UNIVERSITY
OF TRENTO - Italy

Doctoral School in Information and
Communication Technology

EXPLOITING TEXT CORPORA FOR DATA ENRICHMENT IN LANGUAGE AND VISION APPLICATIONS

Dieu-Thu Le

A thesis submitted for the degree of
Doctor of Philosophy

November 2014

ABSTRACT

During the last decade, machine learning techniques have been used successfully in many applications. The performance of these systems depends largely on the quality and quantity of the training data. For many tasks, the data itself is not rich enough. For example, text documents such as user-queries, users-comments and short advertisements consist of only few words. Therefore direct word-based representations are sparse which makes it difficult to measure good similarities for clustering or classification. In many other applications, training data is too expensive to fully obtain. In the task of human action recognition from still images, the total number of possible actions is the cartesian product of objects and verbs. This combinatorial explosion of verb-object relations makes the task of learning human actions directly from their visual appearance computationally prohibitive and makes the collection of proper-sized image datasets infeasible. This thesis proposes a framework to enrich poor data with knowledge automatically extracted from large-scale text corpora. It considers various text modeling techniques to extract knowledge. The data enrichment framework is illustrated in different tasks in both language and vision applications.

For language applications, we apply data enrichment to query classification. A topic model is estimated on external text corpora as a reference set. This model is then used to analyze topics for short queries and categories, generating shared context between them. The experimental results show that the data enrichment process increases the performance of the system, helping to find better categories for a given query.

For vision applications, we employ the knowledge extracted from large scale text corpora to predict objects in context and recognize human actions in images. We investigate the problem of modeling text corpora for knowledge extraction and discuss which model is the most suitable for each particular task. In the first task, we learn the relations between objects from text corpora to predict how different objects often occur together using a probability model. This knowledge is then used to help predict new objects given other objects in the images. In the human action recognition task, we combine the knowledge extracted from external text corpora with the visual features from the images. Based on the visually recognized objects, scenes and relative positions between the human and objects in these images, the most plausible actions are suggested using the knowledge learned from the general external text. This model allows recognizing unseen actions and even outperforms a visual Bag-of-Words model in a realistic scenario where only few visual training examples are available.

ACKNOWLEDGMENTS

I would like to thank, first of all, my advisor Raffaella Bernardi. She has been advising me from the very beginning of my research till now. She has given me a lot of instructions, time and support during the work of this dissertation.

This work would not have been possible without the help and instructions of my co-advisor Jasper Uijlings. His guidance helped me in all the time of research and writing of this thesis, especially in the part of the combination between language and vision.

I would like to thank Massimo Poesio for his support, comments and suggestions for my work in language applications. I would also like to thank Edwin Vald, Eduard Barbu for their contribution to the work of query classification. I would like to thank GALATEAS project for providing me with data, instructions for my research and financial support with travel grants. Thanks also to CLIC lab and members for the feedback during my work.

I would like to thank the university of Trento for the financial support granted through the completion of my PhD. A special thank to our secretary, Andrea Stenico, for helping me to prepare many documents for visa, conferences, etc.

I thank all my friends and family for their support and encouragement. Most of them are living far away in Vietnam, but they have always supported me whenever I needed it. Finally, I thank Michael for always being there cheering me up and standing by me.

CONTENTS

1 INTRODUCTION 11
1.1 Motivation . . . 11
1.2 Contributions . . . 12
1.2.1 Applications in language domain . . . 12
1.2.2 Applications in vision domain . . . 12
2 THE POOR DATA ENRICHMENT FRAMEWORK 13
2.1 Our general data enrichment framework . . . 13
2.1.1 Selecting text collection and databases . . . 14
2.1.2 Knowledge extraction . . . 15
2.1.3 Data enrichment and task application . . . 16
2.2 Language and vision applications . . . 16
2.2.1 Language application: Query classification . . . 16
2.2.2 Vision task applications . . . 17
2.3 Summary . . . 18
3 BACKGROUND 19
3.1 Knowledge acquisition from text data . . . 19
3.1.1 Commonsense knowledge database . . . 19
3.1.2 Semantic space and window-based model . . . 24
3.1.3 Distributional Memory . . . 25
3.1.4 Topic models and extensions . . . 27
3.2 Visual recognizers . . . 30
3.2.1 General Bag-of-Words framework . . . 31
3.2.2 Region extraction . . . 31
3.2.3 Vector quantization k-means . . . 32
3.2.4 Bag-of-words representation and classification . . . 32
3.3 Component integration . . . 33
3.3.1 Introduction . . . 33
3.3.2 Architecture . . . 33
4 APPLICATIONS IN LANGUAGE DOMAIN 37
4.1 Query classification . . . 37
4.1.1 Introduction . . . 37
4.1.2 Related work . . . 38
4.1.3 Our case study: query log of an art image archive . . . 38
4.2 Query enrichment . . . 40
4.3 Building a query classifier with topic models . . . 44
4.4 Experiments and results . . . 44
4.4.1 Datasets . . . 44
4.4.2 Evaluation metrics . . . 46
4.4.3 Experimental settings and results . . . 46
4.5 Chapter summary . . . 51
5 APPLICATIONS IN VISION DOMAIN 53
5.1 Introduction . . . 53
5.2 Related work . . . 54

5.2.1 Using language knowledge to aid visual recognition . . . 54
5.2.2 Human action recognition in images . . . 55
5.3 Our general framework for visual recognition . . . 57
5.3.1 Visual recognizers . . . 59
5.3.2 Distribution extraction from text collections and databases . . . 60
5.3.3 ConceptNet . . . 61
5.3.4 Window model . . . 62
5.3.5 Distributional Memory . . . 62
5.3.6 R-LDA . . . 62
5.4 Datasets . . . 64
5.4.1 SUN dataset . . . 64
5.4.2 89 action dataset . . . 64
5.4.3 TUHOI, the Trento Universal Human Object Interaction Dataset . . . 66
5.5 Object prediction . . . 71
5.5.1 Introduction . . . 71
5.5.2 The SUN data preprocessing for object prediction . . . 72
5.5.3 Experiments and discussion . . . 73
5.6 An integrated system: Action recognition . . . 74
5.6.1 Introduction . . . 74
5.6.2 Human action recognition framework . . . 75
5.6.3 Component integration . . . 76
5.6.4 Model adaptation . . . 77
5.6.5 Classifying human actions based on human-object positions . . . 79
5.6.6 Experiments and results . . . 80
5.7 Chapter summary . . . 90
6 CONCLUSIONS 91

BIBLIOGRAPHY 93

LIST OF FIGURES

Figure 1 A general framework for data enrichment based on text collection and prebuilt databases . . . 14
Figure 2 Applications in language domain: Query classification . . . 16
Figure 3 Applications in vision domain: human action recognition, object prediction 17
Figure 4 An excerpt from the WordNet concept hierarchy . . . 20
Figure 5 An example of a frame *Ride-Vehicle* in FrameNet: its definition, examples of frame elements that are essential to the meaning of the frame *Ride-Vehicle*, i.e. Core FEs (the elements that are essential to the meaning of a frame): Area (the location where the motion takes place, Goal (the endpoint of the trajectory of motion, Path (the trajectory of the motion), etc. . . . 21
Figure 6 A ConceptNet graph . . . 22
Figure 7 Main phases of building ConceptNet from OMCS . . . 23
Figure 8 ConceptNet 5 human evaluation [Speer and Havasi, 2012] . . . 23
Figure 9 A comparison among WordNet, Cyc and ConceptNet . . . 24
Figure 10 pLSA . . . 28
Figure 11 Generative graphical model of LDA (left) vs. ROOTH-LDA (right) . . . 28
Figure 12 Constructing the bag of words for image representation . . . 31
Figure 13 Scale-Invariant Local Features [Lowe, 1999] . . . 32
Figure 14 Designing a loss function: push down on the energy of the correct answer, pull up on the incorrect answers (Picture taken from [Lecun et al., 2006]) . 34
Figure 15 The two level taxonomy of the Bridgeman art library . . . 39
Figure 16 Query enrichment via topic modeling: our framework . . . 40
Figure 17 Enriching queries and categories: (1) Learning a TM from the text collection; (2) Enriching queries and categories with their topic labels . . . 41
Figure 18 Hidden topics derived from WaCKypedia . . . 43
Figure 19 Hidden topics derived from the Bridgeman catalogue . . . 43
Figure 20 Matching QR-CT and TM_{BAL} correct categories against the manual and automatic gold-standards . . . 48
Figure 21 Queries incorrectly classified . . . 49
Figure 22 Effects of TM on the classification task . . . 49
Figure 23 The impact of click-through information with matching-ranking (mr) and learning-based approach (svm) . . . 51
Figure 24 The impact of hidden topics with matching-ranking (mr) and learning-based approach (svm) . . . 51
Figure 25 Applications in vision domain: human action recognition, object prediction 58
Figure 26 List of relations in ConceptNet . . . 61
Figure 27 Images containing actions in the PASCAL VOC 2012 trainval dataset . . 65
Figure 28 The instruction of the crowdflower "Annotate human action in images" . . 69
Figure 29 An example of the interface for action annotation . . . 70

Figure 30 Examples of annotated images: **Left:** (1) play ping-pong, hold racket; (2) use laptop, hold computer mouse; (3) use microphone, play accordion, play guitar, play violin; (4) talk on microphone, sit on sofa, pour pitcher; (5) play trombone; (6) eat/suck popsicle; (7) listen/use/hear stethoscope; (8) ride bicycle, wear backpack; (9) swing/hold racket, hit tennis ball; **Right:** (1) sit on chair, play violin; (2) wear diaper, sit on chair, squeeze/apply cream; (3) sit on chair, play cello; (4) hold/shake maraca; (5) ride watercraft, wear swimming trunks; (6) cook/use stove, stir mushroom, hold spatula; (7) drive/row watercraft; (8) sit on chair, pet dog, lay on sofa; (9) click/type on computer keyboard 70
Figure 31 Many different ways to describe an action in an image 71
Figure 32 SUN dataset preprocessing 72
Figure 33 χ^2-distances between the tested language models and the image model for conditional probabilities of objects $P(O|O)$. 73
Figure 34 Average rank over all images and objects using different language models and ID (image data) 75
Figure 35 Human action suggestion: based on the objects and scenes recognized in an image, the system suggests the most plausible actions. The action models provide the relationships between objects - scenes - verbs 76
Figure 36 An energy-based model for action recognition 77
Figure 37 Probability distributions of scene over object extracted from: (left) image dataset; (right) TypeDM model (as there are many <object - scene> relations, only a few are shown on the Y-axises). The number of relations in the TypeDM is much bigger than in the image model, which shows a more general model than the image one. 81
Figure 38 The average rank over all images based on object setting: O_{gs} when testing on the 89 action dataset 83
Figure 39 The average rank over all images based on object setting: O_{gs} when testing on the TUHOI dataset 85
Figure 40 The number of unseen verb-object pairs in each model when testing on the TUHOI dataset 85
Figure 41 Examples of word-link-word and their weights in the distributional memory 88

LIST OF TABLES

Table 1 An example of a matrix formed by a window with width five for a sentence: "The Horse Raced Past the Barn Fell" . . . 25
Table 2 Links in DepDM . . . 26
Table 3 Parameters and variables of the generation process for LDA . . . 29
Table 4 An example of a metadata in the Bridgeman catalogue . . . 40
Table 5 The Bridgeman browse categories . . . 42
Table 6 Categories used by the annotators . . . 45
Table 7 Experimental Setting . . . 47
Table 8 P, R and F measures – Evaluation . . . 47
Table 9 Matching-based Classifier: number of correct categories found (for 1,049 queries) . . . 48
Table 10 Correct categories checked by the expert for the 270 queries (using the click-through information) . . . 50
Table 11 Correct categories checked by the expert for the 270 queries (without looking at the click-through information) . . . 50
Table 12 Learning-based Classifier: number of correct categories found (for 1,049 queries) . . . 50
Table 13 Examples of relations extracted from ConceptNet 5 . . . 60
Table 14 Random R-LDA topics with the relations between Noun-Object and between Noun-Verb . . . 63
Table 15 The statistics of the dataset used for estimating R-LDA models for each relation type . . . 64
Table 16 19 objects and 15 scenes . . . 66
Table 17 A comparison of available human action datasets in terms of number of objects and actions . . . 67
Table 18 List of the 200 objects in the DET dataset . . . 68
Table 19 The statistics of the DET dataset . . . 68
Table 20 Some statistics of the human action dataset . . . 71
Table 21 $\mathcal{X}^2$ distance for relations between verbs, objects, scenes from different language models to image data . . . 80
Table 22 Average rank over all images AR_I of the human action recognition using different settings: O_{gs}, O_{rec} use only objects (gold standard and object recognizer); S_{gs}, S_{rec} use only scenes, $O_{gs}S_{gs}$ and $O_{rec}S_{rec}$ integrate both objects and scenes together . . . 82
Table 23 Average rank over all images vs. actions of the human action recognition using the $O_{rec}S_{rec}$ setting . . . 83
Table 24 (Mean) Average Precision of Classical BoW and our approach which integrates a Felzen/BoWL object recogniser with TypeDM. The number of training examples for Classical BoW are in brackets. . . . 84
Table 25 Results of the model adaptation with different tailoring function . . . 86
Table 26 Results of the classifier with and without position information . . . 87
Table 27 Objects with higher accuracy when using position information . . . 87

Table 28 Actions that can be disambiguated by positions (Group 1) vs. actions that cannot be disambiguated by positions (Group 2) and their links in the language model 89

1

INTRODUCTION

1.1 MOTIVATION

Long in our history, human has always dreamed of being able to create intelligent computers that can think, reason and understand humans. This dream has been reflected in a number of works of science fiction describing people's imaginative ideas about futuristic science and technology. The past decade has seen a rapid development in the field of artificial intelligence which has taken us closer and closer to fulfilling such dreams. One of the basic required behaviors for a smart computer is learning. As a consequence, machine learning has become a major branch of artificial intelligence, where many different algorithms enabling computers to learn have been developed.

Today machine learning systems are applied successful to many tasks in various fields: from natural language processing (NLP), computer vision, medical diagnosis, to stock market analysis and many more. One of the important steps in building a machine learning system is data collection and representation. The system can then learn from the data and make predictions about new data. Having good and efficient training data is essential in building a good machine learning system. However, depending on the applications, there are many cases that the available data is poor in quality and quantity that it does not provide sufficient information for training.

An example of poor quality data in NLP is short and noisy text data such as users' comments, messages, queries and short advertisements. A common way of representing text documents for building machine learning systems is using a bag-of-words model, where each document is represented by a vector of word frequency, each entry of the vector corresponds to a word in a dictionary. When text documents are short, these vectors become very sparse. Calculating similarities between these documents becomes difficult since their vectors do not share common features. Building machine learning systems upon these poor documents is therefore challenging.

The poor quantity of training data can also result in decreases of performance of the machine learning systems. Taking one of the most popular machine learning task, classification, as examples: when the number of classes increases, new training data for every new class needs to be collected. It requires that each class has a sufficient number of training items for the classifier to work well. For many tasks, the process of annotating new training data is very expensive and even infeasible. In computer vision, there has been a recent interest in action recognition in images. As the number of possible actions can be recognized in images is very large, there are many new classes needed to be added, requiring many more training images. Annotating images to satisfy this growing number of actions becomes prohibitively expensive, while the machine learning system cannot work well without enough training data.

The above given examples have illustrated the problems of poor data in machine learning systems, which happen in many learning tasks. The aim of this thesis is to help such systems to deal with poor data by automatically enriching such data with external knowledge. In particular, we propose a framework that takes advantage of large scale external text data to build general knowledge. This knowledge is then used to enrich poor data in various forms to help increasing the performance of

the machine learning systems. Next, we will summarize the main contributions of the thesis and the applications of the proposed framework in both language and vision domains.

1.2 CONTRIBUTIONS

This thesis presents a general framework that can learn knowledge from large text corpora and uses this knowledge to automatically enrich poor data in various machine learning systems. Firstly, we illustrate how different text modeling techniques can be used in our framework to extract knowledge from text corpora. We examine the difference between these techniques and give recommendation of which models to use in which cases in our framework. Secondly, we show how to integrate this external knowledge with poor data using several combination methods. We illustrate the performance of our framework in both language and vision domains.

1.2.1 *Applications in language domain*

We show that our framework can help dealing with the poor quality of data as given in the first example. We take a query classification task as our case study and present our method in enriching short queries with topics learned from external dataset. We also further integrate this topical knowledge into a support vector machine query classifier to improve the performance of the system.

1.2.2 *Applications in vision domain*

We show that our framework which exploits knowledge from text data can also be useful in computer vision applications. It can help dealing with the poor quantity of data as given in the second example. We introduce a method to enrich human action recognizers with information learned from text data. Such information helps increasing the performance of action classifiers, especially for classes where there are only few training images.

In general, this thesis addresses the problems of poor data in machine learning systems. The performance of the proposed framework is illustrated in both language and vision domains. The key idea of the framework is based on the automatic knowledge extraction from largely available text data and the use of such knowledge in data enrichment. It allows computers to learn from inexpensive data collected on the web, which is growing faster and faster day by day. Small steps in improving the performance of the machine learning systems will allow us to get closer to the dream of creating intelligent computers someday.

2

THE POOR DATA ENRICHMENT FRAMEWORK

During the last decade, machine learning techniques have been widely used in various tasks and achieved satisfactory results in different applications and domains. However, the performance of many machine learning systems largely depends on the quality of data. These systems work well with rich data but usually perform worse when working with poor input data. For example, current document classification or categorization systems generally achieve good performance, but it is usually harder to work with short and poor documents such as queries, text messages, snippets. One of the reasons is that poor data does not provide enough features for building a good classifier.

To deal with the poor data problem, we propose a framework for enriching input data with general knowledge automatically extracted from a large external dataset. This universal knowledge is used to provide the poor data with an informative context in which the data can be better processed and interpreted. To build language models from external dataset, we exploit both structured and unstructured data. For structured data, we take advantage of prebuilt databases where data has already been organized to models that computers can process. For unstructured data, i.e., free form text, we employ two different kinds of models: distributional semantic models and generative models to rearrange the data collections and extract knowledge from them.

Our general framework is flexible and can be used to enrich data in different domains. We illustrate the performance of the framework in various tasks in language and vision applications. In the language application, we present how enriching short and poor text documents help increasing the performance of a query classification task. In the vision application, we show that enriching images with general knowledge learned from text data can help action recognition and recognition by context tasks. In the following sections, we will first discuss our general framework, its components and potential applications. Then we will describe the tasks that we will deal with in language and vision applications to illustrate the flexibility and applicability of the framework.

2.1 OUR GENERAL DATA ENRICHMENT FRAMEWORK

Enriching data with external knowledge has been successfully applied to support artificial intelligence systems in various domains including medicine (Mycin [Buchanan and Shortliffe, 1984]), chemistry (Dendral [Lederberg, 1987]), insurance, credit authorization, scheduling and planning (eXpert CONfigurer, XCON [Prendergast and Winston, 1984]). The most common method used in such systems is based on encoding knowledge in ontologies and rules. It then enables reasoning from the world fact knowledge bases to solve complex problems.

Our general framework is depicted in Figure 1. Instead of relying on rules and encoding knowledge using systems such as ontologies, it takes advantage of available text data in structured and unstructured format for enrichment. The structured databases that can be used in the framework to encode the semantic relations between words, such as commonsense knowledge, lexical and semantic

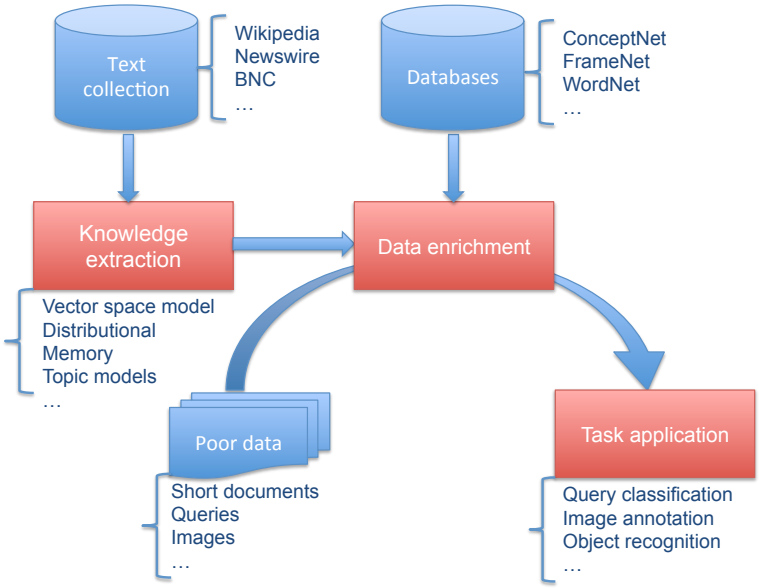


Figure 1: A general framework for data enrichment based on text collection and prebuilt databases

databases (e.g., ConceptNet[1], FrameNet[2], WordNet[3]). The information in these databases has already been organized to allow computers to read and process. Unstructured data on the other hand requires modeling techniques to extract knowledge that computers can process. With the explosion of the World Wide Web, there exists a number of available documents that can be easily collected across the web. The framework takes advantage of this unstructured documents, using different text modeling techniques. The information extracted through text modeling and databases is used for enriching poor data, which will then be fed into different systems for different task applications.

Our general framework consists of the following steps: collecting large text corpora and databases, modeling text collections to extract knowledge and enriching poor data for each applications. We will discuss each step in detail in the following sections.

2.1.1 *Selecting text collection and databases*

The most common source of text data which is rich and easy to collect is the World Wide Web. Many sources are available such as newswires, webpages, chat logs, social networks, etc. This kind of data has a wide variety of genres, structures, content and usually contains a lot of noise. Using this source of text data requires selecting useful discourses and filtering noises. Reliable sources of data which have been used intensively among the computational linguistic community are Wikipedia, DBPedia, the British National Corpus (BNC), etc.

Beside these text collections, researchers in computational linguists have also developed many databases for various purposes. Popular databases are WordNet, FrameNet (a lexical database containing semantic frames), ConceptNet (commonsense knowledge database), and CoreLex[4] (an ontology and semantic database of 40,000 nouns based on WordNet).

1 http://conceptnet5.media.mit.edu/
2 https://framenet.icsi.berkeley.edu
3 http://wordnet.princeton.edu
4 http://www.cs.brandeis.edu/ paulb/CoreLex/corelex.html

Selecting sources of data for building a knowledge base largely depends on the final purpose of the application. For building a general knowledge base with commonly used English words and their relationships, general data sources, such as Wikipedia and BNC can be used. For a specific domain, selecting relevant text documents to build the collection is an important step to aid the performance of the application.

We investigate the problem of domain specific applications by looking at query classification within the art, cultural and history domain. We discuss the problem of choosing the right text collection by comparing models learnt from different data sources in Section 4.2.

2.1.2 *Knowledge extraction*

An important part of this framework is to model unstructured data, i.e., text collections to extract knowledge from it. Big data analytics has recently become a new trend with the requirement of collecting, organizing and analyzing large sets of data. Many techniques to organize and represent knowledge extracted from text have been developed and proposed. One line of research in this area is automatic knowledge acquisition from various sources and its structured representation, such as ontologies. Another growing direction focuses on methods to model text corpora statistically. From n-grams, bag-of-word models to vector space, semantic space [Salton et al., 1975] and probability models such as probabilistic latent semantic analysis (pLSA) [Hofmann, 1999], latent dirichlet allocation (LDA) [Blei et al., 2003] to a multi-purpose distributional semantic model such as Distributional Memory [Baroni and Lenci, 2010].

In our work, we focus on the second line, aiming at discovering the semantic relations between words statistically and use this information as our knowledge base. In Chapter 3, we will present in detail different techniques for representing text corpora and those that can be used in our general framework. Some examples of recent techniques that have been developed and shown to be effective include vector space models, distributional memory, and topic models. For vector space models and distributional memory, the output of this step is the semantic relationship between word pairs. This relationship is determined in a discriminative way, hence using its output itself cannot predict unseen word pairs, although further computation may be used to extend the model. A topic model on the other hands is a generative model, which is flexible and can be able to predict unseen pairs of words. The relationship between words is represented implicitly through the concept of *topic*.

Generally, the output of this step is the semantic relationship between words and words (using distributional memory, vector space model) or words and clusters of words (using topic models), which will then be further extracted for enriching poor data in the next step.

The choices of modeling techniques largely depend on the nature of the data and its application. Text models can be classified into two directions: the first kind of models captures the direct relations between words and model the relations between the two words by analyzing their co-occurences. Examples of such techniques are the Hyperspace Analog to Language model [Landauer and Dutnais, 1997] (using vector space based on weighted co-occurence values within a fixed window), and Distributional Memory. The second kind of models capture better the indirect relations between words by analyzing clusters of words and how the two words are related to each other through their underlying topics (e.g., when using topic modeling techniques). In Chapter 3, we will further discuss the differences between these text modeling techniques. In Section 4.2 and Chapter 5, we will report our choices of modeling methods for different tasks based on their purposes. Our findings in the effect of different text models for a given task will then be summarized and concluded in Section 5.7.

2.1.3 *Data enrichment and task application*

Having learned the relations between words and clusters of words from large corpora through text modeling, we use this knowledge and available databases to enrich poor data. Many common tasks in artificial intelligence (AI) such as document classification, data clustering start from poor data such as short text documents (e.g., queries, web snippets) or have few annotated training data available (e.g., fully annotated images). This framework takes advantage of available text collections and prebuilt databases to enrich poor data. Potential applications can vary from different tasks and different domains. In the next section, we will describe our two application domains.

2.2 LANGUAGE AND VISION APPLICATIONS

In this section, we will describe how we apply the general framework to deal with specific tasks in both language and vision domain.

2.2.1 *Language application: Query classification*

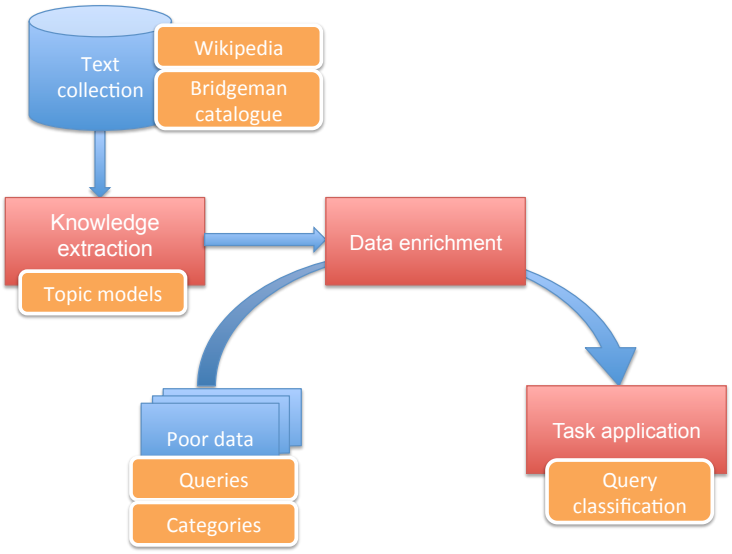


Figure 2: Applications in language domain: Query classification

In the language domain, enriching poor text data with knowledge extracted from general corpora has been studied recently. Various applications including topic enrichment for document classification [Phan et al., 2010], metadata enrichment [Newman et al., 2007], online contextual advertising [Le et al., 2008], author name disambiguation [Bernardi and Le, 2011] have been done for the last few years. In this work, we employ the general framework for query classification, i.e., the task of assigning each user query to a corresponding category (topic) (Figure 2).

Queries are usually short and generally more difficult to classify than longer documents. Our approach is to enrich these queries with information we learned from external data. In particular, we exploit text collection using two different sources: Wikipedia for building general knowledge, and a library catalogue to build domain specific knowledge. For text modeling techniques, we choose to use topic models to estimate topics for the text collection. After that, the estimated model will be

used to analyze topics for both queries and categories before being fed into a query classification system (Section 4.2). We also include this enriched information (i.e., topics) into the training data and integrate it with a support vector machine for query classification in Section 4.3.

2.2.2 *Vision task applications*

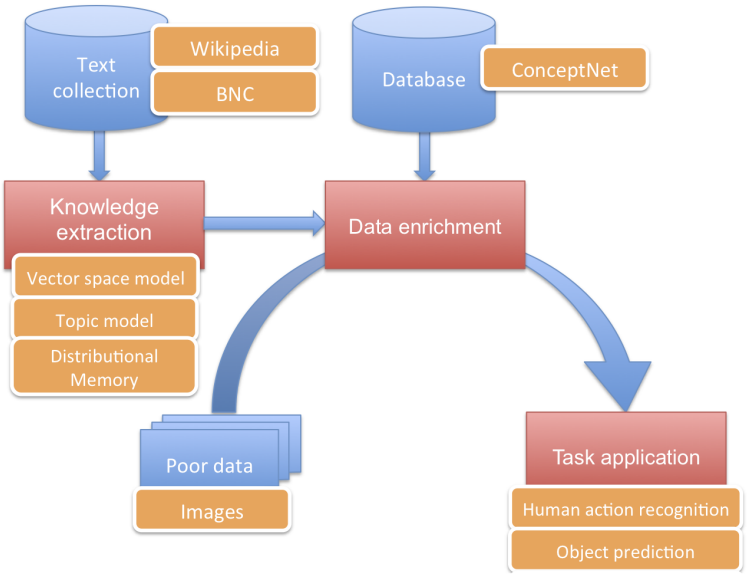


Figure 3: Applications in vision domain: human action recognition, object prediction

In computer vision, object recognition is the task of finding a given object appearing in an image (e.g., book, person, chair, horse). This task can be challenging due to different view points, lighting conditions, sizes/scale of the given objects in different images. The term *scene* is usually referred to common places such as streets, countrysides, dining room, kitchen. Both object and scene recognition are usually performed using machine learning techniques based on an annotated image dataset. Another recent interest in computer vision is the task of human action recognition, which requires more complicated training data and usually performs much worse that the other two tasks due to its complexity. All these visual tasks would require expensive training images and mostly depend on visual features (e.g., shape, textual, color).

To allow the computer analyzing the semantics behind each image rather than purely based on visual features, we propose the use of commonsense knowledge learned from texts and available databases to guide the recognition process. The framework for these vision applications is illustrated in Figure 3. In particular, we use general knowledge by extracting the relations between human, objects, verbs and scenes from text collections such as Wikipedia and BNC. This knowledge extraction is done using different techniques: vector space models, topic models and distributional memories. Given images, the relations extracted from text will help understanding the co-occurences of objects in images, together with scenes and human actions. A detailed description of these tasks and how we use knowledge from text to help better image understanding are presented in Chapter 5.

2.3 SUMMARY

We exploit the knowledge learned from databases and text collections for poor data enrichment and demonstrate its use in different applications both in language and outside the language domain, i.e., computer vision. We use two different kinds of external data for enrichment: structured data (prebuilt databases) and unstructured data (text documents collected on the web). For unstructured data, we employ statistical modeling techniques to represent the knowledge to be enriched to poor data. We examine two types of models for this task: distributional semantic models such as vector space models and generative models such as topic models. The final aim of this text modeling process is to discover the semantic relationships between words represented in unstructured text data. In the next chapter, we will present the background knowledge including the databases, text collections and modeling techniques that will be used in our framework.

3

BACKGROUND

In this chapter, we will discuss some background knowledge and previous work that are helpful to understand the rest of the thesis. First, we will present available commonsense databases and techniques for extracting knowledge from text data and how we extract knowledge from this data, which will be used in chapter 4 and 5. After that, we will discuss techniques for extracting visual features from images. Finally, we will present models that are used in chapter 5 to combine the features extracted from language and vision together.

