

***Textual Inference for Retrieving Labeled Object
Descriptions***

Alicia Tribble

CMU-10-003

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Scott E. Fahlman, Chair
Eric Nyberg
Carolyn Penstein Rosé
Bruce W. Porter, University of Texas, Austin
Vibhu O. Mittal, Root -1 Research

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies*

© 2010, Alicia Tribble

Textual Inference for Retrieving Labeled Object Descriptions

Alicia Tribble

April 2010

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committe:

Scott E. Fahlman, Chair

Eric Nyberg

Carolyn Penstein Rosé

Bruce W. Porter, University of Texas, Austin

Vibhu O. Mittal, Root-1 Research

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy*

Copyright © 2010 Alicia Tribble

Abstract

This thesis presents a knowledge-based solution for retrieving English descriptions for objects, such as images, in a collection. Based on detailed analysis of the errors made by a baseline system relying on surface-level features (i.e. term frequency), we infer that an ideal solution to this problem should use deeper representations of the meaning encoded in textual descriptions.

Applied Textual Inference (ATI) as used in this thesis refers to the class of generic task-based evaluations that address this need. ATI tasks are challenge problems. Because they are intended to drive research on text understanding, the problems are designed to be hard enough to require reasoning. However in order to support cross-site comparisons of results, the problems are evaluated at the surface level. Examples include recognizing textual entailment (RTE), paraphrasing, summarization, word-replacement, and some types of question answering (QA).

This thesis frames the problem of image description retrieval as an instance of ATI, and demonstrates how an inference engine and a set of symbolic knowledge resources in the form of ontologies can improve performance on this task, as measured by Mean Reciprocal Rank.

In the process, we describe the results of several sub-tasks: Introduce an image retrieval task supported by a data set containing over 50,000 images, hand-labeled with multiple descriptions; present a series of parameterizations for calculating the similarity between two descriptions; identify classes of error in a keyword-driven baseline system and use these classes to inform a set of knowledge-based improvements; implement and evaluate the knowledge-based approach.

The success of shared tasks for ATI in the last decade indicates growth in the field of Natural Language Understanding, and in particular a growing interest in deep text representations that can be leveraged by modern machine learning frameworks. The work of this thesis contributes to better understanding of why deep representations are necessary, and how they may be effectively applied.

Acknowledgments

This work would not have been possible without the support of a broad community of collaborators and friends who have been with me over seven years and two hemispheres.

I thank my advisor, Scott E. Fahlman, for making countless contributions to this work while placing so few constraints on it. At every turn he has offered help to make this thesis possible, as well as challenges to make it better. I also thank Penny Fahlman for being a part of the team that made this feel like a true collaboration.

Thank you to the members of my committee: Carolyn Penstein Rosé, Eric Nyberg, Vibhu Mittal, and Bruce Porter. They have persevered with me and continued to invest their time and energy in this work beyond what I could have expected from them.

Thanks to Luis von Ahn and Laura Dabbish, who ingeniously gathered the data set I used in most of my experiments.

Thank you to my classmates at CMU who have all experienced the unique mixture of pride and pain that this effort exacts: Kathrin, Joy, Paul, Kornel, Guy, Arthur, Stephanie, Rosie, Ari, Vasco, Ashish, Yan, Ben, and many, many more. Thank you to Stacy Young and to all of the staff at the LTI, who were able and flexible in following 7 years' worth of bread crumbs, which I blithely scattered on my way to getting this degree. Thank you to Stephan Vogel and to Alex Waibel for giving me responsibilities early in my graduate career that helped me to grow.

Also to my collaborators and friends at the Tsujii Laboratory at the University of Tokyo, who taught me so much and set such wonderful examples for me: Ohta-san, Kim-san, Rune, Kano-san, Miyao-san, and all of the students and researchers. And during the final stretch of this effort I have had still another amazing team to inspire me: thank you to Andre Valente and Lewis Johnson, founders of Alelo, where I have worked for almost two years now while finishing my dissertation. They saw my success as their success from the moment we started working together, which is a common thread among all of these treasured relationships.

Thank you to my family and their families: Hal, Suzie, Scottie, Beth, Paul, Amy, Jerry, Angela, Toshiro, Yuli, Mali, Andre, and Brian. I know this journey was long for you, too.

Finally thank you to my husband Kenji, who assured me that this process would be painful but that it would end happily. Thank you for helping me

endure the first and believe in the second.

All remaining inaccuracies and omissions in this work are my own.

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract NBCHD030010. Any opinions, findings, and conclusions expressed here are those of the author and do not necessarily reflect the views of the sponsors.

Contents

Abstract	2
Acknowledgments	3
List of Figures	11
List of Tables	14
1 Introduction	15
1.1 Background	15
1.2 Applied Textual Inference (ATI)	16
1.3 Current Approaches	17
1.4 Retrieving Object Descriptions	17
1.4.1 Retrieving Images	18
1.5 Thesis Outline	20
1.6 Conclusions	21
2 Related Work	23
2.1 Applied Textual Inference with Semantic Distance	23
2.2 Knowledge Acquisition and Development	25
2.3 Ad-Hoc Retrieval and Question Answering	25
2.4 Image Retrieval	27
3 A Corpus for Description Retrieval	31
3.1 The Phetch Data Set	31
3.2 Comparable Data Sets	33
3.3 Establishing Corpus Sections	34
3.4 Logical Document Structure	36
3.5 Conclusions	36
4 Understanding Image Descriptions	39
4.1 The Language of Descriptions	39

4.1.1	Syntactic Patterns	39
4.1.2	Semantic and Rhetorical Patterns	40
4.2	The Commonness of Descriptions	44
4.2.1	An Uncurated Data Sample from Flickr.com	44
4.2.2	Syntactic Patterns	47
4.2.3	Semantic and Rhetorical Patterns	47
4.3	Conclusions	49
5	Parameterizations and Baseline Results	51
5.1	Retrieval with Keywords	51
5.1.1	Focused Retrieval	52
5.1.2	Indexing and Retrieval Tools	53
5.1.3	Procedure	54
5.1.4	Results	58
5.2	Additional Parameterizations	58
5.2.1	Keyword Refinements	59
5.2.2	Synonyms and Semantic Expansion Terms	60
5.2.3	Dependency Relations	61
5.2.4	Knowledge-Augmented Dependency Relations	62
5.3	Conclusions	62
6	Classifying Errors	67
6.1	Introduction	67
6.2	Counting Retrieval Errors	67
6.2.1	Retrieval Failures	68
6.2.2	Estimating Bounds on Improvement	70
6.2.3	Additional Data	73
6.3	Classifying Retrieval Errors	73
6.3.1	Classes of Precision Error	76
6.3.2	Classes of Recall Error	78
6.3.3	Frequency of Errors by Class	78
6.3.4	Linguistic Features Contributing to Error	80
6.4	Conclusions	85
7	Applied Textual Inference Methods and Results	87
7.1	Introduction	87
7.2	Annotation with Ontology Elements	88
7.2.1	Procedure	88

7.2.2	Results	90
7.3	Graph Distance with Dependency Structures	91
7.3.1	Procedure	92
7.3.2	Reranking	95
7.3.3	Results	101
7.4	Analysis	101
7.4.1	Effect of Query Formulation	101
7.4.2	Effect of Semantic Graph Features	102
7.4.3	Effect of Text within Images	104
7.5	Conclusions	105
8	Knowledge Resources	107
8.1	Scone Knowledge Base System	107
8.2	Retrieval with WordNet	108
8.2.1	Procedure	110
8.2.2	Results	110
8.3	Improved Knowledge Base Structure	112
8.3.1	Upper Levels: WordNet + DOLCE	112
8.3.2	Acquiring Knowledge from Training Data	113
8.4	Retrieval with SconeImage Ontologies	115
8.4.1	Procedure	115
8.4.2	Results	117
8.5	Conclusions	118
9	Conclusions	119
9.1	Experiments and Findings	119
9.2	Contributions	121
9.2.1	Summary of Contributions	121
9.2.2	Refined Vocabulary for Sources of Error	122
9.3	Future Work	124
9.3.1	Summary of Future Work	124
9.3.2	Extension to Other ATI Tasks	126
9.4	Conclusions	130
	Bibliography	131
A	Sample Parameter Files	141

B Upper-level Ontology: DOLCE	143
C Upper-level Ontology: WN+DOLCE	155

List of Figures

1.1	Sample retrieval results for the query “people petting their dogs”	19
3.1	Image from the Phetch data set with descriptions and tags. . .	33
3.2	An sgml representation for structured Phetch documents. . . .	37
4.1	Dependency analysis, as a list and as a visual annotation. . .	41
4.2	Common dependency patterns of 2-4 words.	42
4.3	Frequency of dependency patterns across phrases (col 2) and images (col 3).	43
4.4	Single-word titles with hidden syntactic structure from Flickr.com. 45	
4.5	Common dependency patterns from Flickr titles and descriptions.	46
4.6	Percent of Flickr Descriptions (col 2) and Titles (col 3) where the most common dependency patterns appear.	47
5.1	Matching a query against indexed descriptions based on keywords. Terms shown in bold are features shared by both descriptions.	52
5.2	Indri structured queries.	55
5.3	An example image composed mostly of text. Such images were pruned from the 5A data set for these experiments.	56
5.4	Example of refined keyword representation: spell-correction and stemming. Terms shown in bold are features shared by the query and index descriptions.	60
5.5	Example of semantic expansion. Terms shown in bold are features shared by the query and index descriptions.	61

5.6	Sample dependency annotation. Terms and relations in bold are common to index and query descriptions; relations in dashed-bold are common based on part-of-speech matching.	63
5.7	A second example of dependency annotation. Terms and relations shown in bold are common to this index description and the query shown in Figure 5.6.	64
5.8	Example dependency edges augmented with semantic concepts.	65
6.1	Visual depiction of retrieval outcomes.	69
6.2	Expected gains in MRR from correcting retrieval failures. Relevant images were inserted into the result list at rank $N = 10, 9, 8$, etc. (shown on the x-axis).	71
6.3	Expected gains in MRR from correcting errors. Relevant images found at rank $N = 10, 9, 8$, etc. were inserted into the result list at rank 1.	72
6.4	Expected gains in MRR from additional descriptions.	74
6.5	Comparison of expected gains from correction vs. additional data.	75
6.6	Sample image with <i>visual features</i> in the description.	82
6.7	Sample image without <i>visual features</i> in the description.	82
6.8	Sample image with misleading orthography and non-relevant repetition in the description.	84
7.1	An example of knowledge-base annotation of a topic and index description. Terms shown in bold are features shared by both descriptions.	89
7.2	Retrieval process with graph-based reranking.	93
7.3	KB representation for image descriptions in SconeImage.	94
7.4	Similarity features for a sample pair of edges.	97
7.5	Samples of query encodings.	103
8.1	The file organization of SconeImage knowledge bases and reasoning modules.	109
8.2	WordNet annotation of an image description.	111
8.3	Top level of the SconeImage Ontology, adapted from DOLCE + WordNet. (Gangemi et al., 2003)	114

8.4 Example of knowledge acquisition from training data. KB elements for “woman” “girl” and “female” are added, along with English strings that trigger “woman” and “girl”. Stemming at retrieval time compensates for the singular-plural variation. 115

8.5 WordNet type hierarchy for ‘woman’ and ‘man’. 116

8.6 SconeImage type hierarchy for ‘woman’ and ‘man’. 116

9.1 Screenshot from the web-based annotation tool used for the analysis in Chapter 6. 125

9.2 Diagrammatic description of applied textual inference Tasks. They are recognized to be hard enough to require some amount of language understanding for success, but they are evaluated based on the accuracy of a decision output. Entailment, Paraphrasing, and other well-known tasks are shown here as examples that meet this description. Image-identity is shown as a new example. 127

