, particularly for those already lacking sufficient images.

In order to solve this problem, we chose to focus on the part-of-speech level as it has been found to be an effective way for semantic disambiguation (Wilks and Stevenson, 1998). This solution requires no extra annotation to the image dataset and only requires specialised semantic embeddings. However, it should be noted that this approach does not address the inherent drawbacks of imSitu, which can be observed even with less-polysemous classes, such as PAINTING, where two distinctive visual senses are observed: (1) covering the surface with paint, and (2) composing a picture with paint.

5.1.1 Method

The objective of this experiment is to create distinct representations for verb instances, distinct from non-verb instances, so *play* as as a verb would have a different embedding than *play* as a noun. The hypothesis is that specialised representations for verbs will lead to better accuracy on the downstream task of verb prediction.

Two bespoke (or custom) embedding models are developed: one that specifically handles verbal instances (bespoke-tagged), and another that follows traditional methods as a control model (bespoke-nontagged). Both embedding models are built from scratch using the Continuous Bag of Words (CBOW) algorithm (Mikolov et al., 2013) and trained on the combination of an English Wikipedia archive[3] and Gigaword5 (Parker et al., 2011), a dataset of news articles written in English from specific news agencies.

The training process follows the one that used for training the GoogleNews embeddings with two exceptions: (1) it uses a smaller set of training text, and (2) it applies an additional step prior to training the bespoke-tagged model. The rest of this section provides more details on the training process of the two embedding models.

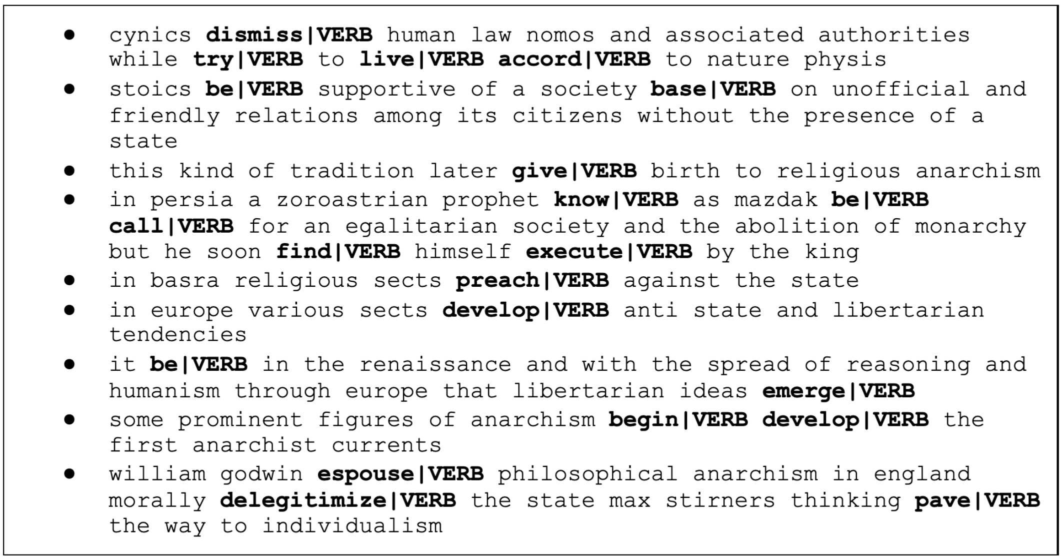
Bespoke-nontagged. The objective is to create a VP model based on the bespoke-nontagged embeddings that act as the control for the experiment, rather than using google-news for comparison against the VP model of bespoke-tagged. Both of the bespoke embedding models follow the same training procedure and are trained on almost identical corpora, with the training corpus for the bespoke-tagged being a slightly modified version of the corpus used for the bespoke-nontagged. The slight modification is due to the tagging process.

Bespoke-tagged. The Bespoke-tagged embedding model creates separate representations for verbs by treating verbal instances in the corpus as distinct from non-verbal instances. This approach is based on the work of Trask et al. (2015), which defines word types at the sense and entity level. To distinguish verbal instances, a pre-processing step is applied to the training text.

The pre-processing step works as follow: The text is tokenised into sentences using Gensim (Řehůřek and Sojka, 2010), and then run through the spaCy part-of-speech tagger (Honnibal and Montani, 2017) to detect verbal instances. If a a token is tagged as a verb, its surface form is converted into the base form, using the spaCy lemmatiser, and appended with a unique substring `|VERB`. For example, `eating` would be converted into `eat|VERB`. The purpose of lemmatisation is to increase training instances and improve the semantic representation (i.e., including all verb tenses yields more training examples than the base form alone). Figure 5.1 shows a sample text after applying this step, ready for training.

However, there is a possibility that this pre-processing step may negatively impact the quality of the embeddings, particularly due to issues with tagging and lemmatisation. The large spaCy language model (en_core_web_lg), which is used for POS-tagging, is reported to achieve 97.2% accuracy. Upon manual inspection of random excerpts, the tagging quality seems to be in line with the reported accuracy. However, issues prompted during lemmatisation are more problematic for embeddings' quality. These issues are caused mainly due to using two different lemmatisers. The first lemmatiser is of the spaCy language model and is applied during

[3] A snapshot archive is downloaded in 2019 using the following link: `http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`



- cynics **dismiss|VERB** human law nomos and associated authorities while **try|VERB** to **live|VERB** **accord|VERB** to nature physis
- stoics **be|VERB** supportive of a society **base|VERB** on unofficial and friendly relations among its citizens without the presence of a state
- this kind of tradition later **give|VERB** birth to religious anarchism
- in persia a zoroastrian prophet **know|VERB** as mazdak **be|VERB** **call|VERB** for an egalitarian society and the abolition of monarchy but he soon **find|VERB** himself **execute|VERB** by the king
- in basra religious sects **preach|VERB** against the state
- in europe various sects **develop|VERB** anti state and libertarian tendencies
- it **be|VERB** in the renaissance and with the spread of reasoning and humanism through europe that libertarian ideas **emerge|VERB**
- some prominent figures of anarchism **begin|VERB** **develop|VERB** the first anarchist currents
- william godwin **espouse|VERB** philosophical anarchism in england morally **delegitimize|VERB** the state max stirners thinking **pave|VERB** the way to individualism

Figure 5.1: A sample of the training text for the bespoke-tagged embeddings after applying the pre-processing step.

the tagging process. The second is a manually curated hash table that assigns imSitu classes (in the gerund form) to vectors (in “lemma|VERB” form), and it is applied before training the VP model. spaCy is also part of the second lemmatiser, but in conjunction with a manual verification and editing process. spaCy could not be used solely in the second case since it expects complete sentences to work accurately. Applying two different methods has created some discrepancies, but only affecting four classes:

- MOULDING: spaCy converts it to *mold*, whereas the manual method converts it to *mould*. Both `mold|VERB` and `mould|VERB` exist in the vocabulary of our trained model, and the similarity between them is high $\approx 86\%$ (in fact, the two vectors are the most similar to each other). The performance of this class is poor on the zero-shot setting, and we believe that averaging the two vectors could improve the performance given the low frequency of the two, especially `mould|VERB`, which has around 2k instances.

- SINGING and SWINGING: singing is lemmatised to either *singe* or *sing*. Both exist in the embedding model, and the similarity between them is high $\approx 88\%$ (both are the most similar to each other). The class performance is good, possibly due to the high frequency of `sing|VERB` ($\approx$ 200k). The same pattern is observed with SWINGING.

- CHEERLEADING: a deliberate decision is made to replace `cheerlead|VERB` with `cheer|VERB`. The training set has only 200 instances of `cheerlead|VERB` and upon inspection of the embedding via the similarity test, it has a poor representation of the actual action or activity.

Table 5.1: The performance of I2A when using bespoke embeddings and how it compares to the google-news from Chapter 4.

Model	**Supervised**		**Zero-shot**	
	Top1	Top5	Top1	Top5
bespoke-tagged	30.43	47.43	15.96	36.52
bespoke-nontagged	30.98	47.76	15.35	37.05
google-news (Ch. 4)	**32.48**	**51.45**	**17.92**	**41.39**

These issues can be resolved by standardising the lemmatisation process (e.g., using the NLTK library (Bird et al., 2009) whenever lemmatisation is needed), but since it only affects one class in the VP task, it was decided not to repeat the whole training process.

5.1.2 Results

The results of the experiment (Table 5.1) indicate that there is no significant difference in performance between the bespoke-tagged and bespoke-nontagged models, indicating that creating a separate representation for verbs does not yield better accuracy on the VP task. Furthermore, both bespoke models perform worse compared to the google-news model, suggesting that google-news has better semantic representations despite not having unique representations for verbs. The results also suggest that using only the gerund form is an effective technique for verb representation. The underperformance of the bespoke models may be due to a lack of semantic representation, potentially caused by the smaller training corpus used.

5.1.3 Discussion

Frequency count in the training corpus. This section analyses the differences in frequency counts among the three embedding models, with a focus on bespoke models that are trained on a smaller and less diverse set of text, primarily comprising of wiki and news articles. The limited size and diversity of the training corpus can result in insufficient counts of certain classes, leading to poor semantic representation of these classes. The data in Table 5.2 illustrates that the bespoke models have a disproportionate frequency distribution, where only a small number of classes have high counts and the majority have low counts. For instance, the class SAYING in the bespoke-tagged has the highest frequency among imSitu classes, with 40.7 million instances, while MANICURING has the lowest frequency with just 235 instances, showcasing the substantial variation among classes in the bespoke-tagged embeddings.

Table 5.2: Figures show the mean, median and standard deviation of token occurrence per class in the different training corpora.

Embeddings type	Mean	Median	SD
bespoke-tagged	332k	30k	2,007k
bespoke-nontagged	69k	11k	181k
google-news	2,958k	2,977k	64k

Correlation between class accuracy and frequency counts. A correlation analysis was conducted to examine the relationship between class accuracy and frequency counts. It was suspected that embeddings trained on a small number of instances would result in "semantically deficient" representations and poor performance in those classes. Kendall's tau was used to measure the correlation between frequency in the training corpus and class accuracy for the three models, using both top1 and top5 metrics, and for both supervised and zero-shot learning settings. The analysis showed a weak correlation between accuracy and counts in all 12 comparisons, with the only noticeable pattern being that the supervised setting had a slightly stronger negative correlation than the zero-shot. This means that classes with more counts in the corpus performed slightly worse in the supervised VP task. The reason for this is that having more training instances may force embeddings to be positioned in a challenging region of the embedding space. Section 5.4.1 discusses the geometry issue and proposes the use of a processing technique to mitigate this issue.

Overall, none of the three models displayed a strong correlation between class accuracy and its frequency in the training text. Some high-performing classes rarely occur in the training corpus (e.g., MANICURING occurs in 235 instances only). Also, some classes perform poorly despite having an abundance of examples in the training corpus (e.g., MAKING has 10 million instances but still scores 0% for top1 and top5).

5.1.4 Summary

Training embeddings on tagged text is highly inefficient since it necessitates training from scratch on a POS-tagged corpus. Tagging large amounts of text is a time-consuming and error-prone process. The method is contrary to the notion of utilising readily available resources for the visual and textual modalities. Nevertheless, the method is introduced to determine the potential of specialised embeddings. The following section presents a more efficient method for obtaining verb-specific embeddings.

5.2 Extracting Embeddings Using All Verb Forms

This section presents another approach to obtaining specialised embeddings for verbs. Instead of training embeddings from scratch on verbal instances as in Section 5.1, representations are directly extracted from pre-existing GoogleNews embeddings used in Chapter 4. This method considers all forms of verbs, instead of just using one form (the gerund) as in the previous chapter. For example, in order to represent the class TILTING, embeddings of all forms of the verb tilt (such as *tilting*, *tilts*, *tilted*, and *tilt*) are considered. This approach is more efficient and easier to implement, as it utilises existing resources and does not require additional training. More importantly, it is expected to produce embeddings with better representation, as it utilises powerful resources that are difficult to create from scratch; better representation ultimately should improve the performance of downstream tasks.

5.2.1 Method

imSitu classes are converted to their lemma forms using the NLTK lemmatiser (Bird et al., 2009). Each lemma is then used to retrieve all possible forms of the verb using pyInflect[4]. For example, class TILTING is converted to the lemma *tilt*, which is then used to retrieve the following set of forms: {*tilting*, *tilts*, *tilted*, *tilt*}. The embeddings for each form of the verb are then aggregated into a unified representation using the average pooling operation. This operation takes the mean of each dimension of the embeddings to create a single vector (Wieting et al., 2016; Adi et al., 2017). These embeddings are referred to as avg-verbs.

5.2.2 Results

The results of using avg-verbs show that it outperforms both the google-news and bespoke-based models, with a significant improvement in the zero-shot setting (as shown in Table 5.3). This indicates that utilising existing resources is more effective than building bespoke embedding models.