3.1 KNOWLEDGE ACQUISITION FROM TEXT DATA

Our general knowledge-based framework introduced in the previous chapter is based on the knowledge extraction from text collections and databases. In this section, we will first review available databases that can be used in the framework. Then we will describe different techniques that are used to extract required knowledge from general text collections, instead of pre-built databases.

3.1.1 *Commonsense knowledge database*

Many tasks in artificial intelligence (AI) require commonsense knowledge in order to allow computers behave intelligently. Commonsense knowledge is facts and information that everybody is supposed to know. For example, everybody knows that a knife is used to cut, a cup can be used to store water. Everybody knows a person can ride a horse or a bike, and can feed a horse but not feed a bike. Most of this knowledge one can perceive through experience without being taught and everyone can easily pick it up. However, how can we represent this commonsense knowledge and teach computers to understand and use it? Researchers in computational linguistics and AI have created many resources and databases that are machine-readable. From lexical databases such as WordNet, FrameNet, to knowledge bases such as Cyc[1], ConceptNet. Following, we will explore some databases encoding commonsense knowledge that can be processed by computers.

WORDNET One of the most commonly used lexical database in computational linguistics is WordNet.[2] Manually constructed for the English language, WordNet includes nouns, verbs, adjectives and adverbs. All of them are grouped into sets called synsets, where each synset represents a unique meaning. Every member of a synset expresses the same concept, although they may not be interchangeable in every context, for example, *automobile* and *car*, *big* and *large*. Based on the type of word, there are different semantic relations that connect synsets together. Totally, there are 117,000 synsets linked to each other by *conceptual relations*. The major relations in WordNet, that

1 http://www.cyc.com/platform/opencyc
2 http://wordnet.princeton.edu/

are not tied to specific lexical categories, are synonymy, as between *cold* and *frigid*, *happy* and *joyful*; antonymy as between *hot* and *cold*, *poor* and *rich*.

For NOUNS, the main relation is hyponymy (or Is-A relation), which is used to organize nouns into hierarchies. From the most general concept *entity*, the noun hierarchies go to more specific synsets. For example, the synset *mailbox, letterbox* is a hyponym of the synset *box*, which is in turn a hyponym of the synset *container*. Other relations include meronym (or Part-Of relation, as *window* is part of *building*), coordinate terms (synsets that share the same parent node, as *motorcar* and *truck* have the same parent node *motor vehicle*).

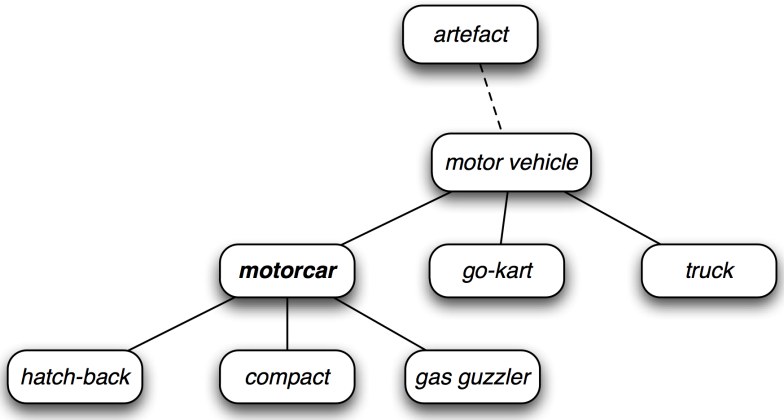


Figure 4: An excerpt from the WordNet concept hierarchy

For VERBS, the most common relation is troponymy, i.e., two synsets that express a similar manner such as *talk* and *whisper*. Other relations are entailment (e.g., *to sleep* and *to snore*), backward entailment (e.g., *divorce* and *marry*).

For ADJECTIVES, there are descriptive and relational adjectives. Descriptive adjectives are organized in terms of antonymy, e.g., *big* and *small*. These pairs of direct antonyms are linked to semantically similar adjectives, such as *wet* to *soggy, waterlogged*. Relational adjectives are mostly connected to their corresponding nouns, for example *atomic, nuclear* are connected to *atom, nucleus*.

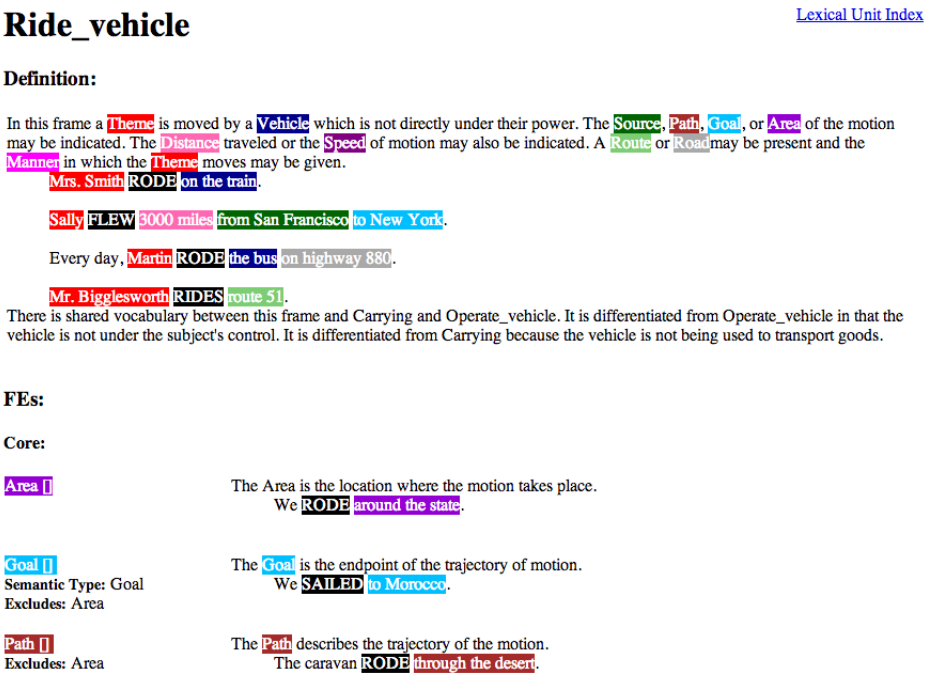
ADVERBS are mostly pointed to their corresponding adjectives that they are derived from, such as *interestingly* to *interesting*.

WordNet can be used as an ontology or thesaurus, in which concepts and relations are well-organized. Users can look up for words with related meanings through a browser. It can also be further used for word sense disambiguation through words' semantic similarities [Brown, 2005]. There is no syntactic information and semantic roles of, for example, nouns functioning as arguments of verbs encoded in WordNet.

FRAMENET Different from WordNet, where no syntactic information is encoded, FrameNet contains this information and represents semantic roles of concepts in a structure called *frame*. A *frame* describes a relation, event, or objects and the set of participants in it. It is based on the theory that the meaning of a word can only be understood against the frame it belongs to [Goddard, 2011]. A typical example is a cooking concept, which usually involves a person doing the cooking (cook), the tool or source of heat used for cooking (heating_instrument), the food to be cooked (food) and the thing used to hold the food during the cooking process (container). Another example about the frame ride-vehicle is given in Figure 5, where some core elements related to this concept are being highlighted (e.g., Area - the location that it takes place, Goal - the end point of the trajectory of

motion, etc.). To represent the frame elements for annotated sentences, FrameNet annotations are composed of triples: frame element name (e.g., food), a grammatical function (e.g., object) and a phrase type (e.g., noun phrase). Totally, there are around 1,200 semantic frames and 190,000 example sentences annotated.

Created at the International Computer Science Institute in Berkeley, California, FrameNet has been regularly annotated with English lexicon, and recently has been further extended to other languages such as Spanish, German and Japanese. Generally, they are often similar across languages since FrameNet is based on semantic meanings. An example given in FrameNet is the activity buying and selling, which often involves the frame elements Buyer, Seller, Goods and Money in all languages.



Ride_vehicle

Lexical Unit Index

Definition:

In this frame a Theme is moved by a Vehicle which is not directly under their power. The Source, Path, Goal, or Area of the motion may be indicated. The Distance traveled or the Speed of motion may also be indicated. A Route or Road may be present and the Manner in which the Theme moves may be given.

Mrs. Smith RODE on the train.

Sally FLEW 3000 miles from San Francisco to New York.

Every day, Martin RODE the bus on highway 880.

Mr. Bigglesworth RIDES route 51.

There is shared vocabulary between this frame and Carrying and Operate_vehicle. It is differentiated from Operate_vehicle in that the vehicle is not under the subject's control. It is differentiated from Carrying because the vehicle is not being used to transport goods.

FEs:

Core:

Area []
The Area is the location where the motion takes place.
We RODE around the state.

Goal []
Semantic Type: Goal
Excludes: Area
The Goal is the endpoint of the trajectory of motion.
We SAILED to Morocco.

Path []
Excludes: Area
The Path describes the trajectory of the motion.
The caravan RODE through the desert.

Figure 5: An example of a frame *Ride-Vehicle* in FrameNet: its definition, examples of frame elements that are essential to the meaning of the frame *Ride-Vehicle*, i.e. Core FEs (the elements that are essential to the meaning of a frame): Area (the location where the motion takes place, Goal (the endpoint of the trajectory of motion, Path (the trajectory of the motion), etc.

FrameNet has been used in various applications in computational linguistic tasks, from question answering, information extraction, paraphrasing to textual entailment. One of a typical task originated from FrameNet, which has become a standard task in natural language processing is the semantic role labeling (SRL) task [Gildea and Jurafsky, 2002].

CYC If WordNet and FrameNet are widely used within the computational linguistic community to analyze the semantic meanings of words, the term *commonsense knowledge base* is more commonly used in the AI community. It refers to a database where all the general knowledge is represented in a way that computer programs can read and process to support AI tasks. One of the earliest system that exploits knowledge base of commonsense knowledge is Cyc, which was started in 1984 by the Cycorp company.[3] It aims to formalize human commonsense knowledge into a logical frame using first-order relationship. It contains over 239,000 concepts and 2,093,000 facts, largely handcrafted by

3 http://www.cyc.com/platform/opencyc

knowledge engineers. These concepts and facts are organized into an ontology, with main classes such as *place, organization, predicate, business related thing, person*. Cyc has been used in various AI tasks, such as health care, facilities status monitoring and alerting, help-desk expertise management and so on.

OMCS Similar to Cyc, Open Mind Common Sense (OMCS) was created within an AI project in the media lab at the Massachusetts Institute of Technology (MIT) in 1999. However, instead of manually building the knowledge base, OMCS was inspired by the success of web-based knowledge projects. The aim of OMCS was to turn volunteer knowledge on the web into a commonsense knowledge base. If Cyc attempts to build a knowledge base in a logical frame, OMCS focus on natural language knowledge representations. For example, objects and events are expressed in natural language phrases, such as "The sun is hot", "A chair is used to sit on", etc. Emotional content is also described in OMCS, for example "Spending time with friends causes happiness". Most of this knowledge has been built by simply asking people on the web to fill in some templates. Knowledge is extracted and collected by analyzing the content that the contributors have typed in and discovering the patterns. OMCS also makes use of data collected by the game "Verbosity".[4] In this game, a player will be asked to describe some concept and the other will try to guess that. For example, questions such as "A coat is used for...?" and the player is supposed to fill in something meaningful, such as "keeping warm". Totally, OMCS contains 30 different activities, with over 700,000 sentences of commonsense knowledge collected from over 14,000 contributors around the world.

CONCEPTNET Containing a large collection of commonsense knowledge expressed in natural language phrases and sentences, OMCS itself is not directly computable. To enable computers to use this database, a semantic network was build from OMCS as a directed graph. Concepts and relations are extracted from sentences using pre-defined rules. There are 1.6 million assertions with over 300,000 nodes forming the semantic network. An example is given in Figure 6: concept *cake* is CreatedBy *bake*, which is MotivatedByGoal *eat*, which is in turn MotivatedByGoal *survive*. This graph was created by first applying a set of rules to the sentences in OMCS. Then a set of normalization and relaxation procedures is applied to optimize the connectivity of the semantic network [Liu and Singh, 2004].

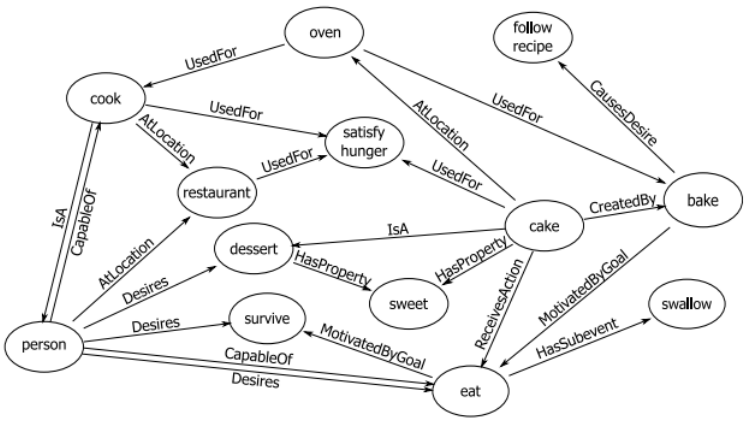


Figure 6: A ConceptNet graph

4 http://www.gwap.com/gwap/gamesPreview/verbosity/

The main process of building ConceptNet is illustrated in Figure 7. The extraction phase involves mapping OMCS English sentences into ConceptNet's binary-relation assertions using around 50 extraction rules. These rules are defined using regular expression patterns, which are based on the provided templates in OMCS, where users fill in the blank. For example, "A [lime] is [a very sour fruit]". Two relations extracted are IsA(lime, fruit) and PropertyOf(lime, sour). The normalization phase includes spell checking and normalization for every extracted nodes. The relaxation phase is applied to merge duplicated assertions, add frequency counts to track how many times a common fact occurs in the corpus. The relation "IsA" is used to create a hierarchy graph for concepts by introducing parent and child nodes. Thematic and lexical generalizations are also applied to map specific knowledge to more general one (e.g., "buy food" to "buy").

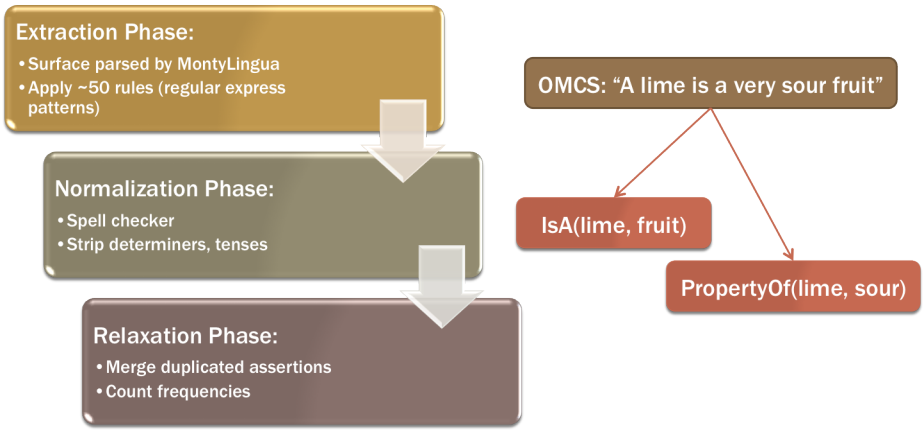


Figure 7: Main phases of building ConceptNet from OMCS

ConceptNet also provides a practical commonsense reasoning toolkit. It supports natural language processing tasks such as computing contextual neighborhood (e.g., given a concept and no other biases, what other concepts are most relevant?), topic generation (e.g., entering *restaurant* would return phrases like *order food* and *waiter* and *menu*), etc.

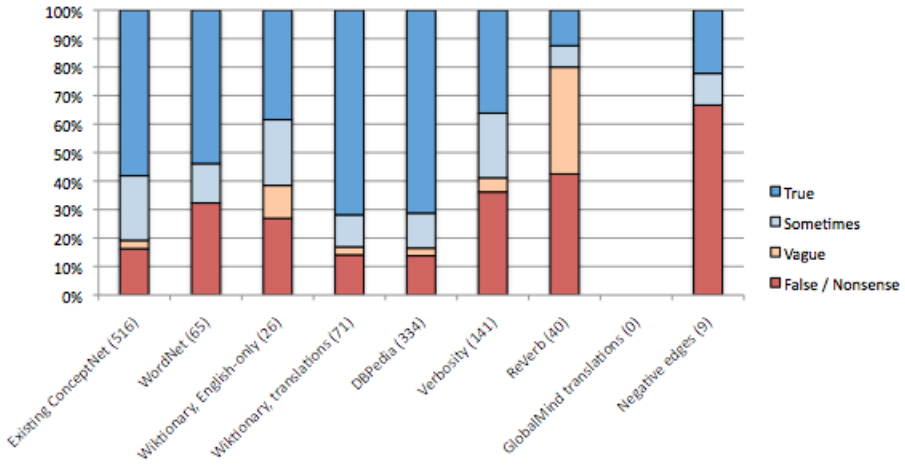


Figure 8: ConceptNet 5 human evaluation [Speer and Havasi, 2012]

The most current version of ConceptNet is ConceptNet 5, which is composed of different sources of knowledge, such as OMCS (in English, Portuguese, Dutch), GlobalMind (translations between assertions), English Wiktionary (word definitions in many languages), WordNet 3.0 (aligns entries in ConceptNet with WordNet), DBPedia and Wikipedia's free text. To evaluate the content of ConceptNet 5, people were asked to classify each random sample of the edges in ConceptNet as "generally true" to "false/nonesense" [Speer and Havasi, 2012]. The relative proportions of evaluating 1,888 statements are given in figure 8. Besides many true statements rated by people, there are also false or nonsense statements (around 17% for ConceptNet database).

Generally, ConceptNet is one of the largest freely commonsense database, which supports practical commonsense reasoning systems using unconventional techniques. The advantages of ConceptNet is that it is easy to use with simple and intuitive structure. The knowledge in the database is presented in natural language and was built automatically. However, as it was built from free text sentences, it contains uncontrolled vocabulary and can be biased in terms of content. It also contains false/nonsense statements that have not been verified.

	Database content	Resource	Capabilities
ConceptNet (2004)	Commonsense	OMCS (from the public) (automatic)	Contextual inference
WordNet (1985)	Semantic Lexicon	Expert (manual)	Lexical categorisation & word-similarity
Cyc (1984)	Commonsense	Expert (manual)	Formalized logical reasoning

Figure 9: A comparison among WordNet, Cyc and ConceptNet

The main different points between ConceptNet and the other two databases WordNet and Cyc are spelled out in figure 9. Both ConceptNet and Cyc are commonsense knowledge bases whereas WordNet is a semantic lexicon. WordNet and Cyc are built based on manually annotation of experts while ConceptNet exploits the contributions of people across the web in OMCS and automatic knowledge extraction from that. WordNet provides lexical knowledge about words and their relations, while Cyc supports formalized logical reasoning and ConceptNet offers contextual commonsense reasoning over natural language texts.

In our general framework presented in the previous chapter, we employ commonsense knowledge databases to enrich poor data. Since ConceptNet is based on free text and provides one of the largest commonsense knowledge databases, we choose to use this in our framework. We evaluate whether the knowledge extracted from it is helpful for solving different problems in computer vision such as object prediction and human action recognition (chapter 5).

Instead of using existing commonsense knowledge bases, we can exploit directly the content of text available on the internet to extract knowledge from it. In the next part, we will discuss most common methods for extracting knowledge from such data.

3.1.2 *Semantic space and window-based model*

SEMANTIC SPACE Analyzing word meanings and their relationships have always been the central research in natural language processing and computational linguistics. The concept of semantic space dates back to 1957, [Osgood et al., 1957] defines "a semantic space as a space in which words and concepts are represented by points, the position of each such point along each axis is somehow related to the meaning of the word". Semantic space encodes the meaning of words in a

high dimensional space, where dimensions represents, for example other words or documents the word has appeared. It can be used to measure the relationship between two words or concepts by calculating the distance between two points representing these words in the space.

Generally, the meaning of a word is represented by a vector, whose dimensions are the vocabulary of the model, and the values are the co-occurrence frequency of the word represented. The word vector is constructed from big corpora and various weighted schema can be applied for better capturing the meaning of a word.

LEXICAL CO-OCCURRENCE: WINDOW MODEL Following the idea of the semantic space, [Lund and Burgess, 1996a] used lexical co-occurrence to represent words in a multi-dimensional space. The idea is to move a sliding window over a corpus, at every window movement, the co-occurrence values of each pair of words are calculated. From this counting, a co-occurrence matrix is constructed, where each row and each column is a word in the whole vocabulary. The size of the matrix is the size of the vocabulary. An example of such matrix is given in figure 1, where the matrix is calculated by extracting the word co-occurrences within a sentence. The distance between two arbitrary points/words can be computed as the distance between two word vectors, for example using the Minkowski family of distance metrics:

$$\mathtt{distance} = \sqrt[r]{\sum (|x_i - y_i|)^r} \tag{1}$$

where $r = 2$ results in the Euclidean distance.

	barn	fell	horse	past	raced	the
<PERIOD>	4	5	0	2	1	3
barn	0	0	2	4	3	6
fell	5	0	1	3	2	4
horse	0	0	0	0	0	5
past	0	0	4	0	5	3
raced	0	0	5	0	0	4
the	0	0	3	5	4	2

Table 1: An example of a matrix formed by a window with width five for a sentence: "The Horse Raced Past the Barn Fell"

The width of the window has an important impact on the presentation of the semantic space and modeling human concept similarity. The choice of the width largely depends on the task and the word relations that it requires. We further discuss about this choice and how it effects the performance of each task in chapter 5.

3.1.3 *Distributional Memory*

Following the ideas of corpus-based semantics, Distributional Memory (DM) [Baroni and Lenci, 2010] provides a multi-purpose framework for semantic modeling. It aims to build a generalized distributional model that does not need to be retrained for each new specific task.

DM extracts $< w_1, l, w_2 >$ tuples from a dependency parse of a corpus, where each tuple is weighted based on the co-occurrence of w_1 and w_2. l (links) represents the type of this co-occurrence relation. The use of the links is the main difference with lexical co-occurrence. The links can be extracted in different ways such as based on frequent n-grams co-occurring within the same document

Links	Sentence	Tuple
sbj_intr	The teacher is singing	<teacher, sbj intr, sing>
sbj_tr	The soldier is reading a book	<soldier, sbj tr, read>
iobj	The soldier gave the woman a book	<woman, iobj, give>
obj (direct object)	The soldier is reading a book	<book, obj, read>
nmod	good teacher	<good, nmod, teacher>
coord	teachers and soldiers	<teacher, coord, soldier>
prd	The soldier became sergeant	<sergeant, prd, become>
verb	The soldier is reading a book	<soldier, verb, book>
preposition	I saw a soldier with the gun	<gun, with, soldier>
	The soldier talked with his sergeant	<sergeant, with, talk>

Table 2: Links in DepDM

or from a dependency parse of a corpus as how DM was built. A scoring function σ is defined to weight the scores of the tuples. This scoring function takes into account the occurrences of the tuples using mutual information.

There are three DM models proposed in [Baroni and Lenci, 2010] corresponding to different ways of constructing the "link" and/or the scoring function. All of these models have been built from around 2.83 billion tokens consisting of the web-derived ukWac, English Wikipedia and the BNC. The concatenated corpus was preprocessed using Tree Tagger[5] and parsed with the MaltParser[6].

The first model in DM is called DepDM, which is the model with the least degree of link lexicalization. It takes into account links which are prepositions, including the relations between noun-verb, noun-noun and adjective-noun links. An example of links with their corresponding sentence and tuple of the DepDM is given in Table 2.

The scoring function for DepDM is the Local Mutual Information (LMI) [Church and Hanks, 1990], computed on the word-link-word co-occurrence counts:

$$\mathrm{LMI} = O_{ijk} \times \log \frac{O_{ijk}}{E_{ijk}} \tag{2}$$

Given the co-occurrence count O_{ijk} of the first word i, the second word j, the link k, and the corresponding expected count under independence E_{ijk} (Equation 2).

LexDM is a heavily lexicalized model, where links are lexicalized dependency paths and lexico-syntactic shallow patterns. It contains complex links with *pattern+suffix*. For example, the relation <soldier, sbj intr+n-the- j+vn-aux-already, shoot> is extracted from the sentence "The tall soldier has already shot", where:

- **sbj intr**: w_1 is the subject and w_2 is a transitive verb
- **n-the-j**: w_1 is a singular noun (n), definite (the), has an adjective (j) that does not belong to the list of high frequency adjectives (high frequency adjectives have been found to largely irrelevant);
- **van-aux-already**: w_2 is a past-participle (van), has an auxiliary (aux), is modified by "already", belonging to the pre-selected list of high frequency adverbs.

5 http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
6 http://w3.msi.vxu.se/?nivre/research/MaltParser.html

TYPEDM This model is mildly lexicalized, laying somewhere between DepDM and LexDM. Its links are as in LexDM but it uses a different scoring function based on pattern type frequency (i.e., suffixes). For example: <fat, of, land> appears more often, but <fat, of, animal> is semantically more informative. Hence, different surface realizations can be counted either as 3 distinct realizations <the fat of the land, the fat of the ADJ land, the ADJ fat of the land> or 9 distinct realizations <a fat of the animal, the fat of the animal, fats of animal, etc. >.

The intuitive idea is that "what matters is not so much the frequency of a link, but the variety of surface forms that express it". Hence, in this model, the scoring function computes on the number of distinct suffix types displayed by a link. In other words, it counts *types* of realizations but not tokens.

To sum up, DM provides a generalized corpus-based framework that represents the semantic relations between words. Its aim is to train a model that can be used in different tasks. Experimental results reported in [Baroni and Lenci, 2010] have shown its successful applications in synonym detection, noun categorization, selectional preferences, relation classification and so on. For our general framework described in chapter 2, we choose to use the model TypeDM as it was reported to achieve the best results for relevant tasks such as selectional preferences.

3.1.4 *Topic models and extensions*

Topic models are statistical models used to discover the hidden structure of a text collection. They are based on the concept of *hidden topics*, where each topic is defined by a word distribution. The word *hidden* reflects the fact that topics in these models are unknown and not labeled. Its assumption is that each document exhibits a number of topics and in turn, each topic is a mixture of words. For example, a document can be about several topics such as computer science, security and jobs. Each of these topics has several "keywords" that describe the topic, such as algorithm, coding, data, software for computer science. The central computational problem of topic models is to discover the hidden topic structure from a set of observed documents, where topics are modeled by latent variables. It can be thought of as a reverse process of generating new documents: which structure is likely to generate the observed collection?

Two examples of topic analysis using latent models are probabilistic Latent Semantic Analysis (pLSA) [Hofmann, 1999] and Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. pLSA, also known as probabilistic latent semantic indexing (pLSI) was developed based on latent semantic analysis (LSA), adding a probabilistic model. It models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions (Figure 10). pLSA can face the problem of assigning the probability to a document outside of the training test. Moreover, it can lead to the linearly growth of number of parameters along with the size of the corpus.

LDA: LDA is a generative graphical model as shown in figure 11 (left). It can be used to model and discover underlying topic structures of any kind of discrete data in which text is a typical example. LDA was developed based on the assumption of document generation process depicted in both figure 11 and Table 3. LDA can be interpreted as follows.

First, each set $\overrightarrow{w}_m = (w_{m,n})_{n=1}^{N_m}$ is generated by sampling a distribution over topics $\overrightarrow{\vartheta}_m$ from a Dirichlet distribution, a family of continuous multivariate probability distributions parametrized by the input vector α, ($Dir(\overrightarrow{\alpha})$), where N_m is the number of words in that set m. After that, the topic assignment for each observed word $w_{m,n}$ is performed by sampling a word placeholder $z_{m,n}$ from a multinomial distribution ($Mult(\overrightarrow{\vartheta_m})$). Then a word $w_{m,n}$ is picked by sampling from the multinomial distribution ($Mult(\overrightarrow{\varphi}_{z_{m,n}})$). This process is repeated until all K topics have been generated for the whole collection. Note that the number of topics K is fixed for each model.

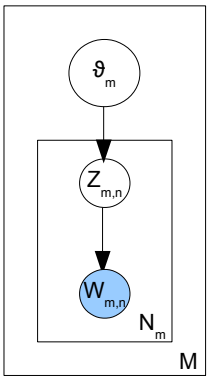


Figure 10: pLSA

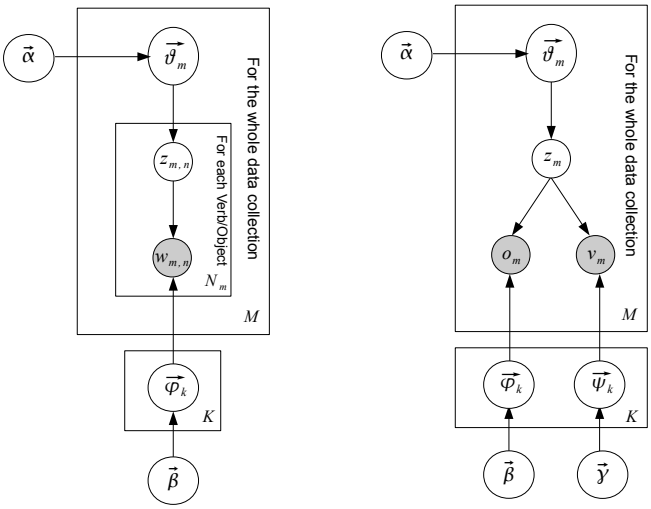


Figure 11: Generative graphical model of LDA (left) vs. ROOTH-LDA (right)

In order to estimate parameters for LDA (i.e., the set of topics and their word probabilities Φ and the particular topic mixture of each document Θ), different inference techniques can be used, such as variational Bayes [Blei et al., 2003], or Gibbs sampling [Griffiths and Steyvers, 2004]. Generally, the topic assignment of a particular word t is computed as:

$$p(z_i = k|\overrightarrow{z}_{\neg i}, \overrightarrow{w}) =$$

$$\left(\frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{v=1}^{V} n^{(v)}_k + \beta_v - 1}\right)\left(\frac{n^{(m)}_{k,\neg i} + \alpha_k}{\sum_{j=1}^{K} n^{(j)}_m + \alpha_j - 1}\right), \tag{3}$$

where $n^{(t)}_{k,\neg i}$ is the number of times the word t is assigned to topic k except the current assignment; $\sum_{v=1}^{V} n^{(v)}_k - 1$ is the total number of words assigned to topic k except the current assignment; $n^{(m)}_{k,\neg i}$ is the number of words in set m assigned to topic k except the current assignment; and $\sum_{j=1}^{K} n^{(m)}_j - 1$ is the total number of words in set m except the current word t. In normal cases,

M	the total number of documents
K	the number of (hidden/latent) topics
V	vocabulary size
$\overrightarrow{\alpha}$, $\overrightarrow{\beta}$	Dirichlet parameters
$\overrightarrow{\vartheta}_m$	topic distribution for document m
$\overrightarrow{\varphi}_k$	word distribution for topic k
N_m	the length of document m
$z_{m,n}$	topic index of nth word in document m
$w_{m,n}$	a particular word for word placeholder [m, n]
$\Theta = \{\overrightarrow{\vartheta}_m\}_{m=1}^{M}$	a $M \times K$ matrix
$\Phi = \{\overrightarrow{\varphi}_k\}_{k=1}^{K}$	a $K \times V$ matrix

Table 3: Parameters and variables of the generation process for LDA

Dirichlet parameters $\overrightarrow{\alpha}$, and $\overrightarrow{\beta}$ are symmetric, that is, all α_k ($k = 1..K$) are the same, and similarly for β_v ($v = 1..V$).

The aim of the sampling process is to estimate the matrix Φ, which gives the topic-word distribution, and the matrix Θ, which gives the document-topic distribution. After finishing Gibbs Sampling, these two matrices are computed as follows.

