Vision and Language: An Application-oriented Perspective

by

Karan Sharma

(Under the Direction of Suchendra Bhandarkar)

Abstract

Vision and language are two of the most critical human faculties. If we are to develop more useful Artificial Intelligence (AI) systems, these modalities will need to work in tandem. Although we are still far from the ultimate goal of synergetic integration of vision and language, several practical applications lying at the intersection of computer vision (CV) and natural language processing (NLP) have experienced a huge upsurge in recent times. This upsurge in the integration of vision and language has been accelerated by recent advances in deep learning and ready availability of both, benchmark and real-world datasets. In this dissertation, we address a few interesting and important applications, such as automated image captioning and classification of objects and actions in images, that lie at the intersection of CV and NLP and have a significant potential impact in important problem domains such as information retrieval and product marketing.

First, we propose an approach to speed up image caption retrieval guided by the top object detected in an image. Second, we propose an approach to classify an action in an image without executing explicit action classifiers on the image. In this approach, we first detect objects in an image and then, with the aid of top objects and associated word embeddings obtained via training on a natural language corpus, we infer the the most probable action in the image. Next, we propose a model to guess objects in an image in situations where the datasets for training classifiers for such objects are unavailable. Finally, we conduct a similarity study on consumer products using both visual and textual features. We believe that these studies and the proposed models will provide practitioners with insights that they could apply in designing AI systems for specific applications.

INDEX WORDS: Deep learning, Computer Vision, Natural Language Processing

Vision and Language: An Application-oriented Perspective

by

Karan Sharma

M.S., Portland State University, 2012

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2018

Vision and Language: An Application-oriented Perspective

by

Karan Sharma

Major Professor:	Suchendra Bhandarkar
Committee:	Tianming Liu Khaled Rasheed

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2018

Vision and Language: An Application-oriented Perspective

Karan Sharma

August 2018

Acknowledgments

First and foremost, I would like to express my sincere thanks to Dr. Suchendra Bhandarkar for his patient guidance over the years, and giving me the freedom to pursue interesting research topics. Your guidance was instrumental in developing my research skills and style. I would also like to thank Dr. Khaled Rasheed and Dr. Tianming Liu for serving as my committee members and teaching me the subtleties of machine learning research.

I would also like to express my gratitude to several external and internal collaborators. I thank Dr. Devi Parikh for introducing me to an interesting and important area of vision and language, and thereby defining my research direction. I thank Dr. Sue Chang for developing my skills in conducting human studies and teaching me attention to detail involved in such studies. Finally, I thank Arun CS Kumar for being an excellent collaborator and team player over the years.

Last, but not least, it was fun working in and around VPCL lab with all my labmates and friends - Somenath, Kyle, Anirban, Vinay, Srijita and Manu.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

Human perception is shaped by two sources of information and knowledge, i.e., *vision* and *language*. If we are to reach the ultimate goal of creating true general-purpose artificial intelligence (AI), these two information and knowledge sources will need to play an all-encompassing and synergetic role. To achieve this goal, significant recent headway has been made in blending the fields of computer vision (CV) and natural language processing (NLP), and as we wait for the ultimate goal of AI to be achieved, it is never too soon to apply recent results and discoveries to practical AI applications.

The paradigm of *Vision and Language* has seen an upsurge in recent times with many practical applications appearing in domains such as automated image captioning. In this dissertation, we study various applications at the intersection of computer vision (CV) and natural language processing (NLP) and propose appropriate models for enabling these applications. We start with automated

image captioning, where we hypothesize that image caption retrieval driven by top-object detection in an image can significantly speed up the captioning process. Next, we propose an approach to predict actions in still images by taking advantage of word embeddings. In a related idea, inspired by the *Fast and Frugal Heuristic* paradigm from decision making, we propose a model that makes decisions based on the detection of a single object in an image, i.e., we classify actions in an image using only the top-object detection results, and then discuss why and when it would be advantageous to use such an approach. In addition, we propose a framework that can guess the presence of other objects in an image from existing object detections (i.e., from context), thus obviating the need to generate comprehensive training datasets. Last but not least, we study how human perceptions of similarity are correlated with visual and textual deep learning features.

A detailed organization of the dissertation is as follows. In the second chapter, we address the problem of image caption retrieval. Given a test or query image, the goal to assign an appropriate caption to the image. For this purpose, the approach of Devlin et al. [5] retrieves the nearest-neighbor images to the given test image using deep learning features where each of the nearest-neighbor images has a caption associated with it. Using the nearest-neighbor image captions, Devlin et al. find a consensus caption or sentence and assign it to the test image. However, this approach is slow because one needs to compare the test image with all training images. We hypothesize that if the nearest-neighbor caption retrieval were driven by top-object detection in the test image, the resulting CPU time savings could be significant.

In the third chapter, we explore the paradigm of action classification in still images. This work, inspired by the verb centrality hypothesis [2], assumes that a verb acts as a binding force between nouns. In other words, a verb defines a relationship between the nouns in a sentence. Hence, if we can detect objects in an image, and infer verbs using language models trained on textual data, we could reliably detect actions in still images, which do not possess the temporal information available in videos. Previously action recognition in still images had been addressed using pose estimation [11], parts-based detection [4], or detection of human-object interactions [12]. However, these approaches require explicit training of action classifiers, which are hard to train for variety of reasons. Our approach obviates the need to train explicit action classifiers, and infers actions from objects detected in an image using word embeddings trained on textual data.

Building on the ideas in the previous paragraph, in the second part of the third chapter, we explore the problem of human activity classification in still images. If an AI agent is asked to judge an action or situation in an image, and it has not been previously trained to judge that action, how would it guess the action? The agent's options are few, but one thing it could do is to identify a significant object in the image and place a bet that the action taking place in the image is one that is most likely to involve the identified object in the real world. Inspired by the *Fast and Frugal heuristic* paradigm in decision theory, we propose that this strategy for guessing an action from a single top-object detection, in the context of human activity classification, is likely to succeed for a wide variety of real-world images.

In the fourth chapter, we explore the paradigm of guessing objects in an image despite not training the relevant classifiers for those objects. We start by clustering word embeddings of objects in which each cluster tends to correspond to objects that co-occur in the real world. We select from the cluster the object that is most representative of that cluster. In an ideal world, this object will have the most training data, whereas other objects within the cluster will have less (or no) training data. Hence, once having trained the classifiers for these representative objects, we run these classifiers on the test images and try to guess the other objects in the images. Our results show that we are able to guess objects in an image reliably for a significant number of situations. In the pre-deep learning era, the context surrounding the object (i.e., denoted by the presence of other objects in the image or by the overall scene characteristics) was used to aid the performance of object recognition systems [3, 7, 8]. Inspired by these approaches, we believe contextual information could be used to guess objects in an image, in situations where the classifier training data for those objects is not available.

In the fifth chapter, we explore the application of a variety of visual and textual features to the product marketing arena. Currently, marketing executives in the industry have to make decisions regarding how similar or dissimilar their products are to competitive products. Such decisions are made manually by looking at product images and product descriptions. It would be beneficial if this process could be automated. To automate this process, it would be crucial to know how different types of features in the image domain and textual domain correlate with human perceptions of similarity. We conduct studies to show that various features

obtained from consumer product descriptions, in both the textual domain and visual domain, are indeed well-correlated with human perceptions of similarity. Previously, various authors have conducted studies in areas such as image popularity [1, 10, 13], image virality [6], and image interestingness [9]. However, we believe that although these areas are relevant to the field of marketing, conducting visual and textual similarity studies on consumer products is more relevant to the broader goals of marketing executives.

Overall, the goal of this dissertation was to address problems that lie at the intersection of vision and language. In the near future, integration of vision and language will be an important aspect of AI systems. Nevertheless, how vision and language will be integrated to meet goals of AI is still not well understood. However, the time is ripe for addressing practical issues in this regard. We addressed several problems: speeding up the process of automated image captioning via nearest-neighbor caption retrieval, action prediction in still images without executing explicit action classifiers, predicting objects in an image when the relevant training sets are not available, and the study of product similarity in a marketing context. We believe that the results from our studies will help in several other domains such as weakly supervised object and action recognition, automated general image captioning, and product branding for marketing.

Bibliography

[1] S. Cappallo, T. Mensink, and C. G. Snoek (2015). Latent factors of visual popularity prediction. In Proceedings of the 5th ACM International Conference on Multimedia Retrieval (pp. 195-202). ACM.

[2] W. L. Chafe. (1970). Meaning and the structure of language. Chicago: University of Chicago Press.

[3] M. Choi, A.Torralba, and A. S. Willsky (2012). A Tree- based Context Model for Object Recognition IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(2), (pp. 240-252).

[4] V. Delaitre, J. Sivic, and I. Laptev (2011). Learning person-object interactions for action recognition in still images. Proc. NIPS 2011, (pp. 1503-1511).

[5] J. Devlin et al. (2015). Exploring nearest-neighbor approaches for image captioning. arXiv preprint arXiv:1505.04467.

[6] A. Deza, and D. Parikh (2015). Understanding image virality. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1818-1826).

[7] S. K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros and M. Hebert (2009). An Empirical Study of Context in Object Detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[8] C. Galleguillos, A. Rabinovich, and S. Belongie (2008). Object categorization using co-ocurrence, location and appearance, Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[9] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and Van Gool, L. (2013). The interestingness of images. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1633-1640).

[10] A. Khosla, A. Das Sarma, and R. Hamid. (2014). What makes an image popular? In Proceedings of the 23rd international conference on World wide web (pp. 867-876). ACM.

[11] S. Maji, L. Bourdev and J. Malik (2011). Action recognition from a distributed representation of pose and appearance. Proc. IEEE Intl. Conf. CVPR, (pp. 3177-3184).

[12] G. Sharma, F. Jurie and C. Schmid, (2017). Expanded parts model for semantic description of humans in still images. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 39(1), (pp. 87-101).

[13] S. Zakrewsky, K. Aryafar, and A. Shokoufandeh. (2016). Item Popularity Prediction in E-commerce Using Image Quality Feature Vectors. arXiv preprint arXiv:1605.03663.

Chapter 2

Automated Image Captioning Using Nearest-Neighbors Approach Driven by Top-Object Detections

2.1 Chapter Outline

The significant performance gains in deep learning coupled with the exponential growth of image and video data on the Internet have resulted in the recent emergence of automated image captioning systems. Two broad paradigms have emerged in automated image captioning, i.e., generative model-based approaches and retrieval-based approaches. Although generative model-based approaches that use the recurrent neural network (RNN) and long short-term memory (LSTM) have seen tremendous success in recent years, there are situations in automated image

captioning for which generative model-based approaches may not be suitable and retrieval-based approaches may be more appropriate. However, retrieval-based approaches are known to suffer from a computational bottleneck with increasing size of the image/video database. With an aim to address the computational bottleneck and speed up the retrieval process, we propose an automated image captioning scheme that is driven by top-object detections. We surmise that by detecting the top objects in an image, we can prune the search space significantly and thereby greatly reduce the time for caption retrieval. Our experimental results show that the time for image caption retrieval can be reduced without suffering any loss in accuracy.

2.2 Introduction

Automated image captioning, i.e., the problem of describing in words the situation captured in an image, is known to be challenging for several reasons. The recent significant performance gains in deep learning coupled with the exponential growth of image and video data on the Internet have resulted in the emergence of *automated* image captioning systems. Two broad paradigms have emerged in the field of automated image captioning, i.e., generative model-based approaches [3], [5], [9], [11], [17] and retrieval-based approaches [1]. Although generative model-based approaches that use the recurrent neural network (RNN) and long short-term memory (LSTM) have seen tremendous success in recent years, there are situations for which retrieval-based approaches may be better suited.

Examples of such situations include:

(1) Situations wherein the training sets are dynamically changing. To keep up with the increasing pace of visual data being constantly uploaded on the Internet, computer vision practitioners face a challenging task of training models that are capable of adapting to constantly changing datasets or reducing the size of the datasets. By reducing the size of the datasets, one runs the risk of discarding useful data resulting in the learning of simplistic models. Adaptive models have the added overhead of requiring constant training or retraining as the underlying datasets change over time. Moreover, adaptive models need to deal with the problem of *concept drift*, i.e., situations where the statistical properties of the target variable or concept, which the model is trying to predict, change over time in unforeseen ways, especially when the new data being uploaded is significantly different from previously observed data. In contrast, retrieval-based approaches, modeled on nearest-neighbor search, do not entail the overhead of constant retraining of models since one can store all the images in the dynamically changing dataset in a database.

(2) Situations wherein one needs to deploy an automated image captioning system with the goal of simultaneously reducing system development time and CPU execution time. Nearest-neighbor approaches lend themselves easily to rapid implementation and deployment since they have very few tunable hyperparameters compared to other approaches. Hence retrieval-based approaches based on nearest-neighbor search are naturally preferred in rapid prototyping situations. However, the potential downside of retrieval-based approaches is that nearest-neighbor

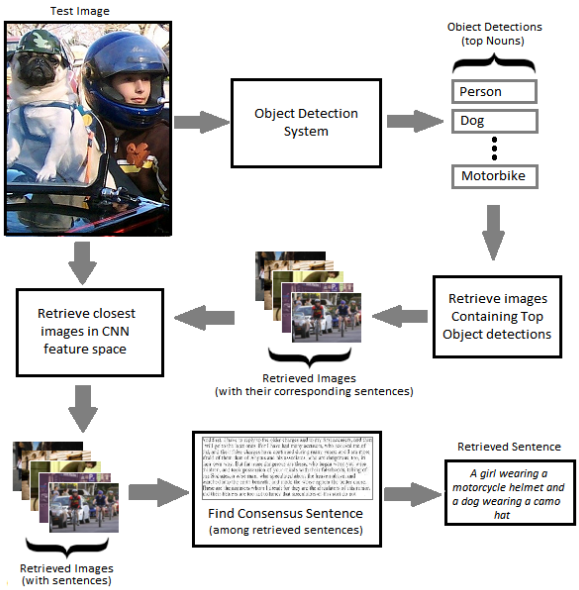

Figure 2.1: Top-object detections used to drive k-NN search.

search can be exhaustive if one has to perform all possible comparisons between the query image and the database entries. Traditionally, techniques such as locality sensitive hashing (LSH) have been used to speed up nearest-neighbor search. However, effective use of LSH requires the proper tuning of several hyperparameters in order to achieve accurate results. In contrast, the proposed approach has very few tunable hyperparameters and hence a much less computationally intensive hyperparameter tuning phase.

Although retrieval-based approaches to automated image captioning have not been as successful as generative model-based approaches, the performance of retrieval-based approaches has been observed to be not very far behind that of RNN- and LSTM-based approaches when addressing the *Microsoft Common Objects in Con-*

text (MS COCO) challenge. Therefore the obvious question arises - in situations (such as the ones described previously) where retrieval-based approaches are called for, how does one speed up the nearest-neighbor search procedure? To this end, we propose a variant of the nearest-neighbor search procedure to speed up image caption retrieval using top-object detections. Specifically, we use the detection of the most significant objects in an image (i.e., the top objects) to speed up the k-nearest-neighbor (k-NN) search for retrieval-based automated image captioning. Although, as noted previously, approaches such as LSH can used to accelerate the retrieval process, LSH entails a hyperparameter tuning procedure that is computationally complex and difficult to implement thereby calling for a significant expenditure of programmers' development time.

2.3 Related Work

Automated image captioning: Automated image captioning systems have grown in prominence owing, in large part, to the tremendous performance gains shown by deep learning in recent times. Existing automated image captioning systems can be categorized as either generative model-based or retrieval-based. Generative model-based systems involve correctly identifying objects, verbs, adjectives, prepositions or visual phrases in an image and generating a caption from these or directly from the representation of the image [3], [5], [9], [11], [17]. Retrieval-based approaches [1], on the other hand, involve retrieving the most suitable caption from a database of captions and assigning it to an image. Currently, generative

model-based approaches that use the RNN and LSTM have been shown to yield the best performance metrics in the context of automated image captioning; however retrieval-based approaches have also proved to be quite competitive in terms of performance. The generative model-based and retrieval-based paradigms are each suited for different kinds of situations and applications. However, in situations where retrieval-based approaches are more appropriate and successful, we propose a scheme to optimize and speed up the caption retrieval process by exploiting the top-object detections in the image.

2.4 Motivation

Although generative model-based approaches that use the RNN and LSTM are regarded as the state-of-the-art in automated image captioning, there are potential situations for which they may not be well suited and hence retrieval based captioning approaches may be called for. However, retrieval-based approaches to automated image captioning can be computationally intensive and slow especially when a query image is compared with all the images stored in the database. However, before we proceed to address the question of how to speed up retrieval-based approaches to automated image captioning, we digress to answer an important related question, i.e., under what potential situations would retrieval-based approaches have an advantage over state-of-the-art generative model-based approaches that use the RNN and LSTM in the context of automated image captioning?

Concept Drift: Consider the problem of automated image captioning in situa-

tions where the underlying datasets are dynamically changing such as when visual data (in the form of images and videos) is being uploaded over the Internet at an extraordinary pace, both on popular online social media (OSM) platforms such as Facebook, Instagram, Snapchat, Google and Twitter, and on websites that contain more structured and specific information such as those dealing with news, sports, art, and technology. Many of the uploaded images and videos have some sort of textual information associated with them, typically in the form of tags, captions, and/or comments. Mining such a large data set is tremendously challenging for most computer vision practitioners. The constant pace of the dynamically changing dataset makes it incredibly difficult to learn reliable computer vision models. The standard assumption underlying most machine learning techniques is that the training data will be similar to the testing or querying data. However, in dynamic situations where the underlying data is continuously changing, it is especially hard to train reliable models. In this paper, we propose a retrieval-based model for automated image captioning for situations wherein the training datasets are very volatile and constantly changing.

One of the problems faced when dealing with dynamic datasets is the problem of *concept drift* where the function learned by a machine learning model is rendered not particularly useful for newly arriving data. For example, near Christmas, people tend to post more pictures or images of their activities around a Christmas-oriented theme on OSM sites. An existing machine learning model may not be in situation to automatically label or caption these images since it has not seen these images previously. One solution is to constantly retrain the existing model

as the new data arrives or use models that are capable of adapting to new data. However, the constant retraining of models could pose significant and, in some cases, an impossibly high computational demand, especially in situations where images are being uploaded at a very rapid pace. Moreover, many adaptive models, in the interest of computational efficiency, subsample the data during retraining. The discarding of data could lead to the learning of overly simplistic models. The interested reader is referred to the work of Gama et al. [7] for a more detailed and comprehensive treatment of the concept drift problem.

For the reasons mentioned above, some of the most popular automated image captioning schemes, based on generative models that use the RNN and LSTM, are seriously disadvantaged in situations where a large proportion of the training data is in a state of constant flux. In such instances, the generative models will learn a classification or prediction function that could account for most cases, but may miss cases that occur only a few times. Moreover, the cases that occur infrequently may contain valuable information. For example, if the training set has millions of images, and only five instances of *Man is biting a dog*, the generative model may simply ignore this infrequent case during the training process, although the case may be of potential interest. Hence, for this reason and reasons described in previous paragraph, generative models are not well suited for image captioning under dynamically changing training datasets. However, retrieval-based approaches, such as ones based on k-NN search do not suffer from such problems. It has been convincingly shown by Hays and Efros [8] that k-NN search is one of the most effective retrieval algorithms if one has a very large dataset. However, exhaus-

tive k-NN search could be computationally very expensive. Although techniques such as LSH have been traditionally used to speed up k-NN search-based image retrieval [4], the hyperparameter tuning procedure needed to optimize the performance of LSH is non-trivial in terms of its computational complexity [4]. The situation is further complicated if we need to retune the LSH procedure in the face of constantly arriving new training data. Thus, retrieval-based automated image captioning techniques suffer from the same disadvantages as their generative model-based counterparts if the former use k-NN search optimized via LSH. In this paper, we propose a simple retrieval-based technique for automatic image captioning that is accurate, reliable and computationally efficient. The proposed technique is based on enhancing the k-NN search by exploiting the top-object detections in an image.

Rapid Prototyping: We use top-object detections to speed up the caption retrieval procedure during automated image captioning. Specifically, we use the detection of the most significant objects in an image (i.e., the top objects) to speed up the k-NN search for retrieval-based automated image captioning. We show top-object detection to be a preferable alternative to the more conventional retrieval-based automated image captioning methods that employ LSH to speed up the k-NN search. It is to be noted that although techniques such as LSH can be used to speed up k-NN search-based image retrieval, the hyperparameter tuning procedure needed to optimize the performance of LSH is non-trivial in terms of computational complexity [4], especially in the case of complex applications such as automated image captioning. Thus, complete automation of the LSH procedure

for automated image captioning is a challenging task. Implementation and proper tuning of LSH also presents a significant expenditure of system development time, which is an important consideration in real-world situations where rapid prototyping is called for.

2.5 k-NN Search Driven by Top-Object Detections

Previously, Devlin et al. [1] have obtained good results for automated image captioning based on k-NN search-based image retrieval. Their approach determines the k-NN images by computing a measure of image similarity between the test/query image and each of the database images. The test/query image is then assigned the caption obtained by computing the consensus of the retrieved k-NN image captions. Performing an exhaustive search of the image database to retrieve the k-NN images using an image feature-based similarity metric is clearly not a scalable approach. We show that, in the context of automated image captioning, by detecting all objects in a test image, selecting the top-n objects (where n is a small number) and retrieving all images that contain at least one of these n objects, one can achieve results comparable to those of k-NN retrieval via exhaustive search while simultaneously obtaining a significant speedup. Fig. 2.2 summarizes the proposed approach. We demonstrate our approach on the MS COCO dataset as a proof of concept. We believe the experimental results on the MS COCO dataset are transferable and generalizable to real-world dynamic datasets. Although the proposed approach involves tuning the parameters of a support vector

machine (SVM)-based classifier for object detection/recognition, it is computationally much less expensive than the LSH hyperparameter tuning procedure used to optimize k-NN search and also yields readily to automation.

Although running various object (i.e., noun) detectors on the test/query image imposes a computational overhead, it is offset by the following considerations: (a) the space of objects (i.e., nouns) is bounded. Also, since objects are concrete entities, generating training sets for object detectors is not very difficult if one uses web-based data coupled with crowdsourcing, (b) sliding windows are not used during the object detection procedure, i.e., the entire test/query image is fed as input to the SVM-based object detector. The computational overhead of object detection in the test/query image is also offset by: (a) the resulting speedup over k-NN image retrieval via exhaustive search and, (b) savings in development time compared to the scenario wherein k-NN image retrieval is optimized using LSH. Additionally, the proposed approach also results in significant savings in CPU execution time as shown in Table 2.1.

Complexity Analysis: Given a set of objects $X = \{x_1, x_2, \ldots, x_n\}$, and a set of images $I = \{I_1, I_2, \ldots, I_m\}$, we make the following assumption regarding the dataset: Each object x_i does not occur in more than k images in the dataset where $k \ll m$. In real world datasets, especially in large datasets, it is expected that no single object category will dominate the images in the dataset. Even generic categories such as *person, car, ...*, would be expected to occur in a significantly small percentage of the total number of images in the dataset. Also, for a small subset $Y \subset X$ where no member of Y occurs in more than r images ($r \ll m$)

in the dataset, the number of comparisons is bounded by $r \cdot |Y|$ resulting in a $O(r \cdot |Y|)$ time complexity. However, what if r is a large number. We argue that in datasets that are sufficiently representative of real world, this will not be the case. For example, consider an image whose top detections are *person, dog, road,* and *building*. Intuitively, in a large dataset representative of many nouns and concepts in the world, we can expect that all the images that contain at least one entity from the set { *person, dog, road, building*} are far fewer than all the images in the dataset thus resulting in an order of magnitude reduction in search complexity.