List of Tables

3.1	Overview of the Phetch data set	31
3.2	Partition of the Phetch corpus into sections for common evaluations.	35
4.1	Overview of the flickr.com sample.	44
5.1	Document size of some TREC 2008 Web Track collections. Phetch is shown for comparison.	53
5.2	Word counts for the 5A data set, non-textual images.	57
5.3	Results of retrieval with the baseline model.	59
6.1	Rank of the relevant image in baseline run of Phetch 5A training queries.	68
6.2	Classes of precision error.	77
6.3	Classes of recall error.	79
6.4	Frequency of error classes.	80
6.5	Error-inducing features in the Phetch 3A data section.	81
6.6	Frequency of error-inducing features.	86
7.1	Results of spelling correction and knowledge-base annotation. ANOVA significance is shown for improvement over the <i>keywords</i> baseline.	91
7.2	Vertex features for graph-based description similarity.	96
7.3	Results of re-ranking. ANOVA significance is shown for the improvement over the <i>keywords</i> baseline.	102
7.4	Results of retrieval with naive vs. structured queries.	103
7.5	Results of retrieval using combinations of syntactic and semantic features for graph-based reranking. ANOVA significance is shown for improvement over the <i>keywords</i> baseline.	104

7.6	Results of retrieval on all images from the 5A data set, vs. the subset without textual content.	105
8.1	Effect of annotation with wordnet synsets on Phetch section 5A, non-textual images. ANOVA significance is shown for improvement over the <i>keywords</i> baseline.	111
8.2	Comparison of knowledge resources for annotated description retrieval. ANOVA significance is shown for improvement over the <i>keywords</i> baseline.	117
9.1	Candidates for general linguistic causes of mismatch between semantically similar texts. Accommodation has two sub-classes, <i>Analogy</i> and <i>Contradiction</i>	123
9.2	Error classes from the original vocabulary that were removed in the general vocabulary.	124

Chapter 1

Applied Textual Inference and Description Retrieval

1.1 Background

The experiments in this thesis address the task of retrieving images based on their short descriptive labels. As in ad-hoc document retrieval, a baseline system using term vectors to represent these labels performs reasonably well (>80% MRR). However, upon inspection of the labels where the baseline system fails, we observe that the most challenging examples for this task may require us to enrich the feature space of our solution, allowing the system to capture more of the deep semantic similarities that humans seem to notice when they make comparisons between images and their descriptions.

As a result, we present a knowledge-based solution for retrieving English descriptions for objects, such as images, from a collection. Based on analysis of the results using keywords alone¹, we infer that an ideal solution to this problem should use deeper representations of the meaning encoded in textual descriptions. This places our work in the landscape of systems for Natural Language Understanding (NLU).

A trademark of such systems is that they convert natural-language text into machine-operable semantic objects, requiring developers to design or select an appropriate data structure to capture the result of conversion. Evaluating these structures is challenging. A researcher might propose a new representation, then evaluate his NLU system by providing it with some ex-

¹Refer to Chapter 6 for a detailed analysis of baseline system errors

ample texts and comparing its output to a set of hand-validated analyses that serve as a gold-standard. This approach has been used for semantic parsing historically (Shank and Tesler, 1969) and in more recent systems (Ge and Mooney, 2006). While such evaluations are critical for driving improvements to a single NLU system, there is a need for complementary evaluations that allow researchers who have chosen different representations, and hence have different gold standards, to compare their results.

1.2 Applied Textual Inference (ATI)

Applied Textual Inference (ATI) as used in this thesis refers to the class of generic task-based evaluations that address this need. ATI tasks are assumed to depend on some level of text understanding and background knowledge, but they are associated with evaluation metrics that abstract away from system-specific representational choices. The flagship example of an ATI task is Recognizing Textual Entailment (RTE). Dagan et al. (2006) describe RTE as follows:

...recognize, given two text fragments, whether the meaning of one text can be inferred (entailed) from another text. (Dagan et al., 2006)

ATI tasks are challenge problems. Because they are intended to drive research on text understanding, the problems are designed to be “hard enough” to require reasoning. However in order to support cross-site comparisons of results, the problems are evaluated at the surface level. Other examples include paraphrasing, summarization, word-replacement, and question answering (QA). A component of an ATI problem definition is an evaluation corpus that can be shared among researchers working on independent solutions. The SemEval-2007 Workshop on Semantic Evaluations² included 18 such tasks, framed as shared evaluations with over 100 systems submitted for evaluation. Tasks included classifying semantic relations (Girju et al., 2007) and temporal relations (Verhagen et al., 2007) between words; identifying valid lexical substitutions (McCarthy and Navigli, 2007); and several cross-lingual word sense disambiguation tasks (Agirre et al., 2007; Orhan et al., 2007; Jin et al., 2007).

²<http://nlp.cs.swarthmore.edu/semeval/>

1.3 Current Approaches

Shared evaluations on ATI tasks have succeeded in generating a wide variety of solutions from researchers. Taking the PASCAL RTE Challenge as an example, we can see that deep representations and even knowledge-based techniques seem to play an important role in state-of-the-art solutions. Of 16 research teams participating in the first challenge in 2005³, 7 used features from WordNet, 3 applied some kind of world knowledge, and 7 applied logical inference engines (9 systems out of 16 used at least one of the three). In 2007 the number of participants and the variety of techniques expanded, with the vast majority of these systems relying on some combination of WordNet, syntactic matching/alignment, and machine learning algorithms; the most successful system in that year applied all of these techniques in addition to a logical inference engine (Hickl and Bensley, 2007).

All of these techniques seem promising for problems in the same class as RTE - problems that can also be described as instances of applied textual inference. However, to adapt them effectively for a particular ATI task, we must first understand that task and the details of why more shallow techniques fail.

1.4 Retrieving Object Descriptions

By framing the task of image-description retrieval as an instance of applied textual inference, we can draw on the growing literature described in Section 1.2, helping us solve the textual understanding problem represented by these challenging examples. We will use the details of the image retrieval problem to demonstrate how ATI techniques can be effectively adapted for a new task.

Textual labels are attached to electronic data in many ways, including filenames, link text, captions, descriptions, and titles. Labels appear almost everywhere that data does. New approaches to retrieving these labels, matching them against user-supplied queries, give us new tools for accessing electronic resources. These tools complement approaches that rely on features extracted directly from the information objects, like Content-Based Image Retrieval (CBIR).

An important feature of object labels assigned by humans is that they often consist of short multi-word phrases. These phrases exhibit syntactic

³<http://pascallin.ecs.soton.ac.uk/Challenges/RTE>

and semantic structure that is not always modeled by information retrieval systems. However, this structural information becomes more prominent in light of the fact that labels, which will make up both queries and documents in our retrieval paradigm, are very short compared to most web pages or passages. Consider a web page devoted to care of beagles. The web page might refer to a beagle in a number of ways, as in: “your pet will need to be fed often,” and “watch out, these dogs like to chew on shoes.” A query like “pets that chew on shoes” could be matched to this page, based on all of the page text. But based on words alone, it would be easy for a retrieval system to miss an image described as “a beagle gnawing on my loafer.” These features make retrieval for image descriptions a specialization of known-item and short-document search. In cases like this, it becomes even more important to leverage syntactic and semantic evidence in addition to keywords.

1.4.1 Retrieving Images

We will approach the problem of image retrieval with two distinct use cases in mind. In a large collection of images or other labeled objects, a user often remembers one that he would like to see or use, as in “show me the one where the guy and the girl are standing in a parking lot.” The user may not recall the filename or other details that would allow him to navigate to the image directly. Still, he can describe some properties of the object and he wants to find it again. This distinguishes the task of *known-item retrieval* from the task of *browsing*, where the query describes a class (“pets playing with toys”), and the user would like the system to retrieve any labeled object that is a member of this class (e.g. “dog catching a frisbee”). Our experiments focus on the first case, but we will explore some properties of both.

As an example of the problem we address in this work, consider an image search using the query “people petting their dogs”. The four images of Figure 1.1 (1.1a, 1.1b, 1.1c, 1.1d) are examples of what we might expect to see in the results.

These four images and their labels reveal interesting features of the problem. First, words that appear in multiple image labels do not guarantee that the images are alike, just as different words do not guarantee that the images differ. For example, three out of the four images feature children, but one of these is labeled “boys and girls.” This lexical mismatch obscures an important similarity.



(a) “drawing of people on the floor petting their dog”



(b) “a boy and girl petting a dog”



(c) “christmas animal celebration child”



(d) “paw of dog on legs of owner”

Figure 1.1: Sample retrieval results for the query “people petting their dogs”

Second, image labels exhibit syntactic and semantic structure. Most of the labels in Figure 1.1 consist of sentence-like phrases that were generated by the owners of the images. Even a specific phrase like “petting their dog” can be found, intact, in more than one image label. The label for Figure 1.1d exhibits a series of nested prepositional phrases. This structural richness seems appropriate for the query: given the label for Figure 1.1a or Figure 1.1b, one could plausibly answer the question “What’s happening in this picture?” This structure adds information that is not captured at the lexical level alone. The effect can be seen in a change as simple as word order. Consider the difference between the image label “paw of dog on legs of owner,” and one with identical words but different structure: “legs of owner on paw of dog.” These two phrases represent different images.

Finally, human interpretation of these labels at the lexical, syntactic, and semantic levels is influenced by background knowledge. As a reader it is easy to see the connections among the terms used to refer to humans in Figure 1.1: *people, a boy and a girl, child, owner*. We can also see connections among the terms used for pets: *dog, animal*. By making these connections available to a retrieval system, we can apply them to the known-object and browsing

tasks. For example, to make the query more specific, substitute *child*, *boy*, or *girl* for *people* in the query. To make the query more general, substitute animal for dog.

1.5 Thesis Outline

This thesis demonstrates how an inference engine and a set of symbolic knowledge resources in the form of ontologies can contribute to performance on the task of retrieving object descriptions, as measured by Mean Reciprocal Rank in an IR-style evaluation on a set of labeled images. Important questions addressed by this research include the cost of developing such a knowledge resource and the quantitative benefit in an end-to-end system for image retrieval. In the process of answering these questions, we will describe the results of several sub-tasks, each of which represents a contribution to the understanding of this problem and to the discovery and implementation of a knowledge-based solution:

- Introduce the image retrieval experiments that are supported by a data set containing over 50,000 images, hand-labeled with multiple descriptions. Compare to other widely available resources for evaluating semantic retrieval. Contrast the description-based retrieval task with tag-based retrieval, including the frequency of tags and descriptions in publicly-available image collections. (Chapters 3, 4)
- Present candidate solutions to the description-representation problem. Describe a series of parameterizations for calculating the similarity between two descriptions. Apply the first of these parameterizations in a baseline retrieval system that represents descriptions as bags-of-words. (Chapter 5)
- Perform a detailed error analysis of the baseline retrieval system. Identify classes of error that occur when the system compares two descriptions, as well as linguistic features of the descriptions that may affect the frequency of these errors. Introduce three hypotheses that connect these error classes to specific knowledge-based solutions. (Chapter 6)
- Implement additional parameterizations from Chapter 5 to test these hypotheses, augmenting the baseline retrieval system with semantic

and syntactic knowledge. Test two configurations of the new system: descriptions annotated with concepts from a hand-constructed knowledge base, and descriptions represented by dependency graphs whose vertices are connected to the same knowledge base. Evaluate the hypotheses from Chapter 4 by calculating the reduction in specific error types when these configurations are used for retrieval. (Chapter 7)

- Describe the structure and development process for the knowledge resources used in Chapter 7. Compare these resources to other widely-available ontologies. Evaluate the effect on retrieval when different knowledge resources are applied. (Chapter 8)

1.6 Conclusions

In the following chapters, we will investigate whether knowledge-based semantic analysis of human-assigned multiword image labels can be performed using NLP techniques, combining syntactic and semantic evidence; further, whether this analysis results in semantic features that are useful in retrieving the images from a large collection. In particular, we will examine whether the resulting semantic representation enables a retrieval system to correctly answer queries that require inference more powerful than that which keyword-based representation supports. In the process, we construct a retrieval paradigm that complements a keyword-based approach, achieving retrieval results that are significantly better than keywords alone, in particular for certain types of inference-dependent queries.