5.2.3 Discussion

This section demonstrated that using all forms of a verb, as opposed to just the gerund form, in pre-existing embeddings (GoogleNews) improves the performance of the downstream task. The size and variety of the training corpus for GoogleNews embeddings is a contributing factor

[4] `https://spacy.io/universe/project/pyInflect`

Table 5.3: The performance of our proposed model when using aggregates of all verb forms (avg-verbs) and how it compares to the google-news from Chapter 4.

Embeddings type	**Supervised**		**Zero-shot**	
	Top1	Top5	Top1	Top5
avg-verbs	**32.90**	**52.24**	**20.21**	**44.75**
bespoke-tagged	30.43	47.43	15.96	36.52
bespoke-nontagged	30.98	47.76	15.35	37.05
google-news (Ch. 4)	32.48	51.45	17.92	41.39

to the success of avg-verbs. We conclude that better semantic embeddings do, indeed, improve the performance of the VP model.

In future research, different weighting techniques can be applied instead of giving equal weight to all forms of the lexeme. One approach is to give higher weightings to certain forms based on the likelihood of them being verbs (e.g., *play* should have less weight than *played*). Another is to assign higher weights based on frequency in a specific corpus, preferably one that focuses on visual descriptions. Additionally, experimenting with cluster-based techniques or different pooling operations such as max-pooling could be studied, but it is doubted that it would make a significant difference in the downstream task.

5.3 Pseudo-Random Embeddings

In previous sections, we have explored the potential of using verb-specific target representation to improve I2A's performance. However, the results were inconclusive regarding the value of using a distributed lexical semantic embeddings to represent the target classes, especially under the supervised setting. Our findings in Section 5.1.2 indicate a weak negative correlation between class accuracy in the supervised setting and class frequency in the training text. In other words, performance using target embeddings of verbs with more instances in the training data is slightly worse that for verbs with fewer instances. We believe that the poor performance of models in Section 5.1.2 is not only due to inadequate semantic representation of verbs by the pretrained embeddings but also due to the positioning of embeddings within the embedding space. Later in this section and in Section 5.4.1, it will become evident that verb embeddings tend to cluster very much in the same space, which makes them hard to separate.

This section proposes the use of embeddings that do not contain any semantic information, i.e. which have not been pretrained on large textual resources, and investigates whether semantics is beneficial for the supervised setting. Additionally, this section will also validate the

Table 5.4: The table compares the performance of I2A using pseudo-random embeddings to other models and baselines discussed earlier in this thesis.

Embeddings type	**Supervised**		**Zero-shot**	
	Top1	Top5	Top1	Top5
pseudo-random	**36.54**	**56.58**	1.65	7.51
avg-verbs (Sec. 5.2)	32.90	52.24	**20.21**	**44.75**
google-news (Ch. 4)	32.48	51.45	17.92	41.39
label-cls (Sec. 4.1.2)	37.04	63.18	NA	NA

significance of semantics in the zero-shot setting.

5.3.1 Method

Zero-semantics embeddings are created using randomly generated numbers, similar to the approach used by Madhyastha et al. (2018), but with a greater emphasis on the distribution of random numbers. A set of 300-dimensional vectors is generated from a normal distribution with a maximum value close to 1, a minimum value close to -1, and an average mean near zero. The values of the vectors are designed to resemble the range of values found in GoogleNews embeddings but not the actual distribution. These vectors are arbitrarily assigned to classes and are referred to as “pseudo-random” embeddings. Unlike semantic embeddings, which have a geometry that reflects the syntax and meaning of words, the space of pseudo-random embeddings is free from any constraints, except for satisfying the normal distribution.

5.3.2 Results

In the supervised setting, the performance of the pseudo-random VP model surpasses all other embedding-based models as shown in Table 5.4. The performance for top1 is comparable to the upper-bound baseline (label-cls), but the zero-shot performance is equivalent to random guessing, which is in stark contrast to its superior performance in the supervised setting. These results confirm that semantic information is essential for predicting unseen classes in the zero-shot setting, but it can also slightly negatively impact performance in the supervised setting. The next section will delve deeper into this negative effect of semantic information on supervised performance.

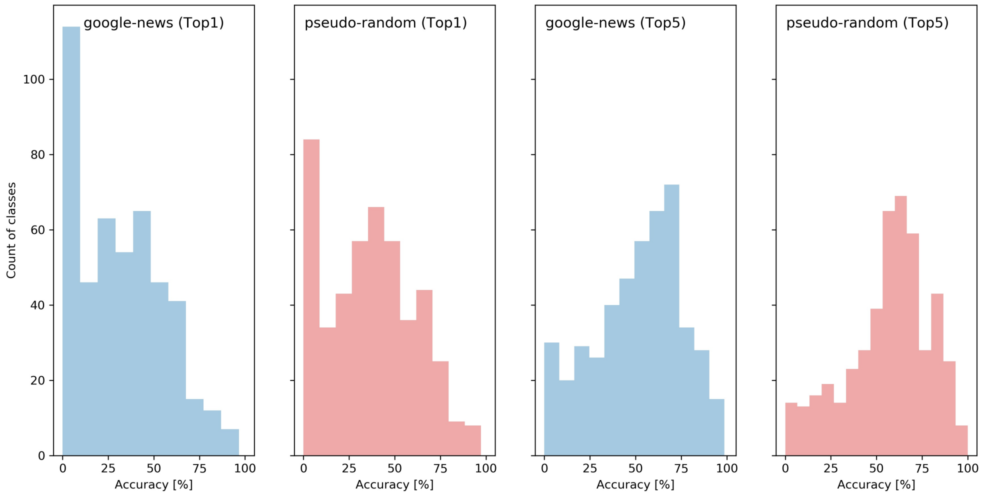


Figure 5.2: The histograms present a comparison of the distribution of class accuracy for two verb prediction models, google-news (blue) and pseudo-random (red). The x-axis displays accuracy values from 0% to 100%, and the y-axis shows the frequency of classes. The left histograms depict top1 accuracy, while the right histograms show top5 accuracy.

5.3.3 Discussion

In the supervised setting, the lack of semantic information has liberated embeddings from getting clustered together in a narrow region of the space, enabling embeddings to be appropriately distanced from one another. Comparing the performance of pseudo-random against google-news shows that pseudo-random performs better on more classes and has a more balanced distribution of class accuracy (Figure 5.2). This suggests that some characteristics of the semantic embeddings play a role in the variance of performance among classes. The following analysis will examine the effect of one of these characteristics on performance.

Embedding similarity. This analysis looks at the correlation between the cosine similarity of embeddings and performance in the supervised setting. More specifically, we measure the similarity of each class embedding with its k neighbours, and examines how this similarity relates to performance. The analysis includes all types of embeddings discussed thus far in this thesis.

The analysis shows that classes in pseudo-random are the least similar to each other, making them more distinctive and more effective for the downstream task. On the other hand, classes in bespoke embeddings are the most similar, which is believed to contribute to their underperformance. Additionally, the analysis finds that bespoke-tagged embeddings have higher

Table 5.5: Average similarity between class embeddings and their neighbouring classes.

Model	**1NN**	**Averaged 5NN**	**Top1 in Supervised**
bespoke-tagged	0.62	0.54	30.43
bespoke-nontagged	0.56	0.49	30.98
avg-verbs	0.53	0.46	32.90
google-news (Chapter 4)	0.52	0.45	32.48
pseudo-random	**0.17**	**0.15**	**36.54**

Table 5.6: The performance of I2A using pseudo-random embeddings of different sizes.

Vector Dimension	**50d**	**100d**	**200d**	**300d**	**600d**	**1024d**
Top1	34.07	36.13	36.73	36.54	36.69	36.04
Top5	46.50	51.94	55.31	56.58	57.11	57.65

similarities than non-tagged ones, likely due to tagged embeddings sharing a common characteristic (being all verbs), making them semantically more close in embedding space. The increase in similarity between class embeddings resulted in reduced performance; the opposite of the desired effect of creating the bespoke embeddings.

Table 5.5 shows that the less similar classes are in the embedding space, the better I2A performs in the verb prediction task. Specifically, it shows that pseudo-random embeddings have the lowest similarity with their nearest neighbours and that well-spaced embeddings improve performance by approximately four points. Section 5.4.1 will aim to tackle the problem of congested areas in the semantic space by introducing a straightforward processing method.

Embedding size. We now consider an experiment that investigates the effect of embedding size on performance in the supervised setting. Specifically, we look at how I2A performs when using larger or smaller embeddings. In addition to the original pseudo-random embeddings of 300 dimensions, five other embeddings of various sizes are generated: 50d, 100d, 200d, 600d, and 1024d. These sizes are chosen to reflect the commonly proposed ideal sizes in the literature (Yin and Shen, 2018).

The results in Table 5.6 indicate that larger embeddings improve performance in the supervised setting, especially for top5. However, performance starts to plateau at 200d for top1 accuracy. The results show that 300d, 600d, and 1024d perform comparably and suggest no significant gain to using a size greater than 300d for pseudo-random embeddings. This is likely due to the fact that the upper bound (label-cls) is approached, and no larger or better embeddings can overcome the limitations of I2A or the difficulty of the task.

5.3.4 Summary

The layout of classes in the embedding space affects the performance of the task in the supervised setting. When the embeddings are more distant or distinct from one another, the model performs better. This section has highlighted that the geometry of the embeddings is crucial for the supervised setting, and semantic information is fundamental for the zero-shot setting. The section also showed that using bigger pseudo-random embeddings does not result in significant improvement for the supervised verb prediction task. However, it should be noted that we cannot generalise this conclusion to semantic embeddings as they possess unique features.

5.4 Embedding Processing

This section covers two methods for modifying pre-existing word embeddings: removing dominant components and retrofitting. Both methods are efficient, require no additional training, and can be applied to various tasks and domains. Section 5.4.1 proposes removing dominant components of embeddings for the purpose of altering the geometry of the embedding space in a way that improves performance for verb prediction models. Section 5.4.2 explores the idea of supplementing embeddings with information from lexical resources.

5.4.1 Removing Dominant Components

Findings from Sections 5.1 and 5.3 support the idea that the geometry of semantic embeddings adversely affects the verb prediction performance in the supervised setting, particularly for verbs which occupy a narrower and densely populated region of the semantic space. Furthermore, embeddings contain information beyond just the semantic meaning of actions, such as syntactic characteristics and social bias (Andreas and Klein, 2014; Garg et al., 2018).

Since this thesis only focuses of a limited set of English verbs with a small portion of verbs in the English language, it may be beneficial to remove information shared across these verbs, such as being all verbs or irrelevant properties like word frequency (Bullinaria and Levy, 2012). We suspect that the similarity in one dominant aspect (e.g., being all verbs or being frequent) can undermine other crucial elements needed for distinguishing verb embeddings. Figure 5.3 illustrates how the embeddings of imSitu classes are pulled into a dominant direction and clustered in a small region of the space.

Method. The technique, adopted from Mu and Viswanath (2018), is used to improve the layout of the embedding space without hindering the semantic representation of individual embeddings. That is, fixing the issue of the global layout without damaging local and fundamental

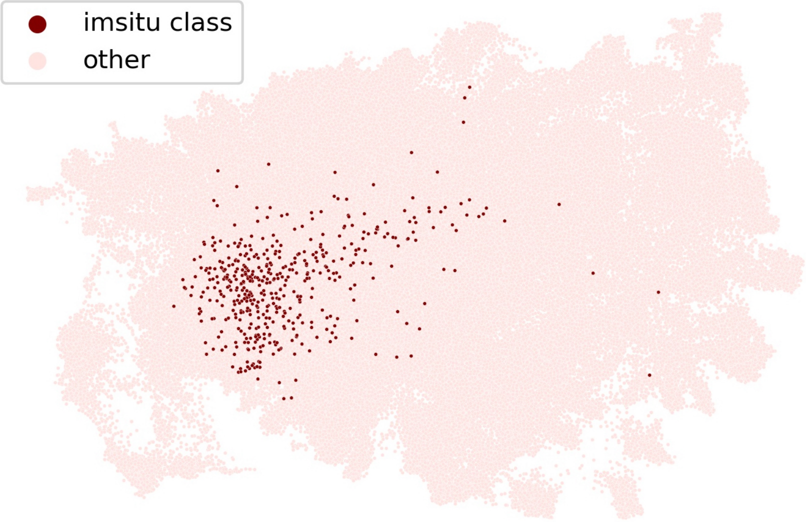


Figure 5.3: A plot that shows the position of imSitu classes (dark dots) relative to the 200k most frequent words in the GoogleNews model (light dots). Embeddings transformed into 2d vectors using the UMAP transformation technique (McInnes et al., 2018).

relationships between embeddings. The technique is applied to pre-trained embeddings; therefore, it requires no further training or finetuning. The technique consists of two steps: (1) subtraction of the mean average vector, followed by (2) eliminating the first D principal components.