Retraining Event Analysis: Assume an image dataset (with associated captions for each image) of size N (i.e., N is the number of data points). Assume this dataset is being constantly augmented with new incoming image data (and the associated captions). Assume that after every w data points (i.e., images) are added to the dataset, there is a concept drift, that requires retraining of the model. In a traditional generative model-based system that uses an RNN, retraining will be needed in two situations after the addition of new data points to the existing dataset:

(a) Changes in concepts, where a concept is any word, which includes nouns, verbs, adjectives and so on. Assume that the concepts change at an average rate of c concepts after w new data points are introduced. Clearly, the space of concepts is far greater than the space of objects (i.e., nouns). Let $tr(c)$ denote the average number of training events required to account for the concept changes after a collection of w new data points is added to the existing dataset.

(b) Changes in concept dependencies. The dependency between two words is a

measure of how much a given word depends on the other word. For example, the word *eating* is dependent on the words *person* and *food*. We need to retrain the model to learn such dependencies after a collection of w new data points is introduced. Assume that the concept dependencies change an average rate of d concept dependencies upon introduction of a collection of w new data points. Again, based on our understanding of the real world, the space of these dependencies is significantly larger than the space of objects alone. Let $tr(d)$ denote the average number of training events required to account for the changes in concept dependencies after a collection of w new data points is added to the existing dataset.

In contrast to a traditional generative model-based system, in the proposed approach, the training events will be required only when new objects are introduced at an average rate of ob objects after w new data points are added to the existing dataset. Clearly, the training events are bounded by the number of objects under consideration. Let $tr(ob)$ denote the average number of training events required after w new data points are added to the existing dataset. Based on our knowledge of the real world and the above arguments, the training events in the proposed approach will be significantly fewer than the training events in a traditional generative model-based system (such as one that uses an RNN), i.e., $tr(ob) \ll tr(c) + tr(d)$.

2.6 Experimental Results

Training: For the purpose of training, we use 80 annotated object categories in the MS COCO dataset [10]. Binary SVM classifiers are trained for each of these 80 annotated categories using VGG-16 *fc-7* image features [13], and the SVMs are calibrated using Platt scaling. For the extraction of fc-7 features, Matconvnet package [15] is employed.

In addition, we store each training image in the MS COCO dataset and its accompanying sentences (5 sentences per image) in our database. We treat these sentences as ground truth captions for the corresponding training image. For testing purposes, we consider the MS COCO validation set consisting of close to 40,000 images.

Testing: For each test image in the MS COCO validation set, we run all the 80 object detectors on the test image. We select the top-n objects from all the detected objects in the image. In our current implementation $n = 5$. The detected top objects are the ones that are deemed to possess the highest probability of occurrence in the image. The probability of occurrence of an object in the image is computed by mapping the classification confidence value generated by the SVM classifier for that object to a corresponding probability value using Platt scaling [12]. From the training dataset, we retrieve all images that contain at least one of the top-n objects detected in the previous step, using the corresponding ground truth captions, i.e., a training image is retrieved if at least one of its associated ground truth captions contains a noun describing the object under consideration. In addition, for the purposes of retrieval, all the synonyms for certain words such as *person*

(synonyms are man, woman, boy, girl, people, etc.) are taken into consideration. Using the cosine distance between the *fc-7* features of each retrieved image and the test image, we select the k-NN images for further processing.

In the current implementation we have chosen $k = 90$ as recommended by [1]. Since each of the k-NN images has 5 associated sentences (captions), we have a total of $5k$ potential captions for the test image. We determine the centroid of the $5k$ potential captions and deem it to represent the consensus caption for the test image. The consensus caption is then assigned to the test image in a manner similar to [1]. The BLEU measure is used to evaluate the similarity (or distance) between individual captions and to determine the centroid of the $5k$ potential captions. We have also implemented image retrieval using exhaustive k-NN search [1] and compared the CPU execution time of the proposed approach with that of image retrieval using exhaustive k-NN search for 2000 random images

.

Results: As shown in Table 2.1, the proposed image retrieval, using k-NN search driven by top-object detections, and the standard image retrieval, that employs exhaustive k-NN search, yield very similar results when the BLEU and CIDEr [16] similarity metrics are used to compare the retrieved captions.

The proposed approach is seen to yield significant gains in CPU execution time when compared to image retrieval using exhaustive k-NN search. Essentially, the proposed image retrieval technique based on k-NN search driven by top-object detections is observed to provide an attractive alternative to LSH for the purpose of speeding up k-NN search-based image retrieval in the context of automated

Table 2.1: Comparison of image captioning results obtained using the proposed approach for image retrieval based on k-NN search driven by top-object detections (Obj-k-NN) and those obtained using conventional image retrieval based on exhaustive k-NN search (Exh-k-NN).

	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	CIDEr	CPU time
Exh-k-NN	65.6%	47.4%	34%	24.7%	0.70%	2.5e+04s
Obj-k-NN	64.6%	46.2%	32.8%	23.6%	0.68%	1.17e+04s

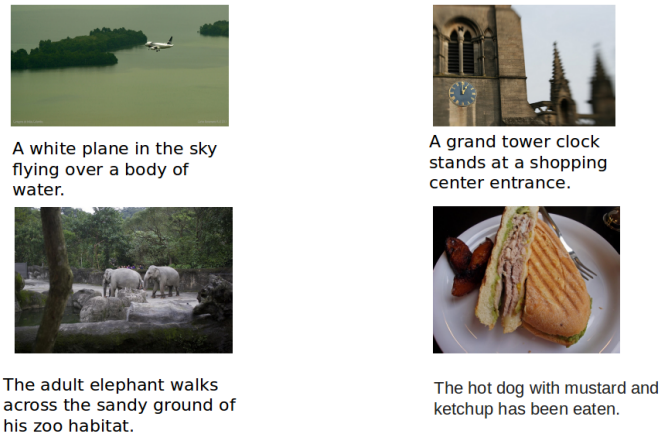

Figure 2.2: Qualitative Results for nearest neighbor driven by top-object detections. Some captions retrieved accurately describe the image while others are partially correct.

image captioning. As a proof of concept, the results of the proposed image retrieval technique based on k-NN search driven by top-object detections on the MS COCO dataset are fairly convincing. We believe that these results could be directly transferred to real-world datasets that are dynamically changing.

These results show that k-NN search driven by top-object detections, even though simple in concept, can provide significant gains in critical situations where the datasets are dynamically changing. This approach requires that we store all the training images along with their associated captions in the database. When dealing with real-world problems, we will store all the image instances in the database and retrieve the relevant images from the database using top-object driven k-NN search. There are three advantages to the proposed approach: We do not need to subsample the dataset by discarding any potentially useful information, we do not need to exhaustively search for the k-NN images, and we do not need to retrain the retrieval models in the face of changing information.

2.7 Conclusions

We have shown that retrieval-based approaches for automated image captioning could be made computationally more efficient if they are driven by top-object detections. The potential advantages of our approach are in situations where the underlying datasets are changing dynamically. In addition, the proposed approach needs much less parameter tuning when compared to the computationally intensive hyperparameter tuning associated with traditional LSH-based optimization

of k-NN search. The proposed approach is a natural candidate for use in rapid prototyping conditions that also call for optimization of CPU time.

Bibliography

[1] Devlin, J. et al. (2015). Exploring nearest-neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*.

[2] Devlin, J. et al. (2015). Language models for image captioning: The quirks and what works. *Proc. ACL 2015*.

[3] Donahue, J. et al. (2014). Long-term recurrent convolutional networks for visual recognition and description. *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (CVPR 2014).

[4] Dong, W. et al. (2008). Modeling lsh for performance tuning. *Proc. ACM Conf. Info. & Know. Mgmt.*, October, pp. 669-678.

[5] Fang, H. et al. (2014). From captions to visual concepts and back. *arXiv preprint arXiv:1411.4952*.

[6] Farhadi, A. et al. (2010). Every picture tells a story: Generating sentences from images. *Proc. Eur. Conf. Comp. Vis.* (ECCV 2010), pp. 15-29.

[7] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys* (CSUR), Vol. 46(4), pp. 44.

[8] Hays, J., and Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics* (TOG), Vol. 26(3), pp. 4, August.

[9] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (CVPR 2015).

[10] Lin, T. Y. et al. (2014). Microsoft COCO: Common objects in context. *Proc. Eur. Conf. Comp. Vis.* (ECCV 2014), pp. 740-755.

[11] Mao, J. et al. (2014). Explain images with multimodal recurrent neural networks. *Proc. NIPS 2014*.

[12] Platt, J. (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, Vol.10(3), pp. 61-74.

[13] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Proc. Intl. Conf. Learn. Rep.* (ICLR 2014).

[14] Slaney, M. et al. (2012). Optimal parameters for locality-sensitive hashing. *Proc. IEEE*, Vol. 100(9), pp. 2604-2623.

[15] Vedaldi, A. & Lenc, K. (2015). MatConvNet-convolutional neural networks for MATLAB. *Proc. ACM Conf. Multimedia Systems* (MMSys 2015).

[16] Vedantam, R. et al. (2015). Cider: Consensus-based image description evaluation. *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (CVPR 2014).

[17] Vinyals, O. et al. (2014). Show and tell: A neural image caption generator. *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (CVPR 2014).

[18] Yang, Y. et al. (2011). Corpus-guided sentence generation of natural images. *Proc. Conf. EMNLP*, pp. 444-454.

Chapter 3

Action Recognition using Natural Language Processing

3.1 Chapter Outline

Detecting actions or verbs in still images is a challenging problem for a variety of reasons, such as the absence of temporal information and polysemy of verbs, all of which lead to difficulty in generating large training datasets for verbs. In this paper, we propose to first detect the prominent objects in the image and then infer the relevant actions or verbs using Natural Language Processing (NLP)-based techniques. The proposed scheme obviates the need for training and using visual action detectors on images, an approach which tends to be error-prone and computationally intensive. This paper provides a valuable insight in that the detection of a few significant or prominent (i.e., top) objects in an image allows

one to extract or infer reliably the relevant actions or verbs in the image.

To this end, in the first part of this chapter, we propose an approach wherein we classify the top-2 most probable objects in an image, and infer the actions based on these top-2 objects using NLP models trained on the captions accompanying the images. For the NLP-based approaches, we rely on the *word2vec* and the *Object-Verb-Object triplet* models for predicting the actions from the top-object detections and also analyze their nuances. Our experimental results show that verbs can be reliably and efficiently detected by exploiting the top object detections in an image.

In the second part of this chapter, we extend the above ideas to situations dealing with human activity recognition under uncertainty, where the term "uncertainty" implies that the system has access to neither pretrained action classifiers nor action labels in its training set. The system simply knows the identities of objects in an image and is required to determine an action based on the identity of the most probable object without access to language models trained on captions that accompany the images in its training set. The techniques proposed in the first part of this chapter, where we learn language models from the captions that accompany the images, could be considered to fall within the camp of *optimization-based methods* since we use extraneous information (in the form of closed world language models trained on caption datasets) to reach the best possible solution. However, in the second part of this chapter, we propose an action recognition technique which is based on the *Fast and Frugal heuristic* (as opposed to being optimization-based). In the proposed action recognition technique we use a zero-shot method to infer the underlying activity or action based on the identity of a

single object (other than a person) in the image by taking advantage of language models trained on a generic text corpus. Here, the term ”zero-shot” implies that we do **not** use any action labels in the training set, however, the object classifiers are trained using the object labels in the training set. Our experiments show that language models trained on a generic text corpus (such as Facebook's FastText which is trained on Wikipedia) are capable of inferring the most appropriate verb, action or activity that could be associated with the identity of the most probable object (other than a person) in the underlying image.

3.2 Action Recognition in Still Images Using Word Embeddings from Natural Language Descriptions

3.2.1 Introduction

Action recognition in still or static images is a challenging problem for a variety of reasons such as the absence of temporal information and the polysemy of verbs which leads to difficulty in generating large verb datasets. In addition, learning high quality verb detectors or classifiers can be difficult because the corresponding decision boundaries are often non-linear and unclear. For example, in the case of the verb *playing*, one would need to train the corresponding detector on a very large dataset comprising of tennis images, baseball images, images of a child playing with a toy and so on. The underlying visual patterns in the images that contain the action of interest may be very hard to capture reliably, thus making

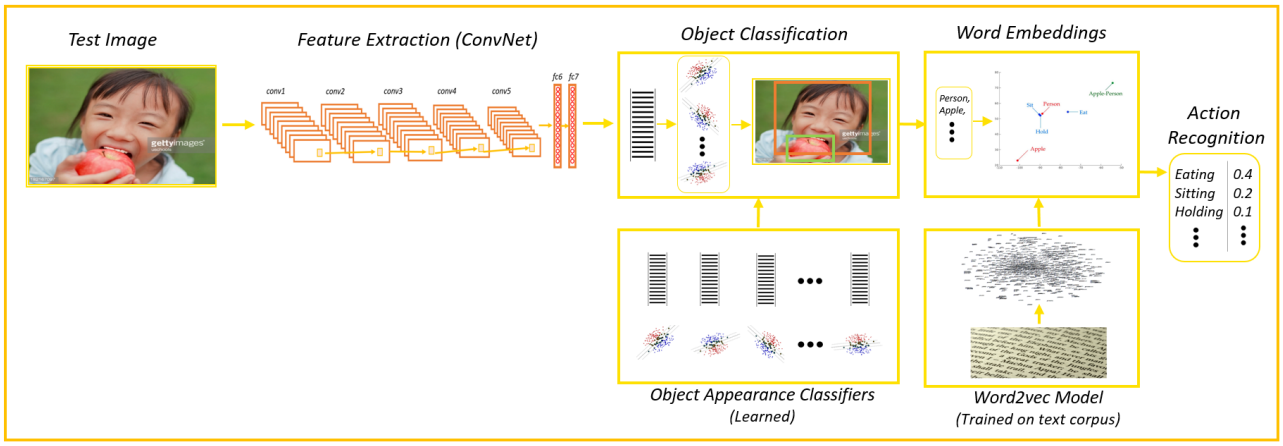

Figure 3.1: Deep learning features are extracted for an input test image using a Convolutional Neural Network (ConvNet), followed by SVM-based classification for each object category. The top-objects are mapped into the word embedding space. The action is inferred from the top-objects in the word2vec space.

it extremely difficult to learn an accurate action classifier. However, if one can reliably determine that there is a *person* and a *tennis racket* in an image, then one can also infer that the corresponding verb is *playing* or *hitting* with high certainty. In this paper, we draw upon the above intuition to detect verbs in a still image without having to explicitly train and employ visual verb detectors. We propose to identify the significant activity (i.e., verb) in an image by taking advantage of the highly plausible object-pair detections in an image.

By drawing upon the recent advances in Natural Language Processing (NLP), we provide a valuable insight in that with reliable detection of the significant or prominent (i.e., top) objects in an image, we can predict the corresponding verb in an image without having to employ visual verb detectors explicitly on the input

image.

We believe that the proposed approach could be potentially useful in various scenarios, especially:

(1) Situations where annotated verb datasets are not available. In this paper, as a proof of concept, we use the sentences in the *Microsoft Common Object in Context* (MS COCO) dataset [25] for training NLP-based models to enable action recognition in still images. However, we believe our results could be extrapolated to situations where ground truth verbs for images are not available and the models need to be learned from a general real-world text corpus such as Wikipedia.

(2) Design of software applications (i.e., apps) for mobile user devices such as smartphones and tablets that are typically resource constrained. Our approach has the advantage of obviating the use of computationally intensive visual activity (i.e., verb) detectors on the input image. In contrast, the detection of significant object-pairs and the corresponding verb assignments in an input image can be made both, reliably and efficiently using the proposed NLP approaches based on word embeddings. The extraction of deep learning features from an input image is, computationally, the most expensive stage in deep learning architectures. As a result, substantial current research effort has focused on speeding up the deep learning feature extraction stage without loss in accuracy with the goal of making the deep learning architecture deployable on resource-constrained mobile devices [1, 21, 22, 23, 24]. In this paper, we offer an alternative perspective that obviates the need to train and deploy separate visual activity detectors. In future, the computationally efficient extraction of deep learning features in conjunction

with the computational savings of NLP model-based action detection could greatly enhance automated image captioning and image annotation systems.

Most previous work has focused on action recognition in video frames [5], with some work on zero-shot action recognition in videos [2, 17, 19, 42]. Most papers that deal with activity recognition in static images, rely on either pose estimation [26], parts-based detection [6], detection of human-object interactions [35] or weakly supervised learning in multi-modal settings [9, 10]. We contend that the need to learn the computationally complex verb detectors described in [6, 26, 35] even in the case of static images can be obviated by just predicting the verb using the top-object detections and NLP-based word embeddings. In addition to computational efficiency and accuracy, we show that the proposed approach has the crucial advantage of conceptual simplicity. Xu et al. [42] exploit word embeddings in a zero-shot framework for action recognition in videos using computationally intensive transductive manifold learning techniques. In contrast, the proposed approach uses word embeddings directly and efficiently to infer the underlying action in static images via detection and recognition of the top objects in the image.

3.2.2 Motivation

We surmise that the top-object detections in an image have enough information for one to reliably infer the relevant actions in the image. First, given the current state-of the art there are pragmatic reasons why such an approach would be practically useful. Second, there are certain inherent properties of the visual and linguistic worlds that make action prediction based on the detection of top objects intuitively

appealing. We list the following reasons for why we believe that action recognition models in static images should be driven by top-object detections:

(1) Objects are cohesive structures, in that, concrete objects, even when malleable, are held together by percepts or parts which gives an object a property of wholeness [13]. This property of wholeness is not a characteristic of relational categories such as verbs. It is this cohesion that makes it fundamentally easy to recognize objects using visual features. On the other hand, correctly recognizing a relational category is often very challenging because it involves recognizing a relationship between disjointed elements [13].

(2) Objects map directly to concrete entities in the world [13]. This direct translation between nouns and objects makes possible a stable assignment of a word to an object in an image. On the other hand, relational categories such as verbs tend to describe relations between disjoint entities. These relational categories may not map directly to the various object configurations present in an image.

(3) Verbs are more polysemous than nouns [11] in that verbs have more senses of meaning than nouns. The verb *sitting* can be used in a variety settings. For example, in the sentences *"John is sitting on a chair."*, and *"The apple is sitting on a table."*, the word *sitting* is used in different senses. This polysemy amongst verbs can make the generation of training datasets for verb detectors very hard.

(4) Verbs are mutable in that verbs adjust themselves in meaning to the context of sentence [12]. For example, in the sentence, *"A person is jumping on pizza."*, the verb *jumping* adjusts its meaning to the nouns in the sentence. This makes it even harder to automatically create large verb training datasets for verb detectors.

(5) On the more pragmatic side, object classification has reached a stage where one can reliably detect objects in an image. Object classification systems have shown impressive results on several challenging datasets in the computer vision research community. As of yet, comparable success has not been met by systems for automatic detection and classification of actions, visual phrases and attributes in still images.

3.2.3 Related Work

Action (verb) recognition

Action recognition in videos is a relatively easier problem and has been addressed by using various spatio-temporal descriptors followed by a classifier [7, 31, 41]. However, in static images, unlike videos, there is no temporal information. Also, the problem is further exacerbated by the lack of reliable annotation data. Hence, several existing works in action recognition on static images use supervised learning of visual information integrated with linguistic information mined from a text corpus [8, 19, 43]. More recently, Gao et al. [9, 10] used word embeddings in conjunction with deep learning features for action recognition in still images; however, their technique still entails the learning of action classifiers in a multimodal setting. In the proposed approach, we argue that even in the case of static images, with reliable recognition of objects in the image, one can successfully use NLP-based word embeddings to describe the underlying action with a reasonable degree of accuracy. More recently, Jain et al. [17] have proposed an approach that explores the efficacy of word embeddings for action recognition in videos using

knowledge of the objects in the video. In contrast, the strength of our approach lies in the fact that we use word embeddings in a relatively simple manner in *static* images and yet obtain good results on a challenging set of *static* images. In their recent work, Xu et al. [42] use word embeddings in a zero-shot framework for action recognition in videos. Their approach is based on learning a mapping between the visual features of the action and a semantic descriptor (i.e., word vector). Determination of the mapping involves a computationally intensive semi-supervised transductive learning procedure which calls for access to testing data in the training phase. We believe that semi-supervised transductive learning may be appropriate for videos because of the complexity of mapping spatio-temporal features to semantic meaning, however, for static images, word embeddings could be used more directly and efficiently to infer the underlying action from the objects. Alexiou et al. [2] have proposed another interesting approach that employs word embeddings for zero-shot action recognition in videos. Their approach shows that the mining and alignment of synonyms from general text data can enrich action word vector embeddings via the introduction of more robust semantic context from a wider range of text domains. Their approach to action recognition, based on directly learning the mapping from the visual action features to the word2vec space, is well suited for video data. In contrast, the proposed approach relies on the detection and recognition of objects which provides more concrete information that can be used to infer the underlying action. We contend that the proposed approach is better suited for action recognition in still images for reasons discussed in the Section 3.2.2.

Word2vec model

The motivation for *word2vec*-based approaches emanates from the distributional hypothesis proposed by Harris [16]. The central idea in the distributional hypothesis is that the words that occur in similar contexts tend to have similar meanings. For example, given the sentences, *A person is eating pizza* and *A person is eating chocolate*, we know from the distributional hypothesis that the words *pizza* and *chocolate* are similar in that they occur in similar contexts. Word2vec [27] is a word embedding scheme that converts a word into a low-dimensional vector. Each word is mapped to a point in a hypothetical space such that words that have similar meanings tend to be closer in this hypothetical space. These word embeddings are used in a variety of applications and have had a significant impact in the fields of NLP, computer vision and information retrieval.

3.2.4 Top-object Detection-driven Verb Prediction Model

In this paper, we present the insight that the determination of top-nouns is enough to predict the relevant verb in an image, hence, separate verb detectors are not required for describing the action in most static images. In support of the above insight, we analyze various NLP techniques where we predict the verb from the top-nouns in an image without explicitly learning a verb detector for that image. Figure 3.1 depicts the computational pipeline for the proposed approach. In the proposed approach, we detect the top objects in an image, identify the most plausible two objects (i.e., object pair) in the image, and then assign the most meaningful action (verb) to this object pair (Figure 3.2). This approach could

prove practically useful in two potential scenarios:

(1) Real world situations where automatic generation of training data for verb detectors is very hard. Objects in an image directly map to concrete entities (i.e., nouns) in the real world [11, 13]. This direct translation between nouns and objects enables the stable assignment of a word to an object in an image. In contrast, relational categories such as verbs tend to describe relations between disjoint entities. Moreover, verbs are also more *polysemous* than nouns, in that verbs have more senses of meaning than do nouns [11]. For example, the verb *running* can be used in a variety of settings, for example, "John is *running* for public office.", and "John is *running* on the field.", use the verb *running* in different senses. Likewise, verbs are also *mutable* in that their meaning can be adjusted based on the context of sentence [12]. For example, in the sentence, "John is *jumping* to a conclusion", the verb *jumping* adjusts its meaning to the nouns in the sentence. The properties of polysemy and mutability make automated generation of training datasets for verbs a very difficult task.