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{v=1}^{V} n_k^{(v)} + \beta_v} \tag{4}$$

$$\vartheta_{m,k} = \frac{n_k^{(m)} + \alpha_k}{\sum_{j=1}^{K} n_j^{(m)} + \alpha_j} \tag{5}$$

ROOTH-LDA: LDA was designed to capture the relations between documents - topics, and topics - words. To capture the relations between two particular types of words such as between objects and objects, verbs and objects, other extensions of LDA have been developed to address this issue. For example, if we want to model the relations between verbs and objects, LDA only gives semantic clusters of verbs and objects separately, but it does not jointly model both verbs and objects together. This joint probability is instead obtained using some LDA extensions such as those described in [Séaghdha, 2010], [Ritter et al., 2010]. Next, we will explain the method of ROOTH-LDA in [Séaghdha, 2010] inspired by [Rooth et al., 1999].

A relation m is a pair of $< v_m, o_m >$, which is generated by picking up a distribution over topics $\overrightarrow{\vartheta}_m$ from a Dirichlet distribution ($\mathrm{Dir}(\overrightarrow{\alpha})$). Then the topic assignment z_m for both v_m and o_m is sampled from a multinomial distribution $\mathrm{Mult}(\overrightarrow{\vartheta}_m)$. Finally, a particular verb v_m is generated by sampling from multinomial distribution $\mathrm{Mult}(\overrightarrow{\psi}_{z_m})$ and a particular object o_m is generated from $\mathrm{Mult}(\overrightarrow{\varphi}_{z_m})$ (Figure 11, right). In this way, the two different verb-topic and object-topic distributions share the same topic indicators. The model can also be estimated using Gibbs Sampling following the sampling method for LDA described in [Heinrich, 2004]. In particular, the topic z_i of a particular verb v_i and object o_i is sampled from the following multinomial distribution:

$$p(z_i = k|\overrightarrow{z}_{\neg i}, \overrightarrow{v}, \overrightarrow{o}) = \left(\frac{n^{(o)}_{k,\neg i} + \beta}{\sum_{o=1}^{V_o} n^{(o)}_{k,\neg i} + \beta V_o - 1}\right) \times \left(\frac{n^{(v)}_{k,\neg i} + \gamma}{\sum_{v=1}^{V_v} n^{(v)}_{k,\neg i} + \gamma V_v - 1}\right) \times \left(\frac{n^{(m)}_{k,\neg i} + \alpha}{\sum_{k=1}^{K} n^{(k)}_{m,\neg i} + \alpha_K - 1}\right), \quad (6)$$

where $\overrightarrow{v}$, $\overrightarrow{o}$ and $\overrightarrow{z}$ are the vectors of all verbs, objects and their topic assignment of the whole data collection; α, β, γ are Dirichlet parameters. $n^{(o)}_{k,\neg i}$, $n^{(v)}_{k,\neg i}$ is the number of times object o and verb v is assigned to topic k accept the current one. Let V_o and V_v be the number of objects and verbs in the dataset and K be the number of topics, the two verb-topic and object-topic distributions are computed as:

$$\varphi_{k,o} = \frac{n^{(o)}_{k} + \beta}{\sum_{o=1}^{V_o} n^{(o)}_{k} + \beta}; \qquad \psi_{k,v} = \frac{n^{(v)}_{k} + \gamma}{\sum_{v=1}^{V_v} n^{(v)}_{k} + \gamma} \quad (7)$$

Finally, the conditional probability of a verb v^j given an object o^i is calculated through the topic indicator z by summing up over z all products of the conditional probability of the corresponding verb and object given the same topic.

$$\begin{aligned} P(v^j|o^i) &= \frac{P(v^j, o^i)}{P(o^i)} \propto \frac{\sum_{k=1}^{K} P(v^j|z = k) \times P(o^i|z = k)}{\sum_{k=1}^{K} P(o^i, z = k)} \\ &= \frac{\sum_{k=1}^{K} \psi_{v^j,k} \times \varphi_{o^i,k}}{\sum_{k=1}^{K} \varphi_{o^i,k}} \end{aligned} \quad (8)$$

To summarize, we have first described the existing databases that encode the semantic relations between words. Then we have moved to representative techniques for knowledge extraction from text collections: the window based method using lexical co-occurrences, the distributional memory and topic models. Each of these models grasps a different characteristic of word relations in corpora. For example, the window method is the most simple yet still effective way of measuring the relationship between a pair of words by looking into their co-occurrences within a sliding window. Distributional memory is more complex in the sense that it further captures the relationship between words by taking into account the "link" between them. Topic models on the other hand are generative models and by nature, they can be able to predict the relations between any word pair that did not occur in the corpus. Distributional memory can also predict unseen pairs but further computation is required and we have not taken it into account in our study.

Since each of these models captures word relations in different perspectives, we will use all of them in our framework and compare the performance of each of them in the next chapters.

3.2 VISUAL RECOGNIZERS

In Chapter 5, we will apply our general framework of using knowledge to enrich data to the domain of computer vision. Therefore, we will review some state-of-the-art methods for visual concept classification that will be later used in chapter 5. We focus on the Bag-of-words (BoW) [Csurka et al., 2004, Sivic and Zisserman, 2003] algorithms, one of the most widely used visual recognition method in computer vision. The process of extracting the BoW feature for image representation includes

local interest point detection, local image patch descriptors and visual word assignment. Finally, classification algorithms are applied for visual concept recognition.

3.2.1 *General Bag-of-Words framework*

First proposed in the text retrieval domain for document analysis, BoW methodology was later adapted to computer vision domain. In document classification, each piece of text (e.g., a sentence, a document) is represented as a bag of its words, taken out word order and grammar. The frequency (occurrence) of each word is used as features for training a classifier. Analogously, computer vision researchers extract local visual features (i.e., descriptors) from images, map them to visual words, and count their frequencies which result in the final image representation. Each image is represented as a vector of occurrence counts of a vocabulary of visual words.

Generally, constructing the bag of words for image representation involves the following steps. (i) First, image regions are sampled from every images in the training set. (ii) From these regions, some single local descriptors are extracted using algorithms such as scale-invariant feature transform (SIFT). (iii) The visual vocabulary is then learned using an unsupervised clustering method such as k-means. Each cluster corresponds to a *word* in the vocabulary. Once this vocabulary is learned, it will be fixed and used for representing all images in the future. (iv) For each image, all local descriptors are mapped to visual words and their frequencies are obtained to serve as the final image representation. Next, we will describe each step and how these visual words are constructed from images.

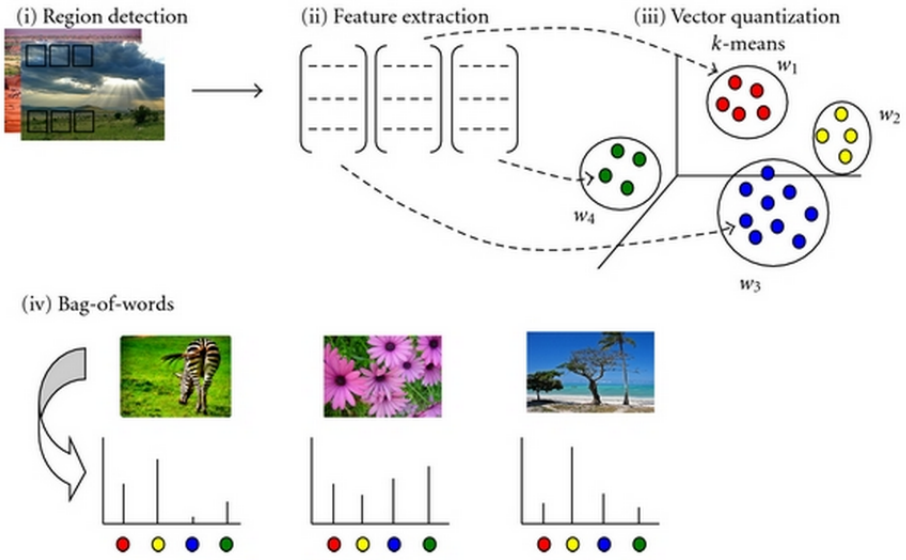


Figure 12: Constructing the bag of words for image representation

3.2.2 *Region extraction*

In the BoW method, the first step to extract visual features from an image is to determine regions from where these features should be extracted. Initially, this was done on interest points, which have a clear, well-defined position in the image space such as high contrast regions, object edges. The

difference of Gaussian (DoG) point detector has been shown to perform well, stable and scalable. It is based on the localization at local scale-space maxima of the difference of gaussian and originally proposed by [Lowe, 1999]. However, later it was found that a dense sampling on a regular grid outperforms local interest points for visual recognition.

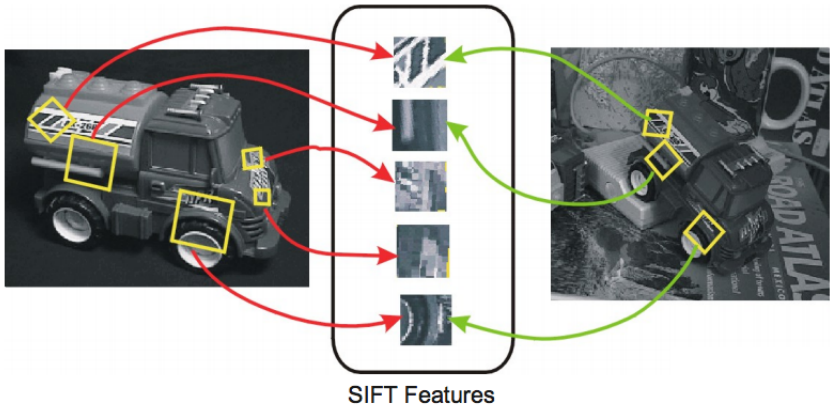


Figure 13: Scale-Invariant Local Features [Lowe, 1999]

3.2.3 *Vector quantization k-means*

Once descriptors are detected, we assign them to a visual vocabulary. Typically, this visual vocabulary is created by using unsupervised learning such as k-means, which has reported to achieve good performance [Zhang et al., 2007, van de Sande et al., 2008, 0003 and Hauptmann, 2008]. The number of visual words generated is the number of clusters k.

The word assignment task involves finding the closest cluster for each data point. In particular, each descriptor from an image is projected to a *visual word* in the predefined vocabulary. For this task, [Uijlings et al., 2010] conducted a comparison between the nearest neighbor assignment and random forests originally proposed by [Moosmann et al., 2008]. Generally, they reported that random forest was much faster in terms of speed and could provide a slightly better result (Mean Average Precision) than k-means. In this thesis we use therefore random forest for visual word assignment.

3.2.4 *Bag-of-words representation and classification*

After the features have been extracted from images, they are fed into a classifier for building or testing a visual concept classifier. Support vector machines (SVM) are probably the most widely used discriminative classifier due to the robustness of the classifier against sparse data and large feature vectors. It has been used successfully in the BoW method. The local feature approach of using BoW model representation learnt by SVM with different kernels has been vastly tested in the area of object recognition. We follow the recommendations of [Uijlings et al., 2013] and use an SVM with histogram intersection kernel, using the fast approximation of [Maji et al., 2008].

For object localization, which not only identifies but also finds the location of the object in the image, we follow the state-of-the-art method of [Uijlings et al., 2013], then use multiple hierarchical segmentation to sample a limited high quality set of possible object locations. From these locations,

we employ the BoW method adopted from [Uijlings et al., 2010] to recognize and localize objects in each image. The scene recognizer is also trained following the BoW method as described above.

3.3 COMPONENT INTEGRATION

In our general framework described in the previous chapter, knowledge from different sources is integrated for each application. In particular, for the vision domain, the features from vision part, i.e., visual recognizers and features from the language part, i.e., language models, have to be combined together. To perform this, we are going to employ an energy-based learning model to optimize the combination parameters. In the following section, we will describe the general ideas and basic concepts of this model.

3.3.1 *Introduction*

In many machine learning problems, to answer questions about unknown variables given observed ones, it is necessary to encode dependencies between variables. Energy-based learning [Lecun et al., 2006] is a framework for capturing these dependencies and solving problems related to prediction, decision making and classification. The general ideas of energy-based learning are based on the association of scalar energies to variables' configuration. The problem of optimizing parameters is therefore formulated as a process of minimizing the energy spending for each configuration.

An *energy function*, denoted as $E(X, Y)$, gives low energies to correct answers and higher energies to incorrect ones, where X is an input variable and Y is the output variable (to be predicted). This function is designed to measure how "good" each possible combination of an output Y given the input X. The inference process searches for the Y that minimizes the energy. The value Y^* chosen from a set $\mathcal{Y}$ is the answer of the model:

$$Y^* = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}}\, E(Y, X) \tag{9}$$

When the set Y has low cardinality, we can perform exhaustive search. In many other cases, when the set is discrete but intractably large or if it is continuous, a strategy called *inference procedure* is usually employed to find or estimate Y that minimizes $E(Y, X)$.

3.3.2 *Architecture*

The main components of an energy-based model (EBM) consist of a family of energy functions, training set, loss function and training phase. Generally, the procedure of finding an energy function that optimizes the configuration to produce Y for each given X is the training process of the EBM. The main components of the EBM are described as follows:

FAMILY OF ENERGY FUNCTIONS: The best energy function will be searched within a family of energy functions ε: $\varepsilon = \{E(W, Y, X) : W \in \mathcal{W}\}$, where W is an indexing parameter

TRAINING SET: A set of training samples S is used to train the model for prediction: $S = \{(X^i, Y^i) : i = 1..P\}$ which contains a set of all inputs and their corresponding outputs

LOSS FUNCTION: In order to assess the quality of each energy function using the training set and the prior knowledge, a quality measure called *loss function* is used, denoted as $\mathcal{L}(E, S)$ or $\mathcal{L}(W, S)$, where S is the training set and W is the index corresponding to the energy function that it measures.

FORM OF THE LOSS FUNCTION: the loss function is generally defined as:

$$\mathcal{L}(E, S) = \frac{1}{P} \sum_{i=1}^{P} L(Y^i, E(W, \mathcal{Y}, X^i)) + R(W) \tag{10}$$

where Y^i is the desired answer, $E(W, \mathcal{Y}, X^i)$ is the energy surface for a given X^i as Y varies and $L(Y^i, E(W, \mathcal{Y}, X^i))$ is a *per-sample loss function*. $R(W)$ is the *regularizer* used to embed our prior knowledge about which energy functions are preferable to others.

LEARNING: The learning procedure involves finding W that minimizes the loss

$$W^* = \underset{W \in \mathcal{W}}{\operatorname{argmin}} \mathcal{L}(W, S) \tag{11}$$

The aim is to design the per-sample loss function such that it gives a low loss to a better energy function (i.e., assigns the lowest energy to the correct answer and vise versa). Generally, that requires shaping the energy function such that the desired value of Y has the lowest energy than other undesired values. Given three types of answers, $Y^i, Y^{*i}, \bar{Y}^i$ correspond to the correct answer, the answer produced by the model and the most offending incorrect answer (the answer with the lowest energy among all incorrect answers) respectively. A *good* loss function is expected to *push down* the energy of the correct answers while *pull up* the energy of other incorrect answers (Figure 14) during the training process.

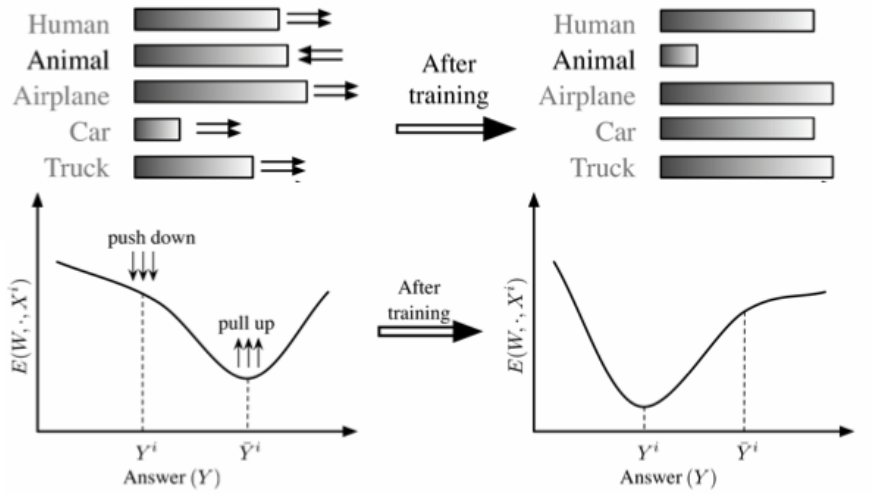


Figure 14: Designing a loss function: push down on the energy of the correct answer, pull up on the incorrect answers (Picture taken from [Lecun et al., 2006])

Building and training an energy-based model generally consist of designing an architecture (a particular form for $E(W, Y, X)$); picking an inference algorithm for Y (find a value of Y that minimizes $E(W, Y, X)$ for any given X); picking a loss function (find $\mathcal{L}(W, S)$ using the training set) and picking an optimization method (find a W that minimizes the loss function over the family of energy functions ε, given the training set).

The central problem of energy-based learning is to design a good loss functions that will make the machine reach the desired behavior.

Generally, defining a loss function largely depends on the learning problem itself. A *good* loss function helps producing the correct answer by pulling up energies of the right points during the

learning phase. In our work, we adopt the concepts of energy-based learning to optimize the parameters when combining the language and vision models together. We define the loss functions based on the metrics used for evaluating the system, i.e., average ranking, which will be later explained in chapter 5. Since the number of parameters is low, we performed exhaustive search to find the best settings. This energy-based framework can allow more advanced learning in the future when integrating with other features in our framework. A more detailed explanation of our framework using energy-based model is described in section 5.6.

4

APPLICATIONS IN LANGUAGE DOMAIN

[1] In this chapter, we will present our knowledge-based framework in language domain, applied to the task of query classification. Section 4.1 defines the query classification problem, related work and our case study. Section 4.1 presents our solution for the query classification problem through query enrichment. Section 4.3 introduces the combination of query enrichment with a support vector machine classifier and section 4.4 reports our experimental results and discussion.

4.1 QUERY CLASSIFICATION

4.1.1 *Introduction*

In Information Science, transaction log analyses have been carried out since its early stages [Peters, 1993]. Within this tradition, lately, Query Classification (QC) to detect user web search intent has obtained interesting results [Shen et al., 2006a, Shen et al., 2006b, Li et al., 2008, Cao et al., 2009]. A QC system is required to automatically classify a large proportion of user's queries into a given target taxonomy: given a set of N unique classes, assign each query to one of those classes. Providing query classification can help the information providers understand users' needs based on the categories that the users are searching for. It will also bring improvement in general web search and online advertising.

A common challenge in QC comes from the nature of user queries, which are usually very short and ambiguous. Most queries often contain only several to a dozen of words. For example, as reported in [Beitzel et al., 2005] for around 5,000 queries, the average words per query is only 2.6. Due to this lack of information, simple matching techniques in Information Retrieval, like comparing overlapping words between queries and categories, do not work in this case. Moreover, one query can be classified to multiple categories and many words in queries are ambiguous. For example, the query "Michael Jordan" can be categorized to both "Sports" and "Computer Science", and the same word "security" in different context can have different categories (e.g., "national security" (Politics), "network security" (Information Technology), or "security market" (Finance)).

The second challenge in QC is the lack of training data. Classification task in general requires a consistent set of annotated data. Insufficient training data would lead to over-fitting or bias model [Li et al., 2008]. However, manually annotating a large set of user queries is very expensive given the fact that a target taxonomy can contain many different domains with dozens to a million categories (e.g., one of the largest online web directory, ODP - Open Directory Project, contains more than 1 million categories). This task becomes even infeasible as the new categories and web content evolve.

To overcome these challenges, we propose an enrichment of queries through a knowledge-based framework as briefly sketched in section 2.2.1. In particular, we extract knowledge from text collection via topic modeling to enrich the queries and categories. We also employ this enrichment process into

1 The material in this chapter is based on articles published in [Le et al., 2011], [Bernardi and Le, 2011], [Le and Bernardi, 2012]

building the training data for the query classifier. In brief, we will address the following research questions: (1) Does the query enrichment through topic modeling help in query classification? (2) How does the choice of text collections for enriching effect the performance of the query classification? (3) Can we exploit the enrichment process to build more informative training data?

In the next section, we will review the literature in the query classification task, then describe our case study.

4.1.2 *Related work*

Initial studies in query classification categorized queries to several different types. [Broder, 2002] considered three different types of queries: informational queries, non informational queries: navigational queries (ask for the site's address that the user wants to reach) or transactional queries (certain transactions that the user can perform, e.g., download, shop, find a map). In the later work, [Rose and Levinson, 2004] introduced a more complex taxonomy based on these three types by splitting each type to more detailed categories. This stream of studies focuses on the type of the queries, rather than topical classification of the queries.

Another stream of work deals with the problem of classifying queries into a more complex taxonomy containing different topics. Our study falls into this second stream. To classify queries considering their meaning, some work dealt with only information available in queries (e.g., [Beitzel et al., 2005] only used terms in queries with three methods: exact match, weighted automatic and rule-based classifier; their combination of the three methods resulted in an outperforming system). As queries are very short, some work has attempted to enrich queries with information from external online dataset, e.g., webpages [Shen et al., 2006a, Broder et al., 2007] and web directories [Shen et al., 2006b]. Our work is similar to them in the idea of exploiting external dataset. However, instead of using search engine as a way of collecting relevant documents, we use different text collections to extract general knowledge as will be discussed in section 4.2.

Context of a given query can provide useful information to determine its categories. Previous studies have confirmed the importance of search context in QC. [Cao et al., 2009] considered the context to be both previous queries within the same session and pages of the clicked urls. In our approach, we will also consider the click through information to enrich the queries and analyze topics.

Besides the different techniques, most studies [Shen et al., 2006a, Shen et al., 2006b, Li et al., 2008, Cao et al., 2009] have focused on the mapping of user queries to a general target taxonomy. However, little has been discussed about learning to classify queries in a specific domain. Our work, on the other hand, will focus on QC in a very specific domain: art, culture and history.

QC has also been a topic for a competition in the KDDCUP 2005 (the ACM knowledge discovery and data mining competition[2]). In our experiments, we adopted the evaluation metrics in this competition to judge our systems.

4.1.3 *Our case study: query log of an art image archive*

In our work, we use a query log taken from an image library, Bridgeman Art Library (BAL).[3] It is one of the world's top image libraries for art, culture and history. It contains images from over 8,000 collections and more than 29,000 artists, providing a central source of fine art for image users.

Works of art in the library have been annotated with titles and keywords (e.g., Table 4). Some of them are categorized into a two-level taxonomy, a more fine-grained classification of the Bridgeman browse menu. In our study, we do not use the image itself but only the information associated with it,

2 http://www.sigkdd.org/kddcup
3 http://www.bridgemanart.com

(-) Ancient and world cultures
- Greek, roman and etruscan
- Egyptian
- Asia
- Middle and near east
- Pre-history and europe
- Oceania
- Africa
- Americas

(-) Business and industry
- Money
- Banking
- Industry
- Shops and markets
- Trades and professions
- Agriculture
- Portraits of people in business and industry

(-) Religion and Belief
- Christianity old testament general
- Christianity old testament personalities
- Christianity new testament life of virgin
- Christianity new testament nativity madonna & holy family
- Christianity new testament life of christ
- Christianity parables / sacraments
- Islam / islamic / moslem / muslim
- Hinduism / hindu
- Buddhism / buddhist
- ...

Figure 15: The two level taxonomy of the Bridgeman art library

i.e., the title, keywords and categories. The taxonomy, catalogue and query log of the library will be explained below.

TAXONOMY The Bridgeman taxonomy has two levels, we use "top-category" and "sub-category" to refer to the first and second level, respectively. Each top-category can contains several sub-categories. A sample of the taxonomy is given in Figure 15. Totally, there are 289 top-categories and 1,148 sub-categories, with an average of ≈ 4 sub-categories for each top-category. Images in BAL are classified into sub-categories. The top-categories are divided into three main groups "Topic", "Object" and "Material".

CATALOGUE The Catalogue contains 324,232 images. For each image the metadata contains the title, a description, keywords and a sub-category from the taxonomy above, besides other information we are not going to consider in this work. The keyword field is for free-text terms (no controlled vocabulary is used), the terms provide physical descriptions, aspects of the image, like the color, shape or the object described, dates, conceptual terms, etc. An example is given in Table 4.

QUERY LOGS Query logs contain information about the queries (usually 1 to max. 5 words each) and the corresponding clicked images (i.e., the image that the user clicked after submitting the query). Via this clicked image, queries can be mapped to the information about the image provided in the metadata.

Title	A Section of the Passaic Class Single-Turret Ironclad Monitor (engraving)
Keywords	design, battleship, weapon, armoured, boat, submarine, warship, naval, cannon, ship;
Description	Transverse section of pilot-house and turret; The Passaic class, single- turret monitors of the U.S. Navy were enlarged versions of the original Monitor ships; the first Passaic was commissioned 5 November 1863;
Sub-category	Sea Battles

Table 4: An example of a metadata in the Bridgeman catalogue

Using the query log and the metadata of the library, our goal is to build a classifier that can automatically categorize each query to a corresponding top-category. In the next section, we will describe our query enrichment framework for building this classifier.

4.2 QUERY ENRICHMENT

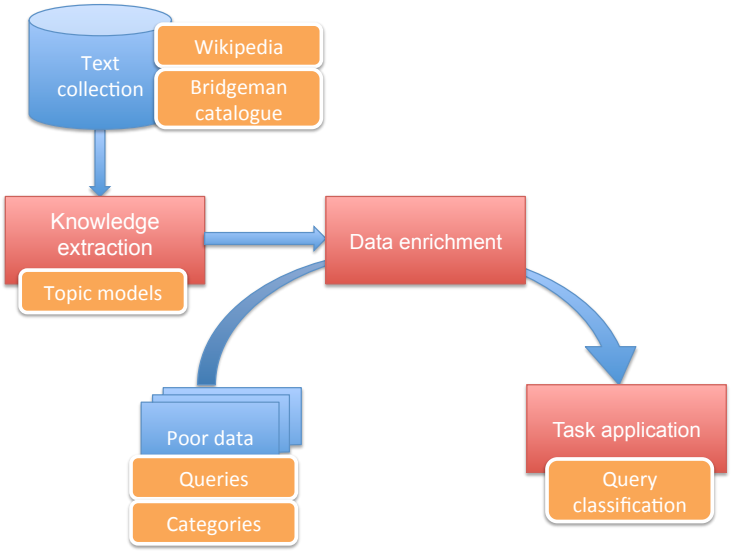


Figure 16: Query enrichment via topic modeling: our framework

Learning to classify queries in this domain, i.e., art, culture and history, is particularly challenging due to the specific vocabulary and the small amount of textual information associated with the images in the metadata.

Powerful classification systems perform badly when given as input just queries since queries are usually very short, the same user's intention can be expressed by different queries, and there can be a big semantic gap between the user queries and the categories in the taxonomy. To deal with these problems, we perform query and category enrichment using topic models. Our general framework for data enrichment in the query classification task through topic modeling is illustrated in Figure 16.

The key idea of this framework is to take advantage of external text collections (e.g., Wikipedia, library catalogue). To represent knowledge from these collections, we use topic models to estimate the data. This knowledge will be used as a reference set for enriching both queries and categories. The enriched queries and categories will help to increase the performance of the query classification

system. In this application, we also enrich queries with an extra information that is related to the information that the user clicks after performing a query, which will be described as the following.

QUERY ENRICHMENT WITH CLICK THROUGH INFORMATION Following the literature on web search classification [Cao et al., 2009], we exploit click-through information. Click-through information is the information coming from the images that a user clicks right after submitting a query. It is assumed that these clicked images are related to the meaning of the query that the user is searching for.

We enrich the query with the words from the title, description and keywords associated with the corresponding clicked image. Furthermore, we enrich the top-category of BAL taxononomy with its sub-categories (as explained in the previous section 4.1.3). We represent the enriched queries and enriched categories as vectors of words.

QUERY ENRICHMENT VIA TOPIC MODELS Beside using click through information to match between the queries and and the top-categories, we further exploit topic modeling to reduce the distance and capture their semantic similarity. The full enrichment process is sketched in Figure 17. First, the queries are enriched with their click-through information (titles, keywords and descriptions of images the user clicked); the top-categories are enriched with their sub-categories, as organized in the BAL taxonomy. Then, the topic model estimated from the text collection, which is composed of documents collected from Wikipedia or using metadata from the Bridgeman catalogue, will be used for analyzing topics for both queries and categories.

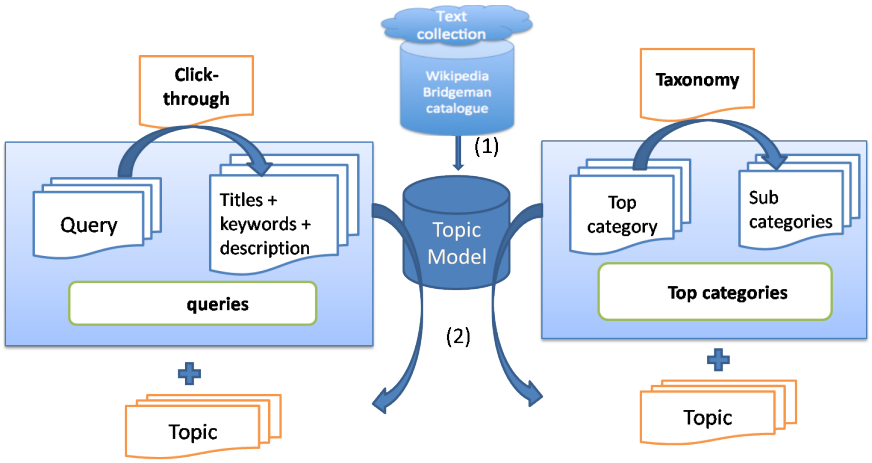


Figure 17: Enriching queries and categories: (1) Learning a TM from the text collection; (2) Enriching queries and categories with their topic labels

To analyze topics for both the queries and categories, we have to estimate a topic model from a large corpus. We estimate the multinomial observations in this model by unsupervised learning (as describe in chapter 3) with GibbsLDA++ toolkit.[4] Following the data enrichment approach in [Phan et al., 2010], we have enriched the query and category with hidden topic. In particular, given a probability $\vartheta_{m,k}$ of document m over topic k, the corresponding weight $w_{\text{topic}_k,m}$ is determined by discretizing $\vartheta_{m,k}$ using the two parameters *cut-off* and *scale*, which are defined as follow:

4 http://gibbslda.sourceforge.net/

The Arts and Entertainment, Ancient and World Cultures, Architecture, Business and Industry, Crafts and Design, Places, Science and Medicine History, Religion and Belief, Sport, People and Society, Travel and Transport, Plants and Animals Land and Sea, Emotions and Ideas

Table 5: The Bridgeman browse categories

$$w_{\text{topic}_k,m} = \begin{cases} round\left(scale \times \vartheta_{m,k}\right), \text{if} \\ \vartheta_{m,k} \geqslant cut\text{-}off \\ 0, \text{if } \vartheta_{m,k} < cut\text{-}off \end{cases} \tag{12}$$

We chose *cut-off* = 0.01, *scale* = 20 as to ensure that the number of topics assigned to a query/category does not exceed the number of original terms of that query/category, i.e., to keep a balance weight between topics enriched and original terms. Further experiments on optimizing the choices of these parameters have not been considered here. Generally, changing the values of these two parameters will change how much weight one wants to put on the topic model when classifying queries.

To discover the set of topics and the distribution of words per topic, we need to choose a text collection. Since we are interested in topics within a rather specific domain, we need to choose a data set that provides an appropriate vocabulary. In particular, we choose two different text collections: a general one and a more domain specific dataset. For the first option, we use a set of selected pages from Wikipedia to produce a general topic model. For the second option, we build a topic model directly from a set collected from the Bridgeman catalogue, which will be described in more detail below.