The mean average vector is thought to contain shared information among vectors, which is irrelevant for highlighting individual characteristics of verb embeddings. The second step applies the principal component analysis (PCA), a dimensionality reduction technique, to identify dominant components. Unlike standard practice, which suggests removing the weakest components and views them as noise signals, this method removes the top components to improve performance. Mu and Viswanath (2018) have experimentally demonstrated that removing top components has a purifying effect and improves performance in several tasks.

Mu and Viswanath (2018) suggest that D should depend on the vector size and states that a good rule of thumb is choosing D to be the vector size divided by 100. For the experiment in this section, four values for D are considered: 1, 3, 10, and 20. Another parameter is the size of the matrix on which the calculations of the two steps are applied. Along with using the whole vocabulary of the original embeddings (i.e., GoogleNews), we experiment with two other matrices that are assumed to be more fitting for the downstream task:

- imSitu classes: A matrix that consists of the 463 class embeddings.
- 5K verbs: A matrix that contains nearly 5000 embeddings of verbs obtained from VerbNet (Schuler, 2006). We hypothesise that a more extensive vocabulary will improve the

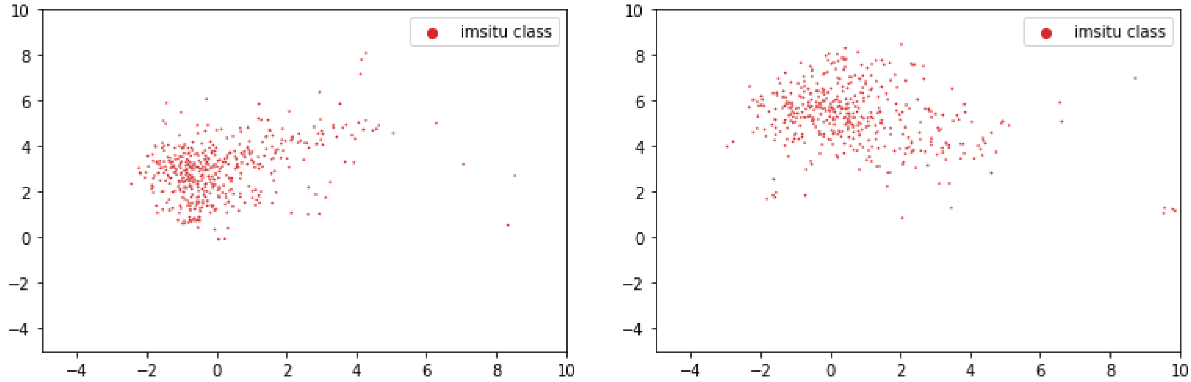


Figure 5.4: The effect of the method on the shape of imSitu embeddings before (left) and after (right), using the matrix of 5K verbs and removing the top 10 components.

processing quality and will assist in calculating a more accurate mean average and principal components. Figure 5.4 shows the effect of the method on imSitu embeddings, using the matrix of 5K verbs.

Results. The results in Table 5.7 indicate that using the matrix of the entire GoogleNews vocabulary has no impact on performance, as finding dominant components and an average vector for a matrix of three million embeddings is unlikely to be effective. On the other hand, using the matrix of imSitu classes not only has no effect in the supervised setting but also negatively impact the performance in the zero-shot, suggesting that this setup is removing important information from the embeddings. This reduction in semantic information is also reflected in the top5 scores for the supervised setting.

In contrast, using the matrix of 5K verbs improves performance for both learning settings, striking a balance between preserving relevant information and improving the geometry. The performance improves as the number of removed components, reaching its peak at $D = 10$ (referred to as ***processed-5K-10C***), and keeps improving for the supervised setting. In fact, it surpasses all other embeddings discussed in this thesis, except for pseudo-random performance in the supervised setting.

SOTA results. We also conducted a comparison between our best-performing model (processed-5K-10C) and the method proposed by Zellers and Choi (2017), as presented in Table 5.8. Their method combines distributional embeddings with learned attributes and/or gold attributes. Despite their use of gold attributes, which may not be a valid setup for zero-shot tasks, our processed-5K-10C outperforms their method when we adhere to their experimental setup. Their experimental setup uses the original dataset without any filtering and employed a different split for training and testing split.

Table 5.7: The performance of I2A using 12 variants of processed GoogleNews embeddings, which were created by varying the matrix of embeddings used for processing and the number of top (dominant) components removed.

Processing a matrix of	**Components removed**	**Supervised**		**Zero shot**	
		Top1	Top5	Top1	Top5
imSitu 463 classes	1	32.65	52.30	18.22	40.72
	3	33.62	53.02	18.08	38.51
	10	33.01	51.63	13.68	29.25
	20	33.26	50.58	10.20	22.94
5K verbs	1	32.23	51.76	18.0	41.18
	3	33.01	52.18	17.99	41.06
	10	34.20	55.08	**20.89**	**44.90**
	20	**34.26**	**55.20**	19.01	40.82
All GoogleNews vocabulary	1	32.65	51.65	17.87	41.11
	3	32.46	51.88	18.02	41.85
	10	33.01	51.63	18.16	41.68
	20	32.94	52.78	18.28	42.04
google-news (Ch. 4)	NA	32.48	51.45	17.92	41.39
avg-verbs (Sec. 5.2)	NA	32.90	52.24	**20.21**	**44.75**
pseudo-random (Sec. 5.3)	NA	**36.54**	**56.58**	01.65	07.51

The exclusion of variants in their method that rely on gold attributes resulted in achieving inferior results, which further emphasises the superior performance of our processed-5K-10C model model.

Discussion. The findings are promising, particularly when comparing our results with the technique's performance on certain intrinsic NLP tasks. Mu and Viswanath (2018) have reported that the technique improves performance by an average of 2.3% on multiple word similarity datasets and only an average of 0.4% on three analogy datasets. The gain in performance on the verb similarly dataset SimVerb-3500 (Gerz et al., 2016), which is more pertinent for our task, is only 0.15%.

An analysis, similar to the "Embedding similarty" in Section 5.3.3, is conducted to assess the impact of removing dominant components on the similarity between the embeddings of imSitu classes. The study measures how the average similarity changes as more dominant components are removed. The average similarity is computed for each variant of the processed embeddings in Table 5.7 (i.e., 12 averages) and calculated by mean averaging the pairwise cosine scores between all classes. The results indicate that the average similarity, monotonically decreases when removing top components, which is desirable as it allows for easier class

Table 5.8: Zero-shot performance comparison of the processed-5K-10C model and three variants of Zellers and Choi (2017). The processed-5K-10C model in this table differs from the one in Table 5.7 as it following the same experimental setup employed by Zellers and Choi.

Three models presented by Zellers and Choi (2017)	**Zero-shot**	
	Top1	Top5
Embeddings only	17.60	39.29
An ensemble of embeddings and predicted attributes	16.75	40.44
An ensemble of embeddings, predicted attributes, and gold attributes	18.15	42.17
Our processed-5K-10C	**22.08**	**43.74**

distinctions. Additionally, it is found that using only imSitu classes as the processing matrix results in a greater decrease in similarity compared to using 5K verbs or the entire GoogleNews vocabulary. However, overfitting to imSitu classes is not recommended as it leads to a loss of useful information, as evidenced by a rapid decline in zero-shot performance.

5.4.2 Retrofitting

This section introduces another post-processing technique, but for a different purpose: enriching word vectors with knowledge from lexical semantic databases. The section applies the seminal work of Faruqui et al. (2015), intending to answer questions about the effectiveness of retrofitting for improving the verb prediction task, the impact of the choice of lexical database, and the characteristics that make a lexicon suitable for a specific task.

Method. The technique produces a relational graph of the lexicon, where nodes are words and edges are relations (e.g., synonym). Machine learners are more accustomed to the graph format. The technique learns new word embeddings that satisfy the following conditions:

(i) Minimising the distance between the original and the new (inferred) embeddings. This condition ensures that inferred embeddings retain most of the information of the original embeddings.

(ii) Minimising the distance between inferred embeddings and their neighbours in the graph. This condition ensures that inferred embeddings reflect the relations in the lexicons and be enriched with semantic information.

The technique uses the graph structure to allow words to connect with their neighbouring words, collect information, and update themselves iteratively. The Euclidean distance is applied to measure the distance. The technique is efficient, requiring only ten iterations to converge, which takes only several seconds to run. In this experiment, we use a tool provided by the original authors, which requires two inputs: a lexicon, and a list of word embeddings to be enhanced. It is noteworthy that the graph is constructed exclusively from the specified set of word embeddings. The remaining part of this section presents experiments where embeddings are retrofitted to three lexical resources.

FrameNet. A manually curated linguistic resource that maps meaning to form through the theory of frame semantics (Ruppenhofer et al., 2006). In theory, FrameNet appears perfectly suitable for the verb prediction task since it is designed for action understanding. It contains schematic representations of events and situations, referred to as frames, evoked by a set of lexical units, including verbs. The meaning of these units can be derived from their frames and other units from the frame. For example, the meaning of *pay* requires an understanding of buying.

However, our primary concern is the size of FrameNet, as it contains only 10K lexical units. Additionally, FrameNet is also underutilised during graph creation, with the graph only expressing one type of relation where lexical units that evoke the same frame are connected. The graph omits the connections between frames, which the database already manifests in a lattice structure. Furthermore, not all frames are used; for instance, *compete* participates in two frames, but only one frame is in the graph. In our experiment, imSitu classes are lemmatised to improve coverage from only 27 to 377 classes.

Paraphrase Database. A collection of word pairs that have been automatically extracted from parallel corpora (Ganitkevitch et al., 2013) and are considered to be paraphrases of each other. The dataset assumes two words are a paraphrase pair if their translations to another language are the same. We believe that this dataset contains semantics that are already captured by word embeddings, which are known for their ability to perform analogy and similarity tasks.

WordNet. A rich linguistic resource in terms of the number of lexical units, the number of relations, and the type of relations (Miller et al., 1990). The database groups words into sets of similar words (synonyms) and connects them to more generic concepts (hypernyms) and more specific concepts (hyponyms). The three types of relations (synonymy, hypernymy, and hyponymy) enable the database to be naturally structured as a graph, making the dataset a perfect fit for retrofitting. Despite having the most extensive coverage among the three lexicons, WordNet might not be perfect for the verb prediction task since it focuses mainly on nouns.

Table 5.9: The performance of I2A when using the GoogleNews embeddings that have been retrofitted with semantic information from three lexical resources.

Lexicon	**Words in the lexicon**	**Supervised**		**Zero-shot**	
		Top1	Top5	Top1	Top5
FrameNet	10,822	30.08	45.19	13.95	33.35
Paraphrase Database	102,902	29.68	47.97	15.83	39.55
WordNet	148,730	30.34	49.33	16.76	39.15
google-news (Ch. 4)	NA	**32.48**	**51.45**	**17.92**	**41.39**

Results. Retrofitting harms performance regardless of which lexicon is used (Table 5.9). The performance is seen to be related to the size of the lexicon, especially in a zero-shot setting. Retrofitting to FrameNet, the smallest lexicon, has the greatest negative impact on performance while retrofitting to Wordnet, the largest lexicon, has the least negative effect.

We conclude that retrofitting to these particular lexicons is not practical. The knowledge they contain is mostly already encoded in the word embeddings, hence retrofitting would not provide any additional benefit and indeed may even be detrimental, as in the case of the verb prediction task addressed in this thesis. That being said, retrofitting would be effective if lexicons contained complementary and perhaps task-specific knowledge, such as expressing relations pertaining to the visual domain.

5.5 The impact of Certain Embedding Characteristics

This section examines the impact of embedding size, corpus size, and corpus genre on the quality of word embeddings and their effect on the verb prediction task. These characteristics are discussed in Sections 5.5.1, 5.5.2, and 5.5.1, respectively.

5.5.1 Embedding Size

In Section 5.3, the optimal embedding size was explored using randomly generated embeddings in a supervised setting. However, this does not provide much insight into the ideal size of embeddings that encode semantic information. As explained in Section 5.4.1, semantic embeddings have unique characteristics that make them different, and the optimal size is dependent on the task, training methods, and corpus size (Yin and Shen, 2018). This section seeks to determine the appropriate embedding size for encoding the semantic information necessary for the verb prediction task.