(2) With the recent proliferation of resource-constrained mobile devices that constitute the Internet-of-things (IoT), it is important to have image analysis and retrieval techniques that could provide significant algorithmic time gains. Hence, by recognizing the *object-pair* and associated *verb* in a time-efficient manner, one could describe the crux of the story underlying an image even in the most resource-constrained environments. Whereas feature extraction and object detection and classification are unavoidable in an automated image annotation or captioning system, we believe that when inferring a relational category, such as the action or

verb, significant algorithmic time gains can be achieved if we can reliably infer a verb from its associated objects in constant (i.e., $O(1)$) time. Although deep learning feature extraction is computationally intensive, substantial research efforts are underway to make deep learning architectures deployable in resource-constrained environments [1, 21, 22, 23, 24]. However, in addition to speeding up feature extraction and other various aspects of deep learning, we believe these research efforts could be greatly assisted by reducing the number of Support Vector Machines (SVMs) (or other classifier functions) employed at test time. Although the computational expense associated with deep learning feature extraction is significantly higher than that associated with SVM-based classification (or with other classifier functions), we believe, with the research efforts currently underway, deep learning feature extraction will get significantly faster in due course. In that case, we contend that the proposed approach, which reduces the number of SVMs (or other classifier functions) needed for action recognition/classification will effectively complement the computationally efficient deep learning feature extraction techniques in the very near future.

In this paper, we analyze and propose two models to predict verbs from top-nouns - the *Object-Verb-Object (OVO) triplet* model and the *word2vec* model. Both models have their advantages and shortcomings depending on the underlying dataset and application scenario.

Figure 3.2: Outline of the proposed top-object detection-driven verb prediction model

3.2.5 Object-Verb-Object (OVO) triplet model

The Object-Verb-Object (OVO) triplet model explicitly models the probability distribution of words based on their co-occurrences. However, there are situations in real world datasets wherein such co-occurrences may not cover all situations. Hence, we need techniques that learn the dependencies between verbs and nouns in a more implicit manner. For example, if we have verb *eating* as predicted by the OVO triplet model using nouns *person* and *pizza*, then the OVO triplet model cannot extrapolate to predicting the verb *eating* for the nouns *person* and *chocolate*. However, the word2vec model explained in the following subsection is capable of handling such situations.

In the OVO triplet model, we predict the relevant verb using the following equation:

$$p(verb|noun_1, noun_2) = \arg\max_i \quad P(verb_i|noun_1, noun_2) \tag{3.1}$$

The probability values in equation 3.1 are computed using the textual data in the training set. At test time, given two objects (i.e., nouns), the most highly probable verbs are determined using equation 3.1 and assigned to the image.

Word2vec model

The word2vec verb prediction model uses the word2vec representation scheme [28] which is based on the embedding of a word in a hypothetical low-dimensional vector space. In the word2vec representation scheme, each word is mapped to a point in the hypothetical vector space such that words that have similar meanings tend to be proximal in this vector space. In our case, we intend to capture the relationships between nouns and verbs, for example *pizza* and *eat*, and noun-pairs and verbs, for example, *Person-dog* and *walk*.

It is interesting to examine why the word2vec representation is able to learn the word embeddings that capture the relations between nouns and verbs, or between noun-pairs and verbs. Note that the word2vec representation groups semantically similar words into proximal regions in the hypothetical vector space, i.e., words that are similar in meaning such as *beautiful* and *pretty* are mapped to proximal points in the hypothetical vector space. In this sense, the word2vec representation treats synonymy, not as a binary concept, but rather one that spans a continuum. However, we hypothesize that even when the words are not obviously synonymous or similar in meaning, the distance between their corresponding points in the vector space can still convey something significant about their relationship.

For the sake of clarification, consider the following example: Assume that we

are given a collection of the following four sentences:
"A person is driving a car on the road". "A car is passing a truck on the road". "A car is parked on the road". "A person is driving a truck".

In the above sentences, the context of the noun *car* is {*person, road, truck, driving*} whereas the context of the verb *driving* is {*person, road, truck, car*}. As the contexts of *car* and *driving* are very similar, word2vec will place the embeddings of *car* and *driving* in close proximity in the vector space although *car* and *driving* are strictly not synonymous words. Based on context, among all verbs, the verb *driving* will tend to be closer to the noun *car* based on their respective embeddings in the vector space. For a more rigorous treatment of why the word2vec embedding tends to capture such linguistic regularities the interested reader is referred to [32].

The problem of determining the closest verb to two given top nouns can be stated as follows. Given a set of verbs V and top nouns n_1 and n_2, the closest verb from the set V to the top nouns n_1 and n_2 is given by:

$$\arg\max_i \ \{SIM(v_i, n_1) + SIM(v_i, n_2)\} \tag{3.2}$$

where $SIM(v_i, n_1)$ and $SIM(v_i, n_2)$ are the cosine similarities of the vector representation of verb v_i to the vector representations of nouns n_1 and n_2 respectively.

One of the problems with above formulation is that certain nouns such as *person* and *apple*, when considered independently, may have multiple verbs that are proximal in vector space. For example, *person* and *apple*, when considered independently, may be proximal to multiple verbs such as *sit*, *hold*, *sleep* and so

on. Simple addition of the cosine similarities as shown in equation 3.2 does not bias the verb prediction towards *eat* when **both** the nouns *person* and *apple* are present in the same sentence. To circumvent the above problem, in the sentence database accompanying the MS COCO training dataset [25], we append each sentence with all object-pairs occurring in that sentence. In other words, we identify all the nouns in a sentence and form all pairs of these nouns before appending them to the sentence. For example, given a sentence *"Person is eating an apple sitting on the table."*, we convert the sentence into following three sentences:
"Person is eating an apple sitting on the table apple-person."
"Person is eating an apple sitting on the table person-table."
"Person is eating an apple sitting on the table apple-table."

This simple preprocessing step potentially captures the dependences between all possible noun-pairs and all verbs in the sentence. Next, we train the word2vec model on the modified sentence database. After the model is trained, it will have learned the verb that defines a relationship between a pair of objects. Figure 3.3 clarifies the above argument using the projection of these word embeddings in a 2D space using the *t-SNE* dimensionality reduction technique [39].

More formally, given the set of verbs V, and noun-pair n_{12}, the closest verb in V to the noun-pair is $np_{j,k} = (n_j, j_k)$ is given by:

$$\arg\max_{i} \; SIM(v_i, np_{j,k}) \tag{3.3}$$

where $SIM(v_i, np_{j,k})$ is the cosine similarity between the vector representations of the verb v_i and noun-pair $np_{j,k} = (n_j, j_k)$.

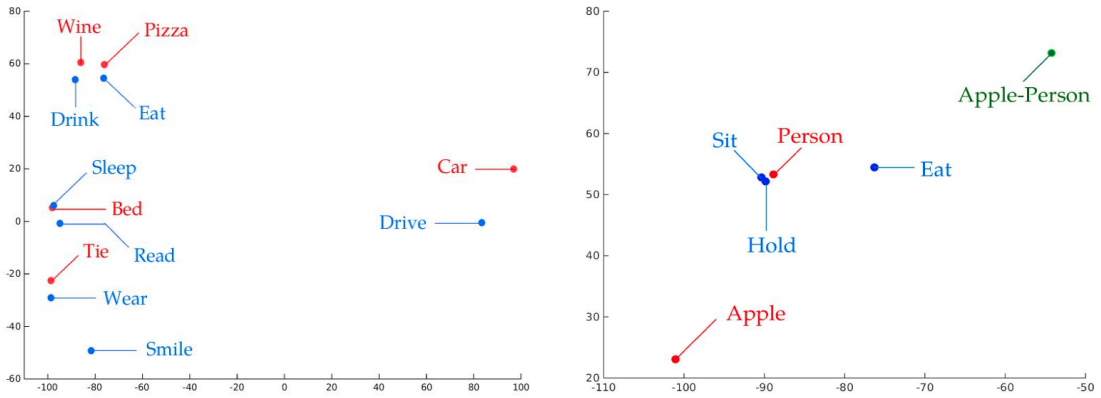

Figure 3.3: Visualization of word embeddings is 2D space using *t-SNE* dimensionality reduction [39]. (a): Most verbs tend to occur closer to their attached nouns. (b): Appended nouns (*apple-person*) occur nearer to verb *eat* than individual nouns *apple* and *person*.

In the above model, once the necessary steps of feature extraction and object detection are performed, verb prediction for a given noun-pair can be achieved in $O(1)$ time. During testing, the top verbs can be easily retrieved in $O(1)$ time using an appropriate hash data structure once the top-2 nouns (objects) in the test image are detected. Since the verb is detected in a zero-shot manner (i.e., without requiring visual training examples of the action underlying the verb), the computational expense of training verb detectors and running them on the image is obviated. After the word2vec model is trained on the modified sentence dataset, we choose plausible verbs that are closest in distance to each object-pair and store the verbs in the database along with the object-pair.

Model training

Training of object detectors

We train an SVM-based object detector/classifier for each of the 80 annotated object categories in the MS COCO dataset [25]. The inputs to the SVM-based classifiers are the VGG-16 *fc-7* image features [36] extracted using the Matconvnet package [27].

Training of the OVO triplet model

The OVO triplet model is trained on the sentences corresponding to the training images (in the MS COCO dataset) using equation 3.1. For each pair of objects that occur together in a particular sentence, we learn the probability value of the corresponding verb in that sentence.

Training of the word2vec model

The word2vec model is trained with a window size of 10 using the implementation of Rehurek and Sojka [34]. This results in a 300-dimensional vector for each word using the skip-gram model for word2vec [27]. In the skip-gram approach, the input to the deep-learning neural network (DLNN) is the word, and the context is predicted from the word.

For example, given the contextual input *eat*, the model will predict {*person*, *pizza*}. To train the skip-gram model, given a sequence of words $w_1, w_2, w_3, ...w_T$,

we maximize the following objective function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{3.4}$$

where c is the context parameter that specifies the number of words to be predicted from a given word [29]. The term $p(w_{t+j}|w_t)$ signifies the prediction probability of the context given the word. The stochastic gradient descent algorithm is used for training the skip-gram model. More details on the skip-gram architecture can be found in [29]. After the skip-gram model is trained, we obtain the word embeddings corresponding to each word in the dataset.

The nouns within each sentence are converted to noun-pairs and appended to the end of the sentence, as explained previously. We also perform a couple of additional preprocessing steps on the entire data. First, we stem each word using Porter's stemmer; for instance, *driving* and *drive* are both converted to *driv*. Additionally, words synonymous to *person* such as *human*, *woman*, *boy*, *girl*, *people* etc. are converted to *person*. Currently, we try to infer only the most frequently occurring verbs in the MS COCO dataset, i.e., we select the top-n (where $n = 51$) most frequently occurring verbs in the MS COCO dataset. The top verbs in the MS COCO dataset are obtained by parsing the training captions using the Stanford parser.

The 40,000 images in the MS COCO validation set are split into two subsets, each containing 20,000 images; one subset is used for validation of the hyperparameter tuning procedure and the other subset for testing. The validation subset is used to learn the hyperparameters and also the other required parameters for

the skip-gram model.

Model testing

Given a test image, we run all the object detectors on it and select the *top-2* highly probable object detections as candidate objects. For this object-pair, we use the word2vec model to obtain the closest verbs. For each test image, we recognize an object-pair, and predict the plausible verbs in the image. Among the comparison measures introduced below, all except one, predict two plausible verbs in an image. If any one of the predicted verbs occurs in any of the ground truth captions of an image, we regard the prediction as accurate. In addition, even if there are multiple verbs in the ground truth captions for a particular image, intuitively, we just need to infer one verb accurately to describe the crux of the story underlying the image. For example, if the two ground truth captions for a particular image are *Person is riding a motorcycle* and *Person is driving a motorcycle*, it would suffice to just get one of the two verbs *riding* or *driving* correct. Hence, when computing the prediction accuracy, we aim to get just one verb correct in the ground truth captions.

We report results on the subsets S_1 and S_2 of the validation set of the MS COCO dataset, which we use for testing purposes. S_1 is a subset of the validation dataset wherein the ground truth captions have at least two objects from the annotated noun set, and at least one verb from top-51 most frequently occurring verbs in the MS COCO dataset. S_2 is a subset of S_1 wherein the top-2 objects have been correctly detected in an image. These results are used to show how

Table 3.1: Comparison of verb prediction accuracy results of the word2vec model (VD_1, VD_2, VD_3) and the OVO model (OVO) with a random baseline (*Rand*), the 1-object baseline (*1-Obj*) and the visual action classifier baseline (*Vis*). DS denotes the data subset. Accuracy is measured based on whether one of the two predicted verbs matches one of the ground truth verbs.

DS	*Rand*	*1-Obj*	*Vis*	VD_1	VD_2	VD_3	OVO
S_1	9%	35.2%	37.7%	36.9%	31.4%	32.8%	55%
S_2	10%	45.81%	41.4%	53.43%	57.74 %	52.35 %	79%

effectively a verb is inferred after the object-pair is correctly detected in an image. We compare the results of the proposed scheme under the following evaluation scenarios:

Random Baseline (Rand): where the two verbs are generated randomly for the top-noun detections in an image.

1-Obj Baseline (1-Obj): where the top-most object (object with highest probability) is used to predict top-2 verbs in an image using word embeddings. The top-2 verbs that are closest in distance to this top-most object are selected.

Vis: where visual action classifiers (such as *walking*, *swimming*, etc.) are explicitly trained using deep learning features followed by SVM-based classification.

VD_1: where the top-2 closest verbs are the ones with the lowest mean distance from the top-2 object detections measured using equation 3.2.

VD_2: where the top-2 closest verbs are the ones with the lowest distance from the appended noun-pair measured using equation 3.3.

VD_3: where the top-2 verbs are assigned as follows: if the closest verb determined using equations 3.2 and (3.3) is the same, we assign this verb to an image, and

the second closest verb is assigned using equation 3.2. Otherwise, one of the top-2 verbs is assigned using equation 3.2 and the other using equation 3.3.

VD_4: where the verbs are assigned using set union between the top three verbs determined using equations 3.2 and 3.3.

OVO: where the verbs are assigned using the OVO triplet model using equation 3.1.

3.2.6 Experimental Results and Discussion

The verb prediction accuracy results under the evaluation scenarios described in Section 3.2.5 are compared in Table 3.1. These results lend support to our claim that top-object detections could be used to infer other information in an image such as verbs. Once we know the plausible object pairs in a static image, we can infer or predict the corresponding verb in $O(1)$ time with reasonable accuracy using the OVO triplet model or the word2vec model as shown in Table 3.1. Also, when either one or both of the top-object detections are incorrect, the word2vec model is observed to underperform the visual action classifiers. However, when both the top-object detections are correct, the word2vec model outperforms the visual action classifiers. This suggests that getting top-object detections correct is important for improving the performance of the word2vec model.

For the OVO triplet model, we see that we are able to predict the verb accurately given that the detection of the top-2 objects is accurate. In the MS COCO dataset, the co-occurence patterns of nouns and verbs are similar for both, the training set and the test set, hence the OVO triplet model works well for the MS COCO dataset. However, in real world datasets, the training set and test set may

Figure 3.4: Qualitative results for the verb prediction model. (a) VD_1: Bad; (b) VD_1: Good; (c) VD_2: Bad; (d) VD_2: Good

exhibit some dissimilarities, and hence we may have to resort to models such as word2vec for predicting verbs. The results of the word2vec-based approach are analyzed in the following paragraphs.

In the case of the word2vec-based approach, if the object-pair is correctly recognized in an image, the results in the case of VD_2 are observed to be slightly better than those in the case of VD_1 and VD_3. Hence the proposed technique of appending the object-pair to the end of the sentence does provide a non-trivial benefit for verb prediction. However, if the object-pair is not correctly identified in an image, then the results in the case of VD_1 are observed to outperform those in the case of VD_2 and VD_3. In other words, finding the closest verb by computing the mean distance to the top-2 nouns (equation 3.2) is better than using the minimum distance to the object-pair (equation 3.3) when at least one of the objects is incorrectly detected. The qualitative results for VD_1 and VD_2 are shown in Figure 3.4.

In the case of the word2vec-based approach, the results of VD_3, where we try

to get the combined benefits of VD_1 and VD_2, were inferior to those of both VD_1 and VD_2. This could be attributed to the fact that results in the case of VD_1 and VD_2 had only a marginal quantitative difference. This appears to suggest that to get actual benefits of both VD_1 and VD_2, we may need to predict more than 2 verbs. Hence, we conducted additional experiments with VD_4 obtaining an accuracy of 56.63% on S_1 and 73.09% on S_2. Therefore, in the case of VD_4, where we predict multiple verbs using both VD_1 and VD_2 we are far more successful in getting at least one verb correct. We believe that besides predicting multiple verbs, there are a couple of other reasons for the relative success of VD_4. There are situations where VD_1 will be successful, and there are situations where VD_2 will be successful; VD_4 denotes the best of both worlds where VD_1 corrects and compensates for weakness of VD_2 and vice versa. Also, the high accuracy of VD_4 suggests if we try to predict a few more verbs (of the order of 3-6), than there is a very high probability of getting at least one of them correct.

Overall, from the results it is clear that just detecting the most prominent objects in an image is enough to predict the underlying action (verb) in an image with competitive accuracy. The proposed NLP approaches based on the OVO triplet and word2vec models successfully beat the baseline results, thus lending support to our claim.

Table 3.2: Verb prediction accuracy for situations where word embeddings are trained on a general text corpus.

DS	VD_2
S_1	33.5%
S_2	51.4%

3.2.7 Effect of word embeddings trained on a general action classification

In the previous sections of this chapter, the word embeddings were trained on the captions accompanying the training images. However, what if such captions are not available? In such situations, it would be interesting to see how the word embedding models trained on a general text corpus would perform. For this purpose, we use the FastText pre-trained word embeddings trained on Wikipedia [3]. We conduct experiments similar to the ones described in the previous section and use VD_2 as our evaluation metric. The object classifiers are trained using the MS-COCO annotations (i.e., captions) in the training set, in the same manner as described in the previous section. However, the actions are not inferred using word embeddings trained on the MS-COCO captions. Rather, the actions are inferred from pre-trained embeddings derived from a general text corpus, which in our case is Wikipedia [3]. Using the same test set as in the previous section, we compute the VD_2 accuracy metric on the test images for this experiment.

From Table 3.2, we see that inferring the action using word embeddings trained on a general text corpus does not harm the performance significantly, and that we

get only slightly worse results in comparison to the results in Table 3.1. Hence, actions can be reliably inferred from word embeddings trained on a general text corpus, thus making this scheme suitable for situations where action labels for training the classifiers are not available. We address such situations in more detail in the second part of this chapter.

3.2.8 Limitations of the Proposed Approach

One of the limitations the proposed approach is that currently we use only two objects for predicting the verb. There are reasons why two objects may yield good results for verb prediction in many real-world situations. A verb is a natural connector between the subject and object in many real-world situations. However, there are situations where multiple objects in an image are needed for predicting the verb accurately. For example, consider a situation where "*A person is baking pizza in an oven.*" In this situation, knowing the nouns *person* and *pizza* would most likely lead us to infer the verb as *eating*; however, knowing the three objects *person*, *pizza*, and *oven*, would most likely lead us to infer that the most appropriate verb is *baking*. In addition, we have not addressed the effect of the relative spatial positions of the objects on the accuracy of verb prediction in our current work. Considering the example above, the relative spatial positions of the objects *person* and *pizza* could be used to disambiguate between the predicted verbs *eating* and *baking*; if the *pizza* is spatially close to the mouth of the *person* then *eating* would be the more likely verb, otherwise *baking* would be more appropriate.

Another limitation of the current work is that global scene context is not in-

corporated in the verb prediction model. Knowledge of the global scene context in conjunction with knowledge of the top objects could potentially enhance the accuracy of verb prediction. In the previous example, if the global scene context is *dining room*, then detecting the objects *person* and *pizza* would lead us to infer the verb *eating* over the verb *baking*. Alternately, if the global scene context is *kitchen*, then the objects *person* and *pizza* would lead us to infer the verb *baking* over the verb *eating*.

Also, in our experiments, the OVO triplet model yielded better results than the word2vec model. This can be attributed to the fact that the training and testing datasets for the MS COCO benchmark are not too different. However, real world situations do not exhibit this characteristic. In the real world, unlike the MS COCO training set, two objects and a verb may not co-occur with one another. For example, we may have never seen instances of *tennis racket* and *person* in single image. In such situations, the OVO triplet model will not be able to assign a probability value to any verb that is associated with the nouns *tennis racket* and *person*. The word2vec model would be more appropriate in this situation. In the word2vec space, *tennis racket* will appear close to other sports-related entities such as *ball* and *baseball bat*, thus facilitating the assignment of an appropriate verb such as *hit*.

3.2.9 Conclusions and Future Work

In this paper we have proposed a scheme to detect the actions (verbs) in a still image by first detecting the prominent objects in the image and then using Natural

Language Processing (NLP)-based OVO triplet and word2vec models to infer the relevant verbs. Our approach obviated the need for training and using visual action detectors which tend to be error-prone and computationally intensive. This paper also provided a valuable insight in that the detection of a few significant (i.e., top) objects in an image allows one to extract the relevant actions or verbs in the image without entailing the learning of an action or verb from visual training data. For this purpose, we proposed NLP approaches based on the word2vec and the OVO triplet models for predicting the actions from top-object detections and also analyzed their nuances. Our experimental results showed that verbs can be reliably and efficiently detected by exploiting the top object detections in an image.

With regard to future work, we plan to extend our work in various directions. We plan to take relative spatial positions of objects in predicting verbs. We also plan to account for situations where more than two objects occur in an image. And in addition to objects, we also plan to incorporate entire scene context in addition to the knowledge of the prominent objects for predicting the verbs. Also, we plan to conduct more robust studies where the efficacy of word2vec approaches is evident, leading to use of word embeddings to resolve all the quirks and nuances of action (verb) recognition based on top-n (where $n \geq 2$) object detection.

3.3 One-Object Decision-Making model: Fast and Frugal Heuristic for Human Activity Classification

3.3.1 Chapter Outline

Consider an uncertain situation where an artificial intelligence (AI) system is called upon to determine a human action or activity in an image or scene. The AI system has not been previously trained to recognize any human action or activity, and has no prior information on pose, parts, spatial layout of the object in an image. In such a situation, what is the AI system supposed to do? Its options are limited, and it must determine the action or activity with the aid of the most probable inanimate object (other than the human actor) that it can detect in the image. The AI system needs to formulate two hypotheses to infer the action or activity in a zero-shot manner; first, that the most probable inanimate object detected in the image is one that is involved in the action or activity, and second, that the most likely action or activity associated with this object in the real world is the one actually occurring in the image. To what extent are these hypotheses valid? We propose that correct detection of the highly probable object and use of natural language word embeddings obtained via training on a general text corpus such as Wikipedia could enable the AI system to determine the underlying human action or activity in an image with reasonable classification accuracy. We conducted studies on the HICO dataset, which is a challenging dataset containing many rare human action/activity categories. Our experimental results show that if the AI system can reliably detect the most probable inanimate object in the image and then infer

the corresponding verb in a zero-shot manner using language models trained on general text corpora, then it has a reasonable chance of correctly guessing the underlying action/activity in an image.