Chapter 2

Related Work

Although the SconeImage system is a pipeline for information retrieval, the work in this thesis builds on current research from several communities, including applied textual inference (recognizing textual entailment, paraphrase identification, question answering), semantic distance, and knowledge acquisition. In this chapter we provide an placement of our work with respect to each of these landscapes.

2.1 Applied Textual Inference with Semantic Distance

Task-based evaluations of information extraction systems were performed in the mid- to late- nineties at the Message Understanding Conferences Grishman and Sundheim (1996), which moved toward textual understanding tasks at MUC-6 with the introduction of the SemEval (semantic evaluation) tasks: word-sense disambiguation, coreference resolution, and predicate-argument labeling. Many of the techniques used in applied textual inference systems today are refinements of approaches developed for MUC.

Since 2005, the flagship venue for the applied textual inference community has been the Recognizing Textual Entailment (RTE) Challenge (Dagan et al., 2006). The target problem for these challenges is described as follows:

The RTE task is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred (entailed) from the

other. This application-independent task is suggested as capturing major inferences about the variability of semantic expression which are commonly needed across multiple applications. (Dagan et al., 2006)

In 2010, the RTE Challenge will be a sponsored track¹ in the Text Analysis Conference (TAC-2010) hosted by the National Institute of Standards and Technology (NIST). TAC-2010 will also sponsor tracks for knowledge base population (KBP) and summarization.

Like the image retrieval task presented in this thesis, detecting a relationship like entailment, paraphrasing, or summarization between two texts can be framed as an application of semantic distance calculations. A great deal of work relevant to this thesis has been done in the areas of applying semantic distance functions, both knowledge-based and empirical, to each of these problems.

An overview of semantic distance functions using WordNet is given by Budanitsky and Hirst (2001). Several of these functions depend on path length through the kb, in similar fashion to the vertexSim function we define in Chapter 7. These functions were compared with corpus-derived semantic distance features in the context of paraphrase detection by Mihalcea et al. (2006), and applied to word sense disambiguation by Patwardhan et al. (2007). These example systems and others support the findings of Chapter 8, that a domain-general resource like WordNet can contribute to better performance on a task that involves text understanding. Our work extends this result by showing that, at small additional development time, a task-specific knowledge base can be developed that yields better results than WordNet alone.

In addition to semantic distance functions, the RTE literature includes relevant work on system architectures and comparison of knowledge resources. Giampiccolo et al. (2008) describes the variety of systems that were submitted to the RTE track of TAC-2008, many of which include inference engines, logical forms, or knowledge bases. Of the twenty-six teams participating in that track, 18 exploited WordNet as a lexical-semantic resource. An analysis of technologies applied in the 2007 challenge shows that many of the best-performing systems from that year applied lexical-semantic features from WordNet in combination with syntactic alignment (Giampiccolo et al.,

¹<http://www.nist.gov/tac/2010/RTE/index.html>

2007). The SconeImage architecture applies some of the lessons learned from these systems to a new type of applied textual inference (retrieving object descriptions).

2.2 Knowledge Acquisition and Development

Knowledge bases are an enabling technology for many of the approaches described in Section 2.1. In this work, we describe one approach to knowledge base development that re-uses existing resources at the top levels of the ontology and performs manual acquisition of domain-specific knowledge by examining and correcting errors found in training runs. In our work we apply the DOLCE upper ontology described in Masolo et al. (2003) and its mapping to WordNet following Gangemi et al. (2003).

This development strategy is similar to the approach taken by Fan et al. (2003). Although the upper levels of the resulting ontology are reusable across multiple systems, this approach implies that knowledge engineers must author the domain-specific layers as appropriate for every new domain (but with portability across unseen users of the system, as we will show in Chapter 7). Although the goal of this thesis is to show that *when you can get it, knowledge helps*, there are many options in the literature that address the acquisition problem.

Some of the most successful of these help non-engineers to assemble knowledge representations from reusable components, as in Clark and Porter (1997), or from scripts, as in Gil and Tallis (1997). These strategies allow Subject Matter Experts to author domain knowledge without requiring them to learn knowledge engineering.

2.3 Ad-Hoc Retrieval and Question Answering

The description-retrieval problem addressed in this thesis is related to, but distinct from, ad-hoc document retrieval. A technical introduction to ad-hoc retrieval is given by Manning, Raghavan, and Schütze in their recent textbook (Manning et al., 2008). As discussed in Chapter 5, images in the Phetch corpus are represented as documents that are very short, in comparison to mainstream document retrieval collections. In addition, the experimental

setting where exactly one image from the collection is relevant to each retrieval topic fits better into the class of known-item retrieval, described by Ogilvie and Callan (2003), among others.

In the broader literature of information retrieval, very relevant techniques have been used in the task of Question Answering (QA). In fact, question answering has also been classified as an instance of applied textual inference². The unit of text returned by answer-candidate retrieval components is usually short. In addition, lexical matching has long been known to cause problems for these systems, since a segment of text that is *similar* to a question does not necessarily *answer* the question. An example from Haghighi et al. (2005) is shown below:

Consider a Question Answer system searching for an answer to When was Israel established? A representation which did not utilize syntax would probably enthusiastically return an answer ... "The National Institute for Psychobiology in Israel was established in 1979." ... it's important to match relationships as well as words ... (Haghighi et al., 2005)

As a result, logical and semantic representations have a long tradition in question answering systems (Clark et al., 1999; Rinaldi et al., 2003; Harabagiu et al., 2000; Hovy et al., 2001). The error-analysis work performed in this thesis helps to establish which techniques from this tradition are appropriate for image retrieval. For example, we will show in Chapter 6 that ontology-related errors were very common in our data set, but that errors attributed to contradictions were rare. As a result, we adopted query-augmentation techniques from a tradition described in Varelas et al. (2005), while saving for future work some of the contradiction-specific methods that have been applied with success in QA (Harabagiu et al., 2006).

SconeImage is one example of an architecture that applies background knowledge for improved retrieval results, but the focus of this work is on exploring and comparing the knowledge resources themselves, and their relationship to retrieval errors. Another thread of related work in question answering focuses on the architecture itself, exploring data structures like the Annotation Graph (Bilotti et al., 2008) for integrating such knowledge. A series of experiments that could be interesting for future work might investigate how these architectures could accommodate SconeImage features.

²<http://art.uniroma2.it/TextInfer2009/>

2.4 Image Retrieval

At the top level, the SconeImage application performs image retrieval. As a strictly description-based system, it is distinct from Content-Based Image Retrieval (CBIR) systems, which use computer-vision techniques to calculate the similarity between images at the pixel level, rather than relying on natural-language annotations. A survey of CBIR systems and the features they use is given by Veltkamp and Tanase (2002). Some hybrid systems have also been developed that take advantage of direct semantic annotation or tags, but not free-text descriptions; examples include Aslandogan and Yu (2000), Wang et al. (2008), and Blei and Jordan (2003).

Knowledge-based image search has been described in Aslandogan et al. (1997), where users could explicitly browse for images labeled with a particular semantic concept, independent of the English strings. That work did establish the utility of attaching to images not merely keywords, but concepts rooted in an ontological knowledge base. More recently, Wang et al. (2008) have shown improvements over baseline results from Google Image Search by performing ontology-driven reranking. However, in that application queries were given as lists of concepts. In this thesis we extend this utility by decoding the concepts from natural-language descriptions. Our work not only confirms prior findings that ontologies can improve retrieval results, but we find that these improvements can be replicated under more realistic retrieval conditions (plain-language queries).

Systems that do perform text-based image retrieval often rely on tags rather than descriptive text, or reduce descriptions to a series of keywords before performing retrieval. Most companies that host mainstream web-search engines sponsor image-specific interfaces that fall into this category. The meta-search website Fagan Finder³ lists nine image search engines, along with three photo blogs, thirteen websites for stock photography, ten for photo sharing, and thirty-four other online image sources, ranging from Biomedical image libraries⁴ to the image catalog of the Library of Congress⁵.

The ImageCLEF track of the Cross Language Evaluation Forum (CLEF) has sponsored shared evaluations on tasks related to description-based image retrieval from 2003 to the present (ImageCLEF 2010). Four teams participated in the inaugural evaluation, with one out of four applying knowledge

³<http://www.faganfinder.com/img>

⁴<http://phil.cdc.gov/Phil/>

⁵<http://lcweb.loc.gov/rr/print/catalog.html>

from WordNet and the remaining systems relying on term-frequency models (Clough and Sanderson, 2003). The most recent ImageCLEF workshop was held in 2009 (Paramita et al., 2009); in that year only one system applied semantic knowledge (from WordNet). There are at least two differences between the ImageCLEF tasks and ours that were likely to discourage knowledge-based approaches: first, historically the tasks were cross-lingual, and off-the-shelf knowledge resources for languages other than English are rare. The work we propose here could be valuable in overcoming this obstacle, considering the low cost of development for adding lexical strings to the existing conceptual structure that we will describe in Chapter 8. Second, the ImageCLEF evaluation applies non-traditional metrics in order to push the state of the art in novel directions. The 2009 metrics rewarded systems for achieving *diversity* as well as precision in their results; while it seems likely that the techniques presented here could help improve the precision of these systems, the effect on synthetic metrics like diversity is beyond the scope of this thesis.

Although the technologies are closely related, the focus of this thesis is slightly different from the task supported by image search engines like Google⁶ and Microsoft’s Bing⁷. These systems support ad-hoc search, where the user seeks “pictures of animals” and any image featuring ducks, rabbits, or puppies will do. The evaluations presented in this thesis more closely replicate the case where a user seeks a particular image (“the one where the puppy is chasing the kitten around that blue chair in the living room”), and must continue the search task until he finds it. This contrast affects the types of inference and query expansion that are appropriate, for example as discussed in Section 7.3.2 with respect to asymmetry in the similarity function.

Nonetheless, findings from this thesis can apply to such systems. The interface of Google Image Search and of Bing Image search offer advanced search options that confirm that knowledge of the task is important in ad-hoc retrieval, as well. These options allow the user to select image features like “size=Medium, Large, Icon..., type=face, photo, clip art, line drawing..., color=full color, black and white, red, orange, green...”. The menu-driven interface circumvents the problem of extracting such features from a plain-text description. However by studying plain-text descriptions, as we do in

⁶www.google.com/images

⁷www.bing.com/images

this thesis, we can discover what users would search for if the interface were unconstrained. This results in a more accurate ontology of features that could be used to improve the menu-driven tools.

For example, we have discovered that users make a critical distinction between features that attach to the contents of an image (“black and white dog”) and features that describe the image as an object (“black and white photo of a dog”). Currently, the semantics of menu-driven features are underspecified in this regard. A user may choose “color=black and white”, but must guess as to whether the filter will be applied at the image or content level. A simple modification of the user interface could apply this finding for improved control over search results.

Chapter 3

A Corpus for Evaluating Description Retrieval

3.1 The Phetch Data Set

The Phetch data set was created by von Ahn and Dabbish (2004). It was collected in the context of an online game where multiple participants compete in teams to identify an image based on one teammate’s typed description. The exercise was repeated for each image in a large collection of JPEG files harvested from the web. Images are roughly thumbnail-sized, with an average file size of 6.3KB and dimensions that range from 72x72 pixels to 224x169 pixels. The total collection (images plus annotations) requires approximately 350MB of disk space. A summary of the corpus size is given in Table 3.1.