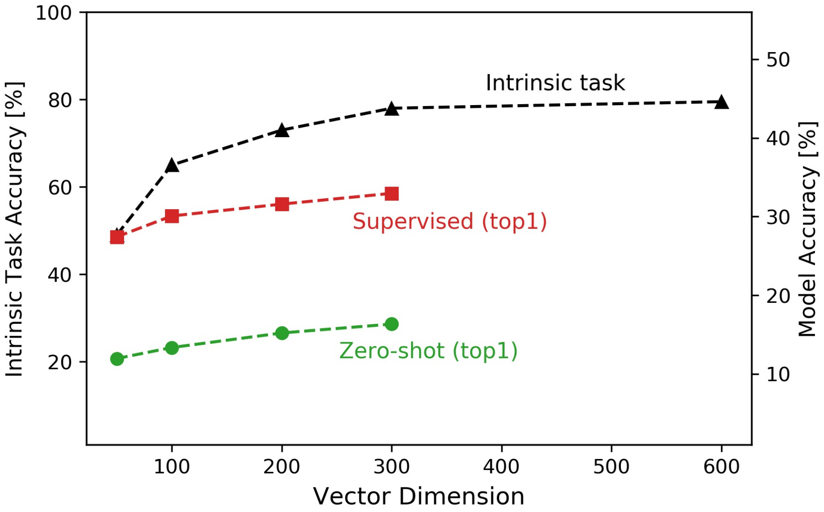


Figure 5.5: The plot illustrates the impact of increasing the size of embeddings on performance. The left scale measures the performance of GloVe embeddings on an intrinsic task, while the right scale measures the performance on the verb prediction task for the two learning settings.

Method. The study employs GloVe embeddings (Pennington et al., 2014) of various dimensons (50d, 100d, 200d, and 300d) that are pretrained using a common training corpus. GloVe is an unsupervised machine learning algorithm that learns word embeddings based on co-occurrence statistics between words. The version of GloVe embeddings used in the experiment is trained of the combination of Gigaword5 dataset (Parker et al., 2011) and an English Wikipedia archive from 2014[5]. A verb prediction model is constructed for each one of these different embedding sizes.

Results. The results of the study are presented in Figure 5.5, which shows the performance of GloVe-based models on a downstream task, as well as the performance of GloVe embeddings on an intrinsic semantic task (as reported by the GloVe authors). The figure illustrates that for the intrinsic task, larger embeddings result in better performance, but this improvement levels off after 300d. The performance of the verb prediction task follows the same pattern in both learning settings. Based on the graph and the information from previous sections, and what we have learned about performance bounds in Sections 4.1.2 and 5.3, it is assumed that there would only be a minimal improvement from using embeddings of size 600d. This would, however, be worth confirming empirically, something we have not done as the 600d embeddings are not publicly available.

[5] The following link was accessed in 2014: `http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`

Table 5.10: The performance of I2A when using two versions of GloVe embeddings, which differ in the size of their training corpus. One version is trained on 840 billion tokens, while the other is trained on only 6 billion tokens.

Size of training corpus	**Supervised**		**Zero-shot**	
	Top1	Top5	Top1	Top5
6B tokens	**32.92**	**50.76**	16.34	38.22
840B tokens	31.41	49.37	**17.62**	**42.41**
google-news (Ch. 4)	32.48	51.45	17.92	41.39

5.5.2 Size of the Training Corpus

This experiment studies the effect of the size of training corpus for embeddings on the performance of the downstream task. add a sentence. It aims to determine if a larger corpus leads to better embedding representation and improved task performance. Two versions of GloVe embeddings are used, with the only difference being the size of the training corpus. The first version is trained on 6 billion tokens from a combination of Gigaword5 (Parker et al., 2011) and an English Wikipedia archive from 2014[6]. The second is trained on 840 billion tokens of web text from the Common Crawl.

Results. As Table 5.10 shows, training embeddings on the large corpus results in slightly better performance in the zero-shot setting, while training on the small corpus yields slightly better performance in the supervised setting. This supports the notion that embeddings with more richer semantic representation aid in identify new concepts in zero-shot, but can also make it harder to learn in the supervised setting, as they complicate the geometry of the embedding space. However, this conclusion should be viewed with caution as the variance in performance in both learning settings is minimal, especially when considering that the large corpus is 143 times bigger than the small one.

5.5.3 Genre of the Training Corpus

The use of embeddings trained on a genre more appropriate for visual-based tasks should yield better performance than training on text from other genres such as news or Wikipedia articles. Novels are rich in descriptive language that we hypothesis may be more helpful in matching images to semantic embeddings.

[6] The following link was accessed in 2014: `http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`

Table 5.11: The performance of I2A when using two embedding variants, which only vary in the type of training corpus used.

Training corpus	**Supervised**		**Zero-shot**	
	Top1	Top5	Top1	Top5
us-novels	**31.93**	**51.56**	**18.62**	**42.59**
small-wiki	28.95	43.52	12.58	31.27
google-news (Ch. 4)	32.48	51.45	17.92	41.39

Method. In this experiment, two embedding models are trained on different genres of text - novels and Wikipedia articles. Both models follow the same training routine, including the same training algorithm and hyperparameters, and are trained on corpora of an equivalent size. The novel-based embeddings are trained on the Chicago Text Lab's proprietary corpus of US novels [7] , referred to as us-novels, which contains nearly 9000 American novels from the period of 1880-2000. The corpus consists of only 700 million tokens. The wiki-based embeddings, which act as the control for this experiment, are trained on a sample of the English Wikipedia archive used in Section 5.1, referred to as small-wiki. The sample is designed to match the number of tokens in the corpus of US novels.

Results. The performance of the us-novels embeddings was found to be superior to that of the small-wiki embeddings in both learning settings. This supports the idea that training embeddings on text that is visually descriptive can be beneficial for vision tasks, particularly those that involve understanding actions. The performance of us-novels is comparable with the main experiment in Chapter 4, which utilised GoogleNews embeddings trained on 6 billion tokens. This suggests that even though genre and corpus quality are significant factors, their impact may be diminished when large corpora are available.

5.6 Summary

In the chapter, we experimented with various techniques and tools for making the representation of target verb classes more suitable for the verb prediction task. We also examined the factors or characteristics of word embeddings that impact verb representation.

Sections 5.2 and 5.4 proposed utilising efficient methods to extract specialised (verb-specific) representations of target verb classes, in line with the thesis objective of utilising existing resources and avoiding additional training. By using the GoogleNews embeddings, these sections

[7] https://textual-optics-lab.uchicago.edu/us_novel_corpus

showed that post-processing of embeddings can improve performance on the verb prediction task, although the improvement was only marginal.

Section 5.1 introduced the idea of having verb-specific representation and attempted creating them from scratch, but the results were not as successful and the reasons for poor performance were identified. Section 5.3 presented experiments to investigate and understand certain properties of word embeddings, and Section 5.5 analysed the contribution of three characteristics of word embeddings.

As future research, we plan to use contextual embeddings to obtain dynamic representations for target labels. This would be advantageous in situations where the definition of the search space (Section 3.2.5) needs to be flexible, such as when a model is trained on single-word labels, but must also handle phrasal verbs during production. Additionally, pretrained contextual embeddings can be beneficial for obtaining representations from text resources that are of a specific genre yet small in size, such as video description datasets, and audio description, and novels.

Chapter 6

Varying the Representation of Images

In the previous chapter, we explored various techniques for extracting verb-specific representations of target labels and assessed their effectiveness in predicting verbs. In this chapter, we explore different approaches for representing input images and extracting key features from them. The techniques are categorised into two sections: visual-based (Section 6.1) and the semantic-based (Section 6.2).

Section 3.2 introduced the two primary types of image feature extraction, with a particular focus on the visual-based method, which is most relevant to this thesis. Figure 3.2 depicted how the image feature extractor, specifically the visual one, processes input images and interacts with other components.

To investigate the effect of image representation on verb prediction, we experiment with varying the feature extraction component in this chapter. We maintain the experimental setups and dataset splits from previous chapters for both supervised (Section 4.1.1) and zero-shot (Section 4.2.2) experiments.

6.1 Visual-Based Representation

This section compares the standard visual feature extractor used for processing images, which is a CNN trained on a subset of ImageNet called ILSVRC, against other CNN models that have different architectures, pretraining image datasets, or pretraining tasks.

We argue that a CNN trained on ILSVRC, despite its widespread use, is not perfectly suitable for the verb prediction task. ImageNet is primarily designed for object classification and features single objects in a canonical view, and many of its classes are not ideally suited for the verb prediction task. We note that many of the ImageNet classes, such as 397 classes of animals and 118 classes of dog breeds, are not relevant for verb prediction, while only a small number of classes, such as *ballplayer*, *groom*, *scuba_diver*, pertain to humans. The lack of classes for humans of different ages, such as babies and children, is also noted as a limitation, given our focus on human actions in this thesis.

6.1.1 Method

This section examines the effect of using different CNN models for extracting visual features from input images on the verb prediction task. Four verb prediction models are presented, each of which employs a distinct CNN model that has been trained on a specific dataset for a specific vision task. These models solve a more difficult task, which is hypothesised to improve the representation of image features. The impact of using larger CNN architectures is also analysed.

Moments in Time (MIT). Monfort et al. (2019) present a 3-second video dataset for action understating, primarily featuring human actions, where each video is annotated with a single label. The authors also propose multiple systems that separately or jointly exploit spatial, auditory, and temporal modality. In this experiment, we extract a CNN from one of their unimodal spatial-based systems. The CNN model is a ResNet50 network initialised on ILSVRC and trained on randomly selected frames from $802,264$ videos to classify 399 actions. It is believed that this CNN can produce better visual representation because it closely resembles the verb prediction task of this thesis.

MSCOCO. It was originally developed to provide a resource for detecting objects depicted in a non-canonical way. The dataset, as describe in Lin et al. (2014), offers various annotations for different tasks. In Section 2.2.2, we discussed the use of MSCOCO as an image captioning dataset, with each image having five captions.

In this section, we will employ MSCOCO as a multi-label classification dataset (multi-label classification dataset for labelling multiple objects in an image. One of the key advantages of MSCOCO is its comprehensive coverage of human-object interactions, with the *person* class having the highest number of instances. This emphasis on human-object interactions and the more complex task of multi-label classification, compared to ILSVRC's single-label classification, makes MSCOCO an excellent choice for generating powerful image representations for verb prediction. However, one aspect that needs to be considered when using MSCOCO as the pretraining dataset is its relatively smaller size in comparison to the other datasets examined in this thesis.

We utilise a CNN model provided by Wang et al. (2019), which is based on a slightly different variant of ResNet50, known as ResNext50. The CNN model is first pre-trained on ILSVRC and then fine-tuned on nearly 82K images [8] for multi-label classifications, rather than MSCOCO's standard detection task.

ImageNet-21K-P. The ImageNet-21K-P (Ridnik et al., 2021) is a filtered version of the original ImageNet dataset that retains most of its content, excluding only classes with insufficient images. The number of images has been reduced from 14 million to 12 million and the number of classes has been reduced from 21K to 11K. More importantly, the dataset has been transformed into a multi-label format through the use of WordNet hierarchy, allowing an image of *swan* to also be labelled as *animal*, *vertebrate*, *bird*, and *aquatic bird*. ResNet50 was initialised on ILSVRC and trained on ImageNet-21K-P.

Despite standardising efforts, the CNN model was trained with a newer PyTorch version and cannot be loaded with the older version used in other experiments in the thesis. The use of the newer version for training the verb prediction model may account for some variations in performance.

ResNet101. The ResNet architecture offers different sizes to cater to various application requirements. ResNet50, being the default choice in this thesis, strikes a good balance between accuracy and efficiency. However, in this study, we opted for a larger architecture, ResNet101, with twice the number of layers as ResNet50, both trained on the ILSVRC dataset. The number 50 and 101 in the name of these architecture represent the total number of layers, mostly convolutional layers.

The primary reason behind choosing ResNet101 is that deeper neural networks can extract more intricate features from images, enabling them to learn complex patterns and deliver better outcomes. We expect ResNet101 to outperform ResNet50 in the verb prediction task.

However, previous research has shown that larger ResNets tend to perform only slightly better on ILSVRC (the pretraining dataset) (He et al., 2016; Kornblith et al., 2019). For example, Kornblith et al. (2019) reported that ResNet101 outperforms ResNet50 only by 1.6% margin on the ILSVRC test set. Despite the small margin of improvement, our goal is to evaluate whether a larger architecture that slightly outperforms in the pretraining task can yield better results in our downstream task. Both ResNet50 and ResNet101 used in this study were sourced from the PyTorch repository (Paszke et al., 2019).

[8] The training set of the official 2014 split. Source: `https://cocodataset.org`

Table 6.1: Performance of the I2A using various CNN models as the feature extractor.