3.3.2 Introduction

Herbert Simon introduced the notion of satisficing; its premise is that humans often rely on good enough decisions rather than optimal ones. Such a decision strategy saves computational resources while operating with limited information. A related notion that is gaining prominence in decision theory is one of *fast-and-frugal heuristics* [14, 15]. The fast-and-frugal heuristics take the notion of satisficing to the extreme, surmising that people make decisions with minimal resources (in terms of available time and information), and yet the decisions are almost as good as, and sometimes better than, the ones that take more information into consideration. Currently, such heuristics have been known to perform well in humans, and authors have argued for the viability of such heuristics in AI systems [14]. Yet, most AI systems, and computer vision systems in particular, have a history of relying on optimization-based search and inference strategies, rather than relying on fast-and-frugal heuristics. Could fast-and-frugal heuristics find some useful applications in computer vision systems? Could there be situations where they could prove helpful?

We believe that the paradigm of fast-and-frugal heuristics has a lot to offer computer vision systems. Within the family of fast-and-frugal heuristics is the concept of one-reason decision making: It is the cognitive heuristic based on the surmise

that people rely on only one primary reason to reach effective decisions [18]. In fact, in many cases, adding more reasons in the form of additional cues or information does not improve the quality of the decision. Inspired by this notion, we propose the one-object decision-making model for the purpose of human action/activity recognition. We believe that, having successfully performed human detection, the knowledge of a single non-human or inanimate object in the image, is enough to reliably estimate the underlying action in an image, for most real-world situations. The real world imposes regularity in the ways that non-human or inanimate objects co-occur with human actions and this regularity is advantageous in modeling fast-and-frugal heuristics. Most non-human or inanimate objects in the real world are associated with only a few human actions; in short, they follow a Zipf distribution. We conduct and present the corresponding Zipfian analysis later in the chapter. Even among those few actions associated with the non-human or inanimate object under consideration the instances of one or two actions significantly outnumber all other actions in most real-world situations. For instance, "a person riding a bicycle" is a much more common occurrence than "a person repairing a bicycle".

Owing to recent advances in deep learning, the reliability of image-based human detection and classification techniques has improved significantly. This can be attributed, in part, to the fact that generating image datasets containing humans for the purpose of training classifiers is not a very difficult task. Standard datasets such as ImageNet and MS-COCO contain many human images. In addition, collecting a large number of human images from online websites is also

not very difficult. Human classification and recognition results on the ImageNet Large Scale Visual Recognition Challenge 2017 dataset are very impressive, and in addition, recent advances in pedestrian detection [30, 37, 38] suggest that reliable human classifiers are not very hard to design given sufficient training data. In our experiments, the human classifiers that we trained on the MS-COCO dataset performed very well on the HICO dataset, yielding an accuracy of almost 95% on human images in the HICO dataset.

It is important to make the following clarification: It may seem that the proposed technique attempts to directly learn the subject-verb-object structure within an image. However, the subject-verb-object structure assumes that a relational structure exists between the subject and the object that we are trying to detect using computer vision and/or natural language processing (NLP) techniques. It should be emphasized that the proposed technique does not attempt to learn any relationship between the subject and the object and that the subject in the form of a human is assumed to be correctly detected in the image.

If the AI system is then to make a guess about the underlying action after both person and bicycle have been detected in the image, then, given the Zipfian nature of object-action co-occurrences, the action riding is a much more optimal guess than repairing. From a systems perspective, such a strategy saves resources in terms of time spent in training and data generation, is advantageous in terms of space and time complexity of the underlying computation, while yielding reasonable accuracy. In addition, such a strategy could also be useful in modeling the human mind from a cognitive science and computational neuroscience perspective.

While one-object decision-making models may not lead to the most optimal solution, they can be added to the repository of strategies that are available to an AI system. Analogously, it has been shown that humans use fast-and-frugal heuristics as one potential strategy and will revert to another strategy if more computational resources are available. On a similar note, fast-and-frugal heuristics is just one strategy that a system should call upon in specific situations. For action recognition and the related problem of image captioning, there already are a variety of strategies proposed in the research literatures such as pose-based action recognition, part-based action recognition, RNN/LSTM-based image captioning, and so on. However, all of these strategies would naturally fall within the optimization-based inference school of thought, rather than Herbert Simon's satisficing school of thought. This is because most of the aforementioned techniques require good training data using which they attempt to optimize various algorithmic and architectural parameters based on several criteria. The fast-and-frugal strategy requires minimal resources in terms of time and data needed for training and input information needed for inference and the inference procedure exhibits very low space and time complexity. It can be counter-argued that space and time complexities of the training and inference procedures will be non-issues for the future; however, we believe that there can still be a place for such strategies if we are to create authentic AI systems. While many of the optimization-based approaches focus on obtaining optimal solutions, there is a place for minimalist satisficing solutions, i.e., solutions that are good enough but not necessarily the very best ones, but ones that require minimal resources in terms of time and data

needed for training and input information needed for inference while exhibiting low time and space complexities for the training and inference procedures.

It is important to clarify that we use the fast-and-frugal heuristic as an inspiration, just as biological neurons were an inspiration for artificial neural networks. In the human decision-making arena, the term "fast and frugal" implies that there is no contemplative phase in the decision making process. Along similar lines, devoid of any knowledge of a particular situation, a hypothetical AI system may indulge in a contemplative reasoning or decision making phase trying to infer an action using unrelated knowledge and other forms of reasoning to deal with an unknown situation. Note that the previous statement is speculative since such an AI system does not exist. Thus, for our purpose, we use the term "fast-and-frugal heuristic" to imply in part that there is no contemplative or reasoning phase involved in making a decision. Hence, we use the term "fast-and-frugal heuristic" as an abstraction and how it translates into space and time complexity for the training and testing procedures would depend on the specific application under consideration.

We digress to explain how a minimalist satisficing strategy could be integrated with optimization-based or Bayesian reasoning in AI. As suggested by [14], under a situation characterized by limited time and resources, a satisficing strategy would suffice. However, when accuracy is paramount, even at the cost of time or other resources, the search or inference procedure would need to be optimized via Bayesian reasoning. This is not an either/or decision [14]; rather, depending on the situation (whether higher accuracy is absolutely critical or not), the system

resorts to the best suited strategy. In the previous chapter, we learned language models from the captions that accompanied the images. Such a strategy could be considered to fall in the optimization-based category because we use extraneous information, in the form of closed world language models trained on the datasets, to arrive at the best possible solution. However, in this paper, we use a zero-shot approach to infer the action or activity from a single object (other than a person) by taking advantage of language models trained on generic datasets. Here, the term "zero-shot" denotes that we do not use any of the action labels from the training set. For this purpose, we use the HICO dataset [4]. In the HICO dataset, most categories are labeled as an object-action pairs. For instance, airplane-fly and boat-repair are two of the categories. We train our system for object classifiers, while ignoring the action part, and essentially assume that the action labels are not present in the dataset. In short, we assume the nonexistence of action labels for training. In a closed world situation, we would be training language models using action labels from the training set. Instead we use a generic language model trained on a general text corpus (such as Facebook's [3] which is trained on Wikipedia) to infer the most appropriate verb or action that could be associated with a particular object. Owing to the linguistic regularities in the real world we believe that this generic language model could capture the verb-object relationships for many real-world images. The action portion of the human-object interaction, therefore, could be considered zero-shot.

In this paper, we argue that, for most types of image datasets, after a person or human is detected in the image, only one additional object could result in

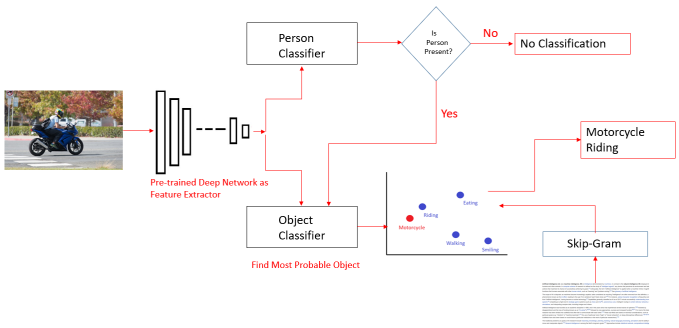

Figure 3.5: Outline of our Fast and Frugal Approach for Human Activity Classification. Note that the general text corpus (Wikipedia) is used to train a skip-gram model (FastText) to obtain the natural language word embeddings.

determining the action. We answer the question "Given a person and another object, what is the most probable action that could be happening in an image?" Surely, there may be no action in the image, but if an action is indeed present, which action would be the most likely, i.e., have the highest probability? For this human activity recognition, we determine the action in an image with the aid of another non-human or inanimate object. For the images involving multiple persons, we will infer the verb that is most relevant in those situations as shown by Zipfian analysis of sentences that describe actions containing multiple people. Using Zipfian analysis, the verbs that occur most frequently are assigned to the images containing multiple persons or humans.

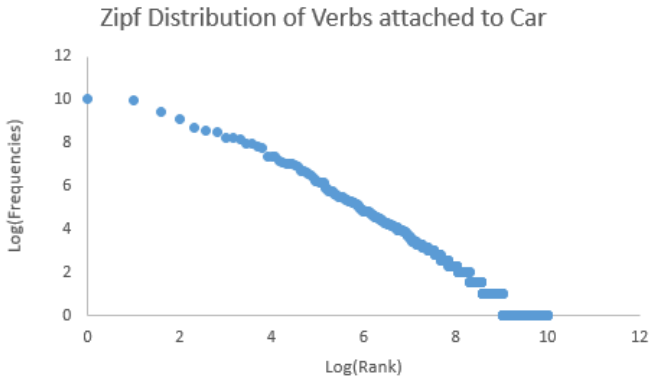

Figure 3.6: The frequencies of verbs conditional on the concept of a car follows the Zipf distribution. Here the rank denotes the rank of verbs associated with object (noun) car with respect to other verbs associated to the same object (noun) car. The frequencies denote the frequency of verbs attached to the noun car.

3.3.3 Motivation

Zipfian Analysis

We claim that the distribution of verbs around nouns follows the Zipf distribution and conduct Zipfian analysis on various categories verbs and nouns in the MS COCO dataset. If the distribution of verbs around nouns indeed follows the Zipf distribution, then it means that each object has only a few verbs that are significantly associated with it for most real-world situations. Figure 3.6 shows the results of Zipfian analysis of verbs that are related to the noun *car* in the MS COCO dataset.

Mathematically, the random variable $X \sim \text{Zipf}(\alpha, n)$ is described by probabil-

ity mass function for the Zipf distribution [20] as given by:

$$f(x) = \frac{1}{x^{\alpha} \sum_{i=1}^{i=n} \left(\frac{1}{i}\right)^{\alpha}} \tag{3.5}$$

for parameters α and n where $x = 1, 2, 3, \ldots, n$, and $\alpha \geq 0$. How is the Zipfian nature of actions associated with objects exploited in word embeddings trained on a real-world text corpus such as Wikipedia? Word embedding models rely on co-occurrence data; hence, they are more likely to embed the object (noun) and actions that occur together closer in a hypothetical space than if they did not occur together. For more details on why this would be the case, the reader is referred to the paper by Pennington et al. [32]. Since most word embeddings rely on co-occurrence data, and we believe that the object-action co-occurrences follow a Zipfian distribution, the resulting word embeddings will necessarily incorporate the Zipfian nature of object-action co-occurrences.

Related Work

The central premise of the fast-and-frugal heuristic paradigm is that, in certain situations, humans make accurate decisions with resource-efficient heuristics and without considering all possible alternatives. In situations where these heuristics are successful, accuracy and speed are not each achieved at the cost of the other, and both can be attained simultaneously. This paradigm is in contrast to the traditional optimization-based school of search and inference and is aligned with Herbert Simon's Bounded Rationality paradigm. For instance, in the one-reason

decision-making model, humans make inferences based on one cue only [15]. If there are multiple cues available to make an inference, then according to this scheme, humans either take the best cue (the cue with highest validity), take the last cue (the cue that worked the last time), or apply a minimalist approach (use a single cue that is randomly chosen) to draw an inference. How do fast-and-frugal heuristics work in human decision making? There is no clear answer to this question. Perhaps some subconscious processes are at play here. Hence, for the purpose of designing AI systems, we draw general inspiration from the fast-and-frugal heuristic paradigm rather than borrowing rigidly from it. The AI analog of fast-and-frugal heuristics would use minimal training data and minimal computational resources, including but not limited to time and space complexity of the inference procedure. We believe that such fast-and-frugal heuristics should be the part of the strategic repository of any AI system. We believe our approach for zero-shot human action classification is a form of fast-and-frugal heuristic because human action could be inferred from just another non-human or inanimate object in an image, at least for most frequently encountered real-world situations. Action classification has been primarily explored for videos and, to a lesser extent, for still images. In videos, action classifiers often take advantage of temporal information [7, 31]. In still images, the most prominent action based approaches are pose-based [26], parts-based [6], and human interactions with objects [35]. Almost all of these approaches entail excellent training data that is labeled and/or computationally intensive algorithms for drawing inference characterized by high time and/or space complexity, and/or intensive programming effort. However, in this

paper, we answer a fundamental question in the context of human action classification: if we know just one inanimate or non-human object besides the human in the image, then how far can we go in predicting the underlying action? A more detailed review of the action classification literature can be found in part 1 of this chapter.

Dataset

To experimentally demonstrate the proposed technique, we chose the HICO dataset [4]. One of the defining characteristics of the HICO dataset is that it contains many rare categories; hence, it cannot be considered representative of the real world. For instance, how many times do categories such as inspecting a tie, inspecting a bicycle, directing an airplane, repairing a boat, grooming a dog, kissing an elephant, lassoing a cow, etc., occur in the real world? In addition, the dataset has many "no interaction" categories where no action is taking place in an image. Both of the above characteristics make the HICO dataset ideal for testing, since if we can show the moderate levels of proficiency for the approach on this dataset, then it would be generalizable to more real-world images.

3.3.4 Experiments and Approach

Consider an AI system that is made to recognize an action in an image, but that has not been explicitly trained to recognize actions within the image. The system has no knowledge about the parts, poses and spatial layouts of the objects within the image, nor does it possess any heuristics to make a judgement about an

action. Since the system has not been explicitly trained to recognize actions what is it supposed to do when confronted with recognizing an action within an image? What are its options? The system clearly needs to make a bet on an action, but how does it make that bet? When forced to make a bet on a human action, a potentially effective heuristic that the system could employ is to recognize the most probable non-human or inanimate object in the image and guess the human action by drawing an inference from a natural language text corpus that is readily available in the public domain. There is no guarantee that such a bet will succeed, but in the absence of any other knowledge, this is the best the system can do.

Since the HICO dataset has same object categories as MS COCO dataset, we use the object classifiers trained on MS-COCO dataset in the same manner as described in the previous part of this chapter. After the non-human and inanimate objects are successfully classified, we use Facebook's FastText [3], which generates word embeddings trained on Wikipedia. Since verbs/actions are obtained via a general text corpus, they are considered to have been detected in a zero-shot manner. Since the proposed system relies on visual deep learning and word embeddings, the details of these techniques and why they work can be found in Chapter 4 and 5.

Facebook's FastText model [3] learns the word embeddings from the Wikipedia dataset and as a result is more likely to learn what non-human and inanimate objects are associated with what actions. The FastText model relies on a distributional hypothesis [16] that states that words that occur in similar contexts will tend to have similar meanings. However, this theory could be extrapolated to sit-

uations where objects and actions occur in similar contexts as well. The proposed approach is comprised of the following stages:

Stage 1: Binary person/human SVM classifiers are run on each image. If the binary SVM classifier indicates that a person/human exists within the image, we consider it a hit. In the proposed approach, this stage is considered separate from the stages of classification of other non-person and inanimate objects. Here we just use the SVM confidence values to tell us about the presence of a person in the image; if the confidence value is positive, we consider it a hit.

Stage 2: Object classifiers other than person/human classifiers run on each image that is considered a hit in Stage 1. The top-object (excluding the person/human) is identified in each such image.

Stage 3: Using this top-object, identify the associated top-actions/verbs in an image in a zero-shot manner using a natural language model (such as FastText) trained on a general text corpus (such as Wikipedia) [3].

Why should the proposed approach be considered fast-and-frugal? This is because the proposed approach makes two bets. The first bet is that the top-object is detected correctly and second bet is that the top-zero-shot actions are correctly inferred from this top-object. The system uses minimal training data except that needed for the training of object detectors. The actions that could be associated with the detected top-object are inferred using a natural language model (such as FastText) trained on a general text corpus (such as Wikipedia).

To measure the top-n accuracy, we consider only the most probable object in the image. However, for actions, we determine the top-n associated actions using

Facebook's FastText language model. For instance, if the most probable object in an image is book, we identify the top-n most associated actions as read, paint, pet, teach and release. A classification is considered accurate if the following three conditions are met:

1. The person/human in the image is detected and classified accurately. If the confidence value generated by the SVM-based person/human detector is positive, we consider it a hit, otherwise a miss.

2. The most probable non-human and inanimate object in an image matches the ground truth object in the image (as specified in the accompanying caption).

3. There is a non-null intersection between the set of classified actions and set of ground truth actions (as specified in the accompanying caption) for an image. For assessing image-wise classification accuracy, only one action per image needs to be correct. For assessing action-wise classification accuracy, we calculate the number of correct actions out of all the actions in the action space.

3.3.5 Results and Discussion

We find that for both image-wise and action-wise calculation of results, the proposed approach is superior to the most-frequent baseline, based on Top-1, Top-3, and Top-5 accuracy results Tables 3.3 and 3.4. Even though the results of the proposed technique are significantly superior to those of the most-frequent baseline, they do not look particularly impressive at first glance. This is because the HICO dataset has many rare categories, such as kissing an elephant, inspecting a tie, etc. However, an AI system is much more likely to encounter actions that

Table 3.3: Comparison of the proposed approach with the most frequent baseline based on action-wise accuracy results using the Top-1, Top-3 and Top-5 accuracy measures

DS	*Top-1*	*Top-3*	*Top-5*
$MostFrequent$	0.4%	4.1%	4.26%
$Our - Approach$	10.2%	14.83%	16.05%

Table 3.4: Comparison of the proposed approach with the most frequent baseline based on image-wise accuracy results using the Top-1, Top-3 and Top-5 accuracy measures

DS	*Top-1*	*Top-3*	*Top-5*
$MostFrequent$	0.4%	4.03%	4.19%
$Our - Approach$	16.05%	24.02%	26.59%

are typically associated with the corresponding non-human and inanimate objects in most real world situations. Nevertheless, obtaining good results on the HICO dataset which is especially biased towards rare action categories, strengthens our argument that betting on most probable non-human and inanimate object and its most likely associated verbs leads to good results, especially in the context of human activity recognition.

In addition, we have noticed that the probability of occurrence of top-object classification had only slight correlation with accuracy of action prediction, exhibiting a Pearson correlation value of 0.24. The Pearson correlation values were slight yet marked, leading to the conclusion that objects that are classified with very high probability in fact have a considerable effect on accuracy but subsequently, the effect of probability of correct classification seems to wane. Another interesting phenomenon that we noticed was when top-object classification was wrong, and

Figure 3.7: Qualitative results for situations where the top-object and one of the verbs is correctly classified.

yet the associated action/verb was correct, as shown in Figure 3.8. What could have led to this situation? One thing we noticed was when the detected object was in the same super-category as the ground truth object. For instance, *orange* was incorrectly detected/classified as *banana*, and yet the verb *eat* was correctly identified. Another situation was when *airplane* was identified as *kite* and yet the verb *fly* was correctly identified. As expected, even if the top-object was correctly detected, we found that the verbs in the rare categories did not make it into the top-5 associated verb categories. For instance, for most situations the verb inspect did not occur with any of the categories such as *tie, car, toilet,* etc.

In addition, we conducted an additional study on the MS COCO dataset to determine how the verbs/actions were affected as we moved from the most probable

Figure 3.8: Qualitative results for situations where the top-object is incorrectly classified, yet the verb is correctly classified.

object in an image to the less probable ones. First, we selected all of the images in the validation set that contained a person/human. We ran object classifiers for the 79 annotated categories on all of these images. Using the most probable classified object in an image, we computed the similarity between this object and all of the verbs that occur in the ground truth sentences/captions corresponding to the validation image. The similarity between an object and the corresponding verbs was computed using Facebook's FastText natural language word embeddings obtained via training on the Wikipedia corpus [3]. Since the MS COCO dataset has 5 sentences associated with each image, we ran an NLTK parser on each of these sentences to obtain the corresponding verbs. For instance, if the most probable object in an image is knife, then we measured the similarity of the object (noun) knife with each of the verbs that occurred in the five sentences/captions

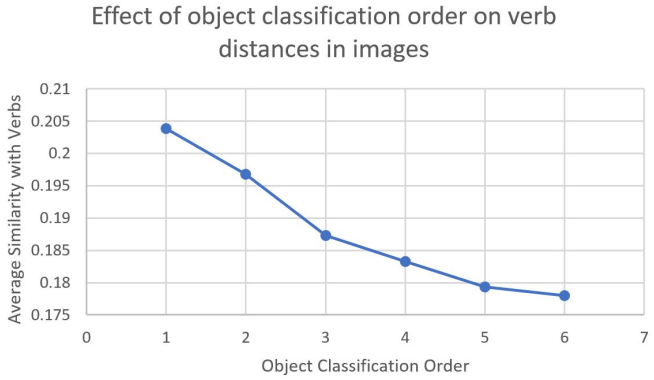

Figure 3.9: The average similarity with verbs decreases as we move from the most probable object classification in an image to less probable objects.

corresponding to the image and averaged these similarity values. We repeated the same process for the second, third, and fourth most probable objects in an image, and measured the average similarity with the verbs contained in the corresponding image captions. The results are shown in Figure 3.9.

We found that the most probable object had the least distance from (or greatest average similarity to) the verbs within the captions accompanying an image, followed by the second most probable object and so on. Hence, such an outcome lends support to our idea that the most probable object could lead to inferring the correct action in an image. It can be argued that the second-most probable object is only slightly less effective in predicting the correct action than the most probable object. However, in many cases the second-most probable object either co-occurs frequently with the most probable object, or is an object that lies within the same super-category as the most probable object, leading one to infer the same

actions/verbs in either case.

3.3.6 Conclusion

Overall, our results lend support to the idea that top-object detection along with its closely associated verbs could help in identifying the human action in most real-world situations, especially when the system needs to make a fast-and-frugal decision. In that case, taking a bet on top-object detection with a bet on associated action that was determined in a zero-shot manner using natural language word embeddings learned via training on a general text corpus could greatly help the system.