In this data set, a single description is a short paragraph written by a single annotator/participant about a single image. Each image in the

Totals	Images	Descriptions	Words		
	54,168	135,561	1,603,567		
	Mean	Sdv	Min	Med	Max
Descriptions per image	2.5	1.5	1	2	13
Description words per image	38	25	1	32	549
Tags per image	8	3.5	2	7	32

Table 3.1: Overview of the Phetch data set

collection has multiple descriptions. Each description is composed of one or more phrases, short segments that contribute to the overall description and are usually connected rhetorically to each other. An example¹ is shown in Figure 3.1.

Image descriptions are different from image tags, which also occur in this data set. The words that appear as tags are unordered keywords that have been associated with the images by committee: each tag was independently produced by a group of participants in another online game, also created by von Ahn and Dabbish (2004). This means that descriptions were not available to the players who generated tags, and tags were not available to players assigning descriptions. As a result the tags come from a restricted vocabulary (since rare or unique words would fail to meet the consensus requirement), and have no syntactic structure as a group. Each description, in contrast, represents one player’s plain-English attempt to express the image contents in his own words. Although the properties of the tags as a corpus may also yield interesting observations, in this thesis we focus on the descriptions and defer a deeper analysis of the tags for possible future work.

Many images in the data set have at least two descriptions assigned to them, each provided by a different participant. This property makes the Phetch data well-suited for evaluation in a labeled-object retrieval task, where the retrieval engine simulates the behavior of a human searching for an image, based on his teammate’s description. One description is used as the retrieval topic, and any remaining descriptions are wrapped as a single document that represents the image in a collection. Because both labels are associated with the same image, we can use image identity as a proxy for relevance judgments and evaluate retrieval on this data set consistently, with no additional annotation. In this setting, the only relevant document is the one describing the same image as the query.

¹In this example punctuation has been added by the author to mark descriptive-phrase boundaries. In the raw data, these boundaries are marked by the carriage return character, since the segmentation is generated by a player pressing ‘Enter’ while typing an image description. As a result, these segments do not always correspond to syntactic constituents. However it is a useful abstraction to think of them as phrases.



Description 1:	“stadium; pageant girls in the foreground”
Description 2:	“olympic stadium; women in foreground; with sashes”
Description 3:	“people at sporting event; pageant contestants”
Tags:	girl face blond blue game crowd model people hair miss

Figure 3.1: Image from the Phetch data set with descriptions and tags.

3.2 Comparable Data Sets

The availability of relevance judgments and the level of detail in textual annotations set the Phetch corpus apart from other data sets that are commonly used to evaluate image retrieval, including the Corel image collection. Corel is in common usage for content-based image retrieval (CBIR) experiments. Corel images are published in *groups*, sub-collections that share a common subject or theme. These thematic groups are used as IR topics in query-by-image experiments; an image is posed as a query to a CBIR system, and images from the same group are considered relevant. A recognized problem with this approach is that the results of such an evaluation depend heavily on the particular query image that experimenters select from each group. Every research team may select queries that play to the strengths of its system. The effect on experimental results, and the resulting inability to compare results across research groups, is described by Müller et al. (2002). In that work the authors establish the need for standardized test sections within widely-used corpora for image retrieval. This type of partitioning is applied to the Phetch corpus in Section 3.3.

In addition to standardized corpus partitions and the availability of rele-

vance judgments, the Phetch corpus contrasts with Corel by including image descriptions. The Corel image collection annotates each image with an unordered set of single-word tags, but not with descriptive titles or captions. While tags are useful for some experiments, they do not meet the needs of researchers who seek to evaluate retrieval based on descriptive text. Shirahatti and Barnard (2005) summarize the problematic assumptions that underlie tag-based evaluation as follows:

These approaches are only indirectly connected to the task that they are trying to measure. For example, there is an implicit assumption that a person seeking an image like one labeled grass will be content with all the images labeled grass and none of the ones not labeled grass (Shirahatti and Barnard, 2005).

Because the Phetch corpus includes descriptions as well as tags, it can support both experimental styles.

A comparable data set that does support description-based image retrieval is the evaluation set used for the ImageCLEF shared tasks. The ImageCLEF evaluation set is derived from the IAPR TC-12 Benchmark of 20,000 images (Grübinger et al., 2006), which has been used in the ImageCLEF image retrieval evaluations since 2006².

In comparison, the Phetch data set contains more than twice as many images, although not all of these images are used in our retrieval experiments³. In addition, the IAPR TC-12 data does not include relevance judgments. Judgments have been added for the ImageCLEF test collection, a subset of 60 images sampled from IAPR TC-12. The resulting data has proven to be helpful for comparative evaluation, but not for system training⁴.

3.3 Establishing Corpus Sections

To be effective as a shared resource, a data set should include sections that allow researchers to perform comparable experiments. The importance of shared experimental conditions, including data and evaluation methods, has

²<http://ir.shef.ac.uk/imageclef/2006/>

³see Chapter 7 for additional detail

⁴The website for ImageCLEF 2006-2008 encourages participants to generate their own training data before submitting retrieval runs, <http://eureka.vu.edu.au/~grubinger/ImageCLEFphoto2007/adhoc.htm>

Section	Descriptions per Image	Total Images	Total Words
1A	1	17,237	470,924
1B	1	17,237	470,924
2A	2	7,264	371,313
2B	2	7,265	371,879
3A	3	5,171	367,284
3B	3	5,171	367,999
4A	4	3,084	283,142
4B	4	3,084	282,486
5A	5 or more	2,946	357,582
5B	5 or more	2,946	355,853

Table 3.2: Partition of the Phetch corpus into sections for common evaluations.

been widely recognized in most scientific communities, including information retrieval (Harman, 1992).

To this end, we present a partition of the Phetch corpus that can be used to clearly identify the experimental conditions reported in this thesis. This partition is supported by the free distribution of scripts that generate the sections⁵, given the full plain-text corpus as input.

At the top level, this partition isolates 5 sections according to the number of descriptions attached to each image. This division supports experimental control for the number of descriptions per image, which affects retrieval accuracy⁶. Each section is further divided into 'A' and 'B' sub-sections. Half of the images from a given section, along with all of their descriptions, are placed in the 'A' sub-section. The other half of the images are placed in the 'B' sub-section. This division supports experimental control for separating training and testing data on the basis of seen and unseen images. A summary of the partition sizes is shown in Table 3.2.

Sections 1A and 1B are included in the partition even though images with only one description are not as appropriate for the evaluation paradigm defined in Section 3.1. With two or more descriptions, one may be used

⁵Send email requests to atribble@cs.cmu.edu

⁶see Chapter 5 for additional detail

as a topic that is meant to retrieve the other. With only one description, there is no meaningful topic/document division. Nonetheless, the examples in sections 1A and 1B could be used in the future for training genre-specific language models or for other parameter-tuning experiments.

3.4 Logical Document Structure

In addition to establishing experimental partitions of the data, we have used the logical structure of Phetch annotations to create a document format that conforms to the TREC⁷ text format. TREC text is a flexible sgml-like formalism. It requires every document to be enclosed in beginning and end markers, and to be labeled with a DOCNO field that encloses a unique document identifier. The body of the document text must be enclosed in a TEXT field.

Within these constraints, additional structure may be added by creating new fields that serve as markup on the content enclosed in TEXT. Such markup may then be modeled, stripped out, or ignored by indexing and retrieval systems. In the case of Phetch images, we propose a document structure that identifies descriptions, phrases, and tags as distinct fields. An example of an image-document is shown in Figure 3.2.

3.5 Conclusions

In this chapter we introduce the Phetch data set, a collection of over 54,000 images that have been richly annotated with descriptive labels. From the raw corpus we have derived training, development, and test sets that are formatted using the trext conventions. The resulting collection compares favorably with comparable corpora for evaluating image retrieval, including the Corel image collection and the ImageCLEF test collection.

Given this landscape of evaluation corpora for image retrieval, in particular description-based retrieval of images labeled by multiple human annotators, the Phetch corpus and its partitioning into replicable subsets are well-timed contributions to the textual inference community.

⁷<http://trec.nist.gov/overview.html>

```
<DOC>
<DOCNO>image-1 </DOCNO>
<TEXT>
<DESCRIPTION>
<PHRASE>stadium </PHRASE>
<PHRASE>pageant girls in the foreground </PHRASE>
</DESCRIPTION><DESCRIPTION>
<PHRASE>olympic stadium </PHRASE>
<PHRASE>women in foreground </PHRASE>
<PHRASE>with sashes </PHRASE>
</DESCRIPTION><DESCRIPTION>
<PHRASE>people at sporting event </PHRASE>
<PHRASE>pageant contestants </PHRASE>
</DESCRIPTION>
<TAGS>girl face blond blue game crowd ... </TAGS>
</TEXT>
</DOC>
```

Figure 3.2: An sgml representation for structured Phetch documents.

Chapter 4

Understanding Image Descriptions

4.1 The Language of Descriptions

Behavioral experiments in image labeling have shown that, in an unconstrained setting, users tend to create short narratives to describe images rather than limiting themselves to unordered lists of keywords (Jørgensen, 2001). Syntactic analysis of the Phetch data set supports this conclusion, as well. The phrase fields that appear in Phetch descriptions are slightly more than three words long, on average. The data set contains over 580,000 such phrases. Phrases of length 1, which might be interpreted as tag-like annotations, make up only 18% of these, with the remaining 82% consisting of 2 or more words.

Among the phrases of length 2 or more, interesting syntactic patterns emerge. To discover these, we applied the Stanford Dependency Parser (Marneffe et al., 2006) to a random sample of 80,000 Phetch phrases, extracted from roughly 20,500 descriptions for 8000 images.

4.1.1 Syntactic Patterns

Typed dependency representation of the kind produced by the Stanford parser has become increasingly popular as an alternative to phrase-structure trees, due to its simplicity and accessibility. The Stanford typed dependency representation has been used as the syntactic representation in systems for Text Mining (Zhuang et al., 2006; Meena and Prabhakar, 2007; Banko

et al., 2007; Zouaq et al., 2006; Chaumartin, 2007; Kessler, 2008) and for Recognizing Textual Entailment (Adams et al., 2007; Blake, 2007; Chambers et al., 2007; Harmeling, 2007; Wang and Neumann, 2007), among other tasks. Figure 4.1 shows an example sentence and its dependency structure in two equivalent representations: as a textual list of word pairs labeled with their relationship, and as visual links that connect words from the sentence.

After parsing, we removed the lexical arguments from each parse result and calculated the frequency of the remaining sequences of dependency labels. The most common sequences, which we interpret as syntactic patterns, involve a single noun modified by an adjective (the *amod* relation) or by another noun, either with a preposition (the *prep* and *prep_** relations) or without (the *nn* relation). These patterns and their frequencies are shown in Table 4.2, where notation is modeled on the Stanford typed dependencies manual¹.

Each pattern in 4.2 represents one entire phrase. Short patterns recur most often, partly due to the fact that the average length of a phrase is only 3.5 words. By examining the overall frequency of individual dependency relations, we can gain an understanding of which syntactic building blocks are most common in phrases of all lengths. Table 4.3 gives these frequencies.

This analysis indicates that the contributors to the Phetch data set did prefer to describe images using short phrases with some recoverable syntactic structure. The overwhelming majority of phrases contain more than one word, and of those only a few appear to be unordered lists of nouns. Dependency patterns consisting only of noun strings occur 5,794 times in our parsed subset, ranging in length from 2 to 7. We might consider these, along with single-word phrases, to be examples where the annotator was using a bag-of-words style. The total of all such examples make up less than 20% of the sample we used for this analysis.