Pretraining Dataset	Dataset Type	CNN Arch	Supervised		Zero-shot	
			Top1	Top5	Top1	Top5
Moments in Time	Single-label actions	ResNet50	34.87	54.40	19.76	43.94
MSCOCO	Multi-label objects	ResNext50	31.86	50.95	18.07	41.38
ILSVRC	Single-label objects	ResNet101	31.24	50.23	17.70	41.42
ImageNet-21K-P	Multi-label objects	ResNet50	**40.54**	**60.01**	**20.60**	**45.43**
ILSVRC (Ch. 4)	Single-label objects	ResNet50	32.48	51.45	17.92	41.39

6.1.2 Results

The results from Table 6.1 indicate that training a CNN with ImageNet-21K-P leads to a substantial increase in performance, surpassing not only other models discussed in this section but also models from previous sections. However, this substantial improvement is only observed in the supervised setting, as the advantage decreases in the zero-shot setting, making the performance comparable to methods that enhance target label representation (as discussed in Sections 5.2 and 5.4.1). Our results suggest a strong connection between the size of the pretraining dataset and the performance of the downstream task, while factors such as architecture size and task difficulty seem to have less impact. ImageNet-21K-P is significantly larger than other datasets in terms of the number of images and classes, being ten times bigger than the second-largest dataset (ILSVRC) and over a hundred times larger than the smallest (MSCOCO). The underwhelming results in the zero-shot setting for ImageNet-21K-P suggest that performance in this setting cannot be significantly improved by simply using an exceptionally large training dataset.

MIT also improves performance when compared to using ILSVRC. This is noteworthy given the main difference between the two is the focus on actions and objects, respectively. Both datasets have roughly the same number of samples (1 million) and their examples are single-labelled. While MIT ranks second among the CNNs in the table, its performance is comparable to, or slightly lower than, the top methods for representing target labels as discussed previously in Sections 5.2 and 5.4.1. Since MIT is the only pretraining dataset of actions, we are concerned that some of its classes exist as well in imSitu, which raises concerns about the reliability of its zero-shot evaluation. The next section elaborates on this issue.

Despite being a more complex task (multi-label vs single label) and having images of a more complex nature (objects in context vs objects in a conical view), MSCOCO fails to surpass ILSVRC in terms of performance. Lastly, utilising a larger ResNet architecture (ResNet101) for the ILSVRC dataset also fails to outperform ResNet50.

Table 6.2: Assessing the performance of a subset of imSitu classes that are not present in MIT (i.e., imSitu − MIT). The results in this table are not obtained from new experiments, but rather a revised evaluation for the two models from Table 6.1.

Pretraining Dataset	**Test Classes**	**Supervised**		**Zero-shot**	
		Top1	Top5	Top1	Top5
MIT	imSitu − MIT	29.67	48.41	13.80	36.51
ILSVRC	imSitu − MIT	29.48	46.78	12.81	34.83
ILSVRC (Ch. 4)	imSitu	32.48	51.45	17.92	41.39

6.1.3 Discussion

Zero-shot learning is based on the premise that the training and test classes are distinct, a condition which is usually met when looking at classes of downstream tasks. However, this may not always the case when considering classes in pretraining datasets. Xian et al. (2019) point out that many of the widely used datasets for zero-shot learning, such as the Animals with Attributes (Lampert et al., 2014), the Caltech-UCSD Birds 200 (Wah et al., 2011), and SUN (Xiao et al., 2010), contain objects that are also present in the ILSVRC (the pretraining dataset for the CNN). If the CNN is exposed to certain classes during pretraining, these classes should be excluded from zero-shot experiments. In this section, we analyse the relationship between MIT and imSitu and assess the the overlap between their classes. MIT is the only action dataset of the three pretraining datasets, making it more susceptible to the issue of ***class leakage***.

Our study reveals that there is an overlap of 258 classes between MIT (pretraining dataset for the CNN) and imSitu (zero-shot dataset), which means that 55% of imSitu classes were already seen by the CNN prior to conducting the experiments in Table 6.1. This raises questions about the validity of MIT's results reported in the table. To address this, we only evaluated classes not found in MIT and the results, displayed in Table 6.2, indicate a substantial decrease in performance. However, it is uncertain if this reduction is solely due to class leakage or if the nature of these classes makes them harder to identify and detect. The table also shows that using a CNN pretrained on ILSVRC results in even further decreased performance, indicating that these classes pose a difficulty.

6.1.4 Summary

The section has experimented with four CNNs of different characteristics, mostly focusing on the charactersitics of the pretraining datasets. Two CNNs, ImageNet-21K and MIT, showed

Table 6.3: Assessing the performance of a subset of imSitu classes that overlap with MIT (i.e., leaky classes). The results in this table are not obtained from new experiments, but rather a revised evaluation for the the models from Table 6.1.

Pretraining Dataset	Test Classes	Supervised		Zero-shot	
		Top1	Top5	Top1	Top5
MIT	imSitu ∩ MIT	38.95	58.97	24.49	49.83
ILSVRC	imSitu ∩ MIT	34.86	55.15	21.98	46.59
ILSVRC (Ch. 4)	imSitu	32.48	51.45	17.92	41.39

improved performance compared to ILSVRC presented in Chapter 4, indicating that the size of the pretraining dataset might be the key factor. The section also addresses the issue of leakage in evaluating zero-shot experiments and proposes a simple method for examining potential leakage.

Future work plans to test newer and more advanced image representation techniques, such as transformer-based visual encoders, which are expected to handle more effectively the original version of imSitu that contains higher-quality images of various sizes. We conducted an experiment using the default configuration of I2A (Chapter 4), but the results were not promising, suggesting the need for a different visual encoder to handle the original imSitu. We will also explore the possibility of combining two visual encoders (e.g., CNNs), each designed for extracting specific features such as objects, scene types, or positional relations (e.g., *on* and *next to*). The next section shifts the focus to a semantic representation of input images, where the features are more abstract.

6.2 Semantic-Based Representation

In complex tasks like verb prediction, extracting meaningful information from raw digital images, such as jpeg files, presents various challenges due to the limitations of traditional techniques for feature extraction. Unlike tasks like object classification of ILSVRC that identifies objects in isolation, verb prediction requires analysing how objects interact in a scene, which further complicates the feature extraction process. In addition, working with imSitu images can be challenging due to quality issues, as previously noted in Section 4.1.6. These quality concerns can negatively impact the accuracy of the results.

To overcome these challenges, a semantic-based representation approach can be used. This method involves capturing higher-level features of the images, such as object names, spatial relations, and scene types, instead of relying solely on raw pixel data. By focusing on these semantic features, the limitations of traditional feature extraction techniques can be overcome, improving the accuracy of verb prediction. Furthermore, this approach provides an upper bound on performance by assuming perfect identification of object types and spatial relations, allowing us to evaluate how the I2A framework performs.

More concretely, this section introduces the idea of using an oracle to alleviate the I2A framework from the task of processing visual information. The oracle is responsible for recognising visual concepts present in images, such as people, physical objects, scenes or locations, and basic relations like *on* and *next to*. The semantic-based extractor consists of two parts: (1) an oracle that labels images with a set of labels of visual concepts, and (2) the extraction process of semantic features of these visual concepts.

The oracle could be either machine-based or human-based. In this section, we use human annotations of imSitu images that are already provided by the dataset authors. As discussed in Section 2.2.1, imSitu provides a wide range of concept types (i.e., fine-grained labels), and the concepts of each image are organised into a frame structure where the role for each concept is specified. For instance, an image of a fireman spraying water on fire is labelled with the following situation: $S = \big($ `spraying`, $\{$(`agent`, `fireman`), (`source`, `hose`), (`substance`, `water`), (`destination`, `fire`), (`place`, `outside`)$\}\big)$.[9] Additionally, the concepts are linked to their respective WordNet synsets, which will be useful in Section 6.2.4.

In the subsequent sections, we discuss different aspects of visual concept labels and their impact on the performance of the semantic-based extractor. Specifically, Section 6.2.1 introduces two techniques for encoding visual concept labels, while Section 6.2.2 presents the results of applying these techniques. Additionally, Section 6.2.3 investigates the impact of concept label granularity on the performance of I2A and the potential for information leakage caused by fine-grained labelling. Lastly, in Section 6.2.4, we propose and evaluate methods for generalising or coarsening the labels. By exploring these topics, we can gain a deeper understanding of the visual concept labels and identify effective ways to utilise them especially for the zero-shot learning.

6.2.1 Method

The section outlines two techniques for encoding semantic information from sets of concept labels, which have replaced raw images as the framework's inputs. These techniques are the semantic feature extractors for I2A, as discussed in Section 3.2.3. The first technique involves using word embeddings, while the second utilises sentence encoders.

[9] This example is taken from the original paper of imSitu (Yatskar et al., 2016)

First technique: Embedding pooling. In this approach, each concept label is encoded using GoogleNews embedding. The embeddings of all concepts in an image are combined into one vector, creating the feature vector of the image (Section 3.2.3). Two methods for combining the embeddings are evaluated: average pooling and max pooling.

- **Average pooling.** The operation takes the mean over each dimension for all word embeddings. Average pooling is used extensively in Computer Vision for aggregating features from CNNs, and later has been introduced for NLP tasks such as sentence encoding (Wieting et al., 2016; Adi et al., 2017).
- **Max pooling.** The motivation behind max pooling is that not all visual concepts contribute equally to final predictions. The technique presumably extracts the most salient features from every dimension. It does that by taking the maximum value along each dimension of the word vectors (Collobert et al., 2011; Shen et al., 2018). This method, however, may amplify the noise signal or spurious features from certain input embeddings.

Matching WordNet synsets to embeddings vocabulary. We found that 87% of synsets in imSitu can be directly linked to the vocabulary of GoogleNews embeddings through their synset name. For the remaining 13%, the following solutions are taken to find a suitable match: (1) using one of the synset's lemmas instead of its name, (2) making slight modifications to the synset name (e.g., hyphenated words), and (3) exploring the WordNet hierarchy (e.g., selecting a parent synset). Through this process, the total number of distinct synsets is decreased from 7.8K to 5.4K.

Second technique: Sentence encoding. Using a sentence encoder as a feature extractor may produce a better representation for the inputs, where the representation consists of salient and highly predictive features. The intuition is driven by the fact that the sentence encoder comprises many complex neural layers, is trained sequentially on two tasks, and has a massive number of parameters. Since this approach expects inputs as sentences, the set of concepts of each image is transformed into a pseudo-sentence but not a complete sentence essentially due to the absence of the main verb, which is the target we aim to predict. The creation of pseudo-sentences is facilitated with the help of manually created templates. The imSitu dataset provides these templates to aid in the annotation process and illustrate the interaction between verbs and their roles. Each verb (class) in the dataset has a unique template; for example, the following template is for the verb *sign*:

```
the AGENT signs the SIGNEDITEM with the TOOL at the PLACE.
```

Table 6.4: A sample of pseudo-sentences ready for encoding.

image 1	`man be contract with hand`
image 2	`people be on sofa at room`
image 3	`man be at hospital`
image 4	`bodybuilder be their body at audience at stage`

For each image, we populate its corresponding template with the set of concepts depicted in the image instance, replacing roles with values from the concept set. The verb is masked by substituting it with the verb *be*. Nouns sharing the verb's root are also removed, as in "*typing with a tape*" and "*pitching on a pitch*". Table 6.4 provides examples of processed templates ready for encoding. These pseudo sentences are transformed into a vector of size 730 dimensions. The feature extractor for this technique is a standard BERT model (Devlin et al., 2019) that is finetuned on a semantic textual similarity task (Reimers and Gurevych, 2019). BERT is a transformer-based model that generates contextual embeddings for each word in the sentence, which are then aggregated via the average pooling operation.

6.2.2 Results

Table 6.5 demonstrates that embedding-based techniques surpass all visual-based (CNN-based) methods, with a significant improvement observed in the supervised setting regardless of the pooling method used. In the zero-shot setting, using average pooling lead to a slightly better performance, indicating that max pooling may lead to overfitting, due to its way of extracting features, as discussed earlier in Section 6.2.1.

Using the sentence encoding techniques on pseudo sentences yields an exceptional performance in the supervised setting, making it the top performer by a substantial margin. However, its performance in the zero-shot setting fails to even reach that of the average embedding pooling, suggesting that the rich textual representation may introduce information leakage as some information in the pseudo sentences is not accessible to the embedding-based methods.

We suggest a new technique to validate that the jump in performance is mostly due to the information present in pseudo sentences and not to the sentence encoder. Instead of representing the input with a pseudo-sentence, we concatenate the image's set of concepts into one string as in "*people sofa room*". This technique, referred to as the ***bag of concepts***, allows us to compare both encoding techniques using the same amount of information. The bag of concepts approach achieves comparable performance to the embedding-based techniques, indicating that the gain observed in the pseudo-sentences approach is entirely due to the additional description and not the encoding mechanism. The result confirms that the pseudo-sentences method relies

Table 6.5: Performance of I2A using two text encoders as replacements for the visual feature extractor. Each one of these text encoders has two distinct implementations, yielding a total of four different models.