Bibliography

[1] S. Anwar, K. Hwang & W. Sung (2015). Structured pruning of deep convolutional neural networks. *arXiv preprint arXiv:1512.08571.*

[2] I. Alexiou, T. Xiang & S. Gong. (2016). Exploring synonyms as context in zero-shot action recognition. *Proc. IEEE Intl. Conf. Image Processing (ICIP 2016)*, pp. 4190-4194.

[3] P. Bojanowski, E. Grave, A. Joulin, & T. Mikolov. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

[4] Chao, Y. W., Wang, Z., He, Y., Wang, J., & Deng, J. (2015). Hico: A benchmark for recognizing human-object interactions in images. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1017-1025).

[5] G. Cheng, Y. Wan, A.N. Saudagar, K. Namuduri & B.P. Buckles (2015). Advances in human action recognition: A survey. Available online *https://arxiv.org/abs/1501.05964.*

[6] V. Delaitre, J. Sivic, & I. Laptev (2011). Learning person-object interactions for action recognition in still images. *Proc. NIPS 2011*, pp. 1503-1511.

[7] I. Everts, J.C. van Gemert & T. Gevers (2014). Evaluation of color spatio-temporal interest points for human action recognition. *IEEE Trans. Image Processing*, Vol. 23(4), pp. 1569-1580.

[8] A. Farhadi, M. Hejrati, M.A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier & D. Forsyth (2010). Every picture tells a story: Generating sentences from images. *Proc. Eur. Conf. Computer Vision (ECCV 2010)*, pp. 15-29.

[9] J. Gao & R. Nevatia (2016). Learning action concept trees and semantic alignment networks from image-description data. *arXiv preprint arXiv:1609.02284*.

[10] J. Gao, C. Sun & R. Nevatia (2016). ACD: Action concept discovery from image-sentence corpora. *ACM Intl. Conf. Multimedia Retrieval (ICMR 2016)* , New York, NY.

[11] D. Gentner (1981). Some interesting differences between verbs and nouns. *Cognition and Brain Theory*, Vol. 4(2), pp. 161-178.

[12] D. Gentner & I. M. France (1988). The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence*, pp. 343-382.

[13] D. Gentner (2006). Why verbs are hard to learn. *Action meets word: How children learn verbs*, pp. 544-564.

[14] G. Gigerenzer, & P.M. Todd. (1999). Fast and frugal heuristics: The adaptive toolbox. In Simple heuristics that make us smart (pp. 3-34). Oxford University Press.

[15] G. Gigerenzer. & D. G. Goldstein. (1999). Betting on one good reason: The take the best heuristic. In Simple heuristics that make us smart (pp. 75-95). Oxford University Press.

[16] Z. S. Harris (1954). Distributional structure. *Word*, Vol. 10(23), pp.146-162.

[17] M. Jain, J.C. van Gemert, T. Mensink & C.G.M. Snoek (2015). Objects2action: Classifying and localizing actions without any video example. *Proc. IEEE Intl. Conf. Computer Vision (ICCV 2015).*

[18] K.V. Katsikopoulos, & G. Gigerenzer. (2008). One-reason decision-making: Modeling violations of expected utility theory. Journal of Risk and Uncertainty, 37(1), 35.

[19] N. Krishnamoorthy, G. Malkarnenkar & R. Mooney (2013). Generating natural-language video descriptions using text-mined knowledge. *Proc. AAAI*, Vol. 1, pp. 541-547.

[20] L. Leemis (2018). http://www.math.wm.edu/ leemis/chart/UDR/PDFs/Zipf.pdf

[21] N. D. Lane, S. Bhattacharya, P, Georgiev, C. Forlivesi & F. Kawsar (2015). An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices. *Proc. ACM Intl. Workshop on Internet of Things towards Applications*, November, pp. 7-12.

[22] N. D. Lane & P. Georgiev (2015). Can deep learning revolutionize mobile sensing? *Proc. ACM Intl. Workshop on Mobile Computing Systems and Applications*, February, pp. 117-122.

[23] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro & F. Kawsar (2016). Deepx: A software accelerator for low-power deep learning inference on mobile devices. *Proc. IEEE Intl. Conf. Information Processing in Sensor Networks (IPSN 2016)*, April, pp. 1-12.

[24] Y. LeCun, J.S. Denker, S.A. Solla, R.E. Howard & L.D. Jackel (1989). Optimal brain damage. *Proc. NIPS 1989*, Vol. 2, November, pp. 598-605.

[25] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar & C.L. Zitnick (2014). Microsoft COCO: Common objects in context. *Proc. Eur. Conf. Computer Vision (ECCV 2014)*, pp. 740-755.

[26] S. Maji, L. Bourdev & J. Malik (2011). Action recognition from a distributed representation of pose and appearance. *Proc. IEEE Intl. Conf. CVPR*, pp. 3177- 3184.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado & J. Dean (2013). Distributed representations of words and phrases and their compositionality. *Proc. NIPS 2013*, pp. 3111-3119.

[28] T. Mikolov, K. Chen, G.S. Corrado & J. Dean (2013). Efficient estimation of word representations in vector space. *Proc. Intl. Conf. Learn. Rep. (ICLR 2013)*.

[29] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado & J. Dean (2013). Distributed representations of words and phrases and their compositionality. *Proc. NIPS 2013*, pp. 3111-3119.

[30] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, & X. Wang. (2017). Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence.*

[31] X. Peng, C. Zou, Y. Qiao & Q. Peng (2014). Action recognition with stacked Fisher vectors. *Proc. Eur. Conf. Computer Vision (ECCV 2014)*, pp. 581-595.

[32] J. Pennington, R. Socher & C.D. Manning (2014). Glove: Global Vectors for Word Representation. *Proc. Conf. Empirical Methods on Natural Language Processing (EMNLP)*, Vol. 14.

[33] J. Platt (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, Vol.10(3), pp. 61-74.

[34] R. Rehurek & P. Sojka (2010). Software framework for topic modeling with large corpora. *Proc. LREC 2010 Wkshp. New Challenges for NLP Frameworks*. Valletta, Malta, pp. 46-50.

[35] G. Sharma, F. Jurie & C. Schmid, (2017). Expanded parts model for semantic description of humans in still images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 39(1), pp. 87-101.

[36] K. Simonyan & A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *Proc. Intl. Conf. Learn. Rep. (ICLR 2014)*.

[37] Y. Tian, P. Luo, X. Wang, & X. Tang. (2015). Pedestrian detection aided by deep learning semantic tasks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5079-5087.

[38] Y. Tian, P. Luo, X. Wang, & X. Tang. (2015). Deep learning strong parts for pedestrian detection. *Proceedings of the IEEE international conference on computer vision*, pp. 1904-1912.

[39] L. Van der Maaten & G. Hinton (2008). Visualizing data using *t-SNE*. *Jour. Mach. Learn. Res.*, Vol. 9, pp. 2579-2605.

[40] A. Vedaldi & K. Lenc (2015). MatConvNet-convolutional neural networks for MATLAB. *Proc. ACM Conf. Multimedia Systems (MMSys 2015)*.

[41] H. Wang & C. Schmid (2013). Action recognition with improved trajectories. *Proc. IEEE Intl. Conf. Comp. Vis. (ICCV 2013)*.

[42] X. Xu, T. Hospedales & S. Gong (2017). Transductive zero-shot action recognition by word-vector embedding. *Intl. Jour. Computer Vision*, pp. 1-25.

[43] Y. Yang, C.L. Teo, H. Daume III & Y. Aloimonos (2011). Corpus-guided sentence generation of natural images. *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 444-454.

Chapter 4

Guessing Objects in Context

4.1 Chapter Outline

The success of deep learning methodologies in object detection and object classification can, in large part, be attributed to the availability of large-scale training data. However, in some real-world computer vision applications, generating sufficiently large training datasets is still a challenge because the required training data may not be readily available for some object categories or may be too expensive to annotate or generate. There is no shortage of applications hampered by the scarcity of training datasets. These applications could be greatly helped by models that could help overcome that. Some examples are annotating images in situations when training data are not available, generating weak labels for weakly supervised learning, or a robot trying to make sense of uncertain environment for which it has not been trained. Practical applications aside, the ability to infer

objects or situations in an image even without being exposed to those objects is of general conceptual interest to the AI community. For instance, humans read about a zebra as "a striped animal similar to a horse". Subsequently, they are able to recognize a zebra without ever having been exposed to one. Thus, humans have the ability to learn and make reasonably accurate inferences even after observing very few instances or even having seen no instances of a particular object category. In contrast, deep learning algorithms must be trained on fairly large-scale data to achieve the same level of success.

4.2 Introduction

In this paper, we propose a model that correctly identifies objects in an image using a limited number of visual classifiers. The proposed model can correctly identify an object in an image even when the object that the visual classifier is trained to detect is not present in the image. For instance, if the visual classifier indicates that a *car* is the most likely object in an image - even when the image does not contain a car - then using a natural language model that exploits word embeddings obtained via training on a publicly available natural language corpus, alternative hypotheses based on other vehicles and/or transport-related entities, such as *train*, *bus*, or even *road* or *pedestrian*, should be offered. Likewise, if the visual classifier indicates that the image contains a *bird*, then natural language-based priors should allow the system to suggest an *airplane* as an alternative hypothesis. In summary, nonspecific visual classifiers could still prove useful in accurately identifying an

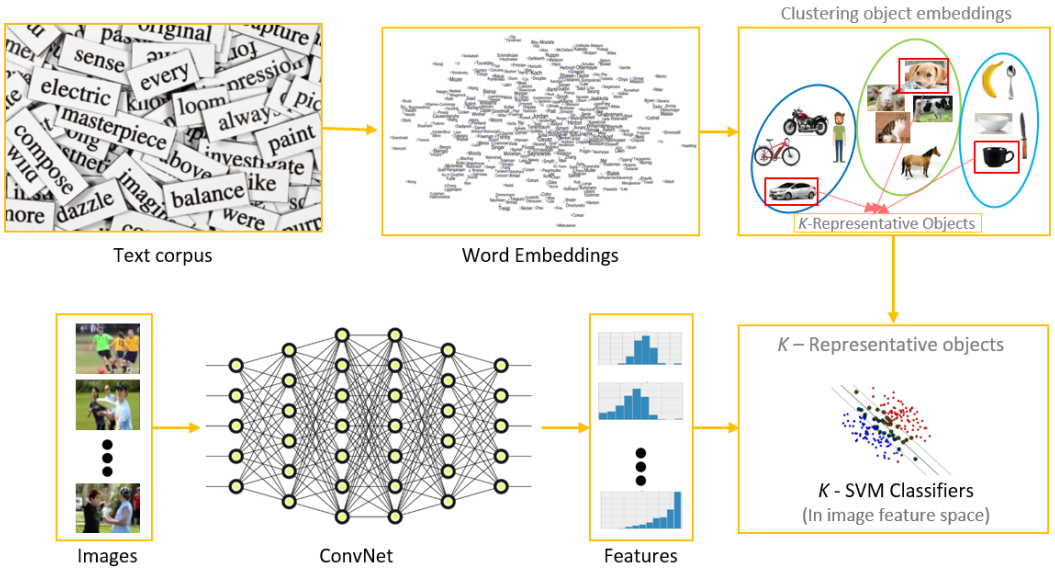

Figure 4.1: Overview of the proposed approach.

actual object in an image, as the most likely object deduced by the visual classifier could belong to the same general category as the actual object in the image. Alternatively, the context of the most likely object deduced by the visual classifier could be similar to that of the actual object in an image, or the most likely object deduced by the visual classifier and actual object in an image could coexist or co-occur in the real world with high probability.

The context of an object is any information that aids in the recognition of that object without explicitly using an object classifier/detector. In this paper, we try to guess an object in an image without using a classifier trained for that particular object. Hence, we do not use the features extracted from an image

to recognize a particular object in an image; instead, we try to use a classifier for another object to guess a particular object in an image. Prior to the deep learning era, various types of contexts were used to aid object recognition, the most prominent were scene-based context and object-object context. Regarding object-object context, for some models, the co-occurrence statistics were learned from text/web data [20]. After the rise of deep learning-based approaches, object recognition systems made a leap in performance and research into improving object recognition using context slowed down. However, several of the ideas developed could still improve deep learning-based approaches. For example, in this paper, we draw upon object-object context to guess objects in an image in situations where the data for training *all* objects is not available to programmers. In fact, data for training most objects are not available. We address situations in which we have data to train *extremely* few classifiers. For this purpose, we take advantage of natural language priors to guess objects in an image. We work under an assumption that real-world natural language could, to a certain extent, provide priors for visual context. Such natural language priors may not always be accurate because co-occurrences in the natural world do not necessarily reflect co-occurrences in the visual world. However, probabilistically, the natural language priors could still yield a reasonable performance in situations when the situation in an image is completely unknown. In other words, we could get something for nothing. For instance, cars and roads may occur together in the visual domain, but in natural language, people do not typically use the word roads in the proximity of the word car often. In addition, it is necessary to mention that we do not require co-

occurrences in the stricter sense with the word embedding models because these models are capable of assigning semantically similar words to nearby space in hypothetical word embedding space. This is because word embedding models are inspired by the distributional hypothesis in which words that share common linguistic context tend to have semantic similarity.

In this paper, to help guess objects in an image, we rely on pretrained word embedding models that were trained on Wikipedia text [1]. These pretrained embeddings provide a natural language context. Word embedding models such as word2vec [13], GloVe [18] and FastText [1] have gained popularity owing to their flexibility in solving challenging natural language processing problems. These models convert each word to a vector in low dimensional space, and these vectors are shown to have some interesting properties, among them, the tendency for vectors to be in close proximity when they are semantically similar. We will return to the details of these approaches in Section 4.5.

In the proposed approach, as a first step, k-means clustering is performed on natural language word embeddings obtained via training on a publicly available natural language text corpus. This results in the formation of object clusters that share some degree of similarity. From each object cluster, a representative object is selected to train the corresponding visual classifier. Typically, the object within the cluster with the greatest number of training instances available is deemed to be the representative object for that cluster. After the visual classifiers for all of the representative objects have been trained, they can be employed on real test images. The visual classifiers for all the representative objects are executed

against the test image to determine the object(s) in the test image. Using the most probable object hypothesis and natural language word embeddings obtained via training on a public text corpus, it is possible to refine the initial object hypothesis to propose alternative object hypotheses for the test image. Previously, Sharma et al. [22] presented a preliminary object detection and classification model based on the above approach with encouraging results. In their approach, Sharma et al. [22] used word embeddings trained on natural language captions that accompany the test images. In this paper, we extend the approach of Sharma et al. [22] by exploiting natural language word embeddings (such as FastText) obtained via training on a general text corpus (such as Wikipedia) and perform a more comprehensive analysis of the results.

While zero-shot classifiers [25] or one-shot classifiers can be used to address the paucity of training data, both involve creating visual classifiers from existing visual classifiers, which is obviated in the proposed approach, thus making it more generally applicable. Moreover, the zero-shot approaches are unlikely to scale well in cases where a unique visual classifier is required for each object category. In contrast, the proposed approach requires only a small number of classifiers and that their guesses be refined by offering related alternatives, which is fundamentally a different problem. The advantage of the proposed approach is that we eliminate the need to generate large visual training sets (i.e., a training set for each object category), which could be difficult to generate for certain object categories or may not be feasible in certain real-world applications.

4.3 Motivation

To accurately classify or guess an object in an image, the proposed approach embodies the following observations about the real world. However, before we proceed, it is necessary to mention that how word embeddings capture the following observations is not well understood. Just as in deep learning, word2vec and related algorithms belong to the "practice precedes theory" paradigm; hence, in the absence of good theory, only intuitive observations can be made. The word2vec family of algorithms convert each word to a vector, and these vectors have some interesting properties. In this paper, we capitalize on the interesting properties of these vectors. However, rigorous theory is currently lacking, although some good attempts exist in the literature [18]. We believe that the following observations about the real world are likely to help the word2vec family of algorithms to guess objects in an image.

1. The first observation is that similarity in the visual deep learning feature space for object categories that belong to a more general category often translates to proximity of the corresponding natural language word embeddings in the hypothetical space. In other words, in the hypothetical space, object categories that belong to the same general category tend to map onto natural language word embeddings that are in close proximity. For instance, the object categories *orange* and *apple* belong to the general *fruit* category and would be in close proximity in the hypothetical space, as would the objects *car*, *truck* and *train*. Similarly, in the visual world, many

object categories that belong to a single general category tend to share the same (or similar) visual deep learning features. This facilitates the identification of objects that belong to the same general category. For instance, if a *car* classifier is used on an image that does not contain a car, but rather a truck, the *car* category would be output. Intuitively, this is expected because when computing deep learning features using a multilayer convolutional neural network (CNN), the lower layers of the CNN capture generic features such as edges and corners that are shared by the object categories *car* and *truck*. Likewise, the middle CNN layers capture features related to object components, which again tend to be shared by the object categories (*car* and *truck*). Thus, deep learning features for visually similar object categories, especially those that belong to the same general category, will tend to exhibit a high degree of similarity or proximity in feature space. Consequently, if a *truck* is present in an image, it could be classified as a *car* if a *car* classifier were employed on the image. Toy Experiment—We conduct a toy experiment on the category *car*. We select a random car image as well as one random image of each annotated object in the MS-COCO dataset [11], ensuring that each image has only one annotated object. Using these images, we compute the similarity between the car image and all the other objects in the visual deep learning feature space. For textual similarity, we extract word embedding features from FastText pretrained embeddings pertaining to car and all the other annotated objects in the MS-COCO dataset. Using these extracted vectors, we compare the similarity between the car and other

objects. Next, we calculate the correlations between similarities in the visual deep learning space and the word embedding space. The correlation turned out to be 0.4, which can be considered moderate, thus suggesting that there is a moderate relation between the visual and textual data.

2. The second observation is that the general semantic context in the form of object co-occurrences in natural language is effectively captured by word embeddings in the hypothetical space. Moreover, object co-occurrences in natural language often correspond to object co-occurrences in the visual world. In their seminar paper on the formulation of global vector-based word representations (termed GloVe) by Pennington et al. [18] provided a formal mathematical theory for the effectiveness of word embeddings in hypothetical space in terms of their ability to capture general semantic context. To revisit the previous example, a local search in the hypothetical space will yield alternative object hypotheses, such as *truck* along with other vehicle categories that are in semantic proximity to the word embedding of the object *car*. In our experiments we use the *FastText* natural language word embeddings obtained via training on a general text corpus such as Wikipedia.

3. The third observation is that objects that share the same general context in the visual world often share the same general semantic context in the natural language world. For instance, *bird* and *airplane* both fly in the sky and hence will tend to share similar visual features because of their shared semantic context, such as wings for flying, and these similarities will be reflected in their common visual deep learning features. Similarly, in the natural lan-

guage world, the word *flying* is often associated with both airplanes and birds, thereby imparting to them the same general semantic context. Hence, if the visual classifier identifies a *bird* as an object in an image when the image actually contains an *airplane*, their corresponding natural language word embeddings would provide an association between the objects *bird* and *airplane* to refine the initial guess. Toy Experiment - We consider the two categories bird and airplane for the purpose of this toy experiment. Again, we select random images from the MS-COCO dataset for each category while ensuring that only one object occurs in an image. We extract one random image for each category. In the visual deep learning world, the mean similarity of *bird* to all categories is 0.22, and the mean similarity of *airplane* to all categories is 0.26. However, the similarity between airplane and bird is only 0.31. In the textual world (FastText embedding space), the mean similarity between bird and all categories is 0.13, and between airplane and all categories it is 0.16. However, the similarity between bird and airplane is 0.19. Thus, in both the visual and textual worlds, the similarity between airplane and bird is greater than the mean similarities of bird and airplane with other categories. This suggests that one could aid in the detection of another. The effect is not particularly strong, yet it is sufficiently substantial to matter in many real-world situations.

4.4 Related Work

In recent years, a wide variety of models for context-based object recognition have been proposed. Divvala et al. [3] identified and categorized various context sources: pixel-level interactions, semantic context, GIST, geographic context, illumination and weather, cultural context, and photogrammetric context, among others. Divvala et al. [3] further demonstrated that incorporation of each type of context leads to moderate improvements in the recognition accuracy. However, two classes of contextual models have gained prominence in recent years, i.e., scene-based contextual models and object-based contextual models [21]. In a scene-based contextual model, the statistics pertaining to the entire scene are used to detect and locate the scene objects, whereas in an object-based contextual model, objects in the spatial vicinity of the target object are used to recognize the target object.

In the Co-occurrence Location and Appearance (CoLA) model, a widely used object-based contextual model proposed by Rabinovich et al. [20], bottom-up image segmentation is followed by a bag-of-words-based object recognition system. Additionally, a conditional random field (CRF) is used to capture the inter-object interactions in the dataset. Although capable of capturing obvious reoccurring patterns in the real world, the CRF-based contextual model cannot identify certain subtle patterns that may characterize similar objects. For example, a rear view of car is often encountered in the spatial vicinity of an oblique view of a building, yet this subtlety not captured by a CRF-based contextual model [12]. To address these shortcomings, Malisiewicz and Efros [12] introduce a visual memex model, which is based on the premise *"Ask not 'what this is, but what this is like'."* In their

visual memex model, Malisiewicz and Efros [12] use a graph-theoretic approach to model real-world images, where similar objects are connected by similarity edges while objects that are contextually related are connected by context edges. Consequently, using the earlier example, different types of buildings are connected by similarity edges, whereas a building and a car are linked via a context edge in the visual memex model [12]. The graph-theoretic model automatically learns the visual memex graph from the input images and is shown to successfully outperform the CoLA model [20] on Torralba's context challenge dataset [27].

Heitz and Koller [9] introduced the Things-and-Stuff (TAS) model, a category-free model that relies on unsupervised learning. In the TAS model, a *thing* is an object that has a concrete shape and size, whereas *stuff* is malleable but has a repetitive pattern and typically contains *things.* For instance, a *car* (thing) is most likely to be found on a *road* (stuff), and likewise a *cow* (thing) on *grass*(stuff). The TAS model is shown to capture regularities not inherent in other contextual models. Another category of contextual models that has gained prominence in recent times is scene-based models. Choi et al. [2] proposed a scene-based contextual model developed using pre-labeled images by optimizing information derived from GIST features, the relative locations of objects, and a co-occurrence tree. The co-occurrence tree is generated using a hierarchical CRF, where a positive edge denotes object co-occurrence, and a negative edge indicates that the corresponding objects do not occur together. Choi et al. [2] used a deformable parts model as their baseline detector on the SUN dataset introduced by Xiao et al. [29] and applied it to the output of the baseline detector. They showed that it outperforms

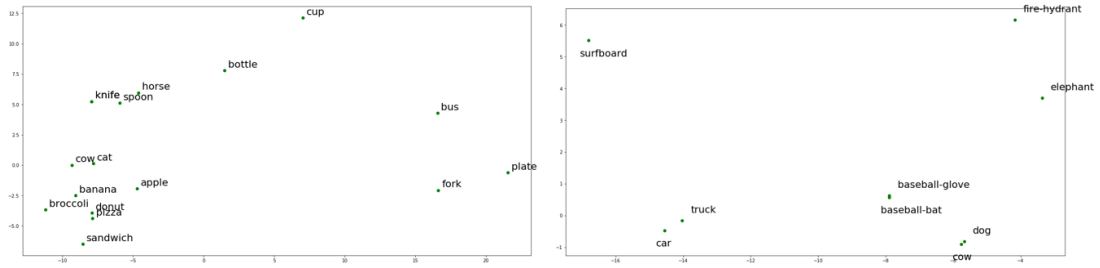

Figure 4.2: tsne Diagram of word embeddings of MS-COCO annotated objects obtained from Fastext trained on wikipedia.

the baseline detector when deformable object parts are present in the input image.