4.1.2 Semantic and Rhetorical Patterns

In addition to characteristic syntactic patterns, descriptions in the Phetch corpus exhibit recognizable semantic patterns. Consider the pattern “*conj_and*” from Table 4.2. A label composed of a string of nouns conjoined by “and” seems effectively equivalent to a basic bag-of-words representation, using the nouns as keywords. However, knowledge of the task informs us that while

¹http://nlp.stanford.edu/software/dependencies_manual.pdf



Description: “guy in black and white pic”

Dependencies (list): amod(pic-7, black-4) conj_and(black-4, white-6)
 amod(pic-7, white-6) prep_in(guy-1, pic-7)

Dependencies
 (visual):

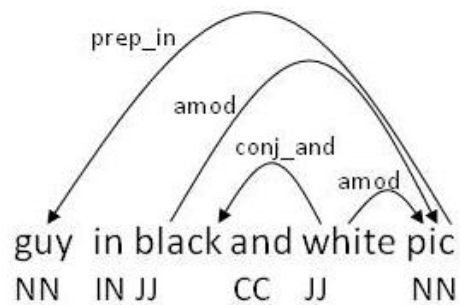


Figure 4.1: Dependency analysis, as a list and as a visual annotation.

Pattern	Examples
amod	“green leaves” “gold lid” “white background” “uncensored video”
nn	“company logo” “cd cover” “picture frame” “caribou antler”
amod nn	“white grave stones” “red paper background” “antique matchbox car” “blue tennis shoes”
conj_and	“black and white” “sword and shield” “belle and sebastian”
prep_in, prep_on, prep_of, prep_towards	“forest in background” “wall on left” “pile of logs” “facing towards photographer”
nsubj amod dobj	“he has short hair” “tub has white rim” “cup has radiating lines”

Figure 4.2: Common dependency patterns of 2-4 words.

Relation	Total Instances	% of Phrases	% of Images
amod	41541	38.1%	93.4%
det	20516	18.7%	73.8%
nn	19892	19.2%	77.6%
dep	17737	16.2%	74.9%
nsubj	14639	15.4%	68.6%
dobj	8778	9.1%	53.5%
conj_and	6532	6.5%	43.2%
pobj	6489	6.6%	45.2%
prep_of	6465	6.8%	45.8%
prep_in	6309	7.0%	44.2%
prep_on	5580	6.0%	40.9%

Figure 4.3: Frequency of dependency patterns across phrases (col 2) and images (col 3).

“sword and shield” describes a pair of entities that appear as the subject of the image, “black and white” is a fundamentally different query, one which specifies meta-information about the image². The distinction between content-descriptions and meta-descriptions is an important semantic/pragmatic feature that can contribute to errors in a bag-of-words retrieval model³.

Descriptions also exhibit rhetorical structure: full sentences sometimes span several phrases. Similar rhetorical features have been observed in comparable corpora but have not yet been exploited for retrieval purposes. An example comes from the creators of the IAPR TC-12 corpus:

The first sentence(s) [in an image description] describe(s) the most obvious semantic information (like “a photo of a brown sandy beach”). The latter sentences are used to describe the surroundings or settings of an image, like smaller objects or background information (“a blue sky with clouds on the horizon in the background”) (Grübinger et al., 2006).

²Out of context, “black and white” could also refer to image content, however instances of this phrase encountered in the Phetch data overwhelmingly refer to the image object itself, as in “black and white picture”.

³see Chapter 6 for additional detail

Totals	Images				Words
	4,061				93,720
	Mean	Sdv	Min	Med	Max
Words per title	3	3	1	2	40
Words per description	15	21	1	8	301
Tags per image	15	15	1	11	399

Table 4.1: Overview of the flickr.com sample.

4.2 The Commonness of Descriptions

Descriptive labels of the kind that we find in the Phetch data set are a common feature of online image collections. As a result, a data set like this one, which supports training and evaluation of description-driven retrieval systems, is relevant to the general problem of leveraging a rich and common information source.

4.2.1 An Uncurated Data Sample from Flickr.com

To establish the frequency of descriptive labels in an uncurated data set, we examine a sample of images extracted from the online photo-sharing site Flickr.com. Using Flickr’s Java API, we downloaded 4,061 photo objects and analyzed their structure. Flickr images can be annotated with a wide variety of meta-data, including technical information about the equipment that was used, geo-tags (latitude and longitude where the image was taken), semantic keywords, titles, and full-text descriptions. For the purpose of this comparison we downloaded image ids, titles, descriptions, and semantic tags only. A summary of this sample is given in Table 4.1.

In our sample we found that almost every image was assigned a set of tags, and 94% of the images were also annotated with a title. Not all images were annotated with descriptions, but over 1,500 images from our sample were, making up 38% of the total.

Examining the titles in detail, 40% consist of a single word, while 60% are multi-word titles. Using the same dependency analysis as we used on the Phetch corpus, we observe that dependency patterns consisting only of noun strings occur more often in the Flickr sample. These patterns have

B-day08
 Blue-Car
 Cat-In-A-Box
 Forget-me-nots
 ObservationGesture30
 Self-Portrait
 Steamed_Crabs_03.30.2009
 Thailand_March_2009
 TheRoadToContemporaryArt
 Tippu_home_temple_4888w
 oldphotos_meandlauren1
 littlechinagirlvint

Figure 4.4: Single-word titles with hidden syntactic structure from Flickr.com.

dependency structures composed of multiple *nn* relations, and they account for 497 of the 3063 unique titles. Combining these instances with the unique single-word titles, we can characterize roughly 1785, or 58%, of unique titles as bag-of-words style annotations. Although this indicates that uncurated data may contain less structure than a curated data set like the Phetch corpus, we still see a significant number of parsable multi-word titles: nearly half of all titles do exhibit multi-word syntactic structure.

This analysis may under-estimate the frequency of descriptive titles, since in some instances authors used creative formatting to assign a descriptive title to an image while using only one “word.” Some examples of the single-word titles that appear in the Flickr sample are shown in Figure 4.4.

These examples speak to an important side-effect that results from framing the tasks of object annotation and retrieval in terms of keyword tags: it may require humans as language users to break or abuse the structure of the task in order to meet their communicative needs. Notice that a tag like “TheRentalCar_InThailand” is informative and transparent to a human reader, but a retrieval system that uses keyword matching will be unable to find this tag in response to queries like “rental car” or “Thailand”. Once we propose to break the tag down into component words, we have started down a path that requires the system to analyze and use multi-word descriptive labels.

Flickr Titles	
Pattern	Examples
amod	“Casual Mohawk2” “Mysterious people”
nn	“Coffee Break” “Easter Bunny” “Belly Button”
conj_and	“Madness and Gladness” “Orange and Red” “Rocks and water”
Flickr Descriptions	
Pattern	Examples
nn	“Chicago Girls” “Animal Lovers” “Lake Martin”
amod	“double exposure” “Abandoned mansion”
det nsubj dobj	“The Judge calls Goldberg” “this picture pleases me”
conj_and, det conj_and	“Julie and Patty” “Shakespeare and the Engineer”
aux nsubj dobj	“I should eat that”

Figure 4.5: Common dependency patterns from Flickr titles and descriptions.

Relation	% of Descriptions	% of Titles
nn	36.2%	27.5%
dep	30.0%	15.1%
nsubj	29.8%	8.6%
det	30.3%	6.9%
amod	27.0%	10.2%
dobj	22.7%	3.5%
advmod	18.1%	1.8%
aux	14.7%	1.5%
poss	13.0%	2.0%
conj_and	12.0%	1.4%
prep_in	11.7%	1.3%
cop	11.3%	1.1%
prep_of	11.0%	1.6%

Figure 4.6: Percent of Flickr Descriptions (col 2) and Titles (col 3) where the most common dependency patterns appear.

4.2.2 Syntactic Patterns

In Table 4.5, as in 4.2, the patterns that we see represent entire titles and descriptions. The most common patterns are short because longer patterns tend to be members of the “long tail” of a Zipfian distribution, appearing only once or twice. Even closer similarity to the Phetch data emerges when we examine the most frequent dependency relations, across titles and descriptions of varying length. These relations are shown in 4.5.

Some features of Flickr titles and descriptions cause difficulties for the parser. For example, 3-word titles commonly include a software-generated image name in addition to the human-generated text, as in “oak tree IMG_1590”. These titles lead to spurious dependencies when the parser attempts to link the image name with the phrase structure of the text.

4.2.3 Semantic and Rhetorical Patterns

As in the Phetch data, we can identify common semantic and rhetorical features in Flickr titles and descriptions. Some of these features are even

more prominent in the Flickr data. Phetch phrases, as discussed earlier, may be categorized according to the context in which they must be interpreted: some are grounded only in the image content (“doggies at play”) while others deal with the image itself as an object (“old lithograph”). In addition to these, Flickr titles and descriptions are sometimes grounded against the user’s personal context (“this picture pleases me”; “us in our favorite spot”).

This third type of description is prominent in Flickr data. To estimate the relative frequency of these classes more precisely, we performed an annotation exercise on a small sample of 100 titles. Each title was coded according to its semantic scope, as follows:

Code 1: The title is semantically grounded in the image contents (“Auckland City”, “Silver Pontiac”)

Code 2: The title is semantically grounded in the image object (“Poster”, “Logo”)

Code 3: The title is semantically grounded in an unavailable context, typically interpreted to be a context of personal relevance to the image owner (“I see you!”, “A Better Little Something”)

Code 4: The title is not semantically grounded; this code was assigned to titles that were automatically generated by digital camera software (“DSC_1326”, “IMG_1751”)

Out of this small sample, 44 titles were category 3 and 23 titles were category 4, meaning that 67% of the titles were unavailable for semantic interpretation, even by a human reader. The remaining 23% of labels conveyed meaningful information that could be interpreted in the context of the image contents or the image object itself.

These observations indicate one of the challenges to transitioning image retrieval technology from development on curated data sets to general application. Developing our retrieval system on the Phetch corpus will not allow us to handle all of these features of uncurated data. However, the similarities we see between these corpora indicate that techniques for improving retrieval performance on Phetch can apply to data in the wild, as well.

4.3 Conclusions

In this chapter we described the syntactic, semantic, and rhetorical structure of image descriptions in the Phetch data set. We also explored the frequency of multi-word image descriptions in an uncurated collection, a sample of over 4,000 images from the online photo sharing site Flickr.com. The annotations included titles, descriptions, and tags. We established that more than half of all unique titles and 96% of unique descriptions were composed of multiple words with recoverable syntactic structure.

Predictably, many features of the Flickr sample demonstrated that uncurated data is less regular and more difficult to analyze than data that is collected in a controlled setting. Variance was higher in the Flickr sample than in the Phetch corpus, both in terms of number of words (for titles, descriptions, and tags) and in the re-use of syntactic patterns. In addition, some of these features indicated that bag-of-words style annotation might be more popular in uncurated data than in the Phetch corpus. Nonetheless, we observed a significant number of titles and descriptions in the Flickr sample that exhibited structure beyond bags-of-words.

We also discovered that the range of syntactic, semantic, and pragmatic structures at play in the uncurated collection is wider than the range that we observe in the Phetch corpus. For example, while descriptions in Phetch may be categorized as dealing with image contents (“a picture of a black and white dog”) or with image objects (“a black and white picture”), in Flickr we observe additional classes, including descriptions that depend on a personal experiential context for their interpretation (“Look out!”) and those with no semantic context at all (“IMAGE2351”). In the process of identifying these differences, we also found similarities that help us to understand the task of description-based image retrieval.

These observations support the claim that the Phetch corpus, which is structured to support automatic evaluation on the task of retrieving descriptive image labels, bears a reasonable resemblance to the data “in the wild” that it is meant to represent. As a result, the task of retrieving images based on their Phetch descriptions is one with real-world significance.

Chapter 5

Parameterizations for Description Retrieval and Baseline Results

5.1 Retrieval with Keywords

In Chapter 3, we introduced an experimental structure for image retrieval using the Phetch corpus: one description per image is used to represent a retrieval topic, while the remaining descriptions of the same image make up a document in an indexed collection. Image identity is used as a proxy for relevance, allowing us to evaluate automatically without further annotation.