Encoding Technique		**Supervised**		**Zero-shot**	
		Top1	Top5	Top1	Top5
Distributional embeddings	avg pooling	51.11	67.65	**22.04**	**49.24**
	max pooling	51.52	67.27	20.99	47.48
Sentence encoder	pseudo sentences	**70.42**	**82.30**	21.50	47.32
	bag of concepts	51.62	67.51	21.18	46.17
Visual-based extractor (i.e., CNN) (Ch. 4)		32.48	51.45	17.92	41.39

on leaked information to achieve its outstanding performance in the supervised setting.

All semantic-based feature extractors perform significantly better than visual-based extractors (or CNNs). The reason for this is probably because fine-grained annotation of visual concepts provides an unfair advantage to semantic-based methods and might be considered a form of data leakage (Kaufman et al., 2012). For example, images of class TEACHING contain leaky labels such as *teacher* and *pupils*, which are strongly predictive of the class. The following section investigates the contribution of annotation granularity contributes to the superior performance of semantic-based methods.

6.2.3 Examining Annotation Granularity

This section investigates the granularity of annotations and its effect on the performance of the verb prediction task, particularly in the zero-shot setting. To explore the annotation quality, we employ Pointwise Mutual Information (PMI) and Decision Trees as analysis tools.

Pointwise mutual information (PMI). PMI measures the dependency between two random variables (Church and Hanks, 1990). A high and positive PMI score indicates that both variables are more likely to co-occur, giving them a predictive power. In this section, we use PMI to measure the relationship between imSitu classes and the labels used to denote visual concepts in imSitu images. We concentrate on pairs with high PMIs as these may reveal leaky labels. After excluding pairs with fewer than ten examples, we calculated the PMI score for 7.2K pairs of the 463 classes and 5.4K unique concepts (or WordNet synsets). Normalised PMIs are calculated to improve the interpretability of scores (Equation 6.1).

Table 6.6: Top 10 pairs with the highest NPMI scores.

Concept	Class	NPMI	Concept	Class	NPMI
1. surfboard	SURFING	0.99	6. blowtorch	WELDING	0.96
2. parachute	PARACHUTING	0.97	7. sheep	SHEARING	0.95
3. scale	WEIGHING	0.97	8. drumstick	DRUMMING	0.95
4. syringe	INJECTING	0.96	9. rocket	LAUNCHING	0.95
5. vacuum	VACUUMING	0.96	10. welder	WELDING	0.94

Table 6.7: Random examples of visual concept labels extracted from the top 100 pairs with the highest NPMI scores. These labels are organised into five proposed categories.

Agent nouns	dancer, punter, massager, beautician, manicurist, cameraman, skydiver
Specific labels	whitewater, launching_pad, skipping_rope, rocking_chair, pier, shaving_cream, marching_band, frying_pan
Specific tool	whip, drum, comb, skate, stapler, masher, shredder, toothbrush
Abstract	injury, mime, prey, voice, song, team, reading
Locations	casino, concert, mine, cockpit, space, recording_studio

$$
\begin{aligned}
npmi(x,y) &= \frac{pmi(x,y)}{-\log p(x,y)} \\
pmi(x,y) &= \frac{\log p(x,y)}{p(x) \times p(y)}
\end{aligned}
\tag{6.1}
$$

Based on our manual examination of the top 100 pairs, we found that most of these pairs reflect a strong relationship between actions and objects (e.g., MICROWAVING and *microwave*), which is an topic already discussed in Section 4.1.7. However, some pairs raise concerns about potential information leakage in the annotation process. For instance, Pair 6 in Table 6.6 exemplifies a situation where the labels are too specific and technical, existing only for a particular class. Pair 10 highlights another issue, where the presence of an agent noun (*welder*) reveals the nature of the action. Our examination of the top 100 pairs resulted into the following categorisation of concept labels: specific labels (11%), specific tools (27%), abstract concepts (9%), agent nouns (10%), and specific locations (6%). We believe that "agent nouns" and "specific labels" result from the level of granularity imposed during the annotation process. This analysis is not intended to establish well-defined categories, but to provide some context for the reader. Table 6.7 presents examples of each category.

A simple PMI-base model is built to quantify the predictive power of visual concepts with high PMI scores. Like other tasks in Section 6.2, the input is represented as a set of concept labels, and the output is the class label. In this experiment, PMI is calculated based on the training instances and without filtering low-count pairs. In the evaluation step, we select the class that maximises the summation of PMI values (Equation 6.2). The method achieves 49.30% accuracy for top1 and 60.30% for top5. Despite having lower performance compared to other semantic-based methods (as shown in Table 6.5), it still managed to be surprisingly close to these more advanced techniques.

$$\hat{y} = \text{argmax} \sum_{i=1}^{k} pmi(o_i, y) \tag{6.2}$$

where k is the number of objects in a given evaluation instance

Decision Trees. In this section, decision trees (Breiman et al., 1984) are utilised as a baseline for I2A, offering an alternative to the PMI-based baseline and come with regularisation techniques to prevent overfitting. Unlike the deep learning algorithms used in I2A, Decision Trees belong to a different class of machine learning that creates simple and easily interpretable models based on meaningful features. This allows for a better understanding of the decision-making process in the task discussed in Section 6.2.1 and highlights the key visual concepts for each class.

Binary classifications (one-vs-rest strategy) are performed by constructing a decision tree model for each imSitu class. The experiments involve testing different tree depths: “depth 1” ($d1$), “depth 5” ($d5$), and “unlimited depth”. Trees with a depth of 1 ($d1$) enable examination of the impact of specific concept labels, while trees with unlimited depth offer a robust comparison to the semantic-based variants of I2A discussed in Section 6.2.1. In this section, we use the terms “feature” to denote use of “visual concept labels” as the variables in decision trees.

Decision stumps, or $d1$ trees, are interesting as they classify instances based on a single feature. This makes it easier to examine how leaky concepts persist in such restrictive models. Out of the 463 decision stumps, only 420 features are utilised; therefore, some features are common. For instance, people is shared features across the following classes: SOCIALISING, CONGREGATING, GATHERING, QUEUING, and INTERMINGLING. Another example is *police*, which is used in the following classes: CLAPPING, PATTING and SHAKING. It is worth noting that classes sharing features tend to perform poorly in testing, which is understandable as only one feature is used to classify multiple classes.

Table 6.8 displays the top ten performing classes. A correlation is observed with the top

Table 6.8: Top 10 performing classes in decision stumps (i.e., one-feature classifiers), along with the accuracy and the feature used for classification.

Class	Top1	Feature	Class	Top1	Feature
1. VACUUMING	0.99	vacuum	6. WELDING	0.91	blowtorch
2. SURFING	0.97	surfboard	7. MOPPING	0.90	floor
3. WEIGHING	0.97	scale	8. INJECTING	0.90	syringe
4. PITCHING	0.96	baseball	9. PHOTOGRAPHING	0.90	camera
5. KISSING	0.96	lip	10. CAMPING	0.90	tent

Table 6.9: The performance of three decision tree classifiers that differ in depth and number of features. The table also shows the PMI-based classifier from the previous section.

Tree depth	Features	Top1	Top5
1	420	28.16	34.76
5	2,251	44.56	57.39
No limit	3,772	47.63	59.83
PMI	5,450	49.30	60.30

10 NPMI scores in Table 6.6. Nonetheless, this table reveals a new issue in the dataset where annotations are made based on prior knowledge of the verb. While *lip* is visually present in many classes, it is only thoroughly annotated in images of the class KISSING. This method is inferior to the PMI-based and other tree variants (as shown in Table 6.9) due to limited features.

The $d5$ trees exhibit a substantial improvement compared to decision stumps. The method employs $2,251$ features, or about 40% of all features, and each classifier typically uses 14 features. Lastly, unconstrained trees show only marginal improvement over $d5$, relatively speaking, despite using a much larger number of features $(3,772)$ and tree size (each model on average uses 43.5 features).

Summary. This section introduced the use of PMI and decision trees not only as baselines against the semantic-based models in Section 6.2.2 but also as diagnostic tools to identify strongly indicative concept labels, where the presence of label l almost guarantees that the image is annotated with class c. While having predictive features is generally desirable, they can also lead to overfitting and a lack of generalisability, as demonstrated in this section.

Our results showed that semantic-based models, as represented in Table 6.5, outperformed PMI-based and decision tree classifiers, despite these baselines performing a simpler task of multi-class classification with discrete features as input.

6.2.4 Generalising Input Labels

This section proposes solutions that can be used jointly or individually to fix the fine-granularity of visual concept labels.

Most generic annotation. imSitu provides three annotations per image. These annotations are structured in verb frames where the role and value of each object are specified. Section 6.2.2 prioritised the richness of input representation by selecting the most populated frame (i.e., the one that has the fewest empty role values). However, in the "most generic" technique, the goal is to label objects with the most generic labels. It is believed that the synset with the shortest distance to the root represents the visual concept in its most generic form.

Coarse human labels. The annotations for people in imSitu images are done at a fine-grained labels, specifying gender, stage in life (e.g., adolescence and baby), occupation (e.g. surgeon and welder), or social status (e.g., husband). Our emphasis on human labelling stems from the significant presence of people in the dataset, with each image depicting a person or a group of people. The dataset contains 628 synsets for labelling humans, and that number is after applying the previous technique (i.e., selecting the most generic annotation). These fine-grained synsets can be problematic, as they enable the machine learner to access information not intended to be part of the input (i.e., information leakage). The conversion process to generic labels is straightforward since the basic-level labels (i.e., the target labels for conversion) are predefined; labels describing an individual human are converted to *person*, and labels describing a group of humans are converted to *people*.

Yolo9000 Labels. Another technique to tackle this problem is to use a well-crafted taxonomy, suitable for the visual domain. More specifically, we paid specific attention to taxonomies designed for object-based tasks such as classification or detection. Such taxonomies tend to satisfy two conditions: visual concreteness of objects and entry-level terminology. This should also allow us to evaluate I2A in a more practical setting where the input is labelled at the same level as current state-of-the-art methods.

Initially, we considered using the MSCOCO taxonomy, but its coverage of imSitu objects is quite lacking, covering only 17% of imSitu labels. Yolo9000 (Redmon and Farhadi, 2017), on the other hand, has broad coverage of objects at different granularities. We used the WordNet hierarchy to rank the granularity of Yolo9000 classes, from the most generic to the most specific. The top 15 generic labels were removed as they were found to be too generic (e.g., *object* and *substance*). This technique attempts to match every concept label in imSitu to a Yolo9000

Table 6.10: The performance of the average pooling technique (detailed in Section 6.2.2) when using four levels of input label granularity (i.e., visual concept labels in the image).

No	Label granularity	Features	Top1	Top5
1	most generic annotation	4,574	50.37	66.88
2	most generic annotation + coarse human labels	4,110	47.68	63.57
3	most generic annotation + coarse human labels + YOLO9000 labels	3,102	42.05	57.13
4	basic-level + coarse human labels	667	31.14	46.72
	Original labels (avg-pooling) (Sec. 6.2.2)	5,450	51.11	67.65

class, starting from the most generic Yolo9000 class at the top of the list.

Basic-level concepts. The basic level refers to the level of abstraction that humans are most familiar with compared to other levels (e.g., *dolphin* is the basic level for *grampus griseus*). Many studies have proposed methods for selecting the basic-level (or the entry-level) term of objects (Ordonez et al., 2013; Izquierdo et al., 2007).

We employed the Izquierdo et al. (2007) tool to convert imSitu concept labels to basic levels. This tool uses the structural properties of WordNet, such as hypernym, synonym, and antonym, to automatically identify the basic concepts. Specifically, the tool considers the total number of relations of each synset and rejects potential basic levels that do not represent a minimum number of synsets. However, we noted that the tool may generate overly generic labels for input labels near the basic-level and may not be effective for very specific levels, resulting in synsets that are still too fine-grained.

Results. For each of the approaches listed in Table 6.10, we replicated the avg-pooling experiment from Section 6.2.2, only varying the input label granularity. The table shows that utilising the most generic technique (first row of the table) led to a slight decrease in performance. The combination of this technique with converting human labels to coarser labels (second row) resulted in a further decrease of three percentage points for both top1 and top5. The combination of the previous techniques with the Yolo9000-based approach (third row) led to a drop in performance by approximately six more percentage points. The fourth row shows the results of combining the basic-level approach with coarse human labels, causing another significant decrease in performance of 11 points.

Table 6.11: Best input and target representation as studied independently. The list shows the top two methods for representing inputs from the previous sections in this chapter, as well as the top two methods for representing targets from Chapter 5.