WIKIPEDIA TOPIC MODEL Wikipedia is a rich source of data that has been widely exploited to extract knowledge in many different domains. We have used a version of it, viz. WaCKypedia,[5] that contains around 3 million articles from Wikipedia segmented, normalized, POS-tagged and parsed. In order to reach a better model for our specific domain, we selected those pages that contain at least one content word of the BAL browse categories (Table 5).

For our vocabulary, we considered only words in the selected WaCKypedia pages that are either Nouns (N.*) or Verbs (VV.*) or Adjectives (J.*) after being lemmatized. We obtain $\approx$ 14K documents, with a vocabulary of $\approx$ 200K words, out of which we computed 100 topics. Examples of random topics and their top words are illustrated in Figure 18. It shows that topics derived from Wikipedia tends to have more topics of all general domains, from finance to sport, plant, war, etc., most of them are normal common terms.

BRIDGEMAN CATALOGUE TOPIC MODEL The most straightforward way of choosing a close domain corpus is to use the Bridgeman catalogue itself. We group together images that share the same sub-categories and consider each group of sub-category as a document. We have 732 documents and $\approx$ 136K words, out of which we computed 100 topics. Examples of topics estimated from this dataset are given in Figure 19.

5 WaCkypedia (http://wacky.sslmit.unibo.it/doku.php)

Topic 0	Topic 4	Topic 19	Topic 33	Topic 45	Topic 89
business	ship	sport	design	japan	plant
company	military	team	designer	japanese	cell
travel	war	world	intelligent	manga	soil
management	force	football	industrial	tokyo	specie
market	army	league	product	ainu	flower
service	navy	play	graphic	shogi	grow
sell	sea	event	interior	textbook	seed
financial	weapon	win	creative	osaka	tree

Figure 18: Hidden topics derived from WaCKypedia

Topic 3	Topic 15	Topic 21	Topic 45	Topic 59	Topic 81
railway	bc	christ	portrait	cotton	wedding
train	century	jesus	king	design	valentine
car	marble	crucifixion	queen	silk	bride
railroad	stone	cross	engraving	tapestry	marriage
carriage	bronze	life	charles	textile	baptism
locomotive	photo	supper	henry	printed	cotract
express	depicting	lord	prince	carpet	mariee
pacific	statue	holy	duke	wool	groom

Figure 19: Hidden topics derived from the Bridgeman catalogue

Let $Q = \{q_1, q_2, \ldots, q_N\}$ be a set of N queries and $C = \{c_1, c_2, \ldots, c_M\}$ a set of M categories. We represent each query q_i and category c_j as the vectors $\overrightarrow{q_i} = \{w_{tq_i}\}_{t \in V}$ and $\overrightarrow{c_j} = \{w_{tc_j}\}_{t \in V}$ where V is the vocabulary that contains all terms in the corpus and w_{tq_i}, w_{tc_i} are the frequency in q_i and c_j, respectively, of each term t in the vocabulary.

To classify each query to a corresponding category, we use matching and ranking. The similarity between each pair of query and category will be computed using cosine similarity. For each query q_i, the cosine similarity between every pair $\langle q_i, c_j \rangle_{j=1..M}$ is computed as:

$$cosin\_sim(q_i, c_j) = \frac{\overrightarrow{q_i}.\overrightarrow{c_j}}{|\overrightarrow{q_i}|.|\overrightarrow{c_j}|} =$$

$$= \frac{\sum_{t \in V} w_{tq_i}.w_{tc_j}}{\sqrt{\sum_{t \in V} w_{tq_i}^2} \cdot \sqrt{\sum_{t \in V} w_{tc_j}^2}}$$

For each query, the top 3 categories with highest cosine similarities are returned. Topics from the Bridgeman catalogue covers words that are more often used in the art, cultural and history domain, such as century (when an artwork was created), or material of sculptures such as marble, stone, bronze, etc. There are also topics that are more related to history such as portrait of kings, queen, charles, henry, prince, etc., which describe the galleries, collections of fine arts.

4.3 BUILDING A QUERY CLASSIFIER WITH TOPIC MODELS

In the above method, we find the best matching category for a given query by calculating their relatedness using cosine similarity, which does not require any training data. In this section, we employ a supervised learning model using training data automatically generated from query logs. As SVM classifier is quite robust and have been used successfully in many text classification problems, we use it to build our query classification. The training set contains queries and their corresponding categories. We enrich these queries with the information mined from the library via click-through links and the information collected from the library metadata via topic modeling as above. This enrichment process is done for both queries in the training set as well as the test set.

Manually annotating queries to create a training set in this domain is a difficult task (e.g., it requires the expert to search the query and look at the picture corresponding to the query, etc.). Therefore, we have automatically generated a training set by exploiting a 6-month query log as follow.

First, each query is mapped to their click-through information to find its corresponding sub-category using the library metadata. Then, from this sub-category, we map the queries again using the taxonomy to find its top-category (i.e., 55 target categories). The distribution of queries in different categories varies quite a lot among the 55 target categories (e.g., there are many more artworks in the library, hence queries, belonging to the category "Religion and Belief" than "Costume and Fashion"). Therefore, we have chosen 15,490 queries randomly by preserving their distribution over the target categories. After removing all punctuations and stop words, we obtained a training set containing 50,337 words in total. Each word in this set serves as a feature in the SVM classifier.

The test set contains 1,049 queries, which is separated from the training set. These queries have been manually annotated by an expert in the BAL (up to 3 categories per query). Note that these queries have also been selected automatically while preserving the distribution over the target categories observed in the 6-month query log.

4.4 EXPERIMENTS AND RESULTS

In this section, we will first describe the datasets that are used in our experiment. Then we present the results of various settings to evaluate the importance of click-through information, the effect of topic enrichment and a comparison between the matching-ranking method (section 4.2) and learning-based method (section 4.3).

4.4.1 *Datasets*

CATEGORIES From a BAL six month log, we extracted all the top category connected to the queries via the click-through information and obtained the list of categories given by group in Table 6.

QUERIES From the six month log we have extracted a sample of 1,049 queries by preserving the distribution of queries per top-category obtained via the click-through information and the taxonomy. We selected only queries with at least one clicked image. Not all image metadata contains title, keywords and a description: for around 60% images the meta-data provides only the title and the sub-category. For each query, we kept only one clicked image randomly selected. We leave for future study the impact the full set of clicked images per query could have on our query classifier.

GOLD-STANDARD: ANNOTATION BY DOMAIN EXPERTS The 1,049 queries have been annotated by a domain expert who was asked to assign up to three categories per query out of the 55

Topics	Land and Sea; Places; Religion and Belief; Ancient and World Cultures; Mythology Mythological Myth; Allegory/Allegorical; People and Society; Sports and Leisure; History; Travel and Transport; Personalities; Business and Industry; Costume & Fashion; Plants and Animals; Botanical; Animals; The Arts and Entertainment; Emotions and Ideas; Science and Medicine; Science; Medicine; Architecture; Photography.
Materials	Metalwork; Silver, Gold & Silver Gilt; Lacquer & Japanning; Enamels; Semi-precious Stones; Bone, Ivory & Shellwork; Glass; Stained Glass; Textiles; Ceramics.
Objects	Crafts and Design; Manuscripts; Maps; Ephemera; Posters; Magazines; Choir Books; Cards & Postcards; Sculpture; Clocks, Watches, Barometers & Sundials; Oriental Miniatures; Furniture; Arms, Armour & Miltaria; Objects de Vertu; Trade Emblems, City Crests, Coats of Arms; Coins & Medals; Icons; Mosaics; Inventions; Jewellery; Juvenilia/Children's Toys & Games; Lighting;

Table 6: Categories used by the annotators

categories in Table 6 and to mark the query as "unknown" if no category in the list was considered to be appropriate. The domain expert looked at the click-through information and the corresponding image to assign the categories to the query. The distribution of queries per group of categories obtained by this manual annotation is as following: 1395, 268, 87 queries have been annotated with a category out of the "topic", "object" and "material" group, respectively.

Out of this sample, 100 queries have been annotated by three annotators, BAL cataloguers, twice: (a) by looking at the click-through information and the image, and (b) by looking only at the query. The agreement between the annotators in both cases is moderate (kappa in average 0.60 for the annotation without click-through information and 0.64 for the annotation done using the click-through information), the agreement is higher for the categories within the "topic" group. For each annotator, using the click-through information and the image has not had a significant impact on the annotation of categories from the "topic" group (kappa in average 0.80), whereas it has increased and changed the annotation of categories from the other two groups, "object" (kappa 0.57) and "material" (kappa 0.62).

GOLD-STANDARD: AUTOMATIC EXTRACTION FROM THE META-DATA OF THE CLICKED IMAGE The top-category associated in the taxonomy with the sub-categories of the image clicked after querying can be extracted automatically exploiting the click-through information. Hence, we created a second gold-standard using such automatic extraction. Though our extraction is automatic, the assignment of the categories to the images is the result of the manual annotation by BAL cataloguers through the years. This annotation was done, of course, by looking only at the images, differently from the previous one for which the domain experts were given both the query and the clicked image. This second gold-standard differs from the one created by domain experts. For instance, the query "mountain lake near piedmont" is classified to the category "Places" by the expert, while using the automatic mapping method, we obtain the category "Emotions & Ideas: Peace & Relaxation". The kappa agreement between the manual annotation and the automatic extraction is 0.52, 0.53, 0.6 for categories within the "material", "object" and "topic" group, respectively.

In our experiment, we will evaluate the classifier against the "manual" gold-standard and use the second one only to select the most challenging queries (those queries the classifiers fail classifying in either cases: when evaluated against the manual or the automatic gold-standard) and analyze them in further detail.

4.4.2 *Evaluation metrics*

To evaluate the classifiers, first of all we compute Precision, Recall and F-measure as defined for the annual Data Mining and Knowledge Discovery competition KDD Cup and reported below.[6]

$$P = \frac{\sum_i \#\,\text{queries correctly tagged as } c_i}{\sum_i \#\,\text{queries tagged as } c_i} \tag{13}$$

$$R = \frac{\sum_i \#\,\text{queries correctly tagged as } c_i}{\sum_i \#\,\text{queries manually labeled as } c_i} \tag{14}$$

$$F-measure = \frac{2 \times P \times R}{P + R} \tag{15}$$

The F-measure average at KDD Cup competition was 0.24, with the best performing system reaching the result of 0.44 F-measure. Differently from our scenario, the KDD Cup task was for web search query classification against 67 general domain categories (like shopping, companies, cars etc.) and classifiers could assign max. 5 categories.

For a more intuitive evaluation, we also report the number of queries that are assigned to the correct category in each of the three positions (Hits # 1, 2, 3). Furthermore, we provide the total number of correct categories found in either position 1, 2 or 3 ($\sum_{Top\_3}$).

4.4.3 *Experimental settings and results*

In this section, we report our experimental settings and results to evaluate our knowledge-based framework, answering our three research questions: whether the query enrichment through topic modeling helps in query classification, how the choice of text collections effect the performance of the system and whether integrating the enrichment process to training data improve the results.

THE EFFECT OF THE ENRICHMENT PROCESS IN QUERY CLASSIFICATION In these experiments, we want to determine the contribution of the enrichment process in query classification. For this purpose, we compare a matching-based classifier with and without the enrichment process. In this matching-based method, each query will be matched to its most relevant category using matching and ranking as described in section 4.2. The different query and category enrichment methods are spelled out in Table 7.

We set up two different configurations: QR (query), where besides the terms contained in the top and sub-categories, V (vocabulary) consists of terms appearing in the queries; QR-CT (query-clickthrough) for which V consists also of terms in the title, keywords, description fields of the clicked images' meta-data. In the case of the classifiers exploiting topic models, both vocabulary is extended with the hidden topics too and both queries and categories are enriched with them as explained in Section 4.2. In particular, TM_{wiki} (topic model - wikipedia) is the classifier based on the model built with the hidden topics extracted from WaCKpedia, and TM_{BAL} (topic model - Bridgeman art library) is the one based on the model built out of Bridgeman metadata.

As can be seen in Table 9, the performance of query classification using only terms in the queries (QR) is very poor. Already enriching the query with the words from the title, keywords and description (QR-CT) increases the $\sum_{Top\_3}$ by nearly 120%.

We conclude that enriching both the queries and categories helps increasing the performance of the query classification task considerably. In the next part, to answer the second research question: "How does the choice of text collections for enriching effect the performance of the query classification?",

6 http://www.sigkdd.org/kddcup/index.php?section=2005&method=task

Setting	Query enrichment	Category enrichment
QR	q	CAT + sCAT
QR-CT	q + ct	CAT + sCAT
TM_{wiki}	q + ct $\oplus$ HT_{wiki}	CAT + sCAT $\oplus$ HT_{wiki}
TM_{BAL}	q + ct $\oplus$ HT_{BAL}	CAT + sCAT $\oplus$ HT_{BAL}

- q: query
- ct: click-through information: title, keywords and description - if available
- CAT: top category
- sCAT: all sub categories of the corresponding CAT
- HT_{wiki}: hidden topics from WaCKpedia
- HT_{BAL}: hidden topics from Bridgeman Metadata

Table 7: Experimental Setting

	Precision	Recall	F-measure
QR-CT	0.11	0.17	0.13
TM_{BAL}	0.26	0.40	0.31

Table 8: P, R and F measures – Evaluation

we will further enrich with topic models learnt from different text collections and discuss the choice of these collections.

THE CHOICE OF TEXT COLLECTIONS The purpose of this experiment is to compare the topic models built from a general text collection and a domain-specific dataset. As explained above, TM_{wiki} is the model built from Wikipedia and $TM_{BAL}$is the model built from Bridgeman metadata. The first model contains general terms since it was extracted from the external dataset. The second model contains more domain-specific terms since they are derived from the library catalogue itself.

As shown in Table 9, the TM estimated from Wikipedia (TM_{wiki}) did not help much in finding the right categories for a query. In comparison to QR-CT classifier, TM_{wiki} decreased the number of correct categories in position 1 and only slightly raised the number of correct categories when considering the three positions.

On the other hand, the TM built from the Bridgeman catalogue (TM_{BAL}) increased the results considerably for each of the three positions. Compared with QR-CT, 399 other correct categories were further found by using topics extracted from the catalogue, giving a raise of 117%.

Figure 20 reports the number of hits in each position 1, 2, 3 for the two settings QR-CT and TM_{BAL}. It clearly shows that TM_{BAL} outperforms QR-CT and matches more correct categories when considering both gold-standards.[7] It is interesting to note that this holds in particular for categories in the first position of the ranked list (Hits #1): it results in a raise of 92% in the first position (from 224 correct categories to 431).

In general, the result shows a positive contribution of the topic model learnt from the library metadata in query classification. It also confirms the importance of choosing relevant text collections for estimating this model: the model learnt from the Bridgeman catalogue outperforms the model learnt from selected pages of Wikipedia.

7 Manual gold-standard: queries are categorized by an expert directly; via-CT gold-standard: queries' categories are decided by the categories of the click-through images

Setting	Hits			
	# 1	# 2	# 3	$\sum_{Top\_3}$
QR	92	38	26	156
QR-CT	183	97	62	342
TM_{wiki}	145	112	88	345
TM_{BAL}	340	257	144	741

Table 9: Matching-based Classifier: number of correct categories found (for 1,049 queries)

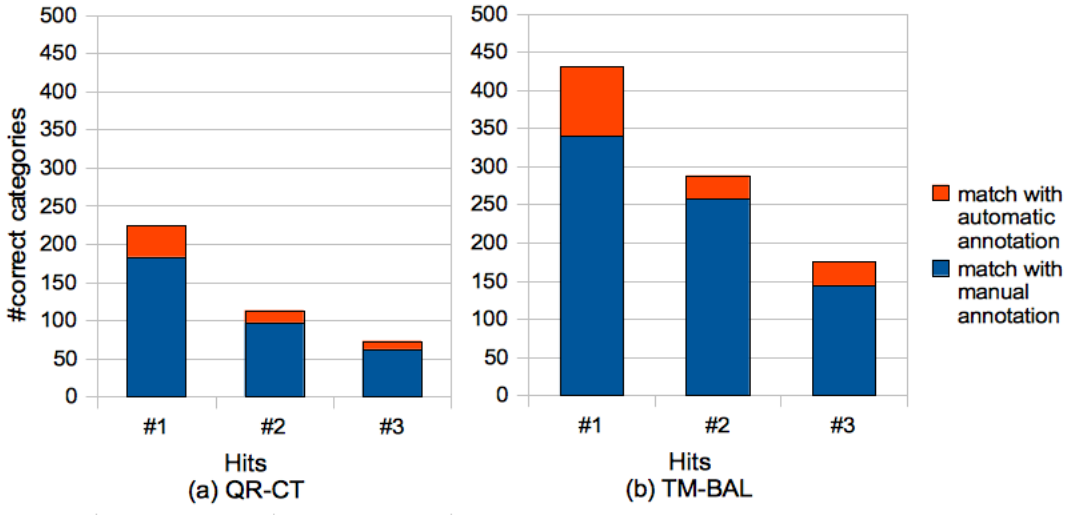


Figure 20: Matching QR-CT and TM_{BAL} correct categories against the manual and automatic gold-standards

ANALYSIS OF WRONG CLASSIFICATION To better understand the results obtained, we looked into the wrong classification. Figure 21 reports the number of queries for which QR-CT and TM_{BAL} have not selected in the top three positions any correct category using either the manual gold-standard and the automatic classification.

We found that there were 692 queries (422+270) for which QR-CT had not found any correct category in the top three positions; whereas 326 queries incorrectly classified by TM_{BAL}, of which 270 queries were in common with those wrongly classified by QR-CT.

We further analyzed the set of 270 queries of Figure 21 which we take to be the most difficult queries to classify since neither of the two classifiers have succeed with them considering either the manual or the automatic gold-standard. These queries and the categories assigned to them by the QR-CT and TM_{BAL} classifier have been checked and evaluated again by the domain expert.

Figure 22 gives an example out of the 270 and the result of the second run evaluation by the domain expert. The top categories assigned to the query "mountain lake near piedmont" by the classifier QR-CT and TM_{BAL} are "Ancient & World Cultures" and "Land & Sea", respectively. The two categories do not match either the correct category assigned by the expert ("Places") or the category assigned by the automatic method ("Emotions & Ideas"). However, after being checked by the expert, it was decided that the category proposed by the TM_{BAL} classifier ("Land & Sea") was also correct whereas the one assigned by QR-CT was not. This query and click-through information do not share any common words with the category "Land & Sea" and its sub-categories, hence it was not possible

Figure 21: Queries incorrectly classified

for the QR-CT classifier to spot their similarity. However, the enrichment with the hidden topics discovered the similarity between the query and the top-category: they share `topic 14` with high probability.

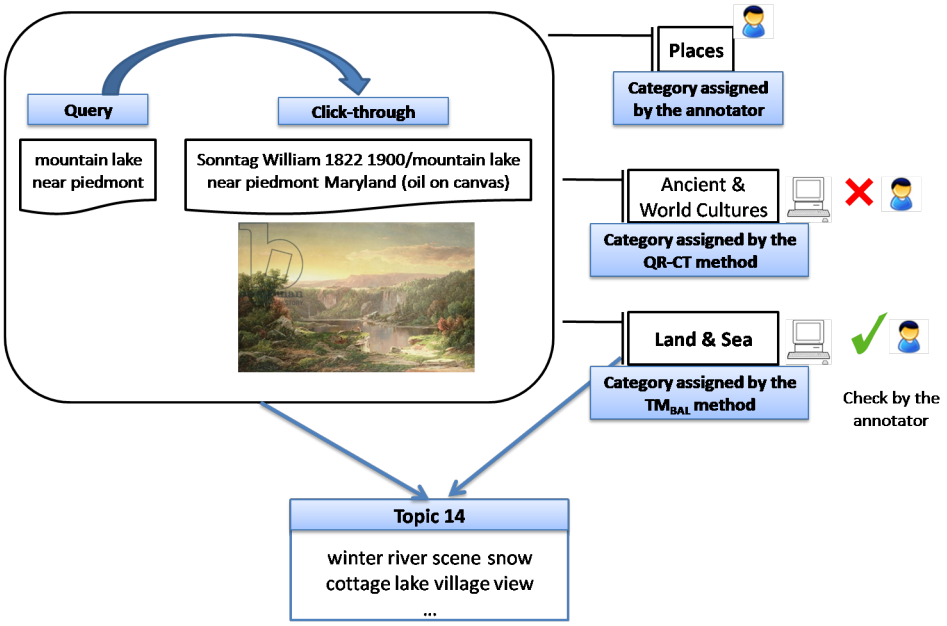


Figure 22: Effects of TM on the classification task

In total, the categories assigned to these 270 queries, were considered to be corrected in 123 cases for the TM_{BAL} classifier and in 45 cases for the QR-CT (Table 10).

Finally, we also measured the effects of click-through information in deciding categories for these 270 queries. In the first experiment, the expert looked at both the queries and their click-through images; in the second experiment, the expert only looked at the query to decide whether a category was correct or not. The result in Table 11 again suggested that the queries did not provide enough information to classify the queries since the number of correct categories decreased considerably.

Finally, the numbers of queries with at least one correct label out of these 270 queries are 39 (14%) for the QR-CT method and 115 (43%) for the TM_{BAL} method.

Setting	Hits			
	# 1	# 2	# 3	$\sum_{Top\_3}$
QR-CT	31	7	7	45
TM_{BAL}	59	43	21	123

Table 10: Correct categories checked by the expert for the 270 queries (using the click-through information)

Setting	Hits			
	# 1	# 2	# 3	$\sum_{Top\_3}$
QR-CT	20	9	8	37
TM_{BAL}	26	22	11	59

Table 11: Correct categories checked by the expert for the 270 queries (without looking at the click-through information)

LEARNING-BASED CLASSIFICATION: SVM The aim of these experiments is to evaluate the performance of the query classifier when integrating the enrichment process into the training data, our third research question. We want to examine whether enriching queries and categories during the training phase can increase the performance of the system or not.

We used the training set as described in section 4.3 to build a SVM classifier. In particular, with all queries enriched with click-through information and topics, we have 15,490 examples and 23,124 features in the training set.

Setting	Hits				
	Manual GS				*via-CT* GS
	# 1	# 2	# 3	$\sum_{Top\_3}$	
QR	207	80	24	**311**	**231**
QR-HT	212	81	25	**318**	**235**
QR - CT	243	107	38	**388**	**266**
QR - CT - HT	289	136	49	**474**	**323**

Table 12: Learning-based Classifier: number of correct categories found (for 1,049 queries)

The results of the experiment are reported in Table 12. As can be seen from the table, the click-through information plays an important role in our classifier. In particular, it increases the number of correct categories found from 311 to 388 (compared with the *manual* GS) and from 231 to 266 (using the *via-CT* GS).

It shows in Figure 23 the impact of the click-through information in the learning-based approach (svm) in comparison with the matching approach. Figure 24 shows the impact of the hidden topics in both cases. We can see that in both cases the learning-based classifier outperforms the matching-ranking one considerably (e.g., from 183 to 388 correct categories found in the QR-CT-HT method).

However, when we use only queries without click-through information, we can see that hidden topics do not bring a very good impact (the number of correct categories found only slightly increases by 7 - using the "manual" gold standard). The result might come from the fact that this topic model was built from the metadata, using only click-through information, but has not been learned from any queries coming from our query log dataset.

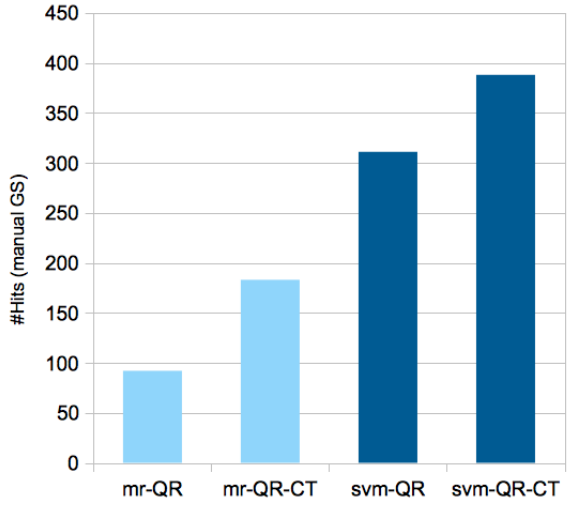


Figure 23: The impact of click-through information with matching-ranking (mr) and learning-based approach (svm)

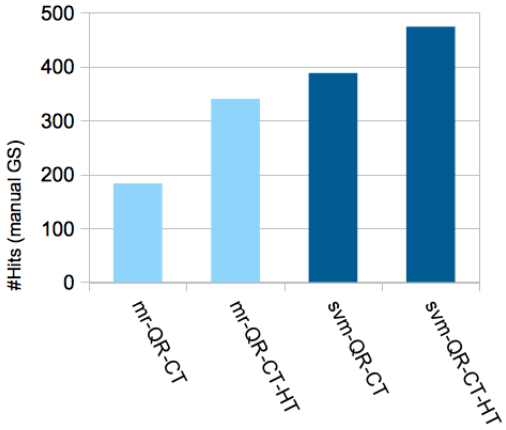


Figure 24: The impact of hidden topics with matching-ranking (mr) and learning-based approach (svm)

To sum up, the results of the experiments have confirmed (1) the contribution of the enrichment process through topic modeling in query classification, (2) the importance of choosing text collections with in-domain vocabularies for building the external dataset for enrichment (in our case study, the library data serves as the best text collection for extracting knowledge from), (3) that integrating the enrichment process to training data improved the performance of query classification considerably.

4.5 CHAPTER SUMMARY

In this chapter, we have presented our knowledge-based framework in the language application: query classification. The task is performed in a specific domain: art, culture and history. Since queries are usually very short, thus difficult to classify, we propose the enrichment of these queries with different

information sources: users' click through information and topic modeling. For building the topic models, we leveraged different text collections: a general universal dataset (using selected pages from Wikipedia) and a domain specific dataset (using the library catalogue). Our findings show the importance of choosing a domain-specific dataset as the text collection used for building topic models in this application. We hence propose the use of metadata of the online library as a source to train topic models for the query and category enrichment. Experiments from the real dataset extracted from the query logs have shown the impact of the click-through information and topic models built from the catalogue in helping to find the correct categories for a given query.

The experimental results also show that integrating hidden topics into an SVM classifier has improved the query classification results considerably. We have described our method of automatically creating a training set based on the selection of queries mapped to the click-through links and their corresponding available categories using a 6-month query log.

The results of this study have confirmed the usefulness of the data enrichment process based on analyzing a text collection in the query classification task in language domain. To further illustrate the flexibility and applicability of our framework in other domains, we are going to move to completely different tasks in vision domain in the next chapter.

5

APPLICATIONS IN VISION DOMAIN

[1] In this chapter, we apply our data enrichment framework to the vision domain. Section 5.1 introduces the problems of visual recognition in images and our approach. Section 5.2 reviews related work that uses language models for visual recognition and main approaches of human action recognition in images. Section 5.3 presents our general framework that exploits external datasets for visual recognition. Sectionsec:data introduces available and our annotated datasets that are used in our experiments. Section 5.5 and 5.6 discuss how we apply the framework to two different visual recognition tasks in computer vision: object prediction and human action recognition, respectively. Section 5.7 sums up our findings and results of the framework applied to the vision domain.

5.1 INTRODUCTION

The problem of visual recognition has always been central to research in computer vision. From recognizing objects such as tables, chairs, lamps, people in images, to classifying images to different scene categories (e.g., countryside, beach, mountain), and to understanding more complicated aspects of visual data, such as recognizing events and human actions in images. Many practical applications in image search, organization and robotics require visual recognition. The most common method for visual recognition in computer vision is using machine learning techniques, which often need a rich set of annotated images as training data. As the number of images and classes of objects, scenes and actions to be recognized increase, the annotation effort quickly becomes prohibitively expensive.

For object recognition, the current available training sets have increased up to thousands of object categories (e.g., the ImageNet large scale visual recognition challenge with 1,000 objects[2]). Although it requires an extensive annotation effort, it is still feasible to build such training data for object recognition. However, recognizing higher level concepts such as human actions is more complicated with many possible actions. Already in the simple case of a human action taken to be composed by a verb and an object, their number of actions grows polynomially as the number of verbs and objects increase. This combinatorial explosion of verb-object relations makes the task of learning human actions directly from their visual appearance computationally prohibitive and makes the collection of proper-sized image datasets infeasible.

To solve the problems of expensive image annotation in computer vision, we propose the use of language resources as an external dataset for extracting knowledge. Computational linguists have created many tools for automatic knowledge acquisition from available text resources. In our framework, we employ different text modeling techniques to outside the language domain, i.e., to the vision domain. The main idea of our approach is to use language as a resource for high level knowledge to guide the visual recognition. Next, we will first review related work and then describe

1 The material in this chapter is based on articles published in [Le et al., 2013a], [Le et al., 2013b], [Le et al., 2014]
2 http://www.image-net.org/challenges/LSVRC/2013/

our general framework in detail, the application of this framework to different visual recognition tasks, namely object prediction, scene prediction and human action recognition.

5.2 RELATED WORK

In this section, we will review the literature about using language models for visual recognition and main approaches in human action recognition in images.

5.2.1 *Using language knowledge to aid visual recognition*

Combining text and image processing has recently received an increased interest in both natural language processing and computer vision communities. One of the common task in image processing is to provide a description for each image. To produce such descriptions, a language model is often used to generate sentences for image annotation. [Yang et al., 2011] took an image description to consist of a noun, a verb, a scene, and a preposition and aim to generate a never seen descriptive sentence for a given image. They combined object and scene detectors from computer vision with language models extracted from a dependency parsed corpus to compute the probability of the action and of the preposition to be associated with the image. In particular, they defined their vocabulary to consist of verbs, nouns, locations, and prepositions. They selected the most likely description by calculating probabilities from co-occurrence statistics from a subset of the Gigaword corpus [Graff and Cieri, 2003]. This work is similar to ours in the idea of using statistics from text corpus. However, instead of using the statistics to produce descriptions for images, we use them to help predict objects and recognize human actions in images. In [Farhadi et al., 2010b], the meaning of images is represented through object-verb-scene triples. A triple works as an intermediate representation of images and descriptive sentences. In our work, we also define a human action to be composed of a verb and an object; and scenes are used to help disambiguating actions. [Ushiku et al., 2012] attempted to generate sentences for images by using an online learning method for multi-keyphrase estimation using a grammar model. [Kulkarni et al., 2011a] also generated image descriptions by employing object recognizers, detecting modifiers (adjectives) and spatial relationships (prepositions) in an image, then integrated the detections with a statistical prior obtained from descriptive text. The sentence generation was done using either a n-gram language model or a template based method.