Figure 5.1 shows how this experiment would work. The image has been annotated with three descriptions. The first description has been used as a query¹, and the remaining two are reserved as `DESCRIPTION` fields in one indexed document that represents this image in the collection. Terms shown in bold represent keyword features that are shared by the indexed document and the query. Retrieval based on keywords generally relies on string-matching² to determine whether the query and the indexed document are descriptions of the same image, as they are in this example.

¹query formulation is discussed further in Section 5.1.2

²refinement via spell-checking and stop-word/synonym lists may be applied



Query Description: “stadium with pageant girls in the foreground”

Index Description 1: “olympic **stadium**, women in front with sashes”

Index Description 2: “people at sporting event; **pageant** contestant girl”

Figure 5.1: Matching a query against indexed descriptions based on keywords. Terms shown in bold are features shared by both descriptions.

5.1.1 Focused Retrieval

This retrieval paradigm can be seen as a type of *focused retrieval*, described in the SIGIR 2008 Workshop on Focused Retrieval as a set of specialized tasks including Question Answering, Passage Retrieval, and Element Retrieval (XML-IR). The defining feature of these tasks is that the object being retrieved is a unit of text within a document:

*Standard document retrieval finds atomic documents, and leaves it to the end-user to locate the relevant information inside... Focused retrieval addresses information retrieval and not simply document retrieval.*³

This definition assumes that the unit of information is smaller than a document in length, for example within a webpage on Elvis Presley where only a sentence or two contains an answer to the question “Where was Elvis born?”

Images in the Phetch collection are encapsulated as documents, rather than sub-spans within documents. However, the alignment of one document to one image means that the unit of information being retrieved is not greater

³<http://www.cs.otago.ac.nz/sigirfocus2008/>

Collection ⁴	Mean Document Size (KB)
GOV	15.2
GOV2	17.7
BLOGS06 Homepage documents	67.1
Phetch	.565

Table 5.1: Document size of some TREC 2008 Web Track collections. Phetch is shown for comparison.

than the user’s information need. Rather, we retrieve units that match the information need in scope.

In addition, the average document in the Phetch corpus is just over 20 words long, and around 270 bytes in size. The text from a typical image with three descriptions is comparable in size to a passage. For comparison, Table 5.1 shows the average document sizes for a selection of TREC test collections from the 2008 Web track.

5.1.2 Indexing and Retrieval Tools

The Indri search engine (Strohman et al., 2005) is a component of the Lemur Toolkit for Language Modeling and Information Retrieval⁵. Lemur is developed by Carnegie Mellon University and the University of Massachusetts, Amherst. Lemur and Indri are freely available and can be downloaded from the Lemur project website. The experiments described in this thesis are all performed with Lemur version 4.5.0, Indri version 2.5.

Indri was designed for scalability, portability, and efficiency, and it is still being supported and extended by IR researchers. These features make it an attractive platform for experiments in information retrieval. In addition, the Indri query language extends Inquery syntax (Callan et al., 1992), supporting a variety of structured operators that are well-suited to the task of focused retrieval. Indri developers describe this aspect of the engine design as follows:

The query language should support complex queries involving evidence combination and the ability to specify a wide variety of

⁵<http://www.lemurproject.org/>

constraints involving proximity, syntax, extracted entities, and document structure.(Strohman et al., 2005)

All of these features play an important role in the experiments presented in this thesis. Proximity features and field restrictions allow us to set a high baseline that takes the known structure of Phetch image-documents into account, while still modeling the content of those descriptions as unordered sets of keywords.

Indri supports these features off the shelf, however the range of fields that are accessible in structured queries are dependent on the document representation we choose. The structure shown in Figure 3.2 is the baseline representation that assumes no knowledge of world semantics, only an understanding of how descriptions have been assigned to images. Experiments in Section 7.2 add semantic annotations to this structure using an sgml field called SEM. Sample Indri queries that take advantage of this structure are shown in Figure 5.2.

5.1.3 Procedure

To establish a baseline for retrieval performance on the Phetch data, we identify a section of the data set, perform indexing and retrieval with Indri.

In our retrieval paradigm the main criterion for success is returning the single image of interest, at the lowest rank possible. Since each description corresponds to precisely one image, we seek a metric that describes, on average, where in the results list that image appeared. This metric is Mean reciprocal rank (MRR). MRR is defined as the average, over all queries, of 1 divided by the rank where the correct document was found. The formula is shown in Equation 5.1.

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{r_i} \quad (5.1)$$

where r_i is the rank at which the first relevant document was found for query i .

Although metrics like Mean Average Precision (MAP) are more discriminative in retrieval settings where multiple documents are relevant to each topic, in our retrieval setting the first relevant document is known to be the only relevant document. For this reason, MRR has been widely used to

```
#combine(pageant girls blond face)
```

Plain Query: Ranks images by their combined match to all these terms

```
#combine(pageant.description girls.description blond.sem face.sem)
```

Field Restriction: Ranks images by their match to “pageant” and “girls” in the description field, and to “blond” and “face” in the sem field.

```
#combine(pageant.(description) girls.(description) blond.(sem) face.(sem))
```

Field Model: Ranks images by their match to all terms, but scores “pageant” and “girls” with a model trained on description fields, and scores “blond” and “face” with a model trained on sem fields.

Figure 5.2: Indri structured queries.



Query Description: “theglobalmuse com elite artist award for musical excellence”

Index Description: “elite artist award - red text”

Figure 5.3: An example image composed mostly of text. Such images were pruned from the 5A data set for these experiments.

evaluate *known-item search* (Ogilvie and Callan, 2003), another example of focused retrieval that closely resembles the description-retrieval problem.

Data

The data set used in this chapter is a subset of the Phetch section 5A. These images have each been annotated with at least five descriptions, allowing us to isolate three sets of topics: one for training, one for development, and one for testing. The remaining two descriptions are used as the index representation of the image-document. In addition, we have pruned from this data set all images that contain text in the image itself; an example is shown in Figure 5.3. This pruning step is motivated by the observation that images with a large percentage of text are poor representatives of the labeled object retrieval problem. In contrast, images that have descriptive labels represent the most difficult and most interesting area of the problem space. The size of the pruned data set is described in Table 5.2. Like the main sections of Phetch, this subset and the test, training, and development queries can be reproduced by applying freely distributed scripts to the full plain-text Phetch corpus⁶.

⁶Send email requests to atribble@cs.cmu.edu

Totals	Images	Tokens
	700	68,867
	Mean Tokens per Description	Total Tokens
Collection	18	32,326
Training Queries	17	12,098
Test Queries	17	12,329
Development Queries	17	12,114

Table 5.2: Word counts for the 5A data set, non-textual images.

Indexing and retrieval

After extracting and formatting the collection and the training queries, we index the collection using the binary program `IndriBuildIndex`⁷. `IndriBuildIndex` can be configured to use built-in document transformation features, including implementations of the Porter and Krovetz stemmers as well as stopword removal. We specify these settings in a parameters file. In addition, to allow the indexing system to capture the corpus-specific sgml fields that we have used to model Phetch documents, we must add each of these fields to the parameters file. An example is shown in Appendix A.

After indexing, we use the binary program `IndriRunQuery`⁸ to execute retrieval for each of the training queries. `IndriRunQuery` takes a set of parameters that are parallel to the build parameters and can be specified in a second parameters file. These parameters specify the location of the index that will be searched, a limit on the amount of memory to allow for the retrieval process, the maximum number of results to return, and optional smoothing rules that are used to tune the retrieval model.

Document model

Indri implements a retrieval model that combines the language modeling approach of Ponte and Croft (1998), which estimates word probabilities, with the inference network approach of Turtle and Croft (1991) for combining beliefs into a single document-level retrieval score. In order to gracefully

⁷<http://www.lemurproject.org/lemur/indexing.php#IndriBuildIndex>

⁸<http://www.lemurproject.org/lemur/indexing.php#IndriRunQuery>

handle data sparseness and rarely- or never-observed language model events, Indri applies smoothing strategies that mix the document-specific scores with a background language model estimated from the entire collection. These options are described in more detail on Indri developer Don Metzler’s web-based documentation of the retrieval model⁹.

Output and Scoring

The output from `IndriRunQuery` conforms to the `trecFormat` style published by the National Institute of Standards and Technology (NIST). For a run that retrieves N documents per query, there are at most $Q \times N$ lines in the result file, one line per returned document for each of Q queries.

To evaluate the results, we apply the `trec_eval` tool, also published by NIST¹⁰. This tool compares the result file to a second file that encodes the known-relevant documents for each query. In our retrieval setting, there is precisely one per query: the document that contains descriptions of the same image as the one described by the query. This feature of the retrieval setting makes Mean reciprocal rank (MRR) a good fit for our evaluations. MRR measures the rank of the first good answer; in the single-relevant-document setting, we are interested precisely in knowing whether the first good answer is also the correct answer.

5.1.4 Results

The results of retrieval using this procedure are shown in Table 5.3. These results serve as the baseline for evaluating refinements to the retrieval process that are reported in Chapter 7. Performance on the training queries and test queries is similar.

5.2 Additional Parameterizations

The system described in 5.1 compares a query description to an indexed document description based on words that appear in both. In this section we describe other feature sets, or *parameterizations* that can be used to determine whether a query and an indexed document describe the same image.

⁹<http://ciir.cs.umass.edu/metzler/indriretmodel.html#estimation>

¹⁰http://trec.nist.gov/trec_eval/

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)	
	Test Set	Training Set
KW (No Text)	0.8241	0.8172
KW (All Images)	0.9304	0.9259

KW=keywords

Table 5.3: Results of retrieval with the baseline model.

Experiments and results of applying these parameterizations are given next in Chapter 7.

5.2.1 Keyword Refinements

Query refinement addresses the limitations of string-matching in keyword-driven document retrieval. These techniques increase the sensitivity of the system to near-matches, in addition to exact matches, for keywords from the query. They include spelling correction, word splitting, word merging, phrase segmentation, and stemming. An overview is given in Guo et al. (2008).

Query refinement (coupled with the corresponding refinement of index documents) allows some of the non-literal matches from the example in Figure 5.1 to contribute to retrieval. The result of stemming our example query is shown in Figure 5.4. The query word "girl" now matches "girls" in our index descriptions. Rarer words like "stadium" are often misspelled, and this type of refinement can correct these errors, as well. In some cases the stemmed keywords are no longer full words; as a result we may refer to them as query terms.

In performing query refinement to find a better match between the query and our index descriptions, we assume that the query already contains the right words, and those words are informative enough to distinguish matching from non-matching index descriptions. The surface forms of the words only need to be corrected or massaged on the morphological level.

Indri natively supports stemming, and spelling correction in SconeImage is applied with a perl-module wrapper to the gnu ASpell utility¹¹. Knowledge-

¹¹<http://search.cpan.org/~hank/Text-Aspell/Aspell.pm>

Query Description:	“stadium with pageant girl in the foreground”
Index Description 1:	“olympic stadium , wom in front with sash”
Index Description 2:	“people at sport event; pageant contestant girl ”

Figure 5.4: Example of refined keyword representation: spell-correction and stemming. Terms shown in bold are features shared by the query and index descriptions.

augmented approaches have also been developed by other researchers for some types of query refinement, including spelling correction (Ruch, 2002). Although our current system does not appeal to the knowledge base during the query refinement phase, the overall architecture lends itself well to future experiments in this area.

5.2.2 Synonyms and Semantic Expansion Terms

Query expansion is the process of adding keywords to the original terms of a query, with the goal of capturing a wider range of surface forms for the underlying concepts in the original query. These terms capture synonyms or other ontologically-related words, including hypernyms and meronyms.