Input representation	**Target representation**	**Supervised**		**Zero-shot**	
		Top1	Top5	Top1	Top5
ImageNet-21K-P (Sec. 6.1)	google-news (Ch. 4)	40.54	60.01	20.60	45.43
pseudo sentences (Sec. 6.2)		**70.42**	**82.30**	**21.50**	**47.32**
ILSVRC (Ch. 4)	avg-verbs (Sec. 5.2)	32.90	52.24	20.21	44.75
	processed-5K-10C (Sec. 5.4)	**34.20**	**55.08**	**20.89**	**44.90**

Discussion. To assess I2A's robustness to input label granularity, we conducted an experiment where the training and evaluation granularities were different. A model trained on the original labels (fifth row) performed poorly when evaluated on coarser granularities (second row in Table 6.10), with a top1 score of 37.30% and top5 score of 53.13%. While reversing the setup improved the results (top1: 42.75% and top5: 58.44%), it still did not perform as well as when the granularity was consistent between training and evaluation (second row). These results suggest that I2A lacks robustness and that training on fine-grained level of granularity can negatively impact performance if that level cannot be replicated in real-world scenarios, as might be the case with the imSitu dataset.

6.3 Combining the Best Image and Target Representations

In the previous sections, we have demonstrated that improving the input representation results in better performance than improving the target label representation, as seen in Table 6.11. In this section, we aim to investigate whether combining the most effective methods for representing the target label with those for representing the image input can further enhance the outstanding performance achieved by the optimal input representations.

6.3.1 Method

We created four different combinations by pairing the two most effective input representations with the two most effective target label representations.

To represent the image inputs, we selected the best-performing technique identified in Section 6.2, which involves using pseudo sentences. However, given (1) concerns voiced above about information leakage arising either from the method of sentence construction or the fine granularity of concept labels, and (2) we have studied both semantic-based and visual-based

Table 6.12: Best individual representation combined. The performance of I2A when combining the best methods for input representation with the best methods for target representation.

Input representation	Target representation	Supervised		Zero-shot	
		Top1	Top5	Top1	Top5
ImageNet-21K-P (Sec. 6.1)	processed-5K-10C (Sec. 5.4)	41.67	62.60	23.86	49.76
pseudo sentences (Sec. 6.2)		71.68	84.12	24.84	51.19
ImageNet-21K-P	avg-verbs (Sec. 5.2)	40.48	60.53	22.88	48.68
pseudo sentences		70.29	82.34	24.63	52.10

representations, we have also decided to include the best-performing visual-based method, ImageNet-21K-P.

For target label representation, we selected the most effective techniques identified in Sections 5.2 and 5.4.1, which outperformed the original Google-News embeddings. To facilitate easy comparison, we summarised the results of these four methods in Table 6.11.

6.3.2 Results

Our analysis of the combined image and target representations reveals several important insights. Firstly, the use of the processed-5K-10C post-processing technique from Section 5.4.1, which involves a matrix of 5K verbs and the removal of the top 10 components, enhances performance in both learning settings, consistent with our prior findings (row 1 Table 6.11 vs. row 1 Table 6.12).

On the other hand, our analysis revealed that using the avg-verbs method for target representation did not result in improved performance in the supervised setting (row 1 Table 6.11 vs. row 3 Table 6.12). However, we did observe a comparable improvement for avg-verbs in the zero-shot setting. This finding is consistent with our earlier findings in Chapter 5, which reported limited benefits of enhancing the semantic lexical representation in the supervised setting but demonstrated potential advantages in the zero-shot setting. These observations suggest that the zero-shot performance can be further improved when combining the best of both input and target representations.

6.4 Summary

This chapter explored the impact of two main types of image feature extraction methods on verb prediction: visual-based and semantic-based. Visual-based extractors operate on raw images and extract low-level visual features such as shapes and textures. In contrast, semantic-based extractors require an oracle, which could be machine-based or human-based, to identify nameable concepts depicted in the image. The semantic-based extractor then extracts features from these concept labels.

In Section 6.1, Convolutional Neural Networks (CNNs) were used as visual-based extractors, with the pretraining dataset being the main variable to determine the characteristics that matter for verb prediction. Specifically, the study explored whether a larger pretraining dataset, a more challenging pretraining task such as multi-label classification, or a more relevant pretraining dataset such as action prediction could improve the performance of the verb prediction task.

The results revealed that using a massive pretraining dataset (ImageNet-21K-P) significantly enhanced supervised performance, outperforming other visual-based image feature extractors and the best techniques from Chapter 5 such as pseduo-random and processed-5K-10C. However, this improvement was not observed in zero-shot, only matching the best technique from Chapter 5 in this setting (processed-5K-10C). Combining the best feature extractor (ImageNet-21K-P) with the best target representation (processed-5K-10C) further improved performance, achieving an impressive 23.86% for top1 and 49.76% for top5. Overall, this section highlights the importance of pretraining datasets in improving verb prediction performance and provides insights into the factors that impact the effectiveness of pretraining datasets.

Section 6.2 focused on the second type of feature extractors, the semantic-based extractors, which operate on a set of labels denoting concepts that appear in the image. In this section, the concept labels provided by the authors of imSitu, annotated by humans, were used. Two techniques were explored for feature extraction from these labels: (1) applying a pooling operation over the distributional embeddings of these concepts, and (2) using a task-specific model, which was a sentence encoder for a similarity task in this case.

The experiments demonstrated that both techniques led to substantial improvements in supervised setting, with accuracy increasing by up to 73.7% and 46.3% for top1 and top5 respectively, compared to the I2A model from Chapter 4. It is worth noting that these figures represent relative improvements, rather than absolute values. However, in the zero-shot setting, the improvements were not as significant, with top1 improving by only 20.6% and top5 by only 17.3%.

The remainder of Section 6.2 aimed to determine the reason for the discrepancy in performance between supervised and zero-shot scenarios. Specifically, the investigation focused on whether the discrepancy was due to approaching the best capability of I2A in the zero-shot setting or the possibility that the concept labels provided an unfair advantage in the supervised setting. This section focused on the latter possibility and explored the level of granularity of the imSitu concept labels and their associations with imSitu classes (verbs). In Section 6.2.3, it was demonstrated that certain concept labels strongly indicate the presence of a particular class. Specifically, the presence of label l almost guarantees that the image is annotated with class c. Section 6.2.4 conducted supervised experiments where the level of granularity of the concept labels was varied. The results showed that as the labels became coarser, performance decreased rapidly.

In future work, it would be useful to replicate the same experiment in the zero-shot setting to determine whether the same rate of decrease occurs or not. The goal would be to investigate whether the fine granularity of the imSitu dataset provides any additional benefit for zero-shot learning. It is possible that coarser labels may be just as effective, or even more so. Therefore, by examining the impact of different levels of granularity in the zero-shot setting, we can gain insights into the most suitable labelling scheme for zero-shot verb prediction tasks.

Chapter 7

Conclusion

7.1 Summary

Over the last decade, the field of artificial intelligence has experienced a remarkable surge in progress, with the development of increasingly large models that require vast computational resources for their training due to the high number of parameters involved. These models have allowed for significant advances in a range of applications, including natural language processing, and computer vision, among others.

In this thesis, we have contributed to this progress by developing a framework that leverages these massive resources to tackle the challenging task of verb prediction. Traditionally, this task has been oversimplified in the literature as predicting a single verb that describes a single interpretation of the action. However, our framework takes a more sophisticated approach by taking into account the multiple possible interpretations of actions.

Our approach overcomes the limitation of supervised learning, which would require training with all possible verbs that could be used for describing actions in images. Instead, our framework can describe image actions using verbs that were not observed during training, making it more practical and efficient.

Concretely, in Chapter 3, we introduced our I2A framework for predicting verbs that can be used for describe image actions. The I2A framework is capable of training models in supervised and zero-shot settings. The chapter discussed the framework's components in detail and also discussed the role of distributed lexical representations.

In Chapter 4, we presented a diverse set of experiments using commonly used off-the-shelf components to evaluate the performance of the I2A framework, serving as a benchmark for subsequent chapters. We conducted extensive analyses to assess the framework's effectiveness on different groups of classes, such as concrete and abstract verbs, providing valuable insights into its strengths and limitations. Furthermore, we carried out a comprehensive set of analyses to understand the framework's failures, identifying the primary causes of errors in the verb predictions. Specifically, we analysed the impact of the pretraining dataset of the image feature extraction component and the properties of imSitu images. We also analysed the impact of distributed representations. These analyses provided valuable insights into the limitations of the current framework and suggest possible avenues for future research to address these challenges.

In Chapter 5, we investigated the impact of varying the representation of target labels (verbs) on the performance of the I2A framework. We introduced several techniques to improve the representation of lexical embeddings, including applying post-processing on existing embeddings and training new ones from scratch. Our experiments demonstrated that simple post-processing techniques are not only efficient but also far superior to building bespoke embeddings from scratch. We showed that by applying one of these techniques to the GoogleNews embeddings, we achieved state-of-the-art performance on the verb prediction task, outperforming the best-performing approach in the literature, so far as we are aware. Moreover, our experiments highlighted the importance of incorporating semantic information into the model to enable effective zero-shot learning of verb prediction. We also analysed the impact of different embedding sizes and training corpora genres and sizes on the framework's performance. Overall, Chapter 5 provides valuable insights into the role of lexical embeddings in the I2A framework's performance and identifies effective techniques for improving the performance of the I2A framework.

In Chapter 6, we investigated the impact of varying the representation of input images, considering two main types of image feature extraction: visual-based and semantic-based. Section 6.1 focuses on the visual-based extractor, which expects the input to be raw images (e.g., jpeg files) and extracts low-level visual features of images such as textures and shapes. We solely focused on one specific type of visual extractor (i.e., CNNs and in particular variants of ResNets) varying only the pretraining dataset. The goal was to determine what characteristics of pretraining datasets matter to the task of verb prediction. Our findings revealed that the size of the pretraining dataset is key for improving the performance of I2A.

Section 6.2 explored semantic-based feature extraction, which involves the use of an oracle to label key visual concepts in input images, such as objects, scene type, and spatial relations. The extraction process then operates on these labels using two techniques: pooling of the distributional embeddings of the labels or utilising a sentence encoder trained to perform a sentence similarity task. Our research showed that semantic-based extractors outperformed visual-based extractors, including the default option from Chapter 4, in the supervised setting. However, we found that the improvements were not as significant in the zero-shot setting.

In addition, we found that both semantic-based approaches performed better than the decision tree classifier, which serves as a robust baseline and was also trained using oracle-supplied visual features. We also used the decision tree baseline and a PMI-based classifier to investigate the relationship between imSitu classes and visual concept labels. These baselines were ideal for the task due to their interpretability. Our study revealed that certain visual concepts were highly indicative of the target class, to the extent that it suggested a form of data leakage. Lastly, we recommended coarsening the visual concept labels and showed that while supervised performance is impressive, it rapidly decreases with coarser labels.

7.2 Future Work

While the research presented in this thesis provides valuable insights into the capabilities of the I2A framework, it is important also to consider future directions that could expand upon these findings. In this section, we outline critical areas that require further investigation, providing potential avenues for future research.

One of the critical areas for future work involves the exploration of combining various visual feature extractors instead of relying solely on a single CNN. The argument is that I2A requires diverse visual features that cannot be provided by a single CNN alone. To tackle this issue, a possible solution would be to develop separate CNNs, each focusing on extracting features related to objects, scene type, and spatial relations. This idea is based on the promising performance of the pseudo-sentence feature extractor discussed in Section 6.2.2, which encodes vital information about objects, the scene, and spatial relations.

An additional area for future research is the definition of the embedding space and the interpretation of its points. In Chapter 4 and most of Chapter 5, the points in the space correspond to the word embeddings themselves, with the main exception being Section 5.2 where points represent the average of verb forms. To further this idea, it may be beneficial to have these points represent the average of multiple definitions or glosses of verbs, or the average of verb instances in various sentences that exhibit the specific meaning we are targeting, as suggested in Chapter 5.

The use of transformer techniques represents a promising avenue for exploration due to their advanced ability to seamlessly integrate multiple modalities. What makes these techniques even more fascinating is their ability to scale to unprecedented sizes, both in terms of the training dataset and model parameters. We have witnessed datasets grow from hundreds of thousands or millions of samples to hundreds of millions or even billions, and model sizes grow from tens of millions to billions of parameters. However, to fully harness the potential of transformers, a significant rethinking of the framework design is necessary. Thus, a mere component replacement of the current framework with transformer-based ones (as demonstrated with the pseudo sentence method in Section 6.2.2) may not result in any further improvements.

It is also crucial to further investigate the set of verbs chosen for training or testing in the context of I2A models. This exploration can provide a deeper understanding of the characteristics of these verbs and complement the findings presented in Section 4.2. To better understand the characteristics of these verb classes, we must address two critical questions. First, we need to explore what factors cause I2A to excel at classifying certain verbs when trained in a supervised setting while struggling with others? Second, we need to determine whether it is necessary or beneficial for I2A models to use all verbs that perform well under the supervised setting to achieve optimal zero-shot performance on a completely new set of verbs?

Finally, it would be worthwhile to examine the impact of training set size on the performance of I2A. By plotting performance against training set size, we can determine whether performance plateaus or continues to increase significantly as we approach the maximum training set size in imSitu. This analysis can have significant implications for determining whether increasing the training set size is a worthwhile investment of effort and resources.