Beside generating descriptions for images, text processing techniques have also been employed to provide high level knowledge for the aiding concept detection in images. Concepts such as objects, scenes, events, actions are usually detected in images using visual features, which are based on extracting edges, colors, textures, motions, etc. of the images. Such features allow computers to know how an object would look like, but not to understand the deeper semantic meanings of the images. For example, with only visual features, computers will not understand that boats often appear in water, but not inside a room. Language can be used as a resource of high level knowledge that guide the recognition process in images. [Yu et al., 2011] provided a framework for scene recognition combining visual recognition and language information. They used an object detector and a scene recognizer, then employed ontological knowledge about relationship between objects and scenes, which could be obtained from a text corpus. Such knowledge was used to propose attentional instructions for the visual recognizers, so that every detected object should maximize the added information for scene recognition. Also using language resources, [Srikanth et al., 2005] aimed at providing automatic image annotation. They used ontologies, in particular the hypernym relations in WordNet, to help defining visual vocabularies for images and to build hierarchical models for automatic annotation. [Teo et al., 2012] used language to resolve ambiguities arising from recognizing actions in cooking videos. First a language model was trained on a large text corpus, then detected

tools (e.g., knife, spoon) were queried into the language model, and the model returned predictions of actions (e.g., cut, pour). Finally, these predictions were compared with action features extracted from the videos, and the beliefs on the action labels were updated. This process stopped when the beliefs was maximized over tools and action features. Similarly, [Yang et al., 2013] learned the co-occurrences of any two words from a text corpus to obtain probability distributions, for example, how likely a noun co-occurs with a certain verb. Robots were "taught" this knowledge to resolve the uncertainty in recognizing objects, human actions in video tracking. The authors also noted that a pre-defined domain corpus was necessary to produce statistics that were relevant to the video content. We also employ such probability distributions in our framework for predicting objects and recognizing actions. Instead of using them to resolve uncertainties, we use them to suggest all possible objects or actions in images. Beside using language knowledge to aid visual recognition, there are also studies that attempt to build multimodal model by combining the knowledge extracted from both language and vision. For example, [Bruni et al., 2012] compared visual models and text models in semantic tasks such as semantic relatedness of words. They showed that visual and textual information capture different aspects of meaning and combining both of them in multimodal models can bring an improvement in performance. Instead of enriching word representations with visual information, [Lazaridou et al., 2014] on the other hand focused on the task of mapping an image of a previously unseen object to its linguistic representation word using existing knowledge to learn about new concepts. Given a new object, they first obtained a visual presentation of it. Then a neutral network is trained to project the image-extracted feature vector presentation of that object to text-based vectors using a cross-modal semantic space. Finally, the model suggested possible words to denote the given object. Generally, they combined the prior linguistic with visual knowledge to learn to associate new objects with words using the cross-modal semantic space and showed that the semantic space can provide sensible guesses of words to denote new objects in images.

5.2.2 *Human action recognition in images*

Initial work in human action recognition focuses on labeling videos consisting of human motions with different actions [Liu et al., 2011, Desai and Ramanan, 2012, Gorelick et al., 2005, Weinland et al., 2006]. Most studies extract image features taken from the videos and assign a segmented sequence of images into one or more predefined human actions. Only recently, there have been more attention in recognizing human actions in still images. First of all, many common human actions such as "riding a horse", "sitting on a chair" can be recognized in a single image without looking at the whole videos. Secondly, while videos contain a lot of information and can be extremely useful, images are still used very widely on the Internet. Not only images occupy less space, are faster to process, but also are more compact, have less distracting elements than videos. When people are overloaded with information on the Web, images are easier to catch their attention since they are instant. Videos take time to be watched and to get the information inside them. Images capture moments, moments are important and the rest is noise. Given such a large portion of images online, there are many potential applications related to labeling images with human actions. Following, we will review different methods to the human action recognition in images.

USING HUMAN POSES The research in human action recognition in still images started in around 2006. Early studies in this topic focus on extracting human shapes and recognize actions mainly based on their poses. One of the first work in 2006, [Wang et al., 2006], attempted to cluster images that have similar human poses together, then manually selected meaningful clusters and assigned them to corresponding human actions. They used the coarse shape of the human figures to make pairwise comparisons between images, then applied spectral clustering to form groups of human

actions. The method has been tested on sport datasets (figure skaters, baseballs and basketballs) and the actions include throwing, swinging, and sliding, etc.

In 2008, [Ikizler et al., 2008] classified human actions in images by parsing human poses and representing them using circular Histogram of Rectangles. Their method was applied to recognize six different human actions: running, kicking, walking, catching, throwing and crouching. Also using human shapes, [Yang et al., 2010] treated the pose information as latent variables in the action recognition model. The human pose is composed of four different parts: upper-body, legs, left-arm and right-arm. The aim of the model is to localize the body parts that are useful for action classification.

USING OBJECT INFORMATION Beside recognizing human actions using poses, many studies [Gupta et al., 2009a, Yao and Fei-Fei, 2010, Yao and Li, 2010, Yao and Fei-Fei, 2012, Desai and Ramanan, 2012, Sener et al., 2012] have also addressed the problem of recognition of interactions between human and objects in images. If the above work focuses on human actions such as running, walking, throwing, recent studies have included the information of objects in defining human actions such as riding a bike, walking a dog. [Gupta et al., 2009a] used object detection and built spatial and functional constraints to find locations where objects are more likely to appear. The location of the object in the human object interaction is constrained by the human location and human pose. Similarly, in [Yao and Fei-Fei, 2010], the authors used objects and human poses as mutual context to each other. They employed an object detector and used pose estimation to improve the results of the object detection. A random field model was used to learn important connectivity patterns between objects and human body parts. Also focusing on human object interactions, [Yao and Li, 2010] introduced an image feature representation called "grouplet", which encoded appearance, shape, and spatial relations of multiple image patches. These grouplets were used as structured information that characterized human and object interactions. Experiments were run on images of people playing seven different musical instruments, with the aim of classification of playing different instruments and discriminating playing from not playing. Also modeling human poses together with interacting objects, [Desai and Ramanan, 2012] further addressed the importance of modeling occlusion in recognizing human object interactions. Due to different viewpoints, many parts of the human may not be visible in images. Such occlusions can generate various changes in appearance, which makes it difficult to recognize human actions. They employed compositional models local visual interactions and their relations, labeling each image with human actions, articulated poses, object poses and occlusion flags. [Delaitre et al., 2011] introduced person-object interaction features based on spatial co-occurrences of individual body parts and objects.

USING CONTEXTS Beside human poses and objects, many studies also considered the importance of context such as scenes, events in defining human actions in images. In [Gupta et al., 2009a], the authors used four different contextual information to recognize human actions: scenes (e.g., cricket ground, tennis court), scene objects (objects that are in the scene and not manipulated by human such as net), manipulable objects (a ball, racket), and human. Similarly, [Delaitre et al., 2010] investigated the role of background scene context in action recognition, integrated the scene context with person-centric description to improve the results. Generally, scenes can provide good information in discriminating different human actions, but they can also have negative effects, particularly when the background is noisy and for actions that can happen in similar backgrounds. Beside scenes, [Khan et al., 2013] proposed the combination of shapes and colors to boost the performance of action classification. They used some selection of the color descriptor and optimized fusion strategy to integrate the color information into the action recognizer. They also noted that a naive combination of colors and shapes may negatively affect the recognition performance.

USING HUMAN-OBJECT LOCATIONS The relative positions between human and objects are crucial in recognizing human object interactions. [Desai et al., 2010] defined the relations including "above", "below", "overlapping", "next-to", "near", and "far" to capture the spatial interactions between human bodies and objects. They integrated object detections with their relative pose and spatial locations as a set of latent variables, which were maximized over during testing. Similarly, considering the locations of humans and objects in their interactions, [Gupta et al., 2009a] introduced two kinds of spatial constraints: connectivity (e.g., manipulable objects like tennis racket should be connected to human) and positional - directional constraints (e.g., a ball should be near the human in a tennis-serve). The positional constraints were learned by modeling positional locations using the object and human body centroid. Similarly, [Prest et al., 2012] considered the relative positions between humans and objects in recognizing human actions. They detected the most prominent human in a given image, then given the detected human, found the best fitting location for the action object.

RECOGNIZING HUMAN ACTIONS BY COMPONENTS Human object interactions can be defined as pairs of verbs and objects, for example: *(ride-horse), (play-guitar), (fix-bike)*. This leads to a large number of possible human actions which are combinations of verbs and objects. Depending on the kind of objects, some verbs are suitable (e.g., *feed-horse*) and some are not (e.g., *feed-bike*). Some verbs such as *ride* are more likely to co-occur with particular objects such as *bike, horse*, but not with objects such as *table*. To learn such relationship, it is important to recognize human actions compositionally. [Yao et al., 2011a] defined human actions using attributes (verbs) and parts (objects and poselets). They jointly modeled attributes and parts by learning a set of sparse bases and reconstructed an action image by a set of sparse coefficients with respect to the bases.

USING EXTERNAL DATASETS As the number of possible human actions is very large, it is expensive to collect enough training images for each human action appearing in images. Using external knowledge to recognize unbound number of human actions is therefore very useful. [Ikizler-Cinbis et al., 2009] presented a framework for learning action representation automatically from the web. Their ideas were to query the web to collect action images and build this collection incrementally. Only images with humans detected were selected, then non-relevant images were further removed using a logistic regression classifier. Once a cleaner image dataset for each human action was built, they learned an action classifier, which included different viewpoints from various sources.

In our work, we have not considered the human poses in recognizing human actions, but used object recognition, scene information, human-object locations and knowledge learned from external datasets to recognize human actions in images. In the next section, we will present our general framework that combines these components for human action recognition.

5.3 OUR GENERAL FRAMEWORK FOR VISUAL RECOGNITION

The general framework for visual recognition has been briefly introduced in section 2.2.2 (Figure 25). The underlying idea is to extract knowledge from a text collection through text modeling or a database to enrich images. We focus on two visual recognition tasks, which involve objects, scenes and human action (e.g., ride a horse, fix a bike) in images. A crucial step of this framework is to extract relations among those objects, scenes and actions from language models. Each components of the framework will be explained below.

TEXT COLLECTION As mentioned earlier in Chapter 2, the text collection can contain raw text collected across the web such as Wikipedia, newswires, online articles, blogs. Such text collection can provide general knowledge about various domains, topics. The advantage of using this data is

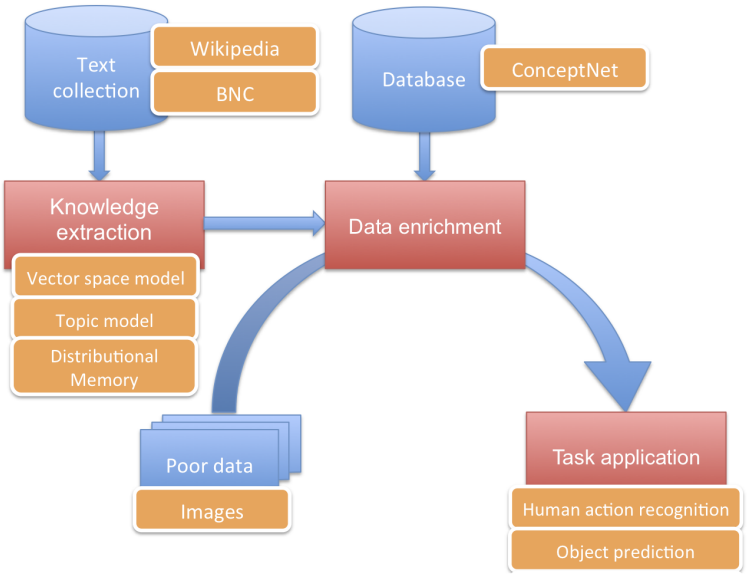


Figure 25: Applications in vision domain: human action recognition, object prediction

that it is easy to collect and available to download with many topics and different genres. However, as it is raw, unstructured and contains noise, this data needs to be modeled and organized in a structural way. Therefore, text modeling techniques are employed to make this text collection more valuable and easy to use.

KNOWLEDGE EXTRACTION To represent and extract knowledge from text collections, different kinds of text modeling techniques can be used such as vector space model, topic model, distributional memory. Details about these techniques are discussed in Chapter 3. Depending on which knowledge and information to extract, one can decide which techniques are useful and work best in each case. In our work, we will evaluate various models for different tasks and determine which language models provide the best knowledge for each case. A thorough comparison and discussion about which models fit best in which case will be given in section 5.6.6.

DATABASES In order to make raw text collections more useful for linguistic research, an annotation process is usually applied such as part-of-speech tagging, lemma base form of words, sentence parsing, etc. Further processing steps are performed to extract for example commonsense knowledge using rules (e.g., ConceptNet). Such processed text collection is often organized and stored in a database for different purposes. For example, as mentioned in Chapter 3 lexical databases that represent the relation among words that are commonly used are WordNet, FrameNet. Such databases can also be used to enrich images for different visual recognition tasks. Later on, we employ ConceptNet to extract information for data enrichment.

DATA ENRICHMENT AND TASK APPLICATION Images with visual features extracted will be enriched with the information learned from text collections and databases. This information will be used to guide the recognition process to predict the most plausible objects/scenes or actions that are present in the image. In particular, we will focus on the two visual scenarios: objects in context and human action recognition in still images. In the first task, we want to predict the most likely

identity of an object given its context as expressed in terms of co-occurring objects. In the second task, we perform human action recognition, defined as a verb-object pair where the human is implicit in the verb, based on objects and scenes recognized in images. For the first task, we want to learn the relationship between objects and objects to see how often they would occur together in texts. For the second task, the relationships between verbs, objects and scenes are extracted from text to help with the prediction of a human action in an image. More details of the word relation extraction will be explained in Section 5.3.2.

5.3.1 *Visual recognizers*

In this section, we will describe the general method for obtaining visual features from images for our specific tasks. In particular, there are two visual recognizers that will be built based on training images: an object recognizer and a scene recognizer, which are both built upon [Uijlings et al., 2013]. The outputs of these recognizers will be further combined with the information obtained from language models to predict the appearance of an object, scene and human action in an image.

OBJECT RECOGNIZER We use two object localization systems: The method of Felzenszwalb et al. [Felzenszwalb et al., 2010] and the method of van de Sande et al. [van de Sande et al., 2011]. We do not want to base our object recognition on a global image impression, such as the common image-based Bag-of-Words representation, as an action is an interaction between a human and an object and less dependent on its surroundings.

The two object localization systems [Felzenszwalb et al., 2010, van de Sande et al., 2011] differ in representation but share similarities in training: Both need as training data images where the objects have been annotated using bounding boxes. In both methods, negative examples are automatically obtained from the training data by finding so-called hard examples: image windows that yield high object probabilities but do not correspond to the object. Given an image, both systems predict the most likely bounding boxes where a specific object o^i is present, together with its probability $P(o^i|I)$.

The part-based method of Felzenszwalb et al. [Felzenszwalb et al., 2010] is based on a sliding window approach and Histogram of Oriented Gradient or HOG-features [Dalal and Triggs, 2005]. For each object class the method automatically determines several poses. For each pose HOG-templates are learned for the complete object and for object-parts, the latter which are automatically determined using a latent, linear SVM. During testing, the HOG-templates are applied to a dense, regular search grid within the image. Locations with the highest template response for both parts and the complete object yield a predicted location with corresponding probability. The framework is widely used and in our experiments, we use their publicly available code (see [Felzenszwalb et al., 2010]).

The method of [van de Sande et al., 2011] is based on the Bag-of-Words (BoW) paradigm [Csurka et al., 2004]. In common BoW, SIFT-descriptors [Lowe, 2004] or variants are extracted from the image on a densely sampled grid. Using a previously learned visual vocabulary (e.g. created by kmeans) each SIFT desciptor is assigned to a specific visual word. The BoW representation is given by a histogram of visual word counts within the image, often using the Spatial Pyramid [Lazebnik et al., 2006a] which regularly divides the image to introduce a rough form of spatial consistency.

In [van de Sande et al., 2011], the authors propose to represent not a complete image but only the object using BoW. However, such representation would be computationally too expensive within a sliding window approach which visits over 100,000 locations. Therefore the authors propose Selective Search which uses multiple hierarchical segmentations to generate around 1500 high-quality, class independent, object locations. The BoW representation for these 1500 locations can be generated within reasonable time. In this paper we model the BoW based localization method after [van de Sande et al., 2011], using the publicly available selective search code. The BoW implementation

LocatedNear		RelatedTo		UsedFor		AtLocation	
oil car	seatbelt car	horse zebra	plant garden	bottle store_liquid	horse race	bus city	car city
chair your_bottom	chair school	horse pony	sheep baa	boat fish	table eat_off_of	bike street	dog city
sheep grass	car car_lot	table chair	dog pet	bicycle transportation	bird watch	horse countryside	bird forest
plant everywhere	muzzle dog	plant green	sheep cloud	dog companionship	chair rest	bird countryside	dog street
trailer car	dog bark_bone	boat ship	cow bull	horse riding	bus travel	car street	chair city
seat chair	bird air	sheep goat	plant tree	car drive	dog keep_you_company	chair kitchen	horse city
salt table	horse cowboy	chair table	horse riding	chair sitting	table eat_meal	cat store	bus city
stool table	carriage horse	dog wolf	sheep farm	chair sit_on	boat travel	car street	chair store
sheep flock	dog human	horse donkey	sofa book	boat sail	dog pet	cow countryside	chair store
pasture cow	horse fence	dog cat	cow milk	car transportation	bottle hold_liquid	car street	bicycle store
cat dog	whisker cat	sheep lamb	table desk	sheep wool	boat float_on_water	bird forest	chair store
horse zebra	desk chair	sheep wool	cat feline	table put_thing_on	table eat_at	car city	bottle store
cat household	train railroad	dog a wolf	dog canine	boat travel_on_water	cat catch_mouse	table kitchen	chair office
plant nature	horse pull_carriage	horse mule	chair sitting	horse ride	cat pet	bicycle street	car street
horsehair horse	sheep wool	cat dog	plant flower	chair sit	cow milk	chair office	chair city

Table 13: Examples of relations extracted from ConceptNet 5

itself is modelled after the fast implementation proposed by [Uijlings et al., 2010]. We denote this localization system as BoWL.

The details of the implementation are as follows. First, we extract SIFT descriptors [Lowe, 2004] and two colour variants, RGB-SIFT and Opponent SIFT [van de Sande et al., 2011] at every single pixel in the image (ultra-dense). We use a single scale of 16 by 16 pixels and a Gaussian derivative filter with sigma = 0.667. Principal Component Analysis is used on the descriptors to reduce their dimensionality by a factor 3. Then each descriptor is assigned to a visual word using a Random Forest based visual vocabulary [Moosmann et al., 2006, Uijlings et al., 2010], which is as accurate as the usual k-means clustering yet is much more computationally efficient. Specifically, we use four trees of depth ten, resulting in 4096 visual words per SIFT variant. The trees are learned beforehand on a random subset of all descriptors in the training set using the global image labels. The visual words and their locations are stored to be able to quickly compute visual word histograms from subregions within an image.

SCENE RECOGNIZER To recognize scenes in images, we employ the Bag-of-visual-word method using the same settings as described above. In localization, we use the visual words over the complete image to create the final BoW representation. However, here we use a spatial pyramid which uses the whole image and divides the image into three horizontal regions, roughly dividing the image into floor, objects and sky. Note that this representation describes the image as a whole. A support vector machine classifier is used to learn the scene recognizer.

Finally, for both object and scene classifiers, the Platt's sigmoid function is used to obtain the final conditional probability of an object given an image $P(O|I)$ and a scene given an image $P(S|I)$, where I refers to the variable image and O, S refers to object and scene, respectively.

5.3.2 *Distribution extraction from text collections and databases*

To enrich the images with general knowledge information, we learn the relations among those recognized components (i.e., objects, scenes) with human actions by exploiting the language models. To extract these probability distributions from language data, we use a commonsense knowledge database ConceptNet, and different text models, namely the Window2 and 20 model, TypeDM and R-LDA (see more in Chapter 3). The aim of this step is to estimate the relationship among objects (O), scene (S) and human action (V) in language. It is based on the observation that in each image, the appearance of an object, scene and human action are related. For example, if there is a person with the location on top of a chair in an office, it is very likely that the person is sitting on the chair.

Knowing this relationship will help better recognize the correct action in the image. In our model, we consider the relationship between each pair of verb and object, verb and scene, and object and scene. Therefore, the four conditional probabilities needed to be estimated are the probability of a verb given an object, a verb given a scene, an object given a scene and an object given another object: $P(V|O), P(V|S), P(O|S), P(O|O)$.

5.3.3 *ConceptNet*

ConceptNet [Speer and Havasi, 2013], as presented in Chapter 3, is a large semantic graph containing concepts and relations between them. It includes everyday basic, cultural and scientific knowledge, which have been automatically extracted from Internet using predefined rules. As ConceptNet contains a large number of concepts with words and phrases of natural language, we use ConceptNet 5 in our framework. In principle, other commonsense knowledge or lexical databases such as WordNet or FrameNet could also be used here although one needs to tailor it to the specific task in hand. As ConceptNet was mined from free text using rules, the database has uncontrolled vocabulary and contains also some false/nonesense statements.

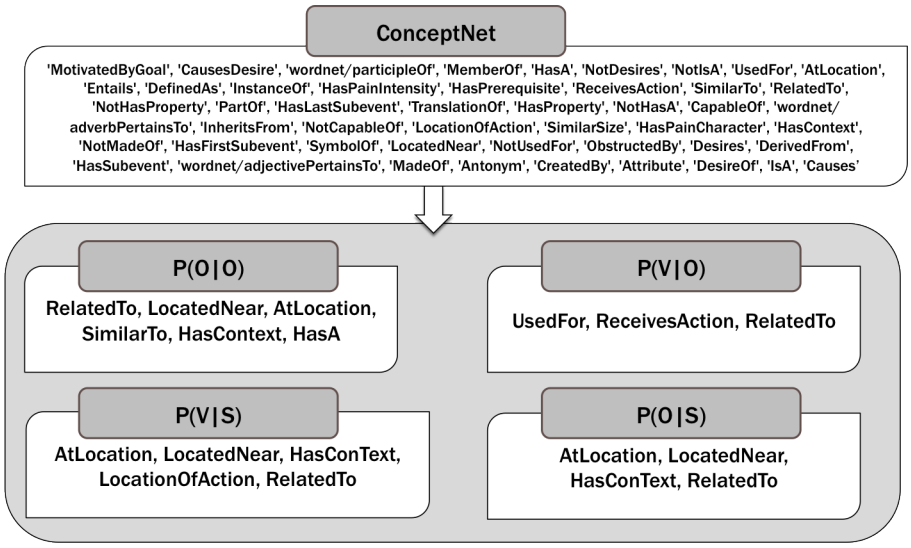


Figure 26: List of relations in ConceptNet

To extract relations from ConceptNet5, we first examine all relations in the database and define those that are relevant to our scenarios (Figure 26). For example, for the conditional probability of objects given scenes, relations such as "At Location", "Located Near" are extracted.

Examples of relations extracted from ConceptNet are illustrated in Table 13, such as: Oil - Located near - Car, Horse - Related to - Zebra. From these relations, we define the four conditional probabilities using their frequency counts, i.e., how many times they occur together in the ConceptNet database through some relations. For example, to compute the conditional probability of an object given a scene $P(o_i|s_j)$, we extract all triples having the form <object, rel, scene>, where "rel" can be "AtLocation", "LocatedNear", etc.

$$P(o_i|s_j) = \frac{freq(< o_i, rel, s_j >)}{\sum_{o_m \in \mathcal{O}} freq< o_m, rel, s_j >} \tag{16}$$

5.3.4 *Window model*

One of the most famous and basic statistical model is based on counting co-occurrences of two words in a corpus within a window of fixed width, which follows the tradition of hyperspace analogue to language [Lund and Burgess, 1996b]. We took the Window 2 and Window 20 models which have been built in [Bruni et al., 2012] using the ukWaC (1.9B tokens) and Wackypedia (820M tokens). As the Window 2 model only looks at 2 words on the left and right of the current one, it reflects the relationships between words occurring near each other, while the Window 20 searches for a broader view of how words are related to each other. The weights of each pairs of words are calculated using the Local Mutual Information (LMI). To compute the conditional probabilities, we use the LMI scoring function provided by the models:

$$P(v_i|o_j) = \frac{LMI_{v_i,o_j}}{\sum_{v_m \in \mathcal{V}} LMI_{v_m,o_j}} \tag{17}$$

$$P(v_i|s_k) = \frac{LMI_{v_i,s_k}}{\sum_{v_m \in \mathcal{V}} LMI_{v_m,s_k}} \tag{18}$$

$$P(o_j|s_k) = \frac{LMI_{o_j,s_k}}{\sum_{o_n \in \mathcal{O}} LMI_{o_n,s_k}} \tag{19}$$

$$P(o_m|o_n) = \frac{LMI_{o_m,o_n}}{\sum_{o_p \in \mathcal{O}} LMI_{o_p,o_n}} \tag{20}$$

5.3.5 *Distributional Memory*

Distributional Memory [Baroni and Lenci, 2010] (DM) is a multi-purpose framework for semantic modeling. This model is more complex than the Window models because it exploits different degrees of lexicalization for each relation (see more about this model in chapter 3). Distributional information is extracted as a set of weighted <word-link-word> tuples obtained from a dependency parse of corpora. In the Window model the relation between each word pair is decided by their co-occurences within a sliding window, while in DM this relation is defined by distributional properties of the two words. These distributional properties are based on a syntactic relation or lexico-syntactic pattern that links the two words. For example, the tuple <marine, use, bomb> encodes that *marine* co-occurs with *bomb* in the corpus, and the word *use* specifies the type of the syntagmatic link.

DM contains three different models, corresponding to different ways to construct the weighted structure through the "link". The first model, LexDM is the most heavily lexicalized model with the most variety of links, whereas the DepDM has the minimum degree of lexicalization, thus having the smallest number of links. TypeDM, which was reported to achieve the best performance in different tasks including selectional preferences, is laying somewhere in the middle of the other two models. It shares the same lexical information as in LexDM but use a different scoring function, which focuses on the variety of surface forms, rather than the frequency of a link. Hence we choose the best model, TypeDM, to learn the relationships between verbs, objects and scenes. As in the window model, we compute conditional probabilities using the LMI scores provided by the model (Equation 17, 18, 19, 20).

5.3.6 *R-LDA*

To model the relationships between verbs, objects and scenes, we adapt the R-LDA model [Séaghdha, 2010] as explained in chapter 3. The R-LDA model has been used for the selectional preference

Topic 0:				Topic 15:				Topic 28:				Topic 54:			
Noun		Object		Noun		Object		Noun		Object		Noun		Object	
attention	.0172	study	.01	child	.052	child	.06	job	.055	worker	.0201	decision	.02	case	.0176
model	.0147	research	.0123	family	.0286	home	.0217	work	.0278	job	.0163	view	.0244	fact	.0096
study	.014	work	.0111	parent	.0251	family	.015	worker	.021	work	.0143	question	.018	question	.0096
role	.0139	chapter	.0085	mother	.0224	life	.015	wage	.016	employer	.0118	issue	.0124	law	.0096
relationship	.0137	analysis	.008	year	.0158	baby	.0147	number	.014	year	.0114	case	.0115	policy	.009
account	.013	problem	.0075	woman	.0158	mother	.0144	employment	.013	union	.0108	evidence	.0104	decision	.0092
approach	.0131	development	.0075	home	.013	parent	.0127	right	.0124	woman	.0104	argument	.01	matter	.0085
analysis	.0123	issue	.006	time	.012	time	.0119	people	.0108	employment	.0094	point	.0099	time	.0067
aspect	.012	system	.0065	life	.0109	year	.008	time	.0099	company	.0088	reason	.0096	issue	.0062
problem	.0106	area	.006	age	.0087	school	.0085	employee	.0083	man	.0081	statement	.0086	evidence	.00617
development	.0104	change	.0058	people	.008	care	.0081	hour	.008	employee	.0077	fact	.007	party	.0058
pattern	.0103	policy	.0055	school	.008	lot	.0081	money	.0076	week	.0071	attention	.0076	point	.0058
issue	.0102	theory	.00551	wife	.0079	problem	.0075	service	.0073	staff	.007	matter	.00738	judge	.0055
range	.0095	way	.0053	day	.0057	contact	.0072	man	.007	trade	.007	policy	.006	statement	.005
area	.009	model	.0051	girl	.0053	right	.0068	action	.0065	wage	.0067	principle	.0066	argument	.0047
method	.0089	structure	.005	daughter	.00524	woman	.0066	skill	.0064	pay	.0063	authority	.0065	opinion	.0046

Topic 8:				Topic 14:				Topic 52:				Topic 99:			
Noun		Verb		Noun		Verb		Noun		Verb		Noun		Verb	
people	.0208	have	.157	year	.0154	win	.109	letter	.02	have	.102	time	.0554	have	.114
job	.0167	work	.108	Cup	.0093	have	.099	information	.0112	send	.067	day	.044	go	.086
work	.0156	make	.0262	team	.0086	beat	0	copy	.0094	receive	.037	night	.0246	come	.041
class	.014	take	.0244	race	.00772	take	.025	form	.0091	write	.0272	home	.0227	leave	.0283
woman	.013	employ	.022	title	.0067	go	.0218	number	.00768	get	.0251	hour	.0219	get	.026
man	.012	go	.0171	championship	.006	play	.0154	message	.0069	take	.0167	week	.016	work	.0199
staff	.0111	pay	.0156	time	.006	run	.0152	office	.0069	ask	.0167	minute	.0118	stay	.017
group	.0089	say	.0146	world	.0058	finish	.0135	detail	.0066	make	.0154	evening	.0106	arrive	.014
way	.0086	get	.0136	game	.0055	make	.0127	order	.00582	know	.0137	way	.0088	take	.0143
service	.008	leave	.0114	champion	.0053	lead	.0122	name	.0055	contact	.0127	school	.0086	tell	.012
year	.0081	know	.0109	victory	.005	come	.0112	card	.0054	return	.0116	year	.008	feel	.0124
company	.0076	run	.0102	seat	.0049	follow	.0098	address	.0049	see	.0111	month	.0072	return	.012
day	.0069	come	.0101	match	.0049	qualify	.008	time	.0047	please	.0099	room	.0066	think	.0115
number	.00615	help	.0088	place	.0047	compete	.0074	school	.0044	come	.009	bed	.0066	wait	.011
hour	.0061	join	.008	round	.0047	hold	.0072	telephone	.0044	sign	.0091	house	.0061	see	.0114
labour	.0061	become	.0075	winner	.0045	include	.007	London	.004	find	.0086	friend	.0052	say	.0104
business	.0055	set	.0072	yesterday	.00406	become	.0067	document	.0041	call	.0083	work	.00435	walk	.0097

Table 14: Random R-LDA topics with the relations between Noun-Object and between Noun-Verb

task in order to obtain conditional probabilities of two words. Each relation $\mathfrak{m}$ of $< w_1, w_2 >$ is generated by picking up a distribution over topics, then both elements of the relation $\mathfrak{m}$ share the same topic assignment $z_{\mathfrak{m}}$, which keep two different w_1-topic and w_2-topic distributions sharing the same topic. The models are estimated by Gibbs sampling following [Heinrich, 2004]. It is also noted that these models are generative, and they can predict the probabilities of tuples that do not occur in the training corpus.

To model the relations between objects and verbs, we use the British National Corpus (BNC) which has been preprocessed and parsed using TreeTagger and Maltparser. Verbs are heads of sentences while objects are either direct or indirect objects related to those verbs by the parser. For the relations between verbs and scenes, we consider also verbs as heads of sentences while scenes are all nouns occurring in the same sentence. We use all nouns because the model will be more general since nouns contain all places denoting scenes that we consider (e.g., living room, countryside, park, etc.). For the relations between objects and scenes as well as objects and objects, we also use all nouns to capture a general model.[3] The statistics of the BNC corpus with their corresponding relations are reported in Table 15.

Samples of topics extracted through R-LDA are illustrated in Table 14. As we can see from the table, Noun and Object share many similar terms in the same topic while Noun and Verb sharing the same topics tend to go often together (e.g., win, cup, beat, race).

3 Different from objects and verbs, which can be defined explicitly from the parsed corpora, scenes can only be defined from more restrained rules (e.g., followed by some prepositions), so here we take all nouns to have the most general model.

	#Relations	#Tokens
Verb - Object	3.3M	6.7M
Noun - Noun	19.8M	39.7M
Verb - Noun	83.4M	166.8M

Table 15: The statistics of the dataset used for estimating R-LDA models for each relation type

Given these relations extracted from the language and the output of the object and scene recognizers, we will now go into detail how this information can be combined and used in each of the visual applications.