The goal of query expansion is to capture words that did not appear in the query, but easily could have. Consider for example the query term “foreground” and the term “front” in Figure 5.4. Two people had the same concept in mind while writing these descriptions, but happened to choose different words.

A knowledge base that provides conceptual distance between surface words seems like the ideal resource to support this step in the retrieval process. In practice, IR researchers have found it difficult to perform query expansion in a way that significantly improves retrieval performance, particularly with knowledge-based techniques that rely on WordNet as their lexical-semantic resource (Voorhees, 1994).

A central hypothesis of our work is that a knowledge base that combines general lexical-semantic knowledge from WordNet with domain-specific and format-specific knowledge (in our case, the format is image descriptions) can be leveraged with success to improve end-to-end retrieval results. The knowl-

Query Description:	“stadium with pageant girl in the foreground {pageant} {ceremony} {person} {female} {image foreground} ”
Index Description 1:	“olympic stadium , wom in front with sash {sash} {pageant} {person} {female} {image foreground} ”
Index Description 2:	“people at sport event; pageant contestant girl {person} {female} ”

Figure 5.5: Example of semantic expansion. Terms shown in bold are features shared by the query and index descriptions.

edge base can be used to annotate the index descriptions as well as incoming queries with concepts that appear in the text. These semantic features allow the retrieval process to condition on semantic as well as orthographic properties of an image description.

Consider our example again, this time with additional features generated by semantic annotation. The result is shown in Figure 5.5 Concepts from the knowledge base are shown in curly braces, as in {concept}.

This example demonstrates how adding semantic expansion terms to our parameterization of the image retrieval problem mitigates some deficiencies of the keywords-only model. For some terms, the effect is identical to stemming: “girl” and “girls” share the same root and they also share the same conceptual entry in the knowledge base. For other terms, the addition of a semantic feature captures similarity that is unavailable from the orthography: “sash” has triggered the expansion concept {pageant}, which appears in the query. “Woman” now matches “girl”.

5.2.3 Dependency Relations

Dependency relations were introduced in Chapter 4 as a way to find syntactic patterns in image descriptions and titles. They can also be used to reveal structural similarities between a query and index description for the same image. Dependency structures have been applied with success in

systems for Information Retrieval, Question Answering, and Textual Entailment. Haghighi, et. al. provide an example of how dependencies contribute to such systems:

Consider a Question Answer system searching for an answer to When was Israel established? A representation which did not utilize syntax would probably enthusiastically return an answer ... “The National Institute for Psychobiology in Israel was established in 1979.” In this example, it’s important to try to match relationships as well as words. In particular, any answer to the question should preserve the dependency between Israel and established. (Haghighi et al., 2005)

In description retrieval, as in question answering, dependency information can help to recapture information lost by keywords alone. An example image annotated with these structures is shown in Figure 5.6. When this query was submitted to the baseline system, the image shown in Figure 5.7 was retrieved. Although more words from the query appear in this alternative description, the dependency annotation reveals the mismatch between “black and white photo” and “black suit white shirt”.

5.2.4 Knowledge-Augmented Dependency Relations

This feature type combines the power of knowledge-base annotations with the dependency structures just described. Semantic annotations function much the same way as they do at the query expansion stage: when added to dependency structures, they provide a layer of abstraction that allows looser, more conceptual matching of these structures. Some additional dependencies can be collapsed as a result of multi-word concepts that have been semantically annotated. An example is shown in Figure 5.8. In this example, the semantic augmentation of reveals that the phrase “black pic” and “white photo”, which commonly occur when annotators describe black-and-white images, are more alike than “white photo” and “white shirt”.

5.3 Conclusions

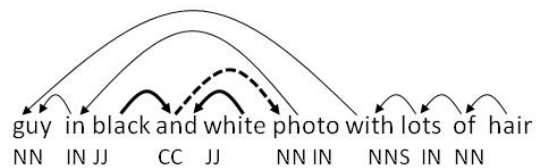
In this chapter we have described a retrieval experiment on the 5A section of the Phetch corpus that uses a bag-of-words representation for queries and



Query Description: “guy in black and white photo with lots of hair”

Dependency

Features:



Index Description: “**black** and **white** pic of a man wearing a jacket”

Dependency

Features:

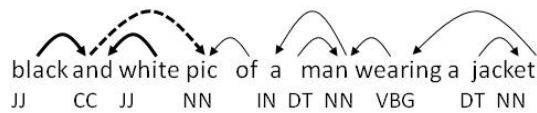
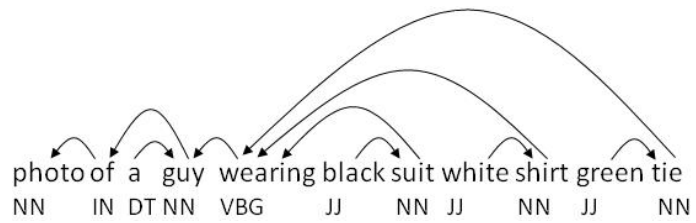


Figure 5.6: Sample dependency annotation. Terms and relations in bold are common to index and query descriptions; relations in dashed-bold are common based on part-of-speech matching.



Index Description: **“photo of a guy wearing black suit, white shirt, green tie”**

Dependency



Features:

Figure 5.7: A second example of dependency annotation. Terms and relations shown in bold are common to this index description and the query shown in Figure 5.6.

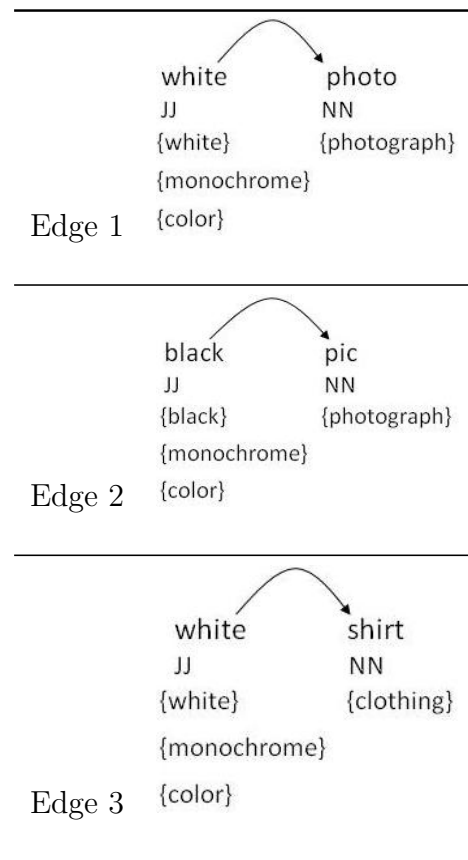


Figure 5.8: Example dependency edges augmented with semantic concepts.

indexed descriptions. We introduce the data, tools, and procedure for building a description-retrieval pipeline based on the Indri Indexing and Retrieval System, part of the Lemur Information Retrieval Toolkit.

The results of this experiment set a strong baseline for the problem of retrieving image descriptions. However we have also explored a series of more sophisticated parameterizations of the problem that may lead to better results: refined keywords, semantic annotations, and dependency relations. Examples have shown that these types of features can bring queries closer to indexed descriptions that should match them, and farther from descriptions that should not.

Next, we will perform a more detailed error analysis of the cases where the baseline system fails. In Chapter 6, we present a classification system for these errors, and hypothesize ways that our advanced parameterizations can reduce them. These hypotheses are tested in Chapter 7, where we implement and test each parameterization.

Chapter 6

A Methodology for Error Analysis in Description Retrieval

6.1 Introduction

Error analysis allows us to understand how well the image retrieval system is working and to prioritize the development of new features for the system. In this chapter we introduce a methodology for analyzing the errors that a description-based image retrieval system can make and for identifying the features of a data set that trigger these errors. This analysis leads directly to hypotheses about how to reduce errors of each specific type, which we evaluate in Chapter 7.

6.2 Counting Retrieval Errors

In this section we investigate errors made by the retrieval system described in Chapter 5 for the data set Phetch 5A. This data set is composed of 2,947 images, each annotated with 5 descriptions. In Chapter 5 one description was used as a retrieval topic while two of the remaining descriptions were indexed as a document in the collection. In this section we will vary the number of descriptions that are indexed, resulting in a series of what-if analyses. We start with a version of the 5A data set where only one description has been retained for each indexed image (5A.1).

Rank	Frequency
1	2525 (85% of queries)
2	151
3	61
4	39
5	17
6	16
7	12
8	13
9	5
10	5
11+ (no result)	100 (3% of queries)

Table 6.1: Rank of the relevant image in baseline run of Phetch 5A training queries.

A retrieval run using the baseline system results in $MRR=0.9259$ on all training queries of 5A data set. Now we will look in more detail at the rank of the relevant image within the result list for each query. If the relevant image was returned at rank 1, no error occurred. A visual representation of this case is shown at the top of Figure 6.1. The error cases in this figure are described in the following sections. A retrieval run with perfect performance, where the relevant image was retrieved first for every query, would achieve an MRR of 1.0. An MRR of 11+ would indicate that no query resulted in the relevant image being returned within the top 10.

6.2.1 Retrieval Failures

To identify the number and severity of errors, we look to the rank of the relevant image in the result list for all of the queries in the current retrieval run. The distribution of these ranks over all of the queries in the 5A data set is shown in Table 6.1.

Table 6.1 shows that 85% of the queries resulted in the correct response from the system, returning the relevant image at the top of the 10-best list. However for 100 queries, the relevant image was not returned at all in the top 10. These queries are of most interest for our error analysis; they

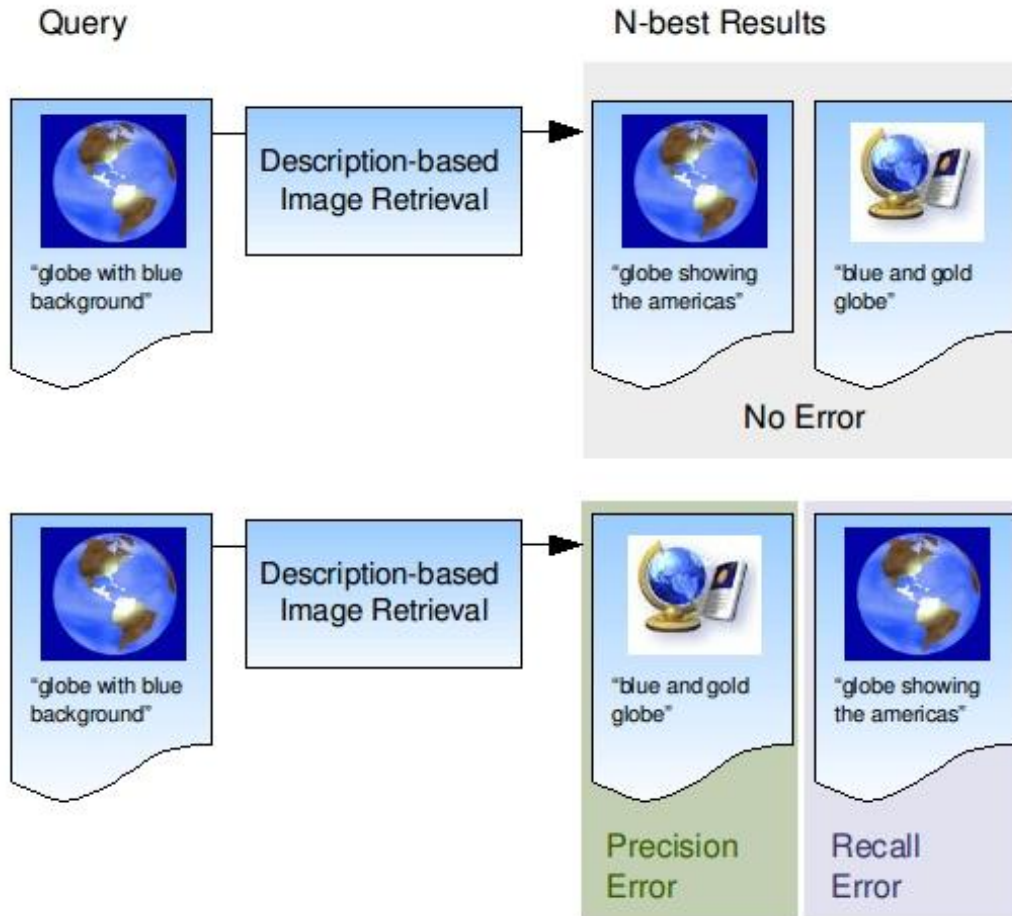


Figure 6.1: Visual depiction of retrieval outcomes.

represent a small but significant percentage of the training set where serious errors occurred. We refer to these errors as retrieval failures. By addressing retrieval failures, we have the opportunity to increase the overall performance of the system.