This section outlined several key areas that could potentially improve the performance of the I2A framework and explored possible avenues for future research, including the combination of different visual feature extractors, the redefinition of the embedding space, and the integration of transformer techniques. Lastly, we argued for further exploration of the set of verbs used for training or testing the I2A model, which should provide insights into the critical factors that determine the success of I2A in the zero-shot setting.

Bibliography

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2016. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(7):1425–1438.

Ziad Al-Halah, Makarand Tapaswi, and Rainer Stiefelhagen. 2016. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pages 5975–5984.

Malihe Alikhani and Matthew Stone. 2019. "Caption" as a coherence relation: Evidence and implications. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 58–67.

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 822–827.

Chris L. Baker, Joshua B. Tenenbaum, and Rebecca R. Saxe. 2009. Action understanding as inverse planning. *Cognition* 113(3):329–349.

Kobus Barnard and Matthew Johnson. 2005. Word sense disambiguation with pictures. *Artificial Intelligence* 167(1):13–30.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(null):1137–1155.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

Hans Boas. 2008. Towards a frame-constructional approach to verb classification. In *Grammar, Constructions, and Interfaces. Special Issue of Revista Canaria de Estudios Ingleses*. volume 57, pages 17–48.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 29.

Leonid Boytsov and Bilegsaikhan Naidan. 2013. Engineering efficient and effective non-metric space library. In *Similarity Search and Applications - 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings*. pages 280–293.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Chapman and Hall/CRC.

John Bullinaria and Joseph Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods* 44:890–907.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal Artificial Intelligence Research* 63:743–788.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*. Springer International Publishing, Cham, pages 213–229.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, pages 169–174.

Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. 2015. HICO: A benchmark for recognizing human-object interactions in images. In *2015 IEEE International Conference on Computer Vision (ICCV)*. pages 1017–1025.

Xinlei Chen, Alan Ritter, Abhinav Gupta, and Tom Mitchell. 2015. Sense discovery via co-clustering on images and text. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 5298–5306.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2023. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1):22–29.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *International Conference on Machine Learning*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(null):2493–2537.

Max Coltheart. 1981. The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology A: Human Experimental Psychology* 33A(4):497 – 505.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science and Technology* 41:391–407.

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

Barry J. Devereux, Lorraine K. Tyler, Jeroen Geertzen, and Billi Randall. 2013. The centre for speech, language and the brain (CSLB) concept property norms. *Behavior Research Methods* 46(4):1119 – 1127.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Luana Fagarasan, Eva Maria Vecchi, and Stephen Clark. 2015. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics*. Association for Computational Linguistics, London, UK, pages 52–57.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. pages 1778–1785.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1606–1615.

Vittorio Ferrari and Andrew Zisserman. 2007. Learning visual attributes. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 20.

J. R. Firth. 1957. A synopsis of linguistic theory, 1930-1955. In *Studies in Linguistic Analysis*. The Philological Society, Oxford, volume 1952-59, pages 1–32.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 2121–2129.

Yanwei Fu, Timothy M. Hospedales, Tao Xiang, and Shaogang Gong. 2015. Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37:2332–2345.

Yanwei Fu, Tao Xiang, Yu-Gang Jiang, X. Xue, Leonid Sigal, and Shaogang Gong. 2018. Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine* 35(1):112–125.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 758–764.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences* 115(16):E3635–E3644.

Spandana Gella, Frank Keller, and Mirella Lapata. 2019. Disambiguating visual verbs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(2):311–322.

Spandana Gella, Mirella Lapata, and Frank Keller. 2016. Unsupervised visual sense disambiguation for verbs using multimodal embeddings. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 182–192.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2173–2182.

Ross Girshick. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, pages 1440–1448.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *2013 workshop on Automated knowledge base construction*. Association for Computing Machinery, San Francisco, California, USA, pages 25–30.

Gabriel Grand, Idan Blank, Francisco Pereira, and Evelina Fedorenko. 2022. Semantic projection: recovering human knowledge of multiple, distinct object features from word embeddings. *Nature Human Behaviour* 6.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42(1):335–346.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 770–778.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 961–970.

Kathy Hirsh-Pasek and Roberta Michnick Golinkoff. 2010. *Action meets word: How children learn verbs*. Oxford University Press.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Nancy Ide and Keith Suderman. 2004. The American national corpus first release. In *Fourth International Conference on Language Resources and Evaluation (LREC'04)*. European Language Resources Association (ELRA), Lisbon, Portugal.

Nazli Ikizler, Ramazan Gokberk Cinbis, Selen Pehlivan, and Pinar Duygulu Sahin. 2008. Recognizing actions from still images. In *2008 19th International Conference on Pattern Recognition*. pages 1–4.

Rubén Izquierdo, Armando Suárez, German Rigau, and Ixa Nlp. 2007. Exploring the automatic selection of basic level concepts. In *International Conference Recent Advances in Natural Language Processing*. Borovets, Bulgaria, pages 298—-302.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: with Applications in R*. Springer.

Dinesh Jayaraman and Kristen Grauman. 2014. Zero-shot recognition with unreliable attributes. In *27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press, Cambridge, MA, USA, page 3464–3472.

Andrej Karpathy. 2016. *Connecting images and natural language*. Ph.D. thesis, Stanford University.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. 2012. Leakage in data mining: Formulation, detection, and avoidance. *ACM Trans. Knowl. Discov. Data* 6(4).

Simon Kornblith, Jon Shlens, and Quoc V. Le. 2019. Do better imagenet models transfer better?

Sachin Kumar and Yulia Tsvetkov. 2019. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In *International Conference on Learning Representations*.

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 270–280.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A comparative evaluation and analysis of three generations of distributional semantic models. *Language Resources and Evaluation* 56(4):1269–1313.

Li-Jia Li and Li Fei-Fei. 2007. What, where and who? classifying events by scene and object recognition. In *2007 IEEE 11th International Conference on Computer Vision*. pages 1–8.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*. Springer International Publishing, pages 740–755.

Jingen Liu, Benjamin Kuipers, and Silvio Savarese. 2011. Recognizing human actions by attributes. In *CVPR 2011*. pages 3337–3344.

Daniel Loureiro and Alípio Mário Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 5682–5691.

Daniel Loureiro, Alípio Mário Jorge, and Jose Camacho-Collados. 2022. LMMS reloaded: Transformer-based sense embeddings for disambiguation and beyond. *Artificial Intelligence* 305:103661.

Max M. Louwerse. 2011. Symbol interdependency in symbolic and embodied cognition. *Topics in Cognitive Science* 3(2):273–302.

Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In *Computer Vision – ECCV 2016*. Springer International Publishing, pages 852–869.

Li Lucy and Jon Gauthier. 2017. Are distributional representations ready for the real world? evaluating word vectors for grounded perceptual meaning. In *First Workshop on Language Grounding for Robotics*. Association for Computational Linguistics, Vancouver, Canada, pages 76–85.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers* 28(2):203–208.

Pranava Swaroop Madhyastha, Josiah Wang, and Lucia Specia. 2018. End-to-end image captioning exploits distributional similarity in multimodal space. In *British Machine Vision Conference (BMVC)*.

Jessica Mange, Cécile Sénémeaud, and Alain Somat. 2015. When the "why" makes you socially useful. *Swiss Journal of Psychology* 74(4):197–206.

Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. 2009. Actions in context. *2009 IEEE Conference on Computer Vision and Pattern Recognition* pages 2929–2936.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* 3(29):861.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods* 37(4):547–559.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 26.

George A. Miller, Richard Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography* 3(4):235–244.

Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. 2019. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pages 1–8.

Rachel L. Moseley and Friedemann Pulvermüller. 2014. Nouns, verbs, objects, actions, and abstractions: Local fMRI activity indexes semantics, not lexical categories. *Brain and Language* 132:28–42.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1059–1069.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. MIT Press, Cambridge, MA, USA, volume 14, page 849–856.

Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2013. From large scale image categorization to entry-level categories. In *2013 IEEE International Conference on Computer Vision*. pages 2768–2775.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In *22nd International Conference on Neural Information Processing Systems*. Curran Associates, Inc., volume 22, pages 1410–1418.

Erwin Panofsky. 1955. *Meaning in the Visual Arts: Papers in and on Art History*. University of Chicago Press.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. *Linguistic Data Consortium, Philadelphia, USA* .

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Curran Associates Inc., Red Hook, NY, USA, pages 8024–8035.

Genevieve Patterson, Chen Xu, Hang Su, and James Hays. 2014. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision* 108(1):59–81.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 2227–2237.

Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. 2008. Query-level loss functions for information retrieval. *Information Processing & Management* 44(2):838–855.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training .

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11:2487–2531.

Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Chuck Rosenberg, and Li Fei-Fei. 2015. Learning semantic relationships for better action retrieval in images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 1100–1109.

Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 6517–6525.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, pages 45–50.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 3973–3983.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *14th International Joint Conference on Artificial Intelligence - Volume 1*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'95, page 448–453.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. ImageNet-21K pretraining for the masses. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Marcus Rohrbach. 2017. *Attributes as Semantic Units Between Natural Language and Visual Recognition*, Springer International Publishing, Cham, pages 301–330.

Josef Ruppenhofer, Michael Ellsworth, Myriam Schwarzer-Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. FrameNet II: Extended theory and practice. Working paper, International Computer Science Institute, Berkeley, CA.

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. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3):211–252.

Mohammad Amin Sadeghi and Ali Farhadi. 2011. Recognition using visual phrases. In *Computer Vision and Pattern Recognition (CVPR)*. pages 1745–1752.

Jorge Sánchez and Florent Perronnin. 2011. High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR)*. pages 1665–1672.

R Saxe and N Kanwisher. 2003. People thinking about thinking people the role of the temporo-parietal junction in “theory of mind”. *NeuroImage* 19:1835–1842.

Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. 2013. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(7):1757–1772.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Sara Shatford. 1986. Analyzing the subject of a picture: A theoretical approach. *Cataloging & Classification Quarterly* 6(3):39–62.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, C. Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 440–450.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6(1):1–48.

Gunnar A. Sigurdsson, Olga Russakovsky, and Abhinav Kumar Gupta. 2017. What actions are needed for understanding human actions in videos? In *2017 IEEE International Conference on Computer Vision (ICCV)*. pages 2156–2165.

Carina Silberer, Vittorio Ferrari, and Mirella Lapata. 2017. Visually grounded meaning representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(11):2284–2297.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 721–732.

Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820* .

Mary. Snell-Hornby. 1983. *Verb-descriptivity in German and English : a contrastive study in semantic fields*. C. Winter Universitatsverlag Heidelberg.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 26.

Otfried Spreen and Rudolph W. Schulz. 1966. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns. *Journal of Verbal Learning and Verbal Behavior* 5:459–468.

Mark A. Thornton and Diana I. Tamir. 2022. Six dimensions describe action understanding: The ACT-FASTaxonomy. *Journal of Personality and Social Psychology* 122(4):577–605.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec — A fast and accurate method for word sense disambiguation in neural word embeddings. *CoRR* abs/1511.06388.

Robin R Vallacher and Daniel M Wegner. 1987. What do people think they're doing? action identification and human behavior. *Psychological Review* 94(1):3–15.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605.

David P. Vinson and Gabriella Vigliocco. 2008. Semantic feature production norms for a large set of objects and events. *Behavior Research Methods* 40(1):183–190.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. 2011. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Huiyu Wang, Aniruddha Kembhavi, Ali Farhadi, Alan Yuille, and Mohammad Rastegari. 2019. Elastic: Improving cnns with dynamic scaling policies. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* .

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations (ICLR)*.

Yorick Wilks and Mark Stevenson. 1998. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering* 4(2):135–143.

Michael Wilson. 1988. MRC psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers* 20(1):6–10.

Michael Wray and Dima Damen. 2019. Learning visual actions using multiple verb-only labels. In *British Machine Vision Conference (BMVC)*. pages 1–14.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(9):2251–2265.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pages 3485–3492.

Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. 2011. Human action recognition by learning bases of action attributes and parts. In *2011 International Conference on Computer Vision*. pages 1331–1338.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 5534–5542.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, page 895–906.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. 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.

Felix X. Yu, Liangliang Cao, Rogério S. Feris, John R. Smith, and Shih-Fu Chang. 2013. Designing category-level attributes for discriminative visual recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pages 771–778.

Rowan Zellers and Yejin Choi. 2017. Zero-shot activity recognition with verb attribute induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 946–958.

Li Zhang, Jun Yu Li, and Chao Ching Wang. 2017. Automatic synonym extraction using word2vec and spectral clustering. *36th Chinese Control Conference (CCC)* pages 5629–5632.

Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. 2023. ViTAEv2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *International Journal of Computer Vision* pages 1–22.