In our experiments, due to the computational limitation, the language models have been trained on datasets with different sizes. The window model was trained on the ukWaC and Wackypedia (around 2.7 billion tokens), and the distributional memory TypeDM was trained on the web-derived ukWaC, English Wikipedia and the BNC (around 2.83 billion tokens). The data used for training these models has therefore similar size. The topic model R-LDA on the other hand was trained on only the BNC corpus with around 95 million tokens.

To verify whether the size of the datasets used for training language models can effect the results of our framework, we have done some preliminary tests on training the TypeDM model on the same dataset BNC as the one used for R-LDA. It shows that the model trained on the BNC corpus only obtains similar results in human action recognition as the one trained on the whole concatenated corpus. Although further experiments must be done in order to fully investigate the effect of data size for training language models in our framework, this preliminary result suggests that when most common relations are extracted from a large-enough corpus, the increase of training data might not add much improvement in the performance of the framework.

5.4 DATASETS

In order to evaluate our framework on different visual tasks, we have used existing dataset (the SUN dataset) and our annotated datasets for the human action recognition task (the 89 action dataset, the TUHOI dataset). The descriptions of these datasets will be given as follows.

5.4.1 *SUN dataset*

The SUN object dataset [Xiao et al., 2010] contains more than 16 thousand images, more than 79,000 objects whose locations are annotated using polygons. The dataset has been annotated by various people who could choose their own object categories, leading to duplicate categories such as “building” and “buildings”, “person” and “person walking”. Furthermore, for some images large parts are not annotated leading to an incomplete context. We therefore considered only images whose content was sufficiently annotated (images with at least 90% of the area are annotated).

5.4.2 *89 action dataset*

Most available datasets used for evaluating action recognizers are restricted to specific domains (e.g., playing musical instruments, sport activities, etc.) or consider a limited number of actions (7 everyday action, Stanford 40 action dataset). Moreover, all these data sets contain many learning examples well distributed over all actions, but this distribution does not reflect the reality where many more possible actions exist for which few examples are available. To overcome these limitations, we have collected

a new dataset from 11.5 thousand images of the PASCAL 2012 VOC trainval set [Everingham et al., 2010] selecting all those images representing a human action, obtaining 2,038 images. In PASCAL 2012 VOC there are in total 20 objects. Figure 27 reports for each object the number of images in total and the number of images that contain human actions.

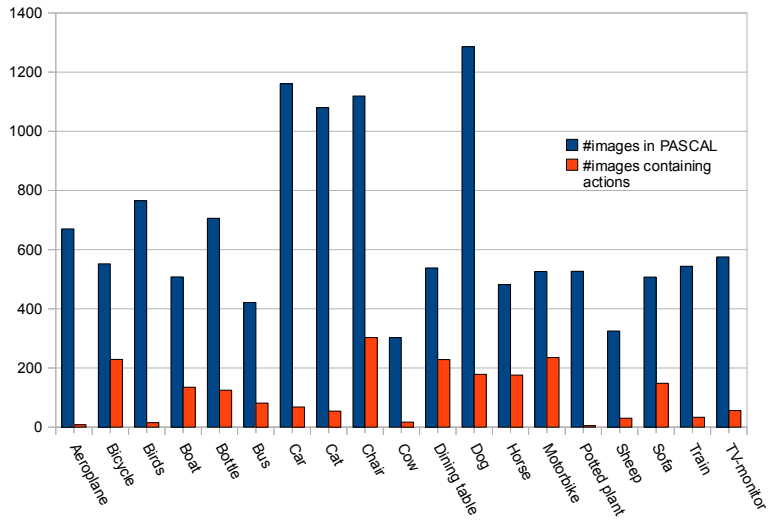


Figure 27: Images containing actions in the PASCAL VOC 2012 trainval dataset

As the images in this dataset were not collected for any specific kind of actions, we believe it gives a general overview of the possible human actions involving the PASCAL objects. Starting from the object label assigned to each image in the PASCAL data set, we manually annotated the 2,038 images with a verb to obtain the label of the human action (verb-object), each image is annotated by one person. The data sets is labeled with 19 objects, human and 36 verbs, that combine into 89 actions. Considering the training vs. validation split used in the PASCAL competition, our human action data set consists of 1,104 images in the training set and 934 images in the validation set[4].

In the data set, there are objects, such as aeroplane, bird, potted plant, which are associated with only few actions (e.g., 8 images with actions related to aeroplanes, 15 images with birds). The objects that are involved in more actions are bicycle (ride, fix), chair (sit), motorbike (ride), bottle (drink). In many pictures, the action is simply a person touching or holding an object.

The 89 action dataset was originally created for the recognition of 20 objects. Afterwards also actions were annotated. Therefore, the actions occurring with these objects are mostly unbiased in the sense that they are not predefined, unlike in other action datasets (e.g., the 7 everyday actions [Delaitre et al., 2010], the Stanford 40 action dataset [Yao et al., 2011b], the PASCAL action classification competition [Everingham et al., 2012]).

After being annotated with human actions, we additionally annotated every image with one of the 15 scenes from the 15 scene dataset [Lazebnik et al., 2006b]. The list of annotated objects and scenes is provided in Table 16.

This dataset is quite domain specific with a limited number of objects (19 objects and humans). due to the availability of large-scale annotated action image datasets.

4 We made the dataset available at http://disi.unitn.it/ dle/pascalaction.php

Objects	bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle chair, dining-table, potted-plant, sofa, TV
Scenes	suburb, coast, forest, highway, city, mountain countryside, street, building, office, bedroom industrial, kitchen, store, living-room

Table 16: 19 objects and 15 scenes

5.4.3 *TUHOI, the Trento Universal Human Object Interaction Dataset*

Visual action recognition is generally studied on datasets with a limited number of predefined actions represented in many training images or videos [Delaitre et al., 2010, Yao and Li, 2010, Wang et al., ,Laptev, 2005]. Though these predefined lists of actions are good for many computer vision problems, this cannot work when one wants to recognize all possible actions. Firstly, the same action can be phrased in several ways. Secondly, the number of actions that such systems would have to recognize in real life data is huge: the number of possible interactions with all possible objects is bounded by the cartesian product of numbers of verbs and objects. Therefore, the task of collecting images or videos of each individual action becomes infeasible with this growing number. By necessity this means that for some actions only few examples will be available. We want to enable studies in the direction of recognizing all possible actions, for which we provide a new, suitable human-object interaction dataset.

To transfer the knowledge from language to vision, it is important that the distribution of the visual actions are sampled similarly as the language data. This requirement is fulfilled when the action frequencies in the dataset mirror the frequencies in which they occur in real life. The 89 action dataset that we have built satisfies this requirement. However, the number of objects in this dataset is limited to 19 objects and each image has been annotated by one person only. Therefore, we aim at building a new human action image dataset TUHOI, the Trento Universal Human Object Interaction Dataset, which can (1) capture the distribution of human interactions with objects in reality (if an action is more common that the other actions, that action is also observed more frequently in the dataset than the others), (2) provide different ways of describing an action for each image (there are many actions that can be phrased in several ways, for example: fix a bike or repair a bike), (3) help with identifying different verb meanings (for example, the word ÔridingÕ has different implications for Ôriding a horseÕ, Ôriding a carÕ, and Ôriding a skateboardÕ). In the following section, we will first describe available image datasets for human action recognition including the above 89 action dataset, and then our new dataset and how it was collected and annotated.

Available image datasets for human action recognition

Most approaches in human action recognition rely on suitable training data for a set of predefined actions: [Gupta et al., 2009b] tests on a 6 sport action dataset, [Yao and Li, 2010] attempts to distinguish images where a human plays a musical instrument from images where he/she does not, [Delaitre et al., 2010] classifies images to one of the seven every day actions, and [Yao et al., 2011c] introduces a dataset containing 40 human actions. Most of these datasets were obtained using web search results such as Google, Bing, Flickr, etc. The number of images varies from 300 to more than 9K images. A comparison of the publicly available datasets with respect to the number of actions and their related objects are given in Table 17: Ikirler [Ikizler et al., 2008], Willow [Delaitre et al., 2011], Sport dataset [Gupta et al., 2009b], Stanford 40 [Yao et al., 2011c], PPMI [Yao and Li, 2010], PASCAL [Everingham et al., 2012] and 89 action dataset [Le et al., 2013c].

Dataset	#images	#objects	#actions	Examples of actions
Ikirler	467	0	6	running, walking, throwing, crouching and kicking
Willow	968	5	7	interaction with computer, photographing, riding bike
Sport dataset	300	4	6	tennis-forehand, tennis-serve, cricket bowling
Stanford 40	9532	31	40	ride horse, row boat, ride bike, cut vegetables
PPMI	4800	7	7	play violin, play guitar, play flute, play french horn
PASCAL	1221	6	10	jumping, playing instrument, riding horse
89 action dataset	2038	19	89	drive bus, sail boat, ride bike, fix bike, watch TV
TUHOI dataset	10805	189	2974	sit on chair, use computer, ride horse, play with dog

Table 17: A comparison of available human action datasets in terms of number of objects and actions

As can be seen in Table 17, the Stanford 40 action dataset contains quite a large number of images with 40 different actions. This dataset is good for visually training action recognizers since there are enough images collected for each actions divided into training and test sets. However, the distribution of the visual actions does not mirror the frequencies in which they occur in real life since the number of training images is collected equally for every action. This makes the dataset not suitable for our purpose. There are some dataset in which human action does not involved any object, these actions are for instance running, walking, or actions where objects are not specified such as catching, throwing. These types of actions are not the target domain of our dataset. We aim at recognizing the human object interactions based on objects. With the same object, some actions are also more common than other actions: for example, sitting on a chair is more commonly observed than standing on a chair. We want to capture such information in our dataset which can reflect the human action distributions on common objects, aiming to sample human actions related to objects in the visual world. Furthermore, how actions can be phrased in different ways, or how verbs can have different meanings when interacting with different objects should also be considered. Some actions can only be performed on some particular objects and are not applicable to some other objects: a person can ride a horse, ride a bike, can feed a horse, but cannot feed a bike. This problem of ambiguity and different word uses have been widely studied in computational linguistics, but have received little attention from the computer vision community.

With the aim of creating a dataset that covers these requirements, we collect our dataset starting from images where humans and objects co-occur together and define the actions we observe in each image instead of collecting images for some predefined human actions. This way of annotating actions in images is more natural and helps creating a more realistic dataset with various human actions that can occur in images generally. Furthermore, we have used crowdflower, where different people can annotate the same image, it can then address the problem that one action can be phrased in different ways.

Recently, some good works attempted to generate descriptive sentences from images [Farhadi et al., 2010a, Kulkarni et al., 2011b]. In our dataset we focus on human actions, which, if present, are often the main topic of interest within an image. As such, our dataset can be used as an important stepping stone for generating full image descriptions as it allows for more rigorous evaluation than free-form text.

The DET dataset: Object categories and labels

We built the TUHOI based on the DET dataset. The DET dataset in the ImageNet large scale object recognition challenge 2013[5] contains 200 objects for training and evaluation. With the idea of starting

5 http://www.image-net.org/challenges/LSVRC/2013/

accordion, airplane, ant, antelope, apple, armadillo, artichoke, axe, baby bed, backpack, bagel, balance beam, banana, band aid, banjo baseball, basketball, bathing cap, beaker, bear, bee, bell pepper, bench, bicycle, binder, bird, bookshelf, bow tie, bow, bowl, brassiere, burrito bus, butterfly, camel, can opener, car, cart, cattle, cello, centipede, chain saw, chair, chime, cocktail shaker, coffee maker, computer keyboard, computer mouse, corkscrew, cream, croquet ball, crutch, cucumber, cup or mug, diaper, digital clock, dishwasher, dog, domestic cat, dragonfly, drum, dumbbell, electric fan, elephant, face powder, fig, filing cabinet, flower pot, flute, fox, french horn, frog, frying pan, giant panda, goldfish, golf ball, golfcart, guacamole, guitar, hair dryer, hair spray, hamburger, hammer, hamster, harmonica, harp, hat with a wide brim, head cabbage, helmet, hippopotamus, horizontal bar, horse, hotdog, iPod, isopod, jellyfish, koala bear, ladle, ladybug, lamp, laptop, lemon, lion, lipstick, lizard, lobster, maillot, maraca, microphone, microwave, milk can, miniskirt, monkey, motorcycle, mushroom, nail, neck brace, oboe, orange, otter, pencil box, pencil sharpener, perfume, person, piano, pineapple, ping-pong ball, pitcher, pizza, plastic bag, plate rack, pomegranate, popsicle, porcupine, power drill, pretzel, printer, puck, punching bag, purse, rabbit, racket, ray, red panda, refrigerator, remote control, rubber eraser, rugby ball, ruler, salt or pepper shaker, saxophone, scorpion, screwdriver, seal, sheep, ski, skunk, snail, snake, snowmobile, snowplow, soap dispenser, soccer ball, sofa, spatula, squirrel, starfish, stethoscope, stove, strainer, strawberry, stretcher, sunglasses, swimming trunks, swine, syringe, table, tape player, tennis ball, tick, tie, tiger, toaster, traffic light, train, trombone, trumpet, turtle, tv or monitor, unicycle, vacuum, violin, volleyball, waffle iron, washer, water bottle, watercraft, whale, wine bottle, zebra

Table 18: List of the 200 objects in the DET dataset

from images with humans and common objects, we chose to use this DET dataset as a starting point to build our human action data.

The 200 objects in the DET dataset are general, basic-level categories (e.g., monitor, waffle iron, sofa, spatula, starfish - see Table 18). Each object corresponds to a synset (set of synonymous nouns) in WordNet. We have annotated both training and validation set for the annotation.

Dataset	#images	#images having "person"	#object instances	#instances/object (min-max-median)	#"person" instances
Training	395,909	9,877	345,854	438 - 73,799 - 660	18,258
Validation	20,121	5,791	55,502	31 - 12,823 - 111	12,823

Table 19: The statistics of the DET dataset

As can be seen in Table 19, there are 15,668 images having human and 31,081 human instances in these images. We select only images having human since we want to annotate this dataset with human object interactions. Objects related to clothes such as bathing cap, miniskirt, tie, etc. are not interesting for human actions (most of the time, the action associated with these objects is "to wear"). Therefore, we excluded all these objects from the list of 200 objects above, which are: bathing cap, bow tie, bow, brassiere, hat with a wide brim, helmet, maillot, miniskirt, neck brace, sunglasses, tie. We are left with 173 unique objects.

Human action annotation

GOAL Our goal is to annotate these selected images containing humans and objects with their interactions. Each human action is required to be associated with at least one of the given 200 object categories. We used the Crowdflower, a crowdsourcing service for annotating these images. The Crowdflower annotators are required to be English native speakers and they can use any vocabulary to describe the actions as they wish. Every action is composed of a verb and an object (possibly with a preposition).

ANNOTATION GUIDELINE For each image, given all object instances appearing in that image (together with their bounding boxes), the annotator has been be asked to assign all human actions associated to each of the object instance in the image (where "no action" is also possible). The instruction is given in Figure 28. The GUI of the annotation task and its instructions are illustrated in Figure 29 and Figure 28.

Every human actions need to have as object one of the object instances given in that image. For example, if the image has a bike and a dog, the annotator will assign every human actions associated to “bike” and “dog”. Every image has been annotated by at least 3 annotators, so that each action in the image can be described differently by different people. Some examples of annotated images in our dataset are given in Figure 30.

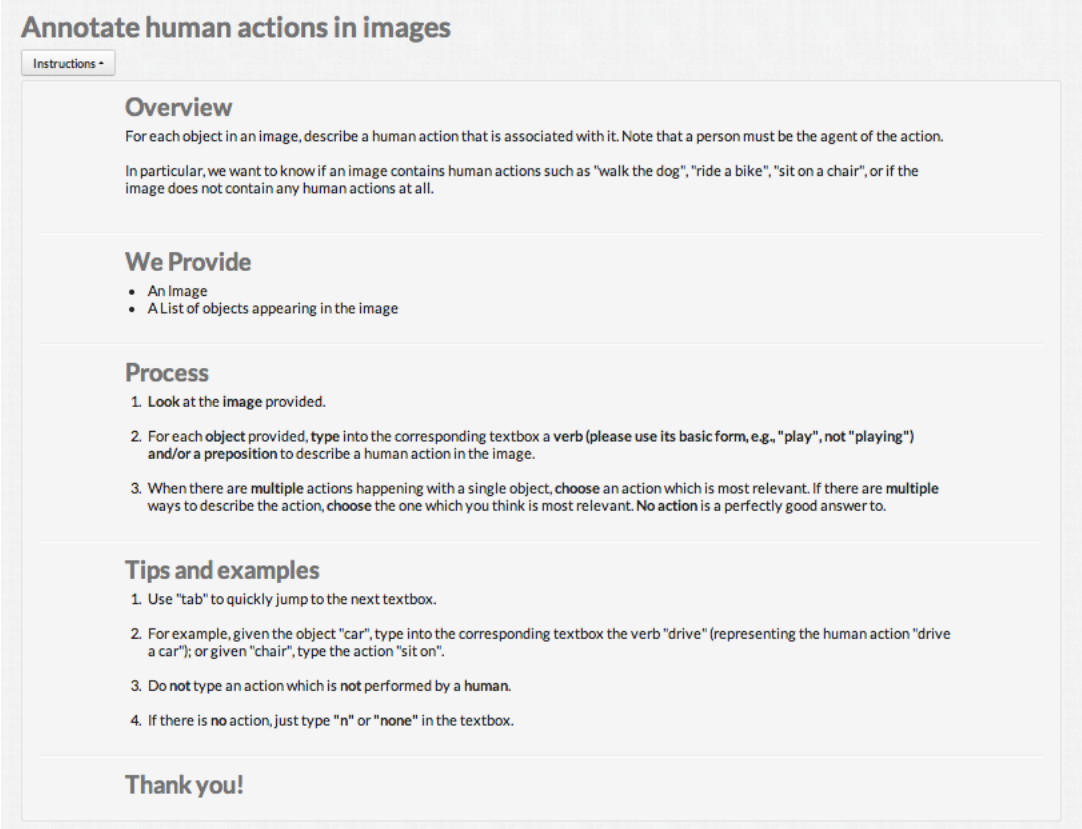


Figure 28: The instruction of the crowdflower “Annotate human action in images”

Results of the annotation and some statistics

In total, there are 10,805 images, which have been annotated with 58,808 actions, of which 6,826 times it has been annotated with “no action” (11.6%). On average, there are 4.8 actions annotated for each image (excluding “no action”), of which there are 1.97 unique action/image. Some other statistics of the dataset are given in Table 20: The number of unique verbs per object ranges from 1 (starfish, otter) to 158 (dog). As dogs occur very often in this image dataset (4,671 times), the number of actions associated to it is also larger than other objects.

For some images, the annotators find many different ways to describe the action in the image. In our data, a set of images was selected to be annotated by more than three people in order to facilitate sanity checks. An example of such image which has been annotated by many people is given in Figure 31. The annotators have found many verbs to describe the action: feeding, leading, running with, touching, giving a treat to, etc.

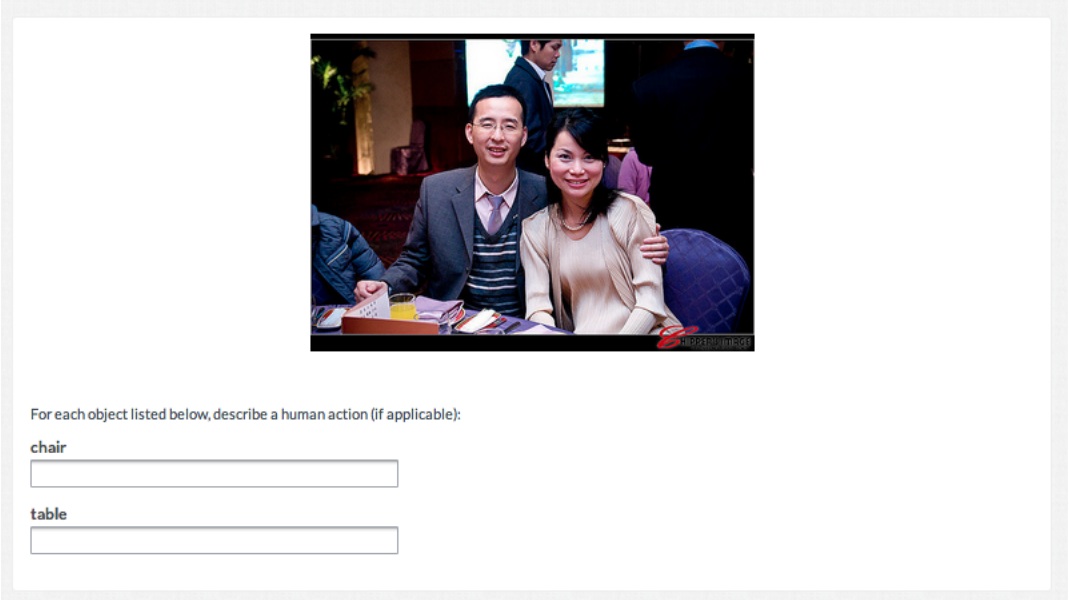


Figure 29: An example of the interface for action annotation

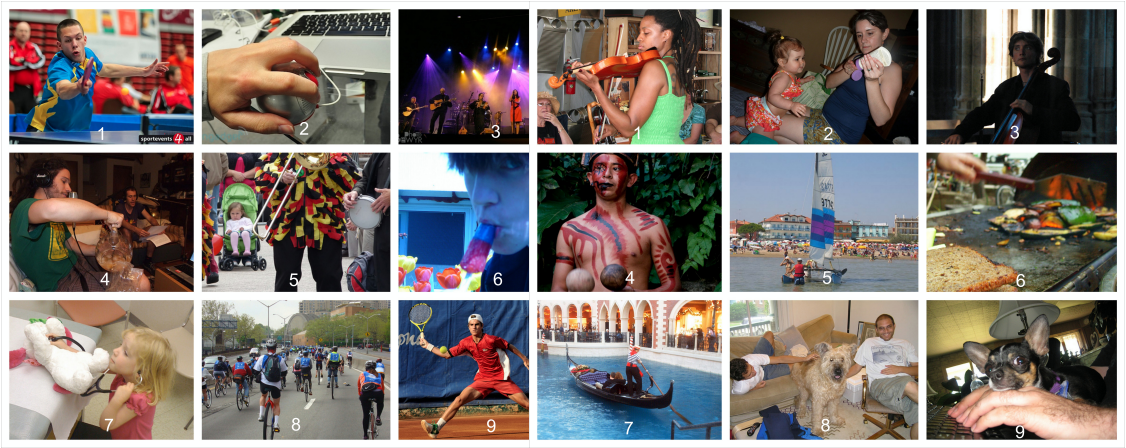


Figure 30: Examples of annotated images: **Left:** (1) play ping-pong, hold racket; (2) use laptop, hold computer mouse; (3) use microphone, play accordion, play guitar, play violin; (4) talk on microphone, sit on sofa, pour pitcher; (5) play trombone; (6) eat/suck popsicle; (7) listen/use/hear stethoscope; (8) ride bicycle, wear backpack; (9) swing/hold racket, hit tennis ball; **Right:** (1) sit on chair, play violin; (2) wear diaper, sit on chair, squeeze/apply cream; (3) sit on chair, play cello; (4) hold/shake maraca; (5) ride watercraft, wear swimming trunks; (6) cook/use stove, stir mushroom, hold spatula; (7) drive/row watercraft; (8) sit on chair, pet dog, lay on sofa; (9) click/type on computer keyboard

Number of unique actions (verb + object): 2,974 actions Number of unique verbs: 860 verbs
Verbs that are used most frequently (verb (#occurrences)): play (13043), hold (7731), ride (4765), sit (3535), sit on (1501) drive (1491), wear (1441), eat (1175), hit (1168), pet (970), use (897), walk (787), stand (756) touch (509), carry (507), blow (384), sail (323), kick (297), lead (290), throw (246), strum (239) stand on (223), run (223)
Verbs that are used least frequently (occur only once): dirty, swing over, twist, beats, walks, ay, curl face, shit, sail in, n', see by, forge, draw, tag10, sling, rides, walk across, no image available, waving drag, award, preform, strumb, died, land, unload, tricks, cooked, time, fasten, fall over, holed, leap over, pull up
Objects go with the largest number of verbs (object (#unique verbs)): dog (158), car (80), table (79), watercraft (68) horizontal bar (56), chair (54), cart (52), whale (50), bicycle (48), cattle (42), soccer ball (41), balance beam (38) band aid (38), motorcycle (37), flower pot (35), ladle (35), guitar (35), horse (35), ski (34), bus (34)
Objects that go with the least number of verbs (object (#unique verbs)): milk can (5), pitcher (5), scorpion (4), bear (4), pretzel (4), sheep (4), frog (4), mushroom (4), printer (4), pineapple (4), ruler (3), guacamole (3), isopod (3), chime (3), plate (rack (3), strawberry (3), porcupine (3), ant (3), toaster (3), bagel (3), jellyfish (3), dragonfly (2), lion (2), zebra (2), goldfish (2), hamster (2), fig (2), squirrel (2), bee (2), centipede (2), koala (bear (2), snail (2), pomegranate (2), armadillo (2), otter (1), starfish (1)

Table 20: Some statistics of the human action dataset

SPLITTING TRAINING AND TEST SET For each object in our human action dataset, we split half of the images for training and the other half is used for testing. The splitting process is done such that actions that occur in test set also occur in training set to guarantee that the training set contains at least one image for each action occurring in the test set.

Figure 31: Many different ways to describe an action in an image

5.5 OBJECT PREDICTION

In this section, we will focus on the task of object prediction in images. Different from recognizing objects using their shapes and visual looks, in this task, the system will predict possible objects that occur in an image given the appearance of other objects. We will explain how we apply the general framework and use the information obtained from visual features and language models to predict objects.

5.5.1 *Introduction*

Our first computer vision scenario is about objects in context. Context is useful in visual recognition for two reasons: Firstly, context can significantly reduce the number of possible object categories simplifying the problem. Secondly, when the object appearance is inconclusive for its identity, context can be used for disambiguation. For example, a grey rectangle on a desk may be recognized as a pen, while a grey rectangle on a table may be recognized as a knife. As the recognition systems are not always reliable, the use of context can greatly improve results.

For this scenario, we choose a theoretical setting in which we want to predict the identity of one object given that the identities of all other objects in the image are known. We believe that our main

conclusions on the linguistic models will transfer to a practical computer vision application where visual recognition systems predict the object identities.

Formally, we can describe this scenario as follows: Given an image I with N objects $\mathcal{O} = \{o_1, o_2, \cdots, o_N\}$, we want to predict the identity of object o_i given all other objects $\mathcal{O} \setminus o_i$. We use a Naive Bayes assumption, leading to:

$$\begin{aligned} P(o_i|\mathcal{O} \setminus o_i) &= \frac{P(\mathcal{O} \setminus o_i|o_i) \times P(o_i)}{P(\mathcal{O} \setminus o_i)} \\ &\approx P(o_i) \times \prod_{o_j \in \mathcal{O} \setminus o_i} P(o_j|o_i). \end{aligned} \tag{21}$$

In this scenario, we need conditional relations $P(o_j|o_i)$ and priors. We obtain these from language data or from images directly.

5.5.2 *The SUN data preprocessing for object prediction*

To evaluate the object prediction task, we use the SUN object dataset as described above. We cleaned the object categories by mapping from around 7,500 objects to over 700 unique object categories, e.g., *Person sitting*, *Person walking*, etc. to the class *Person* . This process is visualized in Figure 32.

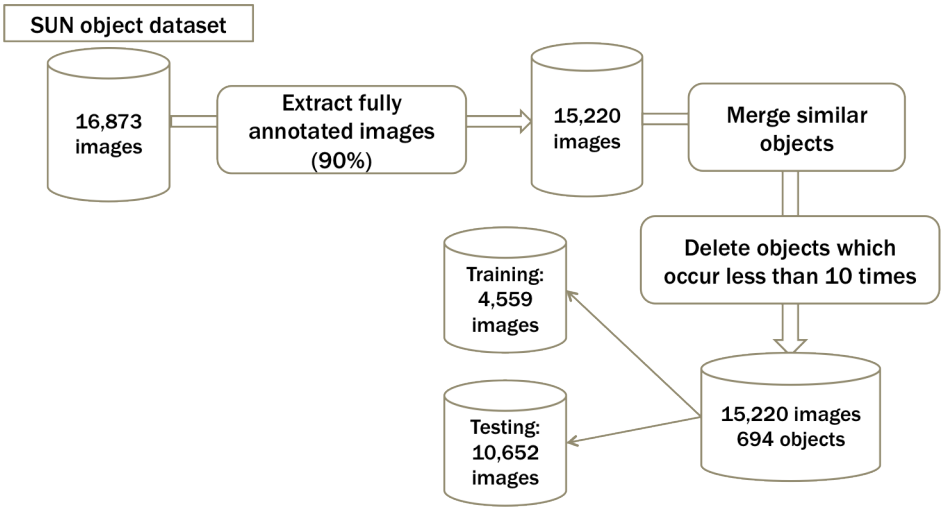


Figure 32: SUN dataset preprocessing

In our experiments, we used the predefined training and testing parts of the SUN dataset and obtained around 4,500 images for learning the object relations and 10,600 images for testing the object prediction. We obtain conditional probabilities $P(o_j|o_i)$ from frequency counts.

For extracting image statistics, the conditional probability of an object given another object is based on their frequencies in the dataset:

$$P_I(o^j|o^i) = \frac{\text{freq}(o_j, o_i)}{\text{freq}(o_i)} = \frac{\#o^i, o^j \text{co-occur in an image}}{\#o^j \text{occurs}} \tag{22}$$

5.5.3 *Experiments and discussion*

For this task, we will perform two experiments: first, we directly compare the statistics of the language models with statistics extracted from the visual domain; second, we compare these language models inside the object prediction task. Our aim is to answer the two questions: (1) Is the knowledge from language compatible with the knowledge from vision? (2) Can the knowledge extracted from language help in object prediction in images?

STATISTICS COMPARISON OF LANGUAGE MODELS AND VISUAL DOMAIN In this section, we compare statistics mined from texts with those mined from visual sources. Ideally, we want statistics from the language models to follow those of the image model, even though not all statistics from images can be reliably measured due to insufficient data. Therefore, we measure how well the estimated language models fit the estimated visual distributions using the χ^2-distance. χ^2-distance measures the difference between two discrete probability distributions:

$$\chi^2 = \sum_{i=1}^{N} \frac{({P_I}^i - {P_L}^i)^2}{{P_I}^i} \tag{23}$$

where P_I and P_L are the probability distribution obtained from the image data and language models respectively.

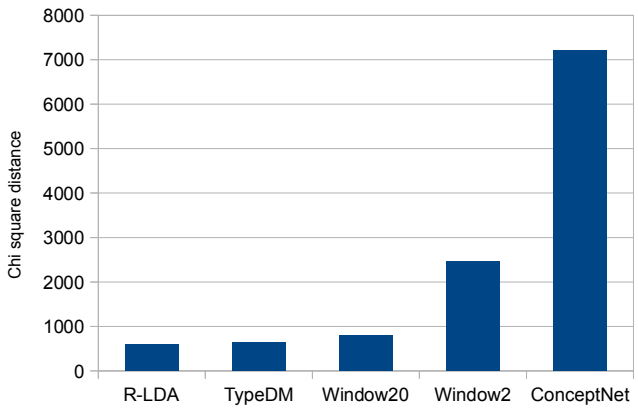


Figure 33: χ^2-distances between the tested language models and the image model for conditional probabilities of objects $P(O|O)$.

For the relations between objects and objects, we use the SUN dataset, a big and general dataset. As shown in Figure 33, R-LDA is most similar to the image model, closely followed by TypeDM and the Window20 model. All these three models are good at capturing broad contextual relations. The Window2 model has a significantly larger distance to the image model as it captures a narrow context of 2 words, which is apparently not enough to find co-occurences of objects. ConceptNet is the most inconsistent with this image data since the relations extracted from it were not enough objects and their relations are extracted from it. To sum up, R-LDA achieves the best performance in modeling the relations between objects and objects among all language models.