6.2.2 Estimating Bounds on Improvement

How much of an increase in performance can we expect from correcting these errors? Figure 6.2 shows the effect on MRR for the entire baseline retrieval run. To generate this analysis, we manipulate the errorful result lists, inserting the correct relevant image. For example, for all result lists where no relevant image was returned, we inserted the relevant image at rank 10. This data point is shown on the far-left side of Figure 6.2. Other data points are generated by inserting the relevant image at rank 9, 8, 7, etc. The experiment indicates that if our knowledge-based retrieval methods can outperform the baseline in these cases, while not disturbing the results in the rest of the data set, we could improve overall MRR by as much as 3.8% on this data set. This analysis indicates that it may be worthwhile to analyze and correct retrieval failures by making improvements to the baseline retrieval algorithm.

For a more complete picture of the projected gain from error correction, we extend this analysis to address errors other than retrieval failures. The goal of making improvements to the retrieval algorithm would be to return the relevant document as near as possible to the top of the 10-best list. When we corrected retrieval failures, we pushed relevant images that were returned at rank $N > 10$ into the 10-best list at rank 1. Figure 6.3 shows the result of correcting errors where the relevant image was returned at rank $N = 10, 9, 8$, etc. For each of these data points, we manipulate the result list to place the relevant images at rank $N = 1$ instead. This experiment projects the potential gain in performance that we could achieve from a re-ranking algorithm that corrects mis-ranked relevant images in the 10-best list returned by the baseline retrieval system.

This projection indicates that after correcting retrieval failures, effective re-ranking of the images found by the baseline system can have a significant impact on MRR. It supports the distribution shown in Table 1, in that a steep gain is predicted if we can correct places where the baseline system failed to differentiate correctly between similar images that were returned at ranks 1, 2, and 3.

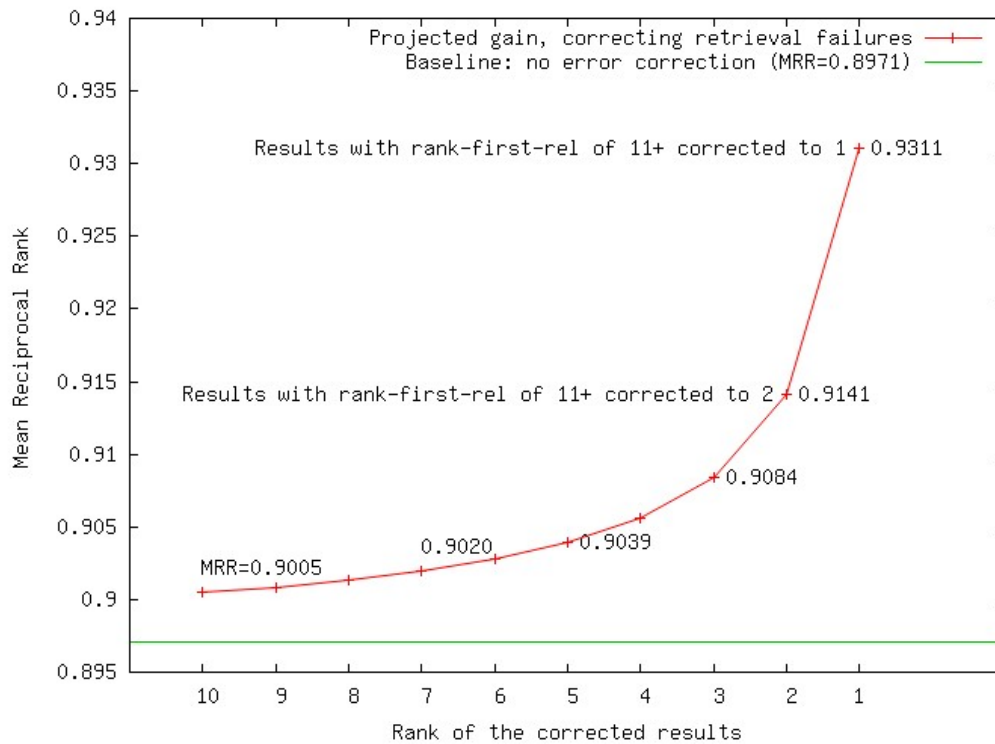


Figure 6.2: Expected gains in MRR from correcting retrieval failures. Relevant images were inserted into the result list at rank $N = 10, 9, 8$, etc. (shown on the x-axis).

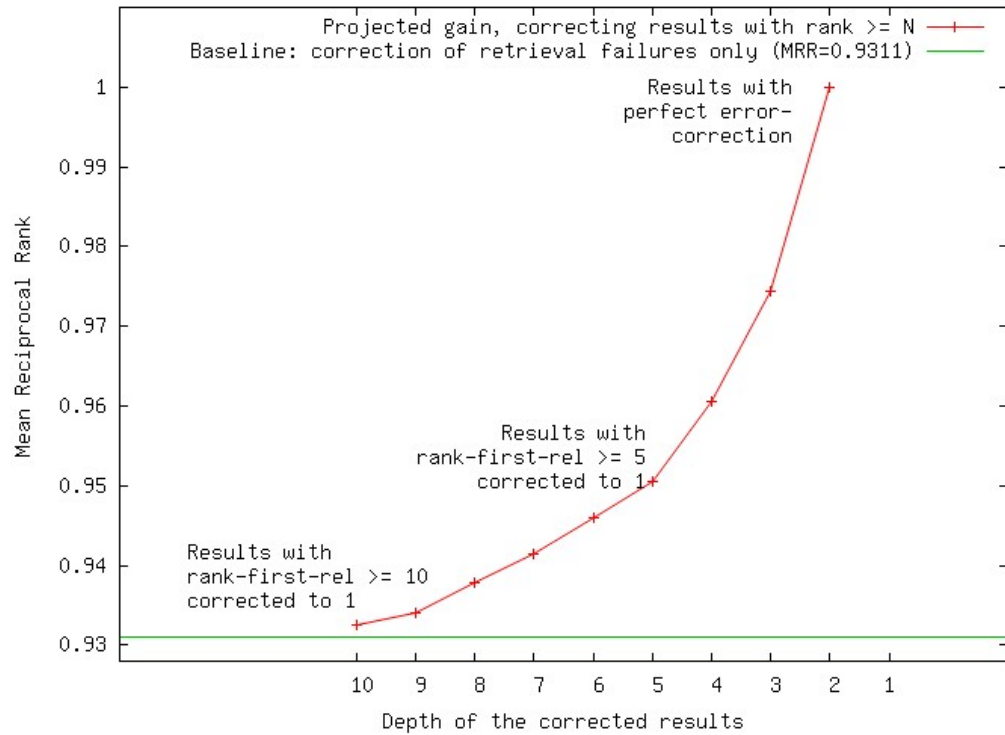


Figure 6.3: Expected gains in MRR from correcting errors. Relevant images found at rank $N = 10, 9, 8$, etc. were inserted into the result list at rank 1.

6.2.3 Additional Data

Many retrieval errors are triggered by mismatches in the language used by individual describers of an image. As we add more annotations from additional describers to an image-document, intuition tells us that it becomes more likely that one of these labels will match the description given by the query. If this is the case, we could also increase system performance on this task by asking more annotators to describe each image in the index. How do the projected gains from error correction compare with the results we would achieve simply by adding more labels to the data set?

To investigate the effect of the number of descriptions on retrieval performance, we re-introduce the descriptions that were ablated from the 5A data set. This results in a series of retrieval runs, all using the same set of queries. In each run we use additional descriptions per indexed image-document. The result is shown in 6.4.

The first data point shows performance when images are indexed using only the tags. Performance increases dramatically when we add a single description to each indexed document. The expected gain begins to plateau between four and five indexed descriptions.

In Figure 6.5 we see these analyses shown together on a single graph. By comparing the curve shown in purple (the result of adding descriptions) to the curve shown in green (the result of correcting retrieval failures to rank 10, 9, 8, etc.), we see that correcting retrieval failures potentially achieves a similar beneficial effect on MRR as annotating image-documents with additional 2-3 descriptions. By annotating with 4 or 5 descriptions, we achieve higher overall performance than we could hope to reach by correcting retrieval failures alone on this data set. However, if we assume that retrieval failures can be corrected and we begin to re-rank the 10-best list as well, we achieve the curve shown in blue, which yields higher performance than additional labels for most points on the two curves.

6.3 Classifying Retrieval Errors

In Section 6.2, we discussed the number of errors produced by the baseline retrieval system and projected the gains in performance we can expect from correcting them. Now we turn to a discussion of what these errors look like and why they occur. The experiments reported in this section were performed

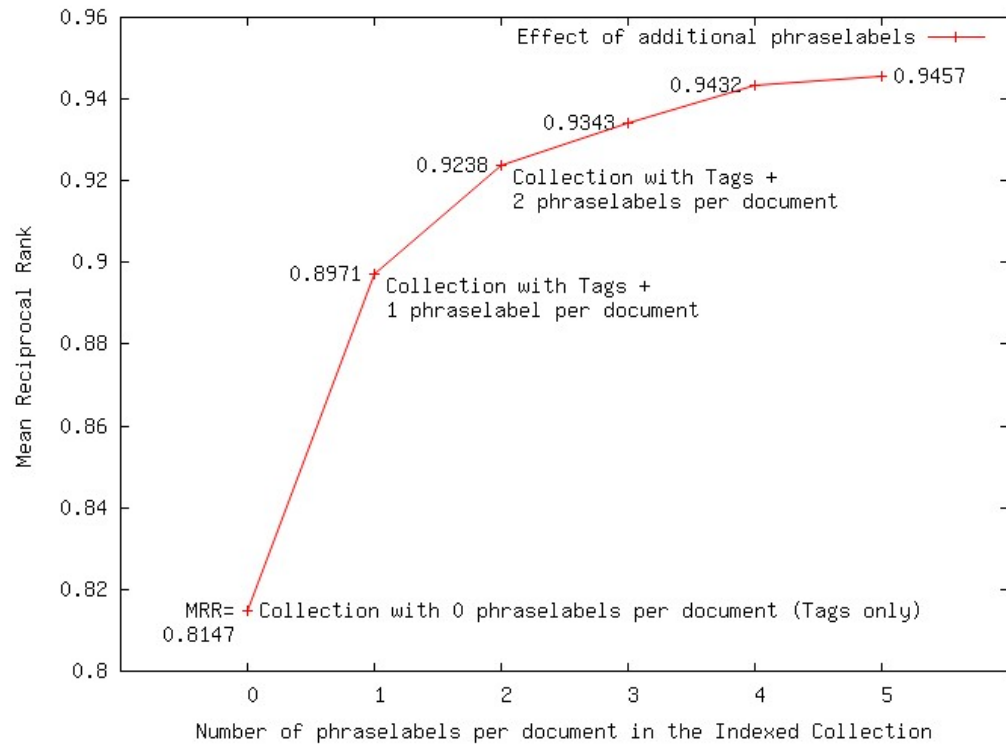


Figure 6.4: Expected gains in MRR from additional descriptions.