More recently, due to the advent of deep learning, several effective attempts have been made to blend context with deep learning methods [7]. For instance, Sun and Jacobs [26] proposed an architecture that can be employed to predict a missing object in an image by exploiting contextual information. Similarly, Zhang et al. [30] integrated 3D context into deep learning for 3D holistic scene understanding, whereas Gonzalez et al. [8] employed a deep learning model to detect object parts by using object context. One disadvantage of all of the aforementioned contextual models is that a very constrained environment is assumed in the model learning phase, i.e., that pre-labeled images are readily available and that the labeled images reliably capture the co-occurrence patterns. The approach proposed in this work is unique in that - instead of using context to improve object recognition performance - the context itself is guessed based on the output from a small number of object classifiers.

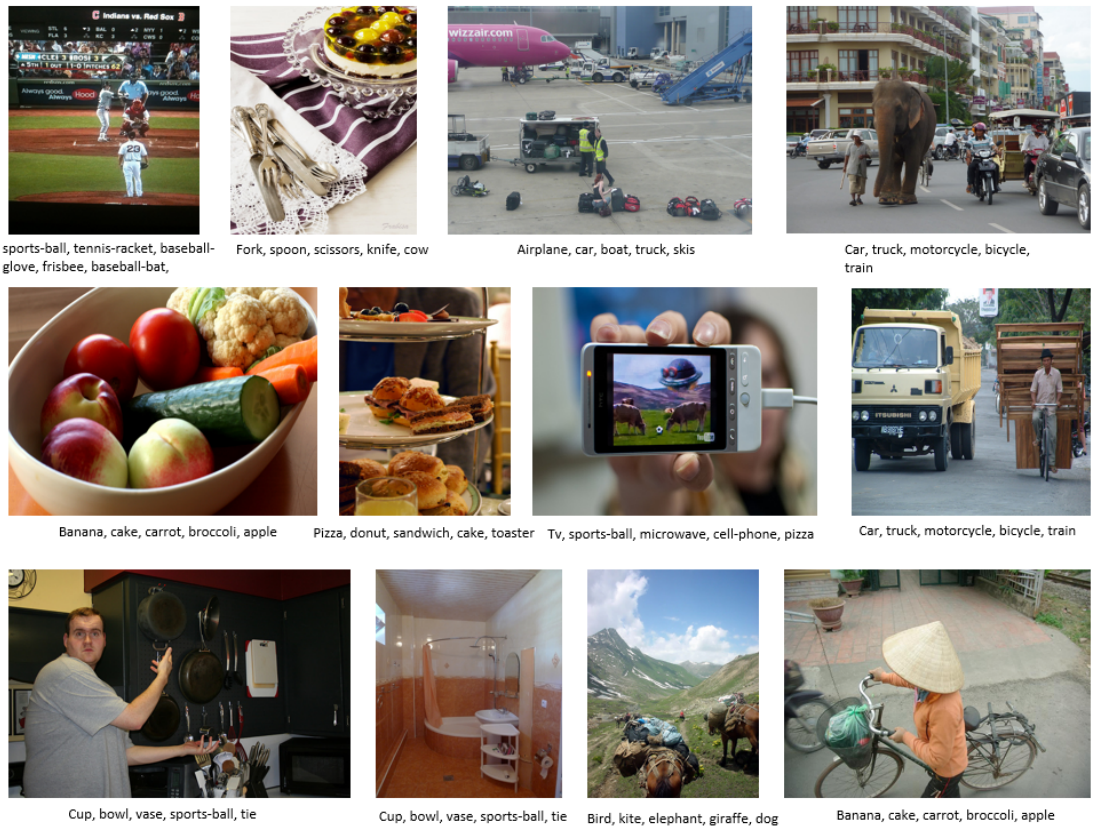

Figure 4.3: Qualitative Results: Top Row: When the correct top-object classification is able to guess at least one more object in an image. Middle Row: When the incorrect top object classification is able to guess at least one more object in an image. Bottom Row: When the incorrect top object classification is NOT able to guess at least one more object in an image.

4.5 Proposed Approach

Our approach is explained step by step in this section. Briefly, in our approach, we train very few classifiers and guess objects in an image using natural language embeddings. The major advantage of this approach is that it eliminates the need to generate training datasets, which is an expensive process in real-world applications. In addition, in situations when reliable datasets are not available for rare categories, our approach could yield some performance improvements. In passing, an additional advantage could be the need to execute only a few classifiers at test time. The proposed approach is composed of the following steps::

1. Extract the word embeddings corresponding to objects in MS-COCO from FastText [1]. These word embeddings are low-dimensional vectors for each word corresponding to an object. The FastText word embeddings were trained on Wikipedia using the skip-gram approach; we used the embeddings provided by [1]. FastText is an enhancement of word2vec [1]. The authors claim that superior embeddings can be created if, unlike word2vec, instead of whole words, the model is trained with n-grams created by splitting each word and then averaging the vectors corresponding to each n-gram of the word as well as the original word, superior embeddings can be created. FastText's emphasis on the morphological structure of words while training makes it an attractive choice for a variety of reasons. In addition, [1] have shown that these vectors yield superior results. Therefore, in our research, we used FastText embeddings pretrained on Wikipedia. Before proceeding to the next stage, we digress to briefly explain word2vec and similar algorithms such as FastText [1] and GloVe [18]. Word2vec converts each word to a low dimen-

sional vector using either the skip-gram approach or the continuous bag-of-words approach. The expectation is that word vectors that are synonymous will be in close proximity in this space. One of the major applications for such word embedding approaches is to recover linguistic regularities from a text in an unsupervised fashion [13, 14]. For example, performing simple vector arithmetic on the word vectors can provide interesting results [15]. One of the classic results involves a word-analogy task. For example, - one interesting result of this approach [15] was:

$vec(king) - vec(man) + vec(woman) \sim vec(queen)$

Another example is the recognition of capital cities:

$vec(Washington) - vec(USA) + vec(UK) \sim vec(London)$

This approach also captures other, subtler relations, such as plurality:

$oranges - orange \sim buses - bus$

Such linguistic regularities were not input into this algorithm; they were learned in an entirely unsupervised manner from the context of the words themselves. Thus, the major advantage of the proposed approach is that we no longer need to provide explicit semantic annotations to understand linguistic regularities. After the advent of word2vec, many new developments, such as GloVe [18], took place in this arena. The main difference between word2vec and GloVe [18] is that word2vec employs a neural network to learn embeddings, whereas GloVe employs a word-to-word co-occurrence matrix of a corpus and learns embeddings from this matrix. Although the GloVe paper [18] successfully explained some of the theory behind the mysterious success of word2vec, practically, the differences between the quality of embeddings produced by the two approaches are minimal and could depend on

the application.

2. Cluster the embeddings obtained in the preceding step using k-means clustering, where the optimal value of k is determined by the performance of the validation set. After k-means clustering, each cluster will encompass a certain number of objects that tend to have particular relationships –they are co-occurring, belong to the same general category, or share the same general context –as explained in the previous section. The optimal value of k is determined by the algorithm's performance on the validation set. We emphasize that all the k-means clustering was performed on word embeddings obtained from natural language public datasets; no visual information was used.

3. From each of the clusters for a particular value of k, an object category that is representative of this cluster is selected. The representative object that is selected for training is the one with the greatest number of instances in the training set in MS-COCO. An analogy could be drawn to the real world where we have unbalanced training sets, with some categories outnumbering others by a significant margin. Thus, our model resembles and is applicable to real-world situations as a proof of concept. It can be counter-argued that there is a possibility that there are no training instances with no representative objects available for a particular cluster. In such a situation, we believe that there are possible workarounds of our proof-of-concept approach, such as determining a different value of k where each cluster has at least one of the object's training instances.

The representative object classifier is trained using deep learning features followed by a support vector machine. These classifiers are further calibrated using

Platt scaling to convert SVM confidences to probability values. Ideally, it is preferable to have clusters that resemble co-occurrence patterns in the images. However, this depends on a variety of factors, such as the extent to which the distribution of unrelated natural language data corresponds to the distribution of the imagery data on which it is being tested. In addition to capturing co-occurrences, categories belonging to the same general category (such as orange and apple) are also useful.

4. Given the test data, run representative object classifiers of all representative object classifiers (aka cluster centers) on an image, and select the most probable object/objects they contain. While the most probable object may not exist, this procedure can still provide useful clues concerning objects that could be present, as explained in Section 4.3.

5. Using these most probable object/objects as the starting point, other objects in an image could be identified using their cosine similarities in the hypothetical word embedding space. In the current implementation, the most probable, the second most likely, and the third most probable detections are utilized for guessing other objects in the image.

In this paper, we conduct proof-of-concept experiments on the MS-COCO dataset; hence, we make a closed-world assumption. We expect that such a model is generalizable to real-world situations. First, most real-world vision applications tend to be closed-world situations, whether in manufacturing, retrieval or robotic navigation. Second, even if we assume a more open-world assumption, the objects that co-occur with reasonable regularity in the real world would cause such a

model to generate at least moderate results. For instance, if our classifier detects a computer keyboard in an image, we could reliably expect a computer and mouse to be in the image, even if the possible object space is huge.

In our approach, we use 1 to 3 top-object classifications to guess other objects. We reiterate that the guesses for the objects in an image are made from representative objects selected using the steps explained above. It could be argued that even in the test set of MS-COCO, the distribution of most representative objects remains the same as the training set. However, even in many real-world applications, the distribution of training and testing sets tends to remain the same. Even when the distribution of training and testing sets is different in real-world applications, we believe there is sufficient regularity in the world so that the system inspired by our proof of concept will yield reasonable results.

At the testing stage, object detectors for all clusters centers (i.e., representative objects) are executed on the image. For 3 top object classifications, given a set of nouns N and top object classifications n_1, n_2 and n_3, the closest guesses from set N are given by:

$$\arg\max_i \ \{SIM(n_i, n_1) + SIM(n_i, n_2) + SIM(n_i, n_3)\} \tag{4.1}$$

where $SIM(n_i, n_1)$ and $SIM(n_i, n_2)$, $SIM(n_i, n_3)$ are the cosine similarities of noun n_i to nouns n_1, n_2 and n_3 respectively.

In contrast, if we use only 1 top object classification for guessing other objects, then, Given a set of nouns N and top object classifications n_1, n_2 and n_3, the

closest guesses from set N are given by:

$$\arg\max_{i} \{SIM(n_i, n_1))\} \tag{4.2}$$

where $SIM(n_i, n_1)$ is the cosine similarities of noun n_i to nouns n_1 respectively.

Using the above approach, it is inevitable that sometimes visually similar objects (or objects in the same general category) may end up in distinct natural language clusters and visually dissimilar objects may end up in the same NL cluster. To circumvent this problem, we search for the value of k that tends to assign objects that have a relationship (either belonging to same general category or co-occurring) to the same cluster and objects that do not have a relationship to different clusters. However, even then, the priors and language models learned from unrelated language datasets could succeed only to the point at which the objects and their relationships in the testing set match those in the language models (in our case FastText on Wikipedia). Nevertheless, it is our belief that there is sufficient regularity in the world to cause language models learned on unrelated datasets to be a reasonable choice. In addition, one of the goals and advantages of the word2vec family of approaches was to *remove explicitness* from the process. The idea is these word vectors, which are learned in an unsupervised manner on an unstructured text without known ontologies or any other categorical information, can still do an excellent job on real-world problems.

In addition, in the real world, we assume that the test set for object recognition, even if different from the training set on which representative object classifiers are trained, could still benefit from regularities in the unstructured text in the form of

language models. Under this assumption, our model could succeed in a reasonable manner in most situations because regularities from the natural language world generalize to the visual world.

4.6 Experiments

Training: We used k-means clustering to cluster objects using FastText embeddings trained on Wikipedia. We experimented with several k values, namely, {3, 5, 7, 9, 11, 13, 15, 17}, which were applied to a validation set of 20,000 images obtained by splitting the 40,000 validation images in MS-COCO into 20,000 images each for validation and testing. The optimal value for k was determined by running experiments on the validation set. The k value that yielded the best results was adopted for subsequent processing. We found that among the k values in the above set, k=17 yielded the best results on validation set. Because k-means clustering is non-deterministic, we used ten iterations of each value of k and calculated the average of the obtained results.

We trained object classifiers using features extracted from the fc-7 layer of the VGG architecture [23], followed by support vector machines, on the MS-COCO training set images. We trained classifiers for 79 objects while excluding the *person* category from our experiments, as a person could co-occur with any type of object.

For each experiment pertaining to a particular iteration for a particular k value, we selected representative objects from each cluster for training. For example, - for a k value of 3, we selected 3 representative objects for training. Hence, k

objects were trained for each experiment. In each experiment pertaining to a particular k value, we assume that other objects in the MS-COCO dataset are not available. Finally, we chose k=17 for reporting our results because on validation set, for values of k from 3 to 17, it yielded the best performance. In addition, k=17 implies about twenty percent of the classes are seen while eighty percent are unseen. This is in contrast to previous work on zero-shot object recognition where most classes tend to be seen and fewer are unseen.

During testing, we ran the classifiers for all the representative objects in a given image. The most likely representative objects that occurred in an image were then chosen to determine the other objects in an image. These other objects were predicted from the FastText embeddings using equation 4.1 (for a guess made with the 3 most probable objects) and equation 4.2 (for a guess made with 1 top-most probable object only).

We evaluated our results on the test set using ground truth object annotations, as well as the aforementioned predictions (guesses), using the following metrics:

Top-1 accuracy: Achieved when the top 1 object determined by our model intersected with at least one object in the ground truth annotations of the test image when all the objects except the representative objects are considered for classification.

Top-1c accuracy: Attained when the top 1 object determined by our model intersected with at least one object in the ground truth of the test image when only the representative objects were considered for classification. For the top one object, it would be interesting to see how simply knowing the representative object

Table 4.1: Comparison of our approach with most frequent baseline for guesses made with three most probable objects in an image for k=17.

	Top-1	Top-3	Top-5
Most-Freq	11%	25%	31%
Ours	22%	46%	53%

alone affects the accuracy vs another object that is predicted by the representative object. It would be important to know how far we can go when executing classifiers for representative objects only.

Top-3 accuracy: Achieved when any of the top 3 objects predicted by our model intersected with at least one object in the ground truth annotations of the test image. This included all objects (the representative object as well as others).

Top-5 accuracy: Attained when any of the top 5 objects determined by our model intersected with at least one object in the ground truth annotations of the test image. This could include all objects (the representative object as well as others).

Most-frequent baseline –The most frequent object/objects were determined by the most frequently occurring objects in the training set. The most frequent objects in MS-COCO (excluding persons) in the training dataset (in descending order) were *chair, car, dining table, cup, bottle*, and *bowl*. For top-n accuracy, the top-n most frequent objects were used for evaluation on the test set. The most frequent baseline is a fundamental baseline widely used by machine learning researchers. This baseline has been shown by several authors to be sometimes difficult to beat [19].

Table 4.2: Comparison of our approach with most frequent baseline for guesses made with two most probable objects in an image for k=17.

	Top-1	Top-3	Top-5
Most-Freq	11%	25%	31%
Ours	26%	47%	55%

Table 4.3: Comparison of our approach with most frequent baseline for guesses made with only one most probable objects in an image for k=17.

	Top-1c	Top-1	Top-3	Top-5
Most-Freq	11%	11%	25%	31%
Ours	35%	13%	47%	55%

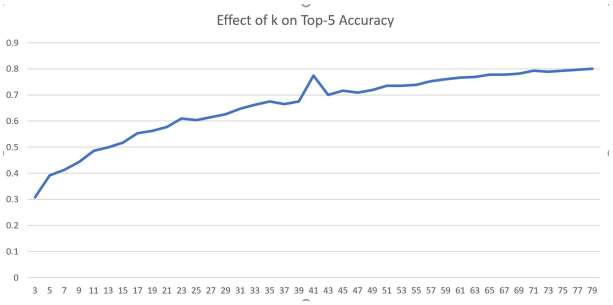

Figure 4.4: Effect of value of k used in k-means clustering on Top-5 Accuracy.

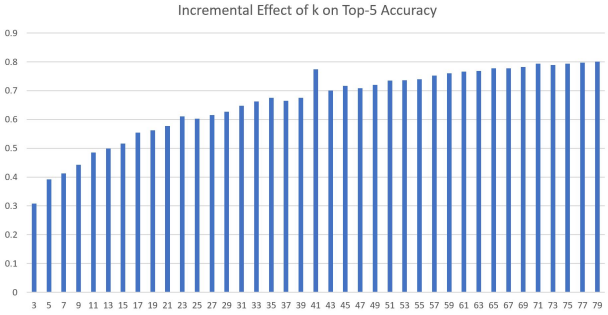

Figure 4.5: Incremental Effect of K on Top-5 accuracy from k=3 to k=79.

4.7 Results and Discussion

1. The results reported here show that our approach is superior to the most frequent baseline when objects are predicted using the top-most likely, second most likely, and third most likely objects. This finding lends support to the idea that using only a few classifiers to guess various objects from language models learned from public datasets is an effective technique.

2. To see how the increase in k effects accuracy, we conducted additional experiments with the k values from k=3 to k=79. Our results further show that the object prediction accuracy increases with the value of k in k-means clustering, as shown in figures 4.4 and 4.5. This increase could be attributed to one or more of the following factors: (1) either performance is improved by increasing the number of classifiers or increasing the number of clusters improves the performance; and (2) a greater number of clusters tends to assign objects possessing some type of relationship into the same cluster and tends to correctly separate unrelated objects.

3. The present investigation also revealed that performance does not improve due to using the most likely object in an image: the same results can be attained with the top-2 or top-3 most likely objects. This implies that the most confident classification is sufficient for making accurate guesses concerning other objects in an image. Importantly, at least for FastText embeddings, including additional information beyond one object classification does not seem to be beneficial when identifying/guessing other objects in an image; hence, this knowledge removes an extra layer of complexity.

4. Because the results obtained using top-5 accuracy were superior to those

cluster	1	cake donut pizza toaster oven carrot sandwich refrigerator banana **spoon** broccoli
cluster	2	skis surfboard snowboard **backpack** boat kite hair-drier
cluster	3	**dining-table** wine-glass toilet potted-plant bench book
cluster	4	laptop keyboard **cell-phone** apple mouse
cluster	5	**bowl** orange
cluster	6	**car** truck motorcycle bicycle airplane suitcase
cluster	7	**clock**
cluster	8	train **bus** stop-sign
cluster	9	**cup** vase
cluster	10	parking-meter **traffic-light** fire-hydrant
cluster	11	frisbee skateboard **umbrella**
cluster	12	**remote** microwave
cluster	13	**dog** cow hot-dog elephant sheep giraffe cat horse zebra bird
cluster	14	**sink**
cluster	15	scissors toothbrush knife handbag bed bottle couch **chair**
cluster	16	baseball-glove sports-ball baseball-bat tennis-racket tie **tv**
cluster	17	bear teddy-bear **fork**

Figure 4.6: Clusters for k=17. The chosen centers are in bold.

related to top-3 and top-1 accuracy, the likelihood of obtaining an accurate classification for at least one object increases when the classifier attempts to guess a number of objects in an image.

4.8 Effect of training word embeddings on captions accompanying images

In this section, we study the effect of using word embeddings trained on captions accompanying training images on the accuracy of guessing objects. For this purpose, we train FastText embeddings[1] using default parameters on these captions and create a 300-d vector for each word. As a preprocessing step, we change all

Table 4.4: Results for guesses made with only one most probable objects in an image, when FastText embeddings are trained on captions accompanying images. COCOemb is FastText embeddings trained on MS-COCO captions. The results are reported for k=17, which approximately represents twenty percent of all categories.

	Top-1c	Top-1	Top-3	Top-5
Most-Freq	11%	11%	25%	31%
COCOemb	34.6%	16.8%	48.8%	57.7%

words referring to a person such as *man*, *woman*, etc. to *person*. Also, we stem all the words using porter stemmer. Other than this, the set up remains same as previous for this experiment.

As shown in table 4.4, we see that our results are only slightly better than if we use pre-trained embeddings trained on Wikipedia dataset. It would be expected that we would see significant gains in accuracy compared to public datasets. However, this was not the case. Training embeddings on captions only lead to marginal improvements. This suggests that natural language real world datasets provide reliable priors for guessing objects in context.

4.9 Failure Analysis

Even though natural language provides good priors for images, it is well known that the natural language world does not necessarily map to the visual world in all situations. In addition, the reasons why word vectors have such interesting properties is currently not well understood. Hence, such limitations are inevitable, and it

is necessary to address those situations. To conduct a failure analysis, we selected one iteration of k=17; the clusters for k=17 are listed in figure 4.6. We analyze the false positive rates for the categories guessed by each most probable-object (also cluster center) for the selected cluster, and then we analyze the categories with the highest false negative rates for the most likely object. For the purpose of this analysis, the false positive category is the category rated as the most-probable object in an image, as one of four guesses made for this object. In the original results, we used top-5 accuracy, which included the most probable object. However, for the purpose of this analysis, we exclude most probable object.

For each most probable object (cluster center), we calculate the false discovery rate FDR for each of the categories in the COCO dataset as follows:

$$FDR = \frac{F_p}{F_p + T_p} \tag{4.3}$$

where F_p represents the false positives pertaining to a particular category and T_p represents the true positives.

From the false discovery rate table, we can observe certain situations. First, rather non-obviously, the categories with the highest FDR turned out to be ones that are highly unlikely to co-occur, such as *bowl* and *baseball-bat*, *fork* and *cow*, and *dining-table* and *toilet*. This could be attributed to the FastText word vectors because the vectors of these objects turned out to be in close proximity, even though they are unlikely to occur together in images. In the second situation, interestingly, the categories with the highest FDR turned out to be ones which that could plausibly co-occur, yet do not. For example, - *backpack* and *laptop*, and

spoon and *toaster*. One reason could be that these categories, although related in the textual world, tend not to visibly co-occur in an image. This could be attributed to the fact that the natural language context is sometimes unable to augment the visual context. Another situation with the highest FDR was when another category from the same general category was retrieved, such as *train* by *car* and *bicycle* by *bus*. Cars and trains, although they belong to the same general category, do not tend to co-occur in the real world; hence, this is another example of the natural language context being unable to assist with visual recognition. Nevertheless, language priors provide reasonable guesses for many, if not all, real-world situations.

Table 4.5: Table reflecting categories with highest FDR for each cluster center when that particular cluster center was the most probable object.

Cluster Center	**Category with Highest FDR**	**False Discovery Rate**
Spoon	Toaster	1
Fork	Cow	1
Bowl	Baseball-bat	1
Bus	Bicycle	0.93
Sink	Boat	1
Dining-Table	Toilet	1
Cup	Sports-ball	1
TV	Sports-ball	1
Remote	Microwave	0.98
Backpack	Laptop	1
Dog	Hot-Dog	1
Traffic-light	Parking-Meter	0.97
Chair	Umbrella	0.97
Clock	Mouse	1
Cell-phone	Microwave	1
Umbrella	Cat	1
Car	Train	0.94

To obtain further insight, we investigated the categories that had the highest

Table 4.6: Table reflecting categories with highest Number of False Negatives for each cluster center when that particular cluster center was the most probable object.