LANGUAGE MODELS FOR OBJECT PREDICTION Let p^i be the position of the correct action found in the ranked list for each image i. To measure the performance of the object prediction task, we use the average ranking over all images (AR_I) and over all objects (AR_O):

$$AR_I = \frac{\sum_{i=0}^{N} p^i}{N}; AR_O = \frac{\sum_{j=0}^{N_o} p_o^j}{N_o} \tag{24}$$

where N is the number of images, N_o is the number of objects and p_o^j is the average rank of all images having object j. The average rank over all image measures the performance over the image dataset, but infrequent objects have little impact on this performance. The average rank over objects gives more weight to rare examples.

For every object in every image in the test set of the SUN database, we guess the identity of an object given the identity of all other objects in the image. In total, there are 78,306 object predictions within 10,652 images.

As shown in Figure 34, the R-LDA model outperforms all other models for average rank both over images and over objects. Interestingly, both R-LDA and TypeDM are better at predicting the correct objects in images than the model learnt from the image training set itself. It shows that for many cases, the relation statistics learnt from language data can help in visual recognition. These language models are even better than the information extracted from general, relatively unbiased image datasets (ID), where annotation is limited. For the limited annotation, this hypothesis is further supported by looking at the average rank over objects, which gives more weight to rarely occurring objects. As seen in Figure 34, all language models except ConceptNet outperform the image model. We conclude that language models can aid visual models in large-scale visual recognition problems which use co-occurrence of objects as their context, especially when the annotation is limited, as is often the case.

We conclude that: the knowledge extracted from the R-LDA and TypeDM language models are the most compatible to the knowledge from visual (the first research question). For the second research question, the experimental results show that the knowledge extracted from language can help in object prediction in images, it even outperform the model where we learn this knowledge from expensive annotated images.

5.6 AN INTEGRATED SYSTEM: ACTION RECOGNITION

In this section, we will present our work in applying the knowledge-based framework in human action recognition in images.

5.6.1 *Introduction*

The problem of action recognition has challenged the Computer Vision community for quite a long time. Currently, research on action recognition in still-images focuses on data sets of around 40 human actions defined by "verb-object" relations, like "playing violin" or "riding a bike", where each action has a good number of training examples. However, the combinatorial explosion of verb-object relations makes the task of learning human actions directly from their visual appearance computationally prohibitive and makes the collection of proper-sized image datasets infeasible. Furthermore, actions are a rather complex semantic concept, since an action is expressed by the combination of a verb with an agent and a patient, as well as other possible elements of what, in computational linguistics and artificial intelligence, is known as a "frame". We assume that one can know an action by knowing the frame it belongs to.

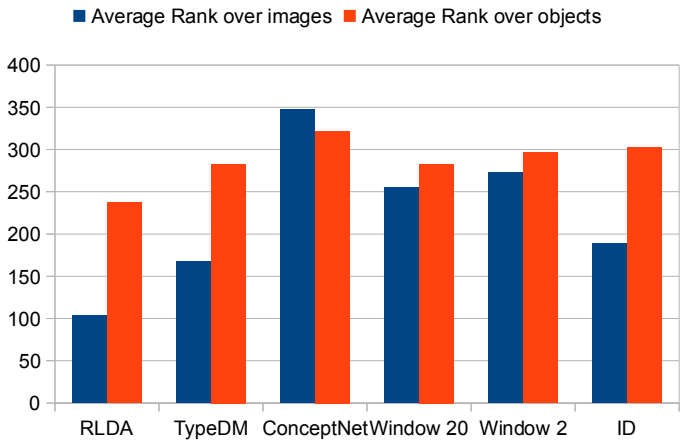


Figure 34: Average rank over all images and objects using different language models and ID (image data)

Therefore, we aim to develop an action recognizer that can recognize *unseen* actions based on their frames, where unseen means that no visual training examples with action labels are available. Having such a system enables us to handle much more actions than currently considered within the Computer Vision community and will guarantee the scalability and stability of results. To this end, we propose a framework in which the knowledge extracted from language models is learned from an open domain and very large text corpora, as briefly introduced in section 5.3.

We want to identify a human action, defined as a <subject, verb, object> triple. We do this by recognizing the human, the object, and the scene and then determine the most likely verb based on these components. Scenes are only used here as features for predicting/disambiguating the human action and the final task is to define the human action triple. As in most work in human action recognition, we simplify the problem by considering only images in which human actions occur. This means that a human is always present, and the problem we tackle is reduced to predicting the verb given the object and the scene. While this may seem like a strong assumption, the possibility of having no action in the image at all is largely unexplored in computer vision due to its difficulty.

5.6.2 *Human action recognition framework*

Our general action recognition framework is presented in Figure 35. Given an image, an object recognizer will predict the probability of each object (e.g, bike, horse) presented in that image. Furthermore, a scene recognizer will provide the probabilities of each scene (e.g., countryside, suburb, forest) given the image. The action model is composed of the conditional probabilities that relate verbs, objects and scenes, which have been learned from training images or language corpora. Given the object and scene probabilities recognized in the image, the action model will guide the action prediction process and finally, the system will suggest the most proper actions (e.g., ride horse, drive car).

As can be seen in Figure 35, the model captures the relationship between Object - Scene, Verb - Scene and Verb - Object containing the probability of an object given a scene $P(o_j|s_k)$, a verb given an object $P(v_i|o_j)$, and a verb given a scene $P(v_i|s_k)$. In one experiment we learn the probabilities from the training images, where each image has been annotated with an object, a verb (of an action)

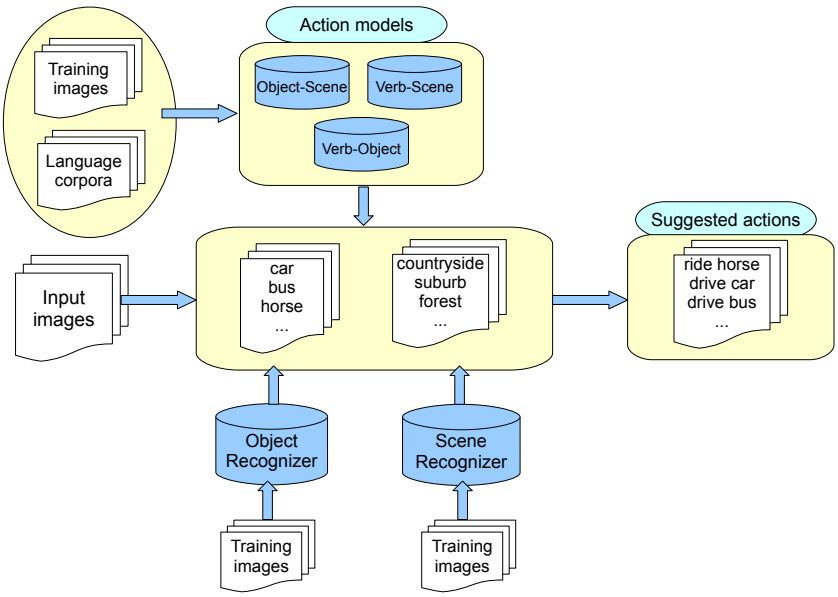


Figure 35: Human action suggestion: based on the objects and scenes recognized in an image, the system suggests the most plausible actions. The action models provide the relationships between objects - scenes - verbs

and a scene. All three probabilities are computed using frequency counts in the training set, for example:

$$P(o_j|s_k) = \frac{\#\text{images having } o_j, s_k}{\#\text{images having } s_k} \tag{25}$$

We aim to replace this learning from annotated training images, which are expensive to obtain, with learning from language corpora. The details of how to extract the probability distributions from language models are explained in section 5.3.2.

5.6.3 *Component integration*

To combine these components in the framework, we use an energy based formulation [Lecun et al., 2006]. We create a model that is visualized in Figure 36, which includes the image I (an observed variable) and object O, scene S, and verb V. This energy-based formulation allows us to set different weights for energies which come from disparate sources (i.e. language and vision) using the Gibbs measure.

Now given an image I, we can compute the score function $S(a_{ij}; I)$ of an action a_{ij} as:

$$S(a_{ij}; I) = S(v_i, o_j; I) = \frac{1}{Z} \exp\left(-\sum_{F \in \mathcal{F}} E_F^{ij}\right),$$

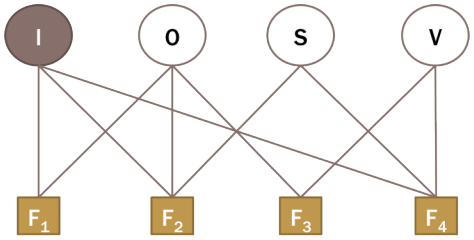


Figure 36: An energy-based model for action recognition

where we define each energy function E_F^{ij} to give lower energies to correct answers and higher energies to incorrect ones, where $\mathcal{S}$ is the set of all scenes:

$$E_{F_1}^{ij}(O, I) = -w_{F_1} \log P(o_j|I) \tag{26}$$

$$E_{F_2}^{ij}(S, I) = -w_{F_2} \log \sum_{S \in \mathcal{S}} P(o_j|S) \times P(S|I) \tag{27}$$

$$E_{F_3}^{ij}(V, O) = -w_{F_3} \log P(v_i|o_j) \tag{28}$$

$$E_{F_4}^{ij}(V, S) = -w_{F_4} \log \sum_{S \in \mathcal{S}} P(v_i|S) \times P(S|I) \tag{29}$$

Let P_i be the position of the correct action in the ranked list of predicted actions for a certain image I_i. The ranked list is sorted in the order of the score S. We evaluate human action recognition in terms of this position average over all images, which we call Average Ranking (AR). Therefore we use Average Ranking as our loss-function:

$$\mathcal{L}(w_F) = AR_N = \frac{1}{N} \sum_{i=0}^{N} P_i. \tag{30}$$

Training the energy model involves finding the factors w_F^* that minimizes the loss:

$$w_F^* = \underset{w_F}{\operatorname{argmin}}\, \mathcal{L}(w_F) \tag{31}$$

As we have only four parameters to learn in our energy model, we do this by performing an exhaustive search and cross validation. We require $w_F \in \{0.0, 0.1, 0.2, ..., 0.9\}$ and set the constraint $\sum_{F \in \mathcal{F}} w_F = 1$. We note that the factor graph formulation of our framework would allow us to use more advanced learning algorithms. In work beyond this thesis, we plan to look into this once the model becomes more complex by adding, for example, information about the position of the objects and the human.

5.6.4 *Model adaptation*

MOTIVATION Verbs describing actions in language are different from those used to describe actions in images. In language, more common verbs are often used to describe actions, for example: take a train, take a car, get a bottle. While in images, people tend to use more specific verbs to describe the actions, for example: a person is sitting on a car, touching a motorbike, holding a bottle. Such verbs (e.g., hold, touch) are usually used to describe actions in some particular states (e.g., a person is getting on a train, holding a dog) that are present in images.

Therefore, when applying language to recognize actions in images, the language model often proposes more general verbs, while the image model suggests more specific verbs. To adapt the language model to the image model, we incorporate a tailoring function $\mathcal{T}(v)$, which diminishes the weight of verbs that are too general in language and increases the weight of verbs that are more specific in language and verbs occur often in images.

$$P'(V|O) \approx P(V|O) \times \mathcal{T}(V) \tag{32}$$

The tailoring function can be learnt from either language data or image data. Ideally, general verbs in language are those that are composed of many actions. For example, "take a train" is an action of waiting a train, getting on a train, sitting on a train. Specific verbs are those that can be used to describe only a single action/state, e.g., "hold a dog", "touch a cat". However, such discrimination is difficult to learn since no existing source is available.

In this work, we define a general verb in language bases on its frequency and on the number of objects it goes with. We define common verbs in images are those that go with many objects in images. The tailoring function decreases the probability of general verbs in language and increases the probability of common verbs in images.

USING LANGUAGE DATA Using language data, we learn the generality of a verb based on its frequency or the number of objects it goes with. In the first case, the tailoring function is defined solely based on the frequency of a verb:

$$\mathcal{T}_{L1}(V) = \log \frac{\sum_{v \in \mathcal{V}} \mathtt{freq}(v)}{\mathtt{freq}(V)} \tag{33}$$

The general idea of this tailoring function is that it will give higher chance for verbs that are not too general, i.e., do not occur too often in the text dataset. In the second case, we define a general verb based on the number of objects that it goes with. In particular, the tailoring function is defined as the number of objects that account for 90% of the total distribution of a verb:

$$\mathcal{T}_{L2}(V) = \log \frac{\text{\#all objects}}{\text{\#objects going with } V} \tag{34}$$

The function $\mathcal{T}_{L1}$ gives lower weight to verbs that occur more often in the corpora while the function $\mathcal{T}_{L1}$ gives lower weight to verbs that can go with many objects.

USING IMAGE DATA Using image data, assuming that we know the number of objects each verb go with, we define the tailoring function:

$$\mathcal{T}_{I}(V) = \log(\text{\#objects going with } V + 1) \tag{35}$$

and

$$\mathcal{T}'_{I}(V) = \text{\#objects going with } V \tag{36}$$

These tailoring functions will increase the rank of verbs which go with a high number of objects in image descriptions and decrease the rank of those that do not occur with many objects.

COMBINATION We combine the tailoring function learnt from language and image together:

$$P'(V|O) \approx P(V|O) \times \mathcal{T}_{L1}(V) \times \mathcal{T}'_{I}(V) \tag{37}$$

This tailoring function favors verbs that are often used in images while decreases the probability of verbs that are found too often in language.

USING FLICKR IMAGE TAGS We use a collection of 3.5 million images with their tags randomly collected from Flickr[6]. The collection contains around 570K unique tags with around 270K unique user ids. From these tags, we learn what verbs are usually used to annotate images and what verbs are less likely used. Following the equation 35, we define the tailoring function such that it will increase the rank of verbs that often occur in the Flickr dataset.

5.6.5 *Classifying human actions based on human-object positions*

EVALUATING HUMAN ACTION CLASSIFICATION BASED ON OBJECTS In this part, we use our newly collected dataset TUHOI for building a general human action classifier based on objects. We analyze the relative positions between humans and objects in each image and use this information to help classifying human actions. Finally, we discuss the relations between human-object positions with prepositions that are used in language for describing human actions.

To evaluate the performance of the human action classification on this dataset, we use two different measurements: the accuracy and the traditional precision, recall and F1 score. The accuracy reflects the percentage of predictions that are correct. We calculate within how many images, the classifier assigns the correct actions for a given object i:

$$\text{Accuracy}_i = \frac{\text{number of images that the classifier predicts correctly}}{\text{total number of images}} \tag{38}$$

If the output of the classifier is one of the three annotated actions by human, then the action predicted is considered to be correct. The accuracy of the whole system is the average accuracy over all objects, with n is the total number of objects.

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \text{Accuracy}_i}{n} \tag{39}$$

This metric gives us the general performance of the system and easy to interpret. However, it gives higher weights to actions that occur more often in the dataset. For example, if there are many actions "ride bike" occurring in the dataset, the accuracy of the whole system depends mostly on the performance of the class "ride bike". For actions that occur more rarely such as "fix bike", then the accuracy of the class "fix bike" will have little effect to the accuracy of the whole system.

To better analyze the results of the system and evaluate each action individually, we use the precision, recall and F1 score for each class in the classifier. More specifically, as this classifier is the multi-class classifier, these metrics are computed using a confusion matrix:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}; \text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}} \tag{40}$$

where M_{ij} is the value of the row i, column j in the confusion matrix. The confusion matrix is oriented such that a given row of the matrix corresponds to the value of the "truth", i.e., correct actions assigned by human, and a given column corresponds to the value of action assigned by the classifier. Finally, the precision, recall and F1 score of the whole system are calculated as the average score over all actions.

In this experiment, we used the Forest Random classification method to classify an image to an action given an object. The features used for this classifier are the positions of the object and of the person appearing in that image. We compare this classifier when using position with a classifier using no position information to see whether position information helps in classifying human actions and in which cases.

6 http://staff.science.uva.nl/ xirong/index.php?n=Research.TagRelevanceLearning

Chi statistics	P(V|O)	P(V|S)	P(O|S)
R-LDA	17.8	11.6	**11.9**
Window2	11.6	**11.4**	32.6
Window20	11.7	**11.4**	23.7
TypeDM	**11.5**	13.3	23.2
ConceptNet	17.5	11.5	34.4

Table 21: $\mathcal{X}^2$ distance for relations between verbs, objects, scenes from different language models to image data

EXTRACTING FEATURES To extract the features of objects and persons' positions in the images, we take the bounding box of the first object instance annotated in that image. There are images with more than one object instance (for example, there are several 'bike' in an image, so we do not know what 'bike' we are talking about). We use the four coordinates of the bounding boxes of the object and person in the image as features for the classifier.

5.6.6 *Experiments and results*

Research questions

Our experiments are designed to answer the four questions: (1) Are the knowledge about actions extracted from text and vision compatible? (2) can we use the knowledge learned from text to aid human action recognition? (3) can we apply the model adaptation as introduced above to help improve the results of the recognizer? (4) how does the size of the datasets effect the results of the framework? and (5) can we map the position information between human and object in images with the prepositions in language to help the action recognition?

Experiments

COMPARISON OF STATISTICS FROM LANGUAGE VISUAL This experiment is designed to answer our first research question: whether the knowledge about actions extracted from text and vision are compatible. We use a similar method as in object prediction experiments. In particular, we measure how well each language model fit the estimated visual distributions using the χ^2-distance.

For the conditional probabilities $P(V|O)$, $P(V|S)$, and $P(O|S)$ we compare language models with image statistics extracted from the 89 human action dataset. Table 21 shows the results. For the relations between verb and scene $P(V|S)$, there is not much fluctuation among different language models. For objects and scenes $P(O|S)$, R-LDA is closest to the image model. This is because R-LDA is good at measuring contextual and indirect relations by design, which is the case for object-scene relations. This also explains why TypeDM and Window20 are further away from the image model, followed by the Window2 model. Instead, human actions are found in language as the relation between verbs and their direct linguistic objects. Indeed, TypeDM is closest to the image model for $P(V|O)$ as it makes explicit use of this linguistic link. The Window2 and 20 models are almost as close to the image model for $P(V|O)$, while R-LDA is considerably further away due to its contextual nature. Finally, ConceptNet is the furthest away from the image model. To conclude, different language models have different degrees of compatibility with the relations extracted from images depending on the relations. TypeDM is best for modeling direct verb-object relations, while R-LDA is better at capturing the more contextual object-scene relations.

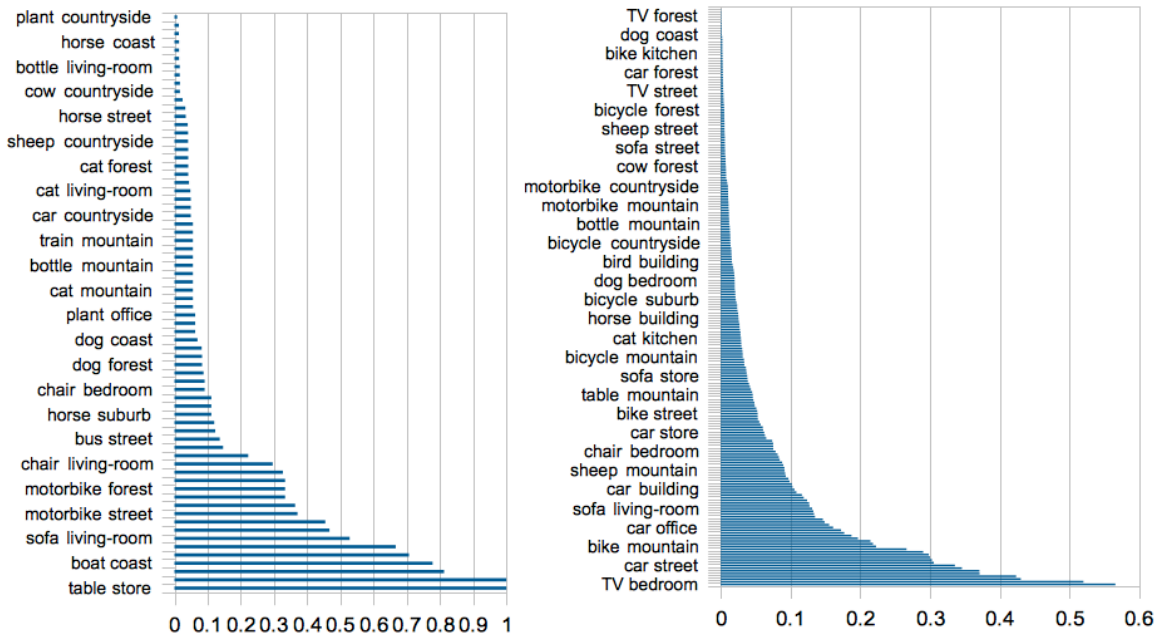


Figure 37: Probability distributions of scene over object extracted from: (left) image dataset; (right) TypeDM model (as there are many <object - scene> relations, only a few are shown on the Y-axises). The number of relations in the TypeDM is much bigger than in the image model, which shows a more general model than the image one.

To look closer at the difference between the statistics obtained from the image and language data, we give an example of the conditional probabilities of an object given a scene $P(O|S)$ in Figure 37. We see that the distribution extracted from language (TypeDM) is much smoother and contains more relations than the image model since it has been trained on general and large text corpora. The distribution from image data on the other hand is more sparse and tailored to this specific dataset. For example, given a "store", the probability that there is a "table" is 1, given "highway", the probability of a "car" is also 1 in the image dataset, while the highest conditional probability of the language model is only less than 60%.

LANGUAGE MODELS FOR HUMAN ACTION RECOGNITION These experiments are designed to answer our second research question: how well the knowledge extracted from text can be used to aid human action recognition. To evaluate how the size of the datasets effect the results of the framework, we test our systems on both the 89 action dataset and the more general dataset, TUHOI. For the 89 action dataset, the action recognition system has been evaluated based on objects and scenes individually, and then study the integration of them. For the TUHOI dataset, since we only have the object and verb annotation (without scene), we only test the human action recognition based on objects.

Testing on the 89 action dataset

The training set of the 89 action dataset contains 1,104 images (for training the image relations) and the test set has 710 images. First, we test how the model predicts an action knowing the actual

	Image	**TypeDM**	**R-LDA**	**Window2**	**Window20**	**C.Net**
O_{gs}	0.3	16.1	63.4	16.4	18.3	86.1
O_{rec}	14.9	26.9	66.7	44.7	54.9	115.6
S_{gs}	35.7	181.7	174.9	168.5	174.8	252.5
S_{rec}	46.8	250.5	348	190.2	189.8	241.2
$O_{gs}S_{gs}$	**0.28**	**10.2**	**15.2**	**13.8**	**13.6**	**81.9**
$O_{rec}S_{rec}$	13.6	26.9	66.7	44.7	54.9	115.6

Table 22: Average rank over all images AR_I of the human action recognition using different settings: O_{gs},O_{rec} use only objects (gold standard and object recognizer); S_{gs},S_{rec} use only scenes, $O_{gs}S_{gs}$ and $O_{rec}S_{rec}$ integrate both objects and scenes together

object and/or scene appearing in an image (given object/scene gold standard), i.e., O_{gs}, S_{gs} and $O_{gs}S_{gs}$ in the settings. After that, we test a complete model which is based on the output of our object recognizer and our scene recognizer (O_{rec}, S_{rec}, $O_{rec}S_{rec}$).

For each setting, we try different action models, either learnt from the training images (Image), or from each of the language models (TypeDM, R-LDA, Window2, Window20, ConceptNet).

Table 22 presents the average ranking over all images. Results show that the action model learnt directly from the training images achieves the best performance in all settings, even if we give more weight to infrequent actions by taking the average ranking over all actions, as presented in Table 23. One explanation may be that the action dataset has a limited domain of only 19 objects, while the language models were learnt from broad knowledge (See Figure 37). To verify this hypothesis, we later run the experiments on the bigger and more general dataset, TUHOI. Another possibility is that verbs used for describing actions in images are more specific than verbs used in language. We will further discuss the use of specific verbs in language and images in the model adaptation experiments.

If we look at the performance of the language models, TypeDM performs best by a significant margin. This makes sense, as the most powerful term for predicting an action is obviously $P(V|O)$, and we saw earlier that TypeDM produces probabilities $P(V|O)$ which are closest to the image model. For the same reason, the second and third best language model are the Window2 and Window20 models, although their performance is significantly lower when using the predictions for objects and/or scenes. This is somewhat surprising considering that TypeDM, Window2 and Window20 are all very close in χ^2 distance to the image model as shown in Table 21. Of course, the distance is just an indication. R-LDA performs poorly because it is much more contextual. Finally, ConceptNet performs the worst.

Another observation is that using the scene identity should theoretically help in human action recognition: Using TypeDM, the use of the gold standard object identity yields an average ranking over all images of 16.1, while using both the scene and object identity yields an average ranking of 10.2, which is significantly better. It means that the use of the scene can disambiguate some actions (e.g. “ride a horse” vs. “feed a horse”). However, when using the recognition system, using the scene does not increase the overall performance (the parameters are optimized based on the average rank; and when including scene does not increase the result, one of the parameter controlling the weight of the scene takes the value 0). This shows that the visual recognition system may not be strong enough for recognizing these 15 scenes. Another problem may be the limitation of 15 scenes only: while annotating we frequently found that it was hard for numerous images to put them into one of the 15 scenes. So a bigger scene database may help.

The main problem with most available annotated human action datasets is that they are very restricted and domain-specific. For example, in this dataset with 19 objects and 15 scenes, there are

many photos of a person riding a motorbike on rocky mountains as a kind of sport. Consequently, the probability of "riding" given "mountain" learnt from the image dataset is high according to the image data (78%) but is uncommon in general. So the image dataset might be too restricted or biased for general knowledge to work well.

	Image	**TypeDM**	**R-LDA**	**Window2**	**Window20**	**C.Net**
AR_I	13.6	26.9	66.7	44.7	54.9	115.6
AR_A	16.4	30.8	64.7	45.3	51.9	131.7

Table 23: Average rank over all images vs. actions of the human action recognition using the $O_{rec}S_{rec}$ setting

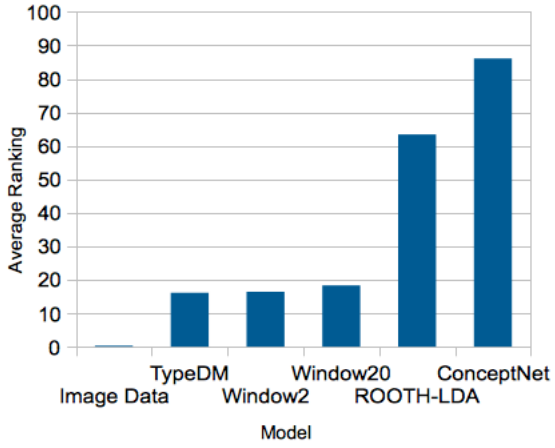


Figure 38: The average rank over all images based on object setting: O_{gs} when testing on the 89 action dataset

Retrieval experiments

In this section we carry out an image retrieval experiment. We compare our system with a state-of-the-art BoW implementation. This BoW implementation represents the complete image using a Spatial Pyramid [Lazebnik et al., 2006a] of 1x1 and 1x3. Results on the Pascal VOC 2007 classification challenge are 60.4 MAP (mean average precision), sufficiently close to the 61.7 MAP reported by Chatfield et al. [Chatfield et al., 2011].

As the BoW method needs training examples, we split our action dataset into two by using the predefined Pascal 2012 training and validation split. To be able to optimize the parameters of the SVM using cross-validation we demand that an action has at least two training examples. For evaluation, an action should have at least one test example. These constraints results in a data set with 44 actions (whereas our model can retrieve all 84 actions found in the language model (89 minus the 5 not present in TypeDM).

Results on the action retrieval task for the BoW approach and our proposed model are reported in Table 24. Surprisingly, our model with BoWL object recognizer outperforms the BoW approach: 0.22 vs. 0.19 MAP, respectively. The BoW method suffers, of course, from a lack of training examples. Yet our method has only seen the objects itself but never how an action looks like. Still, it gives results slightly better than the BoW system.

Action	Classic BoW	Unseen Felzen	Unseen BoWL	Action	Classic BoW	Unseen Felzen	Unseen BoWL	Action	Classic BoW	Unseen Felzen	Unseen BoWL
drive bus (25)	0.717	0.816	0.814	pat dog (10)	0.083	0.050	0.220	watch TV (8)	0.032	0.114	0.243
sail boat (23)	0.822	0.444	0.657	hold bird (3)	0.015	0.013	0.207	feed bird (2)	0.009	0.005	0.068
sit table (111)	0.678	0.352	0.652	walk horse (8)	0.226	0.064	0.201	touch horse (8)	0.040	0.027	0.062
ride motorbike (85)	0.553	0.448	0.609	hold dog (35)	0.210	0.140	0.191	walk dog (16)	0.144	0.088	0.061
ride horse (75)	0.594	0.669	0.607	get bus (6)	0.118	0.122	0.183	take bus (2)	0.362	0.049	0.054
feed sheep (7)	0.040	0.096	0.540	row boat (24)	0.473	0.105	0.182	stay boat (8)	0.024	0.019	0.032
sit chair (148)	0.410	0.406	0.468	touch cat (7)	0.071	0.041	0.173	sit car (7)	0.354	0.068	0.028
sit sofa (59)	0.371	0.299	0.458	touch dog (6)	0.236	0.028	0.164	play dog (11)	0.020	0.011	0.021
hold cat (19)	0.123	0.060	0.395	lay sofa (11)	0.086	0.034	0.160	touch motorbike (14)	0.100	0.020	0.020
ride bike (84)	0.440	0.489	0.378	drive train (4)	0.074	0.417	0.130	drink bottle (15)	0.024	0.013	0.019
drive car (23)	0.204	0.612	0.367	hold bottle (40)	0.158	0.160	0.126	feed cat (4)	0.007	0.172	0.018
take train (8)	0.108	0.149	0.356	sit motorbike (18)	0.132	0.162	0.118	carry dog (2)	0.003	0.008	0.007
get train (3)	0.031	0.181	0.339	hold bike (10)	0.045	0.056	0.087	push chair (2)	0.009	0.001	0.006
walk bike (14)	0.127	0.192	0.280	herd sheep (2)	0.002	0.006	0.077	feed bottle (5)	0.033	0.008	0.004
milk cow (2)	0.006	0.003	0.003	touch sheep (5)	0.032	0.002	0.002	**MAP**	**0.19**	**0.16**	**0.22**

Table 24: (Mean) Average Precision of Classical BoW and our approach which integrates a Felzen/BoWL object recogniser with TypeDM. The number of training examples for Classical BoW are in brackets.

We conclude that our system is able to achieve good performance in image retrieval on unseen actions. In a real-world scenario, where training data is limited, our system even outperforms a state-of-the-art BoW implementation.

Testing on the TUHOI dataset

The previous experiment has been done on a dataset with 19 object. To test how the system perform in a more general scenario, where there are many different objects and verbs can involve, we run the same experiment on the TUHOI dataset.

Since this dataset has been annotated by crowdflower with many different people, they contain some errors and misspellings. Therefore, we have first preprocessed the data as follows.

For verbs, we transform all verbs to the base form (e.g., riding $\rightarrow$ ride, sat $\rightarrow$ sit). Then we manually correct spelling mistakes (e.g., aapply $\rightarrow$ apply, drivw $\rightarrow$ drive, perforn $\rightarrow$ perform) and remove errors or spelling mistakes that are not possible to understand (e.g., pry, ay, prt). After that, we remove all prepositions (e.g., sit on $\rightarrow$ sit, support with $\rightarrow$ support, lay down $\rightarrow$ lay). For verbs that contain more than one word, they are reduced to one word (e.g., hit ball $\rightarrow$ hit, ski downhill $\rightarrow$ ski). We finally check the list of verbs manually and produce a mapping from original verbs taken from crowdsourcing and their corresponding processed verbs.

For objects, we also reduce all objects to one word: objects that contain more than one word (i.e., noun phrases such as computer keyboard, domestic cat, baby bed) are reduced to the head of the noun (keyboard, cat, bed). Finally, objects whose share the same heads (e.g., water bottle and wine bottle) are merged together.

Totally, we obtain 503 verbs and 173 objects, which leads to 173×503 (87,019) possible action combinations. The training and validation split follows the same split as in the Large scale object recognition challenge (DET). Totally, we have 10,843 images, of which 6,226 images are for training and 4,617 images are for testing. Note that only the model trained on image data used this training set, the rest with language model does not use these training images.

Similar to the previous experiment on the 89 action dataset, we calculate the average rank of the correct action found in each image, but for the first, second and third annotation. The results are depicted in Figure 39. It shows that the model trained from language models including TypeDM, R-LDA, Window2, Window20 are better than the model trained from the image data. The model TypeDM is the best model, which is consistent with our previous experiments. It confirms that

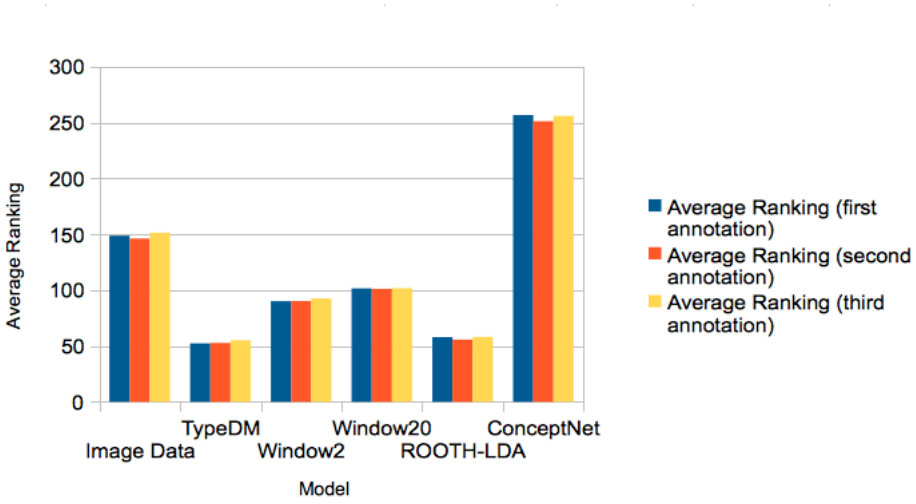


Figure 39: The average rank over all images based on object setting: O_{gs} when testing on the TUHOI dataset

TypeDM works best for the direct relations between verb and object. After that, the R-LDA model follows the TypeDM. After that, the Window2 model works better than the Window20. It also confirms that for relations such as verb and object, the smaller window model works better than the model with bigger window.

Furthermore, the results on three different annotations are consistent, which show that different annotations when testing on a general and big enough data does not matter, whether we test on the first, second or third annotations, the results are similar.

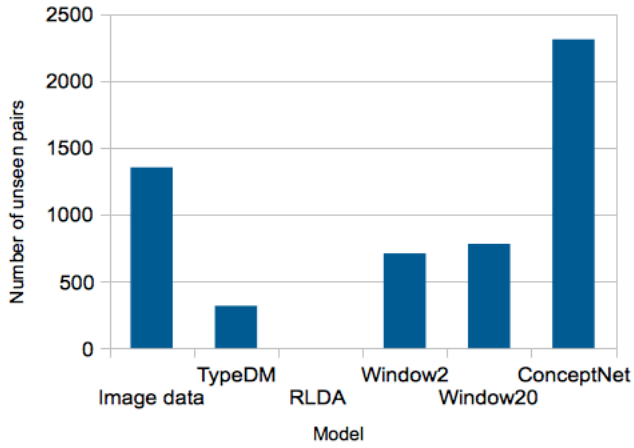


Figure 40: The number of unseen verb-object pairs in each model when testing on the TUHOI dataset

This experiment has shown that when testing on the big general dataset, the model with language model clearly outperforms the model learned from images.

One important reason for models that do not perform well in suggesting correct actions is that the correct verb-object pair in the test set has not been read in the text corpora. Figure 40 shows the

	AR_I	AR_A
Baseline	65.1	75.5
$\mathcal{T}_{L1}$	47.3	52.3
$\mathcal{T}_{L2}$	50.1	51.7
$\mathcal{T}_I$	62.9	66.9
$\mathcal{T}'_I$	43.6	49.0
Combine ($\mathcal{T}_{L1}$, $\mathcal{T}'_I$)	40.4	44.9
Flickr	49.6	56.1

Table 25: Results of the model adaptation with different tailoring function

number of unread[7] pairs when testing on the TUHOI dataset for each language model. Since R-LDA is a generative model, it assigns a probability (>0.0) to every pairs of verb and object, the number of unseen pair is 0. Other models on the other hand only report those that they have seen in the corpus before. When using the training image data, there are many pairs of verbs and objects that do not occur in this training set. The number of unseen pairs in TypeDM is less than those in the Window2 and Window20 model. Finally, the relations extracted from ConceptNet have the most unseen pairs, which leads to the poor performance of this model. However, the number of unseen pairs should not be considered as a main indicator of a good model, as we have seen that TypeDM even works better than R-LDA regardless of its higher number of unseen pairs.

MODEL ADAPTATION To answer our third research question about model adaptation, we conduct the following experiments. We examine whether adapting the language models could help increase the result of the human action recognition in images. We compare the baseline, which is the previous $O_{rec}S_{rec}$ setting using TypeDM language model with the adapted model using different tailoring function. In this experiment, we do not limit the verbs as to only those appearing in the training set as the previous experiment, but use the whole list of verbs the language model proposes.

The results reported in Table 25 show that all adapting functions work well in helping to improve the results of human action recognition in images. Especially, the combined model where verbs are re-ranked based on $\mathcal{T}_{L1}$ (learning the generality of a verb based on its frequency of the number of objects it goes with) and $\mathcal{T}'_I$ (learning the number of objects each verb go with using the image data from PASCAL) work best with decreasing the average position from 65.1 to 40.4. Using out-domain image dataset, i.e., Flickr, we also obtained an increase in the performance (from 65.1 to 49.6). However, the use of Flickr does not outperform the use of general language data.

Generally, model adaptation is promising in increasing the performance of the human action recognition system. It can be used to adapt the language models to fit better the image data that we apply to.

HUMAN ACTION CLASSIFICATION WITH POSITION INFORMATION The purpose of the following experiments is to evaluate whether the position information between human and object in images can be mapped to prepositions in languages to help recognizing human actions. First, to compare whether position information can help in recognizing actions or not, we design a naive classifier which learns from the probability of a verb given an object to assign an action for each image from the training image dataset.

7 we use "unread" actions to denote pairs of a verb and an object that never occur in *text data*, and "unseen" for actions that never occur in training *image data*.

	Accuracy	Precision	Recall	F1
Without position	74.2%	0.40	0.26	0.29
With position	72.1%	0.65	0.29	0.36

Table 26: Results of the classifier with and without position information

Object	Without position	With position	Object	Without position	With position
baseball	0.36	0.52	bus	0.57	0.73
face powder	0.33	1	hair spray	0.73	0.74
harmonica	0.07	0.97	horizontal bar	0.42	0.45
hotdog	0.29	0.57	motorcycle	0.80	0.82
turtle	0.43	0.71	water bottle	0.56	0.65

Table 27: Objects with higher accuracy when using position information

The results of the systems with and without position are report in Table 26. It shows that the accuracy of the classifier without position is higher than when including the position (74.2% in comparison to 72.1%). However the precision, recall and F1 of the classifier using position are all higher than without position. It's due to the fact that the classifier without position blindly assigns each image to the most probable action (i.e., actions that occurs most often with a given object learned from the training set), so it obtains better overall accuracy when testing on all images. However, for other possible actions, this classifier is unable to disambiguate actions and the performance of this classifier on less frequent actions is worse than when including position information into the classifier. Generally, when taking into account all possible actions, the position-based classifier has better average precision, recall and F1 score (28.6% without position in comparison with 35.8% using position).

To further analyze which objects and actions, the position information helps better, we compare the accuracy of each individual objects. Table 27 reports main objects that have higher accuracy when using position. We want to be able to predict which kind of actions that positions will help in recognizing it through the knowledge we learn from language. This prediction will help us to learn how to include the position information inside our human action recognizer since not all actions can be disambiguated by positions. We divide the actions into two groups: one group for which we found position information increase the classification results, a second group for which we found position information to decrease the classification results. Results about these two groups are reported below.

FROM PREPOSITIONS IN LANGUAGE TO RELATIVE POSITIONS BETWEEN HUMAN AND OBJECT IN IMAGES The aim of this experiment is to learn whether the prepositions in language can be mapped to the position information in images to help with human action recognition. In particular, we want to learn if and how prepositions in language can be used to determine which positions are useful in action classification, i.e., if they belong to the first group or the second group in the previous experiment.

The relative positions between human and object in images are useful in analyzing their interactions. For example, when a person is riding a horse, the person is usually on the top of the horse, and when a person is feeding a horse, then the person is usually standing next to the horse. In English, sometimes prepositions can be used as an indicator to the relations between human and object positions.

We want to exploit the connection between human-object positions in images and prepositions that link human, verb and object in language. Intuitively, if an action implies a strong positional relation

between the human and the object, we expect to find specific, distinguishing prepositions in language. For example, in language you usually say "sit *on* chair", where the preposition *on* suggests a specific spatial relation between the human and the chair. When an action does not imply a strong positional relation, such as "play", we expect no specific prepositions.

Link *in language models*

To test this hypothesis, we use TypeDM [Baroni and Lenci, 2010], a distributional memory that has been built from large scale text corpora. This model contains weighted <word-link-word> tuples extracted from a dependency parse of corpora. The relations between words are characterized by their "link". Some of these links are prepositions that connect verbs and objects together. Examples of some tuples with word-link-word and their weights are provided in Figure 41.

bicycle-n	by	ride-v	11.2994
bicycle-n	in	ride-v	6.7795
bicycle-n	of	ride-v	2.4167
bicycle-n	on	ride-v	278.4273
drum-n	against	play-v	3.5056
drum-n	behind	play-v	4.7656
drum-n	by	play-v	2.4393
drum-n	in	play-v	8.9440
drum-n	of	play-v	2.9940
drum-n	on	play-v	185.8888
drum-n	over	play-v	2.8841
accordion-n	on	play-v	174.7606
ant-n	over	hold-v	3.3807
apple-n	in	hold-v	0.3519
apple-n	on	hold-v	1.1309

Figure 41: Examples of word-link-word and their weights in the distributional memory

Number of links and link entropy

We want to determine whether there is any correlation between human-object relative positions in images and the associated prepositions from language models. To do this, we record two metrics: the number of links, where we count how many different links that connect verbs and objects in the language model; and the entropy of each action A^i verb-object pair (where the human is implicit) is $H(A^i)$ defined by: $H(A^i) = -\sum_{l_j \in L^i} p(l_j) \times \log p(l_j)$

where L^i is the set of all links that occur between verb and object of action i; $p(l_j)$ is the probability of the link l_j of the action A^i:

$$p(l_j) = \frac{weight(l_j)}{\sum_{l_k \in L_i} weight(l_k)} \tag{41}$$

where $weight(l_j)$ is the weight given by the TypeDM of link j in action i.

Generally, the entropy for each action allows seeing whether a link is predictable for a given pair of verb-object or not: when a link is predictable, the entropy is expected to be low (contain little information), which might correspond to the case that the position information will be useful in predicting actions and the other way around.

RESULTS The result shown in Table 28: for the first group (with position is better), the average number of links per relation (verb - object) is 8 and the average entropy is 1.05; the average number of links per relation for the second group is almost twice more, 15.3, and their average entropy is also higher, 1.36. It shows that verbs which have many different ways of linking to an object might not have a *representative* relative position between the person and object, hence more difficult to be

	Number of links	Entropy
Group 1 (position helps)	8	1.05
Group 2 (position doesn't help)	15.3	1.36

Table 28: Actions that can be disambiguated by positions (Group 1) vs. actions that cannot be disambiguated by positions (Group 2) and their links in the language model

classified based on their positions. Verbs that have less links to an object tend to have more *fixed* relative positions between persons and objects, hence it might be helpful to use position information in classification.

A qualitative analysis

We further examine actions where this statement does not apply, i.e., actions with high number of links and high entropy but belong to group 1 (position information helps) and actions with low number of links and entropy belonging to group 2. For the first case, typical actions which have high number of links/entropy are: ride car, ride bus, ride train, pull cart, light lamp. The large number of links of these actions seem to come from relations which do not describe the human/object interaction itself. For example, the links associated with 'ride bus' do not all actually refer to 'ride a bus' but to ride another object in a position with respect to the bus: ride after bus, ride behind bus, ride before bus. These cause extra links which are not related to the action itself. Similarly, actions pull of/around/behind/below/on cart, there is another object which is moved to a specific position with regards to the cart.

For the second case, examples of typical actions with low links but for which positions information doesn't help are hold harmonica, wear diaper, hold ladle, spread cream, hold racket, apply lipstick. These actions are related to objects, for which their positions depends a lot on the human pose (e.g., hold something). These actions in the language model do not contain many links as we expected: the most possible link between hold, harmonica is *in*, which probably means hold harmonica *in* your hand.

Instead of looking at actions, we look into typical verbs where position information helps in classifying actions and verbs where position information doesn't help. For the first group, the most frequent verbs are: chop, cut, drink, feed, lean, sit on, sleep, look at, put on, shake, shoot, wash, catch. For the second group, the most frequent verbs are: clean, cook, lift, punch, sing, spray, spread. It can be observed that verbs related to some particular poses or relative positions between human and object are better with the position information (chop, drink, sit on, sleep), and verbs related to more various human poses and unspecific are not helped by the position information (cook, sing, spray, clean).

Generally, there is a relation between prepositions in language and the relative positions between human-object in images. Although this statement does not hold in every cases, for example when the prepositions refer to the positions between another action (e.g., ride) and that object (e.g., after a bike), this can be potentially solved by better NLP parsing and analyses of verb phrases. Furthermore, actions that cannot be disambiguated by positions are usually related to different human poses, while actions that have some particular human poses can be classified using position information.

ANSWER TO THE RESEARCH QUESTIONS To sum up, the above experiments have shown that: (1) The knowledge about objects and scenes extracted from the language model R-LDA is the most compatible to the knowledge extracted from the images. The knowledge about the relations between verbs and objects extracted from TypeDM on the other hand is the most compatible to

the one from the images. (2) The knowledge from text can aid human action recognition, in a realistic scenario where few training examples are available, our system based on language models outperforms a state-of-the-art Bag-of-Words approach. (3) The model adaptation can help to further improve the results of the recognizer, by adjusting the language model to propose words that are more frequently used to describe images. (4) When evaluating the system on a big and general dataset with more objects and verbs involved, the system learned from text can even outperform the model with the knowledge learnt from expensive annotated images. (5) The position information is valuable in helping to determine the correct actions in images. However, how to efficiently learn the relations between the position information and actions is still an open question. Mapping between positions and preposition in language is promising in helping the system to recognize better actions.

5.7 CHAPTER SUMMARY

In this chapter, we have presented our knowledge-based framework in the computer vision domain. In particular, we investigated the problem of applying knowledge learnt from language corpora to visual recognition. We compared statistics of various language models mined on general corpora with statistics observed in image datasets. It shows that the generative R-LDA model is good at relating contextual relations (e.g., object - object, object - scene), while the syntactic based distributional model TypeDM is good at representing direct relations such as verb - object in images.

We have evaluated the performance of the language models in two visual scenarios: human action recognition and object prediction. It suggests that the language models need some tailoring when applied to restricted datasets. However, predicting objects and recognizing human actions learned from language model when testing on a big, general dataset outperforms the model learned from annotated images. This shows that language models built from available text corpora can be used for visual recognition instead of expensive annotated image data.

6

CONCLUSIONS

In this thesis, we tackle the problem of dealing with poor data in machine learning by enriching it with knowledge automatically learned from large, general text corpora. We propose a general framework that takes advantage of available text data to improve the performance of machine learning systems that work with poor data in quality or quantity. To extract the knowledge from text data, we have employed different text modeling techniques, such as distributional semantics, topic models. We have illustrated the framework in machine learning tasks in language and vision applications.

For language applications, we have exploited topic models estimated on large text corpora to enrich short queries with the knowledge learned from general text to improve query classification. The results show an increase in the performance of the classification system when including the knowledge we exploited through topic modeling.

For vision applications, we have applied the general framework to two tasks: object prediction and human action recognition in images. We have studied the problem of selecting proper techniques to extract knowledge from text for each particular task and how to include this knowledge to the visual system. Many visual recognition tasks require sufficient training annotated images, while these annotations are expensive and sometimes even infeasible. Our results have shown that using the knowledge learned from text can be used instead of expensive annotated images, especially for recognition tasks that require many training samples and such samples are not fully available.

In the future, we want to apply and extend this framework to other visual recognition tasks that require semantic knowledge such as event recognition, human - human interaction recognition. We plan to further study the link between language and vision to use the knowledge learned from language to aid visual recognition: for instance, the link between prepositions in language and positions in vision, adjective in language and colors/shape in vision.

BIBLIOGRAPHY

[0003 and Hauptmann, 2008] 0003, J. Y. and Hauptmann, A. G. (2008). (un)reliability of video concept detection. In Luo, J., Guan, L., Hanjalic, A., Kankanhalli, M. S., and Lee, I., editors, *CIVR*, pages 85–94. ACM.

[Baroni and Lenci, 2010] Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

[Beitzel et al., 2005] Beitzel, S. M., Jensen, E. C., Frieder, O., and Grossman, D. (2005). Automatic web query classification using labeled and unlabeled training data. In *In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 581–582. ACM Press.

[Bernardi and Le, 2011] Bernardi, R. and Le, D.-T. (2011). Metadata enrichment via topic models for author name disambiguation. *Advanced Language Technologies for Digital Libraries, Hot Topic series, Springer*.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[Broder, 2002] Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36:3–10.

[Broder et al., 2007] Broder, A. Z., Fontoura, M., Gabrilovich, E., Joshi, A., Josifovski, V., and Zhang, T. (2007). Robust classification of rare queries using web knowledge. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 231–238, New York, NY, USA. ACM.

[Brown, 2005] Brown, K., editor (2005). *Encyclopedia of Language and Linguistics, 2nd edition*. Elsevier Ltd, Oxford.

[Bruni et al., 2012] Bruni, E., Boleda, G., Baroni, M., and Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL. ACL.

[Buchanan and Shortliffe, 1984] Buchanan, B. G. and Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Buchann and Shortliffe.

[Cao et al., 2009] Cao, H., Hu, D. H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., and Yang, Q. (2009). Context-aware query classification. In *SIGIR'09, The 32nd Annual ACM SIGIR Conference.*

[Chatfield et al., 2011] Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*.

[Church and Hanks, 1990] Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29.

[Csurka et al., 2004] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual Categorization with Bags of Keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

[Delaitre et al., 2010] Delaitre, V., Laptev, I., and Sivic, J. (2010). Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[Delaitre et al., 2011] Delaitre, V., Sivic, J., and Laptev, I. (2011). Learning person-object interactions for action recognition in still images. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 24*, pages 1503–1511. Curran Associates, Inc.

[Desai and Ramanan, 2012] Desai, C. and Ramanan, D. (2012). Detecting actions, poses, and objects with relational phraselets. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 158–172, Berlin, Heidelberg. Springer-Verlag.

[Desai et al., 2010] Desai, C., Ramanan, D., and Fowlkes, C. (2010). Discriminative models for static human-object interactions. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–16.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. IJCV.

[Everingham et al., 2012] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[Farhadi et al., 2010a] Farhadi, A., Hejrati, M., Sadeghi, A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010a). Every picture tells a story: Generating sentences for images. In *ECCV*.

[Farhadi et al., 2010b] Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010b). Every picture tells a story: generating sentences from images. In *European Conference on Computer Vision ECCV*. Springer.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *TPAMI*.

[Gildea and Jurafsky, 2002] Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288.

[Goddard, 2011] Goddard, C. (2011). *Semantic Analysis: A Practical Introduction*. Oxford Textbooks in Linguistics. OUP Oxford.

[Gorelick et al., 2005] Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *In ICCV*, pages 1395–1402.

[Graff and Cieri, 2003] Graff, D. and Cieri, C. (2003). English gigaword. In *Linguistic Data Consortium*.

[Griffiths and Steyvers, 2004] Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235.

[Gupta et al., 2009a] Gupta, A., Kembhavi, A., and Davis, L. S. (2009a). Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1775–1789.

[Gupta et al., 2009b] Gupta, A., Kembhavi, A., and Davis, L. S. (2009b). Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10).

[Heinrich, 2004] Heinrich, G. (2004). Parameter estimation for text analysis. Technical report.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

[Ikizler et al., 2008] Ikizler, N., Cinbis, R., Pehlivan, S., and Duygulu, P. (2008). Recognizing actions from still images. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4.

[Ikizler-Cinbis et al., 2009] Ikizler-Cinbis, N., Cinbis, R. G., and Sclaroff, S. (2009). Learning actions from the web. In *ICCV*, pages 995–1002. IEEE.

[Khan et al., 2013] Khan, F. S., Rao, M. A., van de Weijer, J., Bagdanov, A. D., Lopez, A., and Felsberg, M. (2013). Coloring action recognition in still images. *International Journal of Computer Vision (IJCV)*, 105(3):205–221.

[Kulkarni et al., 2011a] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A., and Berg, T. (2011a). Baby talk: Understanding and generating simple image descriptions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1601–1608.

[Kulkarni et al., 2011b] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A., and Berg, T. (2011b). Babytalk: Understanding and generating simple image descriptions. In *CVPR*.

[Landauer and Dutnais, 1997] Landauer, T. K. and Dutnais, S. T. (1997). A solution to platoÕs problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.

[Laptev, 2005] Laptev, I. (2005). On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123.

[Lazaridou et al., 2014] Lazaridou, A., Bruni, E., and Baroni, M. (2014). Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1403–1414.

[Lazebnik et al., 2006a] Lazebnik, S., Schmid, C., and Ponce, J. (2006a). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, New York.

[Lazebnik et al., 2006b] Lazebnik, S., Schmid, C., and Ponce, J. (2006b). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition CVPR*, volume 2, pages 2169–2178.

[Le et al., 2013a] Le, D., Bernardi, R., and Uijlings, J. (2013a). Exploiting language models to recognize unseen actions. In *ICMR*.

[Le et al., 2014] Le, D., Bernardi, R., and Uijlings, J. (2014). TUHOI: trento universal human object interaction dataset. In *Vision and Language Workshop at COLING*.

[Le et al., 2013b] Le, D., Uijlings, J., and Bernardi, R. (2013b). Exploiting language models for visual recognition. In *EMNLP*.

[Le and Bernardi, 2012] Le, D.-T. and Bernardi, R. (2012). Query classification using topic models and support vector machine. In *Proceedings of ACL 2012 Student Research Workshop*, ACL '12, pages 19–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Le et al., 2013c] Le, D. T., Bernardi, R., and Uijlings, J. (2013c). Exploiting language models to recognize unseen actions. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, ICMR '13, pages 231–238, New York, NY, USA. ACM.

[Le et al., 2011] Le, D.-T., Bernardi, R., and Vald, E. (2011). Query classification via topic models for an art image archive. In *Recent Advances in Natural Language Processing, RANLP, Bulgaria*.

[Le et al., 2008] Le, D.-T., Nguyen, C.-T., Ha, Q.-T., Phan, X. H., and Horiguchi, S. (2008). Matching and ranking with hidden topics towards online contextual advertising. In *Web Intelligence*, pages 888–891. IEEE.

[Lecun et al., 2006] Lecun, Y., Chopra, S., Hadsell, R., Huang, F. J., Bakir, G., Hofman, T., Schölkopf, B., Smola, A., and Eds, B. T. (2006). A tutorial on energy-based learning. In *Predicting Structured Data*.

[Lederberg, 1987] Lederberg, J. (1987). How dendral was conceived and born. In *Proceedings of ACM Conference on History of Medical Informatics*, HMI '87, pages 5–19, New York, NY, USA. ACM.

[Li et al., 2008] Li, X., Wang, Y.-Y., and Acero, A. (2008). Learning query intent from regularized click graphs. In *SIGIR'08*.

[Liu and Singh, 2004] Liu, H. and Singh, P. (2004). Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22:211–226.

[Liu et al., 2011] Liu, J., Kuipers, B., and Savarese, S. (2011). Recognizing human actions by attributes. In *Conference on Computer Vision and Pattern Recognition CVPR*.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA. IEEE Computer Society.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60:91–110.

[Lund and Burgess, 1996a] Lund, K. and Burgess, C. (1996a). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28(2):203–208.

[Lund and Burgess, 1996b] Lund, K. and Burgess, C. (1996b). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*.

[Maji et al., 2008] Maji, S., Berg, A. C., and Malik, J. (2008). Classification using Intersection Kernel Support Vector Machines is Efficient. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Moosmann et al., 2008] Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized Clustering Forests for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:1632–1646.

[Moosmann et al., 2006] Moosmann, F., Triggs, B., and Jurie, F. (2006). Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992.

[Newman et al., 2007] Newman, D., Hagedorn, K., Chemudugunta, C., and Smyth, P. (2007). Subject metadata enrichment using statistical topic models. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '07, pages 366–375, New York, NY, USA. ACM.

[Osadchy et al., 2007] Osadchy, M., Cun, Y. L., and Miller, M. L. (2007). Synergistic face detection and pose estimation with energy-based models. *J. Mach. Learn. Res.*, 8:1197–1215.

[Osgood et al., 1957] Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1957). *The Measurement of Meaning*. University of Illinois Press.

[Peters, 1993] Peters, T. A. (1993). The history and development of transaction log analysis. *Library Hi Tech*, 11(2):41–66.

[Phan et al., 2010] Phan, X.-H., Nguyen, C.-T., Le, D.-T., Nguyen, L.-M., Horiguchi, S., and Ha, Q.-T. (2010). A hidden topic-based framework towards building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints).

[Prendergast and Winston, 1984] Prendergast, K. A. and Winston, P. H. (1984). *The AI business: the commercial uses of artificial intelligence; 2nd ed.* MIT, Cambridge, MA.

[Prest et al., 2012] Prest, A., Schmid, C., and Ferrari, V. (2012). Weakly supervised learning of interactions between humans and objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):601–614.

[Ritter et al., 2010] Ritter, A., Mausam, and Eztioni, O. (2010). A latent dirichlet allocation method for selectional preferences. In *Association for Computational Linguistics*.

[Rooth et al., 1999] Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via em-based clustering. In *Association for Computational Linguistics*.

[Rose and Levinson, 2004] Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 13–19, New York, NY, USA. ACM.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

[Séaghdha, 2010] Séaghdha, D. O. (2010). Latent variable models of selection preference. In *ACL*.

[Sener et al., 2012] Sener, F., Bas, C., and Ikizler-Cinbis, N. (2012). On recognizing actions in still images via multiple features. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *Computer Visionâ ECCV 2012. Workshops and Demonstrations*, volume 7585 of *Lecture Notes in Computer Science*, pages 263–272. Springer Berlin Heidelberg.

[Shen et al., 2006a] Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., and Yang, G. (2006a). Query enrichment for web-query classification. *ACM Transactions on Information Systems*, 24(3):320–352.

[Shen et al., 2006b] Shen, D., Sun, J.-T., Yang, Q., and Chen, Z. (2006b). Building bridges for web query classification. In *SIGIR'06*.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*.

[Speer and Havasi, 2012] Speer, R. and Havasi, C. (2012). Representing general relational knowledge in conceptnet 5. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *LREC*, pages 3679–3686. European Language Resources Association (ELRA).

[Speer and Havasi, 2013] Speer, R. and Havasi, C. (2013). Conceptnet 5: A large semantic network for relational knowledge. In *The People's Web Meets NLP*. Springer Berlin Heidelberg.

[Srikanth et al., 2005] Srikanth, M., Varner, J., Bowden, M., and Moldovan, D. (2005). Exploiting ontologies for automatic image annotation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 552–558, New York, NY, USA. ACM.

[Teo et al., 2012] Teo, C., Yang, Y., Daume, H., Fermuller, C., and Aloimonos, Y. (2012). Towards a watson that sees: Language-guided action recognition for robots. In *2012 IEEE International Conference on Robotics and Automation ICRA*.

[Uijlings et al., 2010] Uijlings, J. R. R., Smeulders, A. W. M., and Scha, R. J. H. (2010). Real-time Visual Concept Classification. *IEEE Transactions on Multimedia*, 12.

[Uijlings et al., 2013] Uijlings, J. R. R., van de Sande, K., Gevers, T., and Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*.

[Ushiku et al., 2012] Ushiku, Y., Harada, T., and Kuniyoshi, Y. (2012). Efficient image annotation for automatic sentence generation. In *ACM MM*.

[van de Sande et al., 2008] van de Sande, K. E., Gevers, T., and Snoek, C. G. (2008). A comparison of color features for visual concept classification. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, CIVR '08, pages 141–150, New York, NY, USA. ACM.

[van de Sande et al., 2011] van de Sande, K. E. A., Uijlings, J., Gevers, T., and Smeulders, A. (2011). Segmentation as Selective Search for Object Recognition. In *ICCV*.

[Wang et al.,] Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. Dense trajectories and motion boundary descriptors for action recognition.

[Wang et al., 2006] Wang, Y., Jiang, H., Drew, M. S., Li, Z.-N., and Mori, G. (2006). Unsupervised discovery of action classes. In *CVPR*.

[Weinland et al., 2006] Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Comput. Vis. Image Underst.*, 104(2):249–257.

[Xiao et al., 2010] Xiao, J., Hays, J., Ehinger, K., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition CVPR*.

[Yang et al., 2010] Yang, W., Wang, Y., and Mori, G. (2010). Recognizing human actions from still images with latent poses. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2030–2037.

[Yang et al., 2011] Yang, Y., Teo, C. L., Daumé, III, H., and Aloimonos, Y. (2011). Corpus-guided sentence generation of natural images. EMNLP, Stroudsburg, PA, USA.

[Yang et al., 2013] Yang, Y., Teo, C. L., Fermüller, C., and Aloimonos, Y. (2013). Robots with language: Multi-label visual recognition using NLP. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 4256–4262.

[Yao and Fei-Fei, 2010] Yao, B. and Fei-Fei, L. (2010). Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24.

[Yao and Fei-Fei, 2012] Yao, B. and Fei-Fei, L. (2012). Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703.

[Yao et al., 2011a] Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L., and Fei-Fei, L. (2011a). Human action recognition by learning bases of action attributes and parts. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1331–1338, Washington, DC, USA. IEEE Computer Society.

[Yao et al., 2011b] Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L. J., and Fei-Fei, L. (2011b). Action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision ICCV*.

[Yao et al., 2011c] Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L. J., and Fei-Fei, L. (2011c). Action recognition by learning bases of action attributes and parts. In *ICCV*.

[Yao and Li, 2010] Yao, B. and Li, F.-F. (2010). Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, pages 9–16.

[Yu et al., 2011] Yu, X., Fermuller, C., Teo, C. L., Yang, Y., and Aloimonos, Y. (2011). Active scene recognition with vision and language. In *Proceedings of the 2011 International Conference on Computer Vision*, International Conference on Computer Vision ICCV.

[Zhang et al., 2007] Zhang, J., MarszaÅĆek, M., Lazebnik, S., and Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.