Cluster Center	**Category with Highest False Negatives**
Spoon	Dining-table
Fork	Dining-table
Bowl	Dining-table
Bus	Handbag
Sink	Bottle
Dining-Table	Pizza
Cup	Dining-table
TV	Laptop
Remote	Chair
Backpack	Snowboard
Dog	Surfboard
Traffic-light	Car
Chair	Tennis-racket
Clock	Bird
Cell-phone	Tie
Umbrella	Boat
Car	Bench

number of false negatives for each cluster center. A false negative occurs for a category when it exists in an image, but our approach was unable to guess the category. First, surprisingly, categories such as *spoon, fork*, and *bowl* were not able to retrieve dining table. Similarly, the category traffic-light was not able to retrieve car. This could again be attributed to natural language context providing insufficient assistance to visual context in some situations, especially when words are mapped to hypothetical space.

4.10 Conclusion

Our models and results lend support to the concept that - even using a limited number of object classifiers - other objects in an image could be successfully guessed, even from language priors learned on unrelated datasets such as Wikipedia. Moreover, even the ability to classify incorrect objects in an image can yield useful cues that could help in guessing correct objects in an image. However, such natural language priors are not always helpful, as described by the results of our failure analysis. Nevertheless, word embeddings trained on unrelated public datasets can still yield effective priors - at least good enough to matter in uncertain situations. Future work will involve incorporating feedback mechanisms to improve classification. In addition, many weakly supervised object detection and image captioning algorithms could benefit from our approach. Additionally, this approach could be used by AI practitioners to help conceptualize human intelligence because humans tend to recognize numerous categories without ever having been exposed to them.

Bibliography

[1] P. Bojanowski, E. Grave, A. Joulin, & T. Mikolov (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

[2] M. Choi, A.Torralba, and A. S. Willsky (2012). A Tree- based Context Model for Object Recognition IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(2), 240-252.

[3] S. K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros and M. Hebert (2009). An Empirical Study of Context in Object Detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[4] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman (2010). The PASCAL Visual Object Classes (VOC) Challenge, International Journal of Computer Vision, 88(2), 303-338.

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan (2008). A Discriminatively Trained, Multiscale, Deformable Part Model Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[6] C. Galleguillos, A. Rabinovich, and S. Belongie (2008). Object categorization using co-ocurrence, location and appearance, Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[7] G. Gkioxari. (2016).Contextual Visual Recognition from Images and Videos. University of California, Berkeley.

[8] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari. (2017). Objects as context for part detection.arXiv preprint arXiv:1703.09529.

[9] G. Heitz and D. Koller (2008). Learning spatial context: Using stuff to find things, Proc. European Conference on Computer Vision.

[10] J. Liu, R. Hu, M. Wang, Y. Wang, and E. Y. Chang (2008). Web-scale image annotation. In Advances in Multimedia Information Processing-PCM 2008 (pp. 663-674). Springer Berlin Heidelberg.

[11] T. Y. Lin, et al. 2014. Microsoft COCO: Common objects in context. *In Proc. ECCV*, (pp. 740-755).

[12] T. Malisiewicz, A.A. Efros (2009). Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships, Proc. Neural Information Processing Systems.

[13] T. Mikolov et al. 2013. Distributed representations of words and phrases and their compositionality. *In Proc. NIPS*, (pp. 3111-3119).

[14] T. Mikolov. 2013. Efficient estimation of word representations in vector space. *In Proc. ICLR*

[15] T. Mikolov, W. T. Yih, and G. Zweig. (2013). Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 746-751).

[16] A. K. McCallum (2002). "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu.

[17] A. Oliva and A. Torralba (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision, 42(3).

[18] J. Pennington, R. Socher, and C. Manning. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[19] J. Preiss, J. Dehdari, J. King, and D. Mehay. (2009, June). Refining the most frequent sense baseline. In Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (pp. 10-18). Association for Computational Linguistics.

[20] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora and S. Belongie (2007). Objects in Context, Proc. International Conference on Computer Vision.

[21] A. Rabinovich and S. Belongie (2009). Scenes vs. objects: a comparative study of two approaches to context-based recognition, International Workshop on Visual Scene Understanding, Miami, FL.

[22] K. Sharma, A. Kumar and S. Bhandarkar. (2016). Guessing objects in context. In ACM SIGGRAPH 2016 Posters (p. 83). ACM.

[23] K. Simonyan and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *Proc. Intl. Conf. Learn. Rep. (ICLR 2014)*.

[24] J. Sivic, B.C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros (2008). Unsupervised discovery of visual object class hierarchies. In Computer Vision and Pattern Recognition, 2008.

[25] R. Socher, et. al. 2013. Zero-shot learning through cross-modal transfer. *In Proc. NIPS*, (pp. 935-943).

[26] J. Sun, and D. W. Jacobs. (2017). Seeing What Is Not There: Learning Context to Determine Where Objects Are Missing.arXiv preprint arXiv:1702.07971.

[27] A. Torralba, The context challenge. http://web.mit.edu/torralba/www/carsAndFacesInContext

[28] L. Van Der Maaten, and G. Hinton. 2008. Visualizing data using *t-SNE*. *In JMLR*, 9 (2579-2605), 85.

[29] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba (2010). SUN Database: Large-scale Scene Recognition from Abbey to Zoo. Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[30] Y. Zhang, M. Bai, P. Kohli, S. Izadi, and J. Xiao. (2016). DeepContext: context-encoding neural pathways for 3D holistic scene understanding.arXiv preprint arXiv:1603.04922.

Chapter 5

A Study of Similarity using Visual and Textual Features

5.1 Chapter Outline

Marketing companies and managers are under constant pressure to introduce new products and relaunch existing versions, as this helps them to remain competitive and ensure that their existing customers remain loyal. When companies relaunch their brand, it would be useful to know how different a new version and a old version are. This is typically done before allocating resources to research and development, design, and manufacturing. Most importantly, similarity/dissimilarity across products must be determined in the most objective manner. For this purpose, automated techniques should be utilized, ideally incorporating consumer perceptions as an input variable. In this work, visual deep learning and textual

features are examined to establish their correlation with human perception related to consumer products. Our findings reveal that visual deep learning features are highly correlated with human perceptions, whereas color and textual features are only moderately correlated with human perceptions. In the visual domain, even though deep learning features turned out to be clearly superior, it is difficult to ascertain which aspects of the product led to their superiority. The correlations for color were only moderate, suggesting that they do not influence human perceptions in a stand-alone fashion, yet do seem to have a marked effect on human perceptions of similarity. In the textual domain, both TF-IDF (Term Frequency - Inverse Document Frequency) and word2vec features had a moderate impact on humans' perceptions of product similarity.

5.2 Introduction

In the marketing arena, managers and companies, for a variety of reasons, are under constant pressure to introduce new products and upgrade their products. New products need to be introduced for a variety of reasons [14, 22] - such as to deter other competitors from covering different niches, to target a different segment of customers based on different preferences, and not to make consumers get bored with existing products. For a successful product launch, companies need to effectively decide how to differentiate new products from their existing products and evaluate their similarity to existing products. They need to make this determination either before allocating resources to design/manufacture/research

products or at a stage later after the product has already been designed. In either case, companies need to determine similarity/dissimilarity or more loosely, the level of innovation of the new products compared to their own existing products. Hence, establishing the extent of product similarity or dissimilarity between the old and new versions in an objective manner would be very useful. In this, we test automated techniques that measure similarity/dissimilarity between within-brand products and examine how different they are compared to the consumers' real perception. Hence, we investigate visual deep learning and textual features and their correlation with human perception for product similarity for the smartphone and shampoo categories.

In the recent computer vision literature, various ideas that have direct relevance to the field of marketing literature have been extensively explored, such as image popularity [4] [12] [28], image virality [5], and image interestingness [8]. Although interestingness, popularity, virality are important for marketing, these studies have been conducted on standalone images, while neglecting the relative similarity or dissimilarity of different products. In the real world, consumers often compare new products with the existing ones while making purchasing decisions. Thus, it is important to examine the correlation between relative product similarity, as judged by human subjects, with that judged by visual and textual features. The findings in this study could have potential implications for business as in the area of product innovation.

In attempting to meet the aforementioned goal, a dataset including the image and text information for within-brand smartphone and shampoo categories was

collected. In the next step, for each consumer product category, various similarity matrices pertaining to several feature descriptors were constructed, based on visual deep learning, shape, color, TFIDF, and word2vec respectively. Next, we conducted a consumer survey to measure consumer perceptions of product similarity and compute the correlations between consumer perception and visual and textual features. The aim of these investigations was to address the following research questions:

1. How is consumers' perception of product similarity correlated with visual deep learning features?

2. How is consumers' perception of product similarity correlated with shape features?

3. How is consumers' perception of product similarity correlated with color features?

4. How is consumers' perception of product similarity correlated with textual features?

5. What are the relative effects of visual and textual features on human perception of similarity?

6. How do the correlations differ for durable and non-durable products? In the context of this paper, durable products such as smartphones imply products that are intended to be used by consumers for longer periods of time, while nondurable products such as shampoo are intended to be used by consumers for shorter periods of time.

7. For textual features, such as word2vec [18, 19], what is the effect of context

window and feature size embeddings on the correlations between deep learning features and consumer perception?

5.3 Literature Review

5.3.1 Computer Vision

In the computer vision literature, researchers have explored various ideas that have a direct relevance to the marketing literature, such as image popularity [4], [12], [28], image virality [5], and image interestingness [8]. Khosla et al.'s [12] seminal work is particularly relevant in this context, as the authors demonstrated that the popularity of images is correlated with low-level image content features, such as color, texture, gradient, and deep learning features. In addition, their findings revealed that high-level features, such as objects and social cues, are correlated with image popularity. In other words, both low-level and high-level information can serve as image popularity predictors. In a more recent study conducted in an e-commerce environment, Zawkersky et al. [28] examined the effect of image quality features (simplicity, spatial edge distributions, hue, contrast, blur and rule of thirds) on image popularity, and found that it was positively correlated with all analyzed variables. In particular, their results suggest that products in which textual and visual features are used in conjunction tend to be perceived to be of higher quality.

Another related phenomenon from the marketing perspective is image virality [5]. In their work, the authors aimed to ascertain the effect of different features

on image virality. Unlike the authors of the popularity studies discussed above, they reported that low-level features - such as HOG (Histogram of Gradients) - have minimal effect on virality. Owing to the absence of correlation, the authors argued that virality could not be predicted without considering high-level information. Specifically, they posited that animal, synthetically generated, (not) beautiful, explicit and sexual are the features that most accurately predict virality.

In a related work, Gygli et al. [8] explored interestingness of image. Their findings revealed that, irrespective of their personal preferences, most individuals tend to agree on what makes images interesting. Thus, the authors developed specific computational measures of interestingness, based on which unusualness and aesthetics were identified as reliable predictors of interestingness. In particular, unusualness was highly correlated with interestingness. In this context, unusualness is defined as the test image being outlier in feature space or patches inside image are statistically different from patches of similar images. Similarly, image aesthetics was defined by color, contrast, complexity and edge distribution. Among these features, unusualness was found to exhibit the strongest effect on interestingness. In a more recent investigation, Lu et al. [16] used a deep learning-based framework for rating image aesthetics using both global and local information in an image, and found that these features were predictive of aesthetics. Even though interestingness, popularity and virality are important for marketing, the aforementioned studies were conducted using standalone images, thus failing to examine the relative value of different products for intra-product similarity. This gap in extant research is addressed in the present study, as consumers typically compare new

products with old products when making evaluations and purchasing decisions. Thus, the aim of this investigation was to ascertain how relative product similarity, as judged by human subjects, correlates with visual and textual features. The findings yielded by this analysis could have potential applications for marketing, especially in the domain of product innovation.

5.3.2 Text Mining

There is a wide variety of text mining techniques that could be effectively adopted in marketing applications. For example, Latent Dirichlet Allocation could be used to find themes in user discussions, while clustering analysis could be employed to group various users/products based on relevant criteria, and supervised classification of textual data is particularly interesting, as it could have many potential applications, such as sentiment analysis [24], analysis of product attribute effects on sales [1], hotel demand estimation [7], and market structure assessment [21]. In contrast to these works, in the present study, text mining is employed to predict the correlation between the users' judgment of similarity between products and automated measures of text similarity obtained using word embeddings.

5.4 Algorithms/Techniques

The techniques adopted in this research are briefly described below.

5.4.1 Visual Deep Learning

Deep learning has recently grown in prominence and is presently used in a wide range of computer vision applications. Deep learning systems are implemented as a hierarchical neural network with multiple layers of neurons. These systems have existed since the 1980s and are inspired by the structure and functionality of the human brain. However, owing to the rapid technological advances, they have only recently grown in popularity. These systems use convolution filters as neurons, followed by max pooling layers. The idea behind this approach is that convolution filters, when combined with max pooling, can extract features from images that are invariant to size and location, while remaining sufficient discriminatory information for the purpose of object recognition. As an analogy, in deep learning systems such as convolutional neural networks (CNNs), the lower layers resemble low-level generic information, such as lines and edges, the middle layers resemble middle-level concepts such as dog-head, cat-legs, etc., and the higher layers represent high-level objects, such as dog, cat, person, etc. Various deep learning architectures have been proposed recently, such as Alexnet [13], Googlenet [25], VGG [23], Resnet [9], and Densenet [10]. These architectures comprise of repeatable layers of building blocks such as convolution, pooling, RELU, dropout, and other layers. The convolution layers tend to capture discriminatory information in an image, whereas pooling layers serve a dual purpose of dimensionality reduction and building spatial invariance. The RELU layer is used to expedite the training process, whereas the dropout layer prevents data overfitting. A brief description of the aforementioned layers and their functionality is given below.

Convolution layer

The convolution layer consists of filters that slide over the image. The weights in the filter are multiplied by the pixel values in an elementwise manner to produce an output. Such outputs across the image result in a feature map, which is then used in subsequent stages. Unlike traditional filters, the parameters or weights characterizing this filter are learned from the input data in an unsupervised manner. Convolution layers are capable of capturing a variety of discriminative information from the input images, such as edges and color at low levels, object part-based information at middle levels, and more conceptual information is captured at high levels in the multilayer convolution architecture. Generally, at every layer, multiple convolutional filters are applied to an image, thus ensuring that different types of information are extracted at each stage.

Pooling layers

Since convolution layers produce high-dimensional feature maps, these are usually downsampled using pooling layers. In addition to reducing dimensionality by downsampling, pooling layers are also known to help in scale invariance. max, average, and sum, etc., are examples of downsampling pooling functions, with max pooling being one of the most common forms of pooling.

Rectified linear units (RELU)

RELU layers were introduced to remove disadvantages associated with sigmoid/tanh layers. The RELU function is given by f(x) = max(0,x), and is known to yield

a computational advantage of faster training, since it speeds up the convergence of the stochastic gradient descent algorithm. In addition, the computations do not involve any expensive operations, such as addition, subtraction, multiplication, etc. Moreover, while sigmoid/tanh layers are prone to the vanishing gradient problem, this issue does not affect RELU layer, since the RELU layer does not squash the positive values. However, it cannot backpropagate error signals if the node output is a negative value, since all negative values are converted to 0. To address this shortcoming, several workarounds such as Leaky RELU [17] have been introduced.

Dropout

Dropout is a regularizer adopted to prevent overfitting. During training, if the training data contains noise, the neurons will tend to learn the noise together (since neurons are co-dependent), and will thus overfit. To mitigate this issue, certain percentage (dropout rate) of neurons is sterilized (not participating in training for that particular iteration) during training, thereby rendering them immune to noise (which is thus not learned). In each training iteration, different neurons are sterilized, ensuring that they do not learn similar noise patterns.

Training

The network learns its parameters by optimizing the objective function employed to calculate the error between the predicted output and the target output. The network architecture goal is to select the weights that would ensure that this

objective function is minimized. Stochastic gradient descent with backpropagation is the standard technique employed for this purpose.

Pretrained deep neural network models

It has been shown by several researchers [6] that pretrained deep neural network models could yield competitive results when applied to real-world images. For instance, if the model is trained on ImageNet, which has 1000 object categories, and features are extracted from this pretrained model, the features tend to be generalizable to many real-world categories. These superior results are possible because lower layers in the CNN architecture tend to learn edges and corners, which is domain-generalizable information, whereas the middle CNN layers tend to learn part-based information, which is also generalizable. For example, dog paws and cat paws are similar. However, even for the top CNN layers, at least for real-world entities, the computed features are transferrable because intuitively (and implicitly), real-world entities tend to share attributes such as shape, color, pose, texture, etc.

The major advantage of using pretrained models is that they eliminate the computational expense of training deep learning models anew, and also the need to generate large training datasets required for training these models. Hence, effective features can be extracted with a substantially reduced number of training images and far fewer computational resources. In general, the features extracted at different depth level of the CNN will yield different outcomes. Extracted features from higher CNN layers are useful when the novel categories bear some vi-

Figure 5.1: An example of image survey question for Shampoo category.

sual similarity to categories used for pretraining the model. For instance, a CNN trained on tables could reliably extract features for a chair or a bed, but will not successfully generalize features for a medical tumor. However, extracted features from lower CNN layers, while less informative, will likely allow for generalization among dissimilar categories, such as table to tumor used in the example above. This generalization is possible because lower CNN layers tend to capture low-level information, such as edges and corners, which may be transferable even to seemingly unrelated categories. Nevertheless, the features extracted from high-level CNN layers models of ImageNet would be expected to most real-world categories, but will not generalize to categories in medical and aerial images.

AlexNet [13] was one of the first revolutionary CNN architectures in deep learning to achieve remarkable results on ImageNet. This convolutional neural network (CNN) consists of a convolution layer, followed by pooling layers, and fully con-

nected layers. It has 11×11 filters in the lowest layer, resulting in a significant number of parameters. Owing to the success of AlexNet, various other enhancements to CNN architectures were subsequently proposed, including VGG, since VGG has smaller filters compared to AlexNet, the number of training parameters is reduced. A further benefit of this is architecture stems from capturing more non-linearities, and thus more information. In addition, VGG is a much deeper architecture, resulting in more effective capture of non-linearities, again leading to better performance. However, in spite of impressive results, VGG is computationally expensive, both temporally and spatially. To mitigate these limitations, Szegedy et al. [25] introduced Googlenet, which has less computational overhead and yields enhanced performance. The key change in the GoogleNet architecture stems from the use of an inception module, which consists of differently sized filters (1×1 vs 3×3 vs 5×5) in a single layer. The results from these filters and the pooling layer are concatenated before proceeding to the next layer. The inception layer further benefits from differently sized filters, as these can capture different information, resulting in a richer representation. In fact, one of the motivations behind the development of GoogleNet was to take advantage of local correlations between pixels, which is effectively achieved via the inception module. Moreover, by placing 1×1 convolutions before 3×3 and 5×5, the computational cost is significantly reduced.

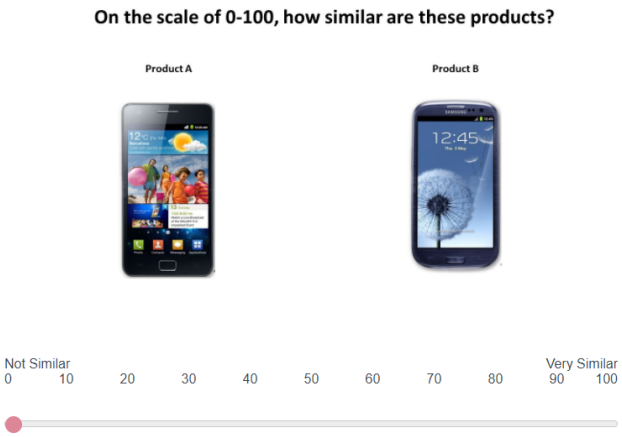

Figure 5.2: An example of image survey question for Smartphone category.

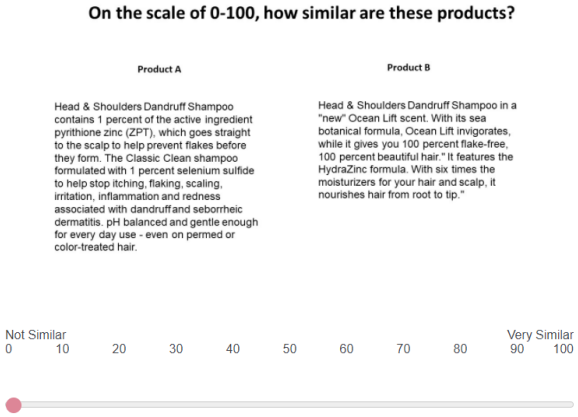

Figure 5.3: An example of textual survey question for Shampoo category.

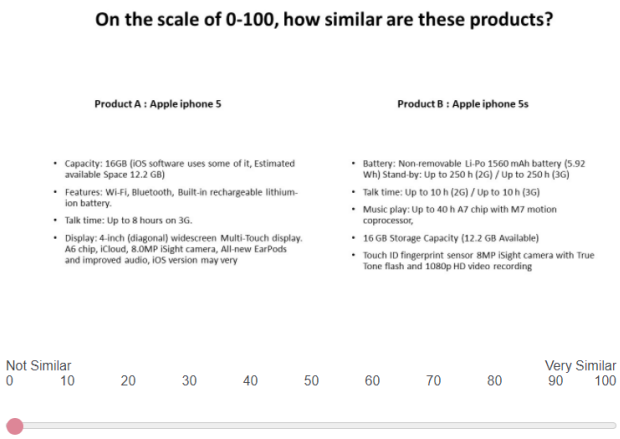

Figure 5.4: An example of textual survey question for Smartphone category.

5.4.2 Textual Features

To acquire textual features, we chose TF-IDF and word2vec features [18, 19]. The TF-IDF features convert a product's descriptive text to a vector feature space, in which each word in the entire corpus represents a feature, and a weight is assigned to a particular word in a product description based on two criteria: term frequency, which describes how often a term occurs in a product description, and inverse document frequency, which describes how infrequent the term is in other documents. Although TF-IDF effectively captures the similarity/dissimilarity between descriptions, it does not consider the semantic similarity between words. For instance, the words pretty and beautiful are treated differently. To overcome this shortcoming, we use word2vec, which tends to place semantically similar words in closer proximity in a hypothetical space. Word2vec was explained in detail in Chapters 3 and 4. For the purposes of this paper, we choose word2vec as represen-

tative of a family of word embedding models, such as GloVe. The word embedding models, even though they differ in their specific implementations, tend to behave similarly. Hence, we believe that our results are generalizable and representative of all word embedding families.

5.5 Hypotheses

Hypothesis 1: Visual deep learning features are correlated with human perception of similarity between products.

This hypothesis was tested by conducting an experiment using images for the shampoo and smartphone categories. We believe that these two categories are sufficiently representative of a wide range of products: smartphones are representative of durable products and shampoos are representative of nondurable products. In addition, we conducted a pretest in which consumers were asked to rate 4 durable and 4 nondurable categories in which they used textual and visual information. The results showed that shampoos and smartphones are the most appropriate representative products. After selecting these two categories, we conducted a survey for these two categories only. In the survey, we asked the study participants to rate product similarity. Next, similarity matrices for each of these categories were constructed by computing cosine similarity between each pair of product for each deep learning feature (visual and textual). Finally, we performed a correlation analysis between human subject ratings and visual deep learning features. An example of survey questions is shown in Figures 5.1 and 5.2.

Hypothesis 2: Color features are correlated with human perception of similarity between products.

This hypothesis was tested in a similar manner to the previous hypothesis. The similarity between the images was determined by comparing the color naming descriptors [26], discriminative color descriptors [11], and color histogram. In color naming descriptors [26], the authors use 11 dimensional descriptor where each dimension corresponds to a color label in the real world. The descriptor is computed by measuring probability of occurrence of each of those 11 colors in a region of interest. In discriminative color descriptors [11], the authors take an information-theoretic approach, where descriptors are learned for their discriminating ability on a particular dataset. The authors make their descriptor generalizable by training it on multiple datasets. As a third descriptor, we also compare color histogram similarity of images in the RGB space where pixels take value in the range 0-255.

The human similarity ratings were compared to the image-based color similarity measures, as computed by above descriptors. We expect that the color alone are not highly correlated with the actual perception of consumers compared to visual deep learning feature.

Hypothesis 3: Shape features are correlated with human perception of similarity between products.

The similarity between the images was determined by matching shapes using opencv [3], and shape context [2]. First shape measure is based on Hu-moments, and second on shape context. The Hu-moment descriptors use Hu-moments which

are invariant to translation, rotation and scale. The shape context descriptor [2] is computed by sampling points of the contour of an object where each point's relative position to every other point on contour is encoded in the descriptor. For the purpose of testing this hypothesis, we blacken each of the smartphone and shampoo images to convert them to a binary image, while preserving their outer borders and edges. For smartphone images, their buttons would be preserved. For shampoo images, the outer shapes of the container would be preserved. Again, we believe that people are not solely influenced by holistic shape in their determination of similarity. We expect the shape to play a significant but not all-encompassing role in determining similarity that correlates well with human perception.

Hypothesis 4: Language features (in the form of word2vec and TFIDF) are correlated with human perception of similarity between products.

This hypothesis was tested by conducting an experiment using the same sets of products as those employed in the preceding experiment. In this case, the participants were given product descriptions, and were asked to rate them on the scale from 0 to 100. An example of survey questions is shown in Figures 5.3 and 5.4.

Hypothesis 5: The correlation of visual features is higher than that of textual features.

Another interesting phenomenon we plan to study is the relative effects of visual and textual features. The question, whether visual or textual features affect consumers' perception to a greater degree is of paramount importance for managers.

Although we expect both types of features to play an important and complimentary roles, we believe that a quantification of the relative effects of visual and textual features would be necessary in certain kinds of situations.

Hypothesis 6: Nondurable products correlate more highly with human perceptions of product similarity than do durable products.

We expected the nondurable products to have higher correlations between human perceptions of similarity and visual/textual features. Nondurable products are introduced at a much higher frequency than are durable products, and manufacturers tend to differentiate nondurable products using a wide variety of criteria. Novel durable products tend to be introduced at a slower rate and tend to have more stable features that the manufacturers can use to differentiate between products. In other words, there are more variations in product similarity among nondurable products than among durable products because consumers use more information to make brand choice decisions for nondurable products. In other words, features extracted from images or text of nondurable products should have more variance than features extracted from durable products. Due to the high variance in the feature space for nondurable products, owing to the high variety in image and text, humans will have an easier time assessing the similarity between these products. In other words, we expect that visual deep learning and textual features should work better for categories in which consumers use *less* objective product information and they should vary more across consumers. For instance, the shapes of durable products such as smartphones do not differ significantly. Additionally, colors and

textures inside the phones are not significantly different between products. Hence, the distances between the deep learning features of smartphones will tend to be comparatively insignificant, resulting in reduced correlations with human perceptions. On the other hand, the shapes, color, and other information for shampoo products will differ significantly, and the distances between the deep learning features will result in high correlations with human perceptions. The same argument holds in the textual domain, where shampoos are described using a much wider variety of words, which should result in significant distances in textual feature space that should correlate well with human perceptions.

5.6 Experimental Methodology

Datasets: For the smartphone experiments, we collected the data from the consumer website Amazon.com and located the web page corresponding to a particular product. From this page, we extracted the smartphone image and the text featuring the product description. Using the smart phone images and product descriptions, we created the survey questions similar to ones shown in Figures 5.2 and 5.4. For the shampoo experiments, we collected data from product launch analytics [20], and created survey questions using that dataset.

Survey Design: To test our six hypotheses, two product surveys were created, pertaining to images and text, as explained in the previous section. Each survey question was related to two products that are potential competitors within the same brand. For the image-based survey, participants were asked to compare the

images of two products and rate their similarity on a 0-100 scale. For the textual survey, participants were required to compare the textual descriptions of two products and rate their similarity on a 0-100 scale. The surveys were created in Qualtrics and were run on Mturk Prime. Mturk Prime is Amazon's crowdsourcing platform, and it can be used for a variety of purposes. It is frequently used by researchers in various fields to conduct survey-based research. In our case, each user was shown a set of questions asking them to rate the similarity between products. For shampoos, 9 questions each were shown for the textual and image categories, and for smartphones, 9 questions randomly chosen from a set of 16 questions were shown to users. The questions were randomized to have an even distribution. Overall, 229 users participated in the survey for smartphone, and 144 for shampoo. Once the survey was completed, the user responses to each question were averaged and the averages were used in the final correlational analysis. Finally, we calculated Spearman correlations of averaged survey responses for each of the product categories. The Spearman correlation was chosen because by the virtue of being rank correlation, it is robust to outliers. For the correlational analysis, we compared image survey responses with visual product similarity using visual deep learning features and other image features (color and shape), while textual survey responses were compared with textual product similarity using textual features.

For deep learning, the similarity between products based on their images was computed using visual deep learning features extracted from VGG architecture [23], which was pre-trained on ImageNet. The features were extracted using Matconvnet package [27] and similarity was computed by using cosine similarity.

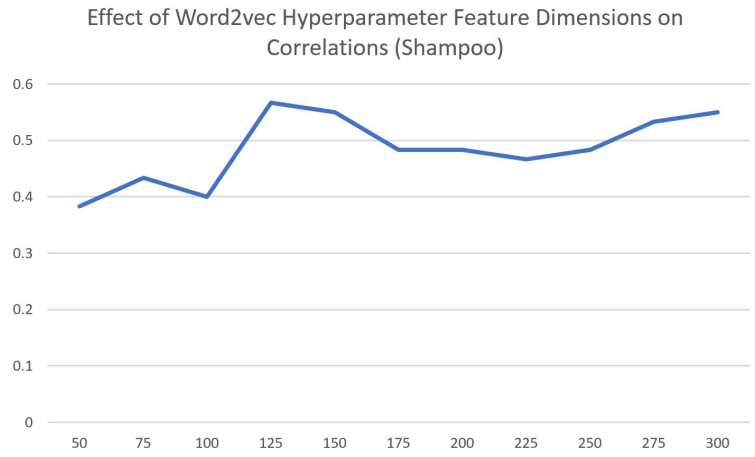

Figure 5.5: Effect of word2vec Hyperparameter Feature Dimension on Correlations (Shampoo) when Context Window Size is constant at 9.

We also measured similarity based on color and shape features. For color we used color histogram, color naming descriptors [26], and discriminative color descriptors [11]. For shape, we used hu-moments from opencv [3], and shape context [2]. In addition, we also computed similarity using SIFT features for comparison purposes.

For textual features, the similarity was computed using TF-IDF and word2vec features. For word2vec features, the vector of each word in first product's textual description was compared to each word in the second product's textual description (using cosine similarity), and an average score was calculated.

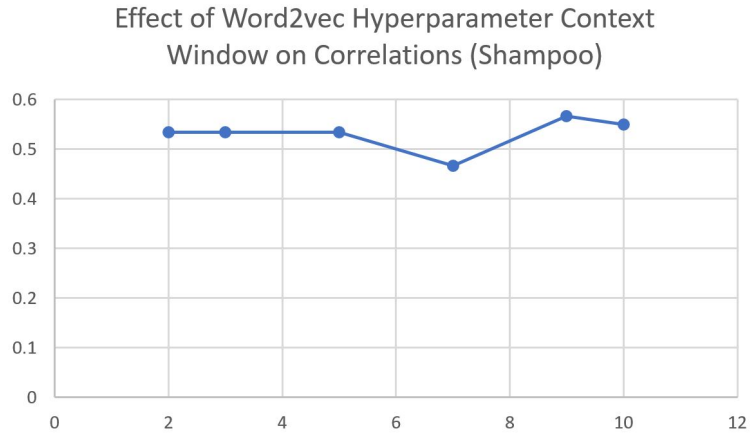

Figure 5.6: Effect of word2vec Hyperparameter Context Window on Correlations for Shampoo when Feature Dimension Size is constant at 125.

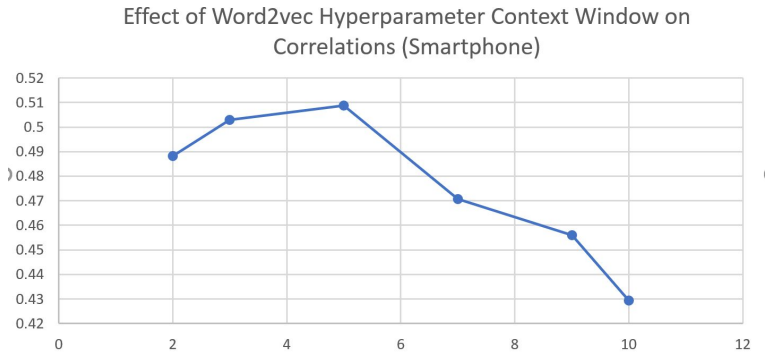

Figure 5.7: Effect of word2vec Hyperparameter Context Window on Correlations for Smartphones when Feature Dimension Size is constant at 150.

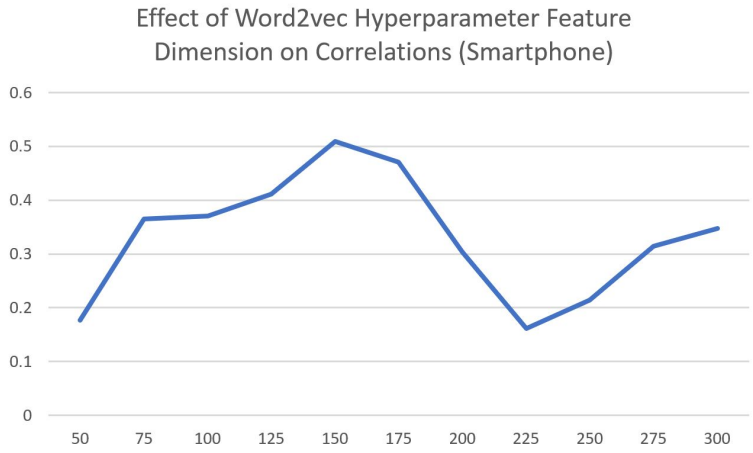

Figure 5.8: Effect of word2vec Hyperparameter Feature Dimensions on Correlations for Smartphones when Context Window Size is constant at 5.

5.7 Results and Discussion

1. As shown in table 5.1 and 5.2, our findings revealed that the images and textual features were correlated with consumers' perception of within-brand products. The companies can use this information to objectively determine the similarity/dissimilarity of their upcoming products or the degree of innovativeness in new products. Such objective measures could help companies to evaluate the difference between the old and new versions before launching a new version and develop more innovative product. In addition, managers can utilize the similarity in deep learning features for their product portfolio diversification and R&D decisions, which will be important to avoid any cannibalization among the products within a company.

2. Overall, among image features, deep learning features was clear winner,

Table 5.1: Spearman Correlations for Shampoo Data

$Image - DeepLearning$	0.9
$Image - ColorDiscriminative$	0.48
$Image - ColorNaming$	0.35
$Image - ColorHistogram$	0.1
$Image - ShapeHu$	-0.28
$Image - ShapeContext$	0.51
$Image - Sift$	0.25
$Text - TFIDF$	0.48
$Text - Word2vec$	0.56

suggesting that deep learning features capture richer information. However, even color in isolation, had a non-significant effect on correlations. This is expected because consumers could be influenced by plethora of information in an image (some subtle and some not so subtle). Deep learning features would be capable of richly capturing those features whereas color capture only one aspect of perception. Regarding shape features, it turned out that the correlations for smartphone were negative, and for shampoo, the correlation for only shape context was positive at 0.5. Hence, we conclude that hand engineered shape features are not correlated with human perception. This could be attributed to the fact that capturing shape in descriptors at fine grained level is much harder problem, especially when done with old school features. On the contrary, we believe that deep learning features can implicitly capture shape information too, thus making them more robust and reliable for practical applications.

3. Our analysis revealed that other than deep learning features, textual fea-

Table 5.2: Spearman Correlations for Smartphone Data

$Image - DeepLearning$	0.81
$Image - ColorDiscriminative$	0.42
$Image - ColorNaming$	0.2
$Image - ColorHistogram$	-0.44
$Image - ShapeHu$	-0.82
$Image - ShapeContext$	-0.35
$Image - Sift$	0.06
$Text - TFIDF$	0.56
$Text - Word2vec$	0.5

tures had higher effect on consumers' perception of similarity. The textual features studied in this paper were still inferior to visual deep learning features. However, compared to color and shape in isolation, they have higher correlation with consumers' perception of similarity. Textual features capture complimentary information to visual features. Managers and marketers could use both textual and visual features for determining product similarity for different purposes.

4. The analyses also revealed that correlations for shampoo products within a brand (representative of non-durable product category) were higher than the correlations pertaining to the smartphone category. This difference could be attributed to the fact that, the for durable products such as smartphone, more objective attributes play a role in consumers' perception of similarity, whereas they tend to rely more on subjective attributes when assessing non-durable products such as shampoo. Since smartphones describe their product attributes based on the objective (numeric) attributes such as screen size and memory size, there is not

enough variation in product description or image of smartphones. However, we see more variety in product description of shampoo products compared to that of smartphones. The deep learning features and textual features seem to work better when there is more variation in the visual and textual data.

5. We also examined whether the hyperparameters of word2vec affect the correlations. Word2vec and similar algorithms in this category were recently successfully employed in variety of NLP applications. However, to ensure that word2vec is effective in determining similarity between products, various hyperparameters need to be considered. Word2vec is very sensitive to the choice of parameters, such as context window and feature dimensions. The context window signifies at what level of granularity could we capture the co-occurrences for effective embeddings. Hence, we conducted our study on textual data pertaining to smartphones and shampoos. We experimented with feature dimension sizes (50, 75, 100, 125, 150, 175, 200, 225, 250, 275, and 300), and context window sizes (2, 3, 5, 7, 9, and 10). For shampoo, the highest correlation was obtained with the context window of 9 and feature dimension size of 125. For smartphone, the highest correlation was obtained with the context window of 5 and feature dimension size of 150. As shown in graphs, the correlation is the greatest when the context window size of 5 is chosen for smartphone and 9 for shampoo. This finding suggests that, appropriate context window is specific to dataset. Similarly, the feature dimension of size 125 -150 produced the highest correlation for shampoo, and 150 - 175 for smartphone. Prior empirical evidence indicates that feature dimension size in the 200-500 range tends to produce the most optimal results. Thus, the results

presented above suggest that word2vec may require some experimentation to find the optimal parameter values. Specifically, for small datasets, context window may need to be determined experimentally and for feature dimensions, low-to-moderate dimensional vector sizes seem to be the best choice. For larger dataset sizes and more product categories, the word2vec hyperparameters such as context window and feature dimension sizes would likely require even more tuning.

5.8 Conclusion

The analyses conducted as a part of this work indicate that, for consumer products, visual and textual features are correlated with human perception of product similarity and could indeed be used for evaluating product similarity/dissimilarity without expensive and time-consuming human-subject research. Companies need to remain innovative to maintain and/or increase their market share. As shown here, by using visual and textual features before innovating and manufacturing a product, companies could bring some objectivity to the process of determining similarity/dissimilarity and save some R&D costs. In addition, the findings reported here have shown that different factors play a role in human assessment of durable and non-durable products, thus requiring particular attention when designing similarity/dissimilarity measures. Finally, when determining the word2vec hyperparameters, it is important to consider the specific dataset, as this will determine the context window size and number of feature dimensions that would yield optimal correlation values across products.

Bibliography

[1] Archak, N., Ghose A., & Ipeirotis P.G. (2011) Deriving the pricing power of product features by mining consumer reviews. Management Sci. 57(8).

[2] Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence, 24(4), 509-522.

[3] Bradski, G., & Kaehler, A. (2000). OpenCV. Dr. Dobb's journal of software tools, 3.

[4] Cappallo, S., Mensink, T., & Snoek, C. G. (2015, June). Latent factors of visual popularity prediction. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (pp. 195-202). ACM.

[5] Deza, A., & Parikh, D. (2015). Understanding image virality. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1818-1826).

[6] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature

for generic visual recognition. In International conference on machine learning (pp. 647-655).

[7] Ghose A., Ipeirotis P. G. (2011) Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics.IEEE Trans. Knowledge Data Engrg. 23(10).

[8] Gygli, M., Grabner, H., Riemenschneider, H., Nater, F., & Van Gool, L. (2013). The interestingness of images. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1633-1640).

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[10] Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2016). Densely connected convolutional networks. arXiv preprint arXiv:1608.06993.

[11] Khan, R., Van de Weijer, J., Shahbaz Khan, F., Muselet, D., Ducottet, C., & Barat, C. (2013). Discriminative color descriptors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2866-2873).

[12] Khosla, A., Das Sarma, A., & Hamid, R. (2014, April). What makes an image popular? In Proceedings of the 23rd international conference on World wide web (pp. 867-876). ACM.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[14] Kwaku Atuahene-Gima (2005) Resolving the Capability—Rigidity Paradox in New Product Innovation. Journal of Marketing: October 2005, Vol. 69, No. 4, pp. 61-83.

[15] Lee, C. H., & Schluter, G. (2002). Why do food manufacturers introduce new products? Journal of Food Distribution Research, 33(1), 102-111.

[16] Lu, X., Lin, Z., Jin, H., Yang, J., & Wang, J. Z. (2015). Rating Image Aesthetics Using Deep Learning. IEEE Transactions on Multimedia, 17(11), 2021-2034.

[17] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML (Vol. 30, No. 1, p. 3).

[18] Mikolov T. et al. 2013. Distributed representations of words and phrases and their compositionality. *In Proc. NIPS*, (pp. 3111-3119).

[19] Mikolov. T. 2013. Efficient estimation of word representations in vector space. *In Proc. ICLR.*

[20] Marketing Intelligence Service Ltd, & Datamonitor (Firm). (n.d.). Product launch analytics. Retrieved August 1, 2018, from `https://www.galileo.usg.edu/scholar/uga/databases/pru9-uga1/`.

[21] Netzer, O., Feldman, R., Goldenberg, J., & Fresko, M. (2012). Mine your own business: Marketstructure surveillance through text mining. Marketing Science, 31(3), 521-543.

[22] Rust, R. T., Zeithaml, V. A., & Lemon, K. N. (2004). Customer-centered brand management. Harvard business review, 82(9),

[23] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

[24] Sonnier GP, McAlister L, Rutz OJ (2011) A dynamic model of the effect of online communications on firm sales. Marketing Sci. 30(4):702âĂŞ716.

[25] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[26] Van De Weijer, J., Schmid, C., Verbeek, J., & Larlus, D. (2009). Learning color names for real-world applications. IEEE Transactions on Image Processing, 18(7), 1512-1523.

[27] Vedaldi A. & Lenc K. (2015). MatConvNet-convolutional neural networks for MATLAB. *Proc. ACM Conf. Multimedia Systems (MMSys 2015).*

[28] Zakrewsky, S., Aryafar, K., & Shokoufandeh, A. (2016). Item Popularity Prediction in E-commerce Using Image Quality Feature Vectors. arXiv preprint arXiv:1605.03663.

Chapter 6

Conclusion and Future Work

If we are to create human-like AI, vision and language will both need to blend appropriately to achieve this formidable goal. However, while we pursue such a goal, the applications at the intersection of computer vision and natural language processing will tend to appear as a by-product. In this dissertation, we addressed projects that blend computer vision and natural language processing to achieve real world goals. Below is the summary of what we have achieved in various chapters:

In chapter two, we addressed the issue of speeding up of image caption retrieval, where we use top objects detected in an image to prune the candidate image/caption pairs in the training set for further processing. In the next step, the k-nearest neighbor is used to retrieve nearest neighbor images in the visual deep learning feature space. These retrieved images have captions associated with them. Using this final set of captions, we find consensus sentence, which is assigned

to a test image.

In chapter three, we addressed action classification in images aided by natural language processing. In the first part of this chapter, we inferred action from top-objects in an image, where the action was inferred using word embeddings trained on natural language corpus. In the second part of the chapter, we addressed the action classification in a fast and frugal manner, using the minimum information to infer action from the most probable object classified in an image, using natural language word embeddings trained on Wikipedia.

In the fourth chapter, we addressed the problem of guessing objects in an image, where the datasets for training classifiers for those particular objects are not available. In these situations, we train classifiers for few categories whose training sets are available, and from these classifications, using word embeddings trained on Wikipedia, we guess what other objects could be present in an image.

In the fifth chapter, we studied the visual and textual features in the domain of consumer products. Managers and marketers need to determine how their new products differ from existing products in the manner that is correlated with consumers' perception. Before assigning resources to manufacturing and R & D, currently, managers subjectively determine the visual and textual similarity of their products. However, it would be convenient if such a similarity could be automated. In the pursuit of this goal, we studied the variety of visual and textual features such as visual deep learning, color, shape, TF-IDF and word2vec and how they correlate with human perception.

6.0.1 Future Work

In guessing objects in images for situations where datasets for training classifiers are not available, we could combine explicit real world knowledge in the form of ontologies or logical rules along with word embeddings to improve the classification results. Some work in this direction already exists [1], however, all the existing work has the majority of seen classes and very few unseen classes. In our approach, we are addressing a different problem where majority of classes are unseen. In such a situation, it needs to be seen how blending explicit real world knowledge and word embeddings learned from unstructured text could improve performance.

In the action classification domain, state-of-the-art visual action classifiers could be blended with word embeddings to improve performance. In the Fast and Frugal heuristic domain, such a model could be implemented in robots that operate in highly uncertain environments, and could inspire human-like AI to infer actions and other information using minimum information/resources. Also, from an application perspective, such a model could inform actions in hopeless situations where absolutely no information is available.

In the marketing domain, a more comprehensive study involving many other consumer products could be conducted in the future. Other domains such as clothing, automobile, food, etc. could also benefit from correlational studies between visual and textual features with human perception. In addition, we could make several automated measures of innovation using visual and textual features.

Bibliography

[1] Wang, X., Ye, Y., & Gupta, A. (2018, March). Zero-shot Recognition via Semantic Embeddings and Knowledge Graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6857-6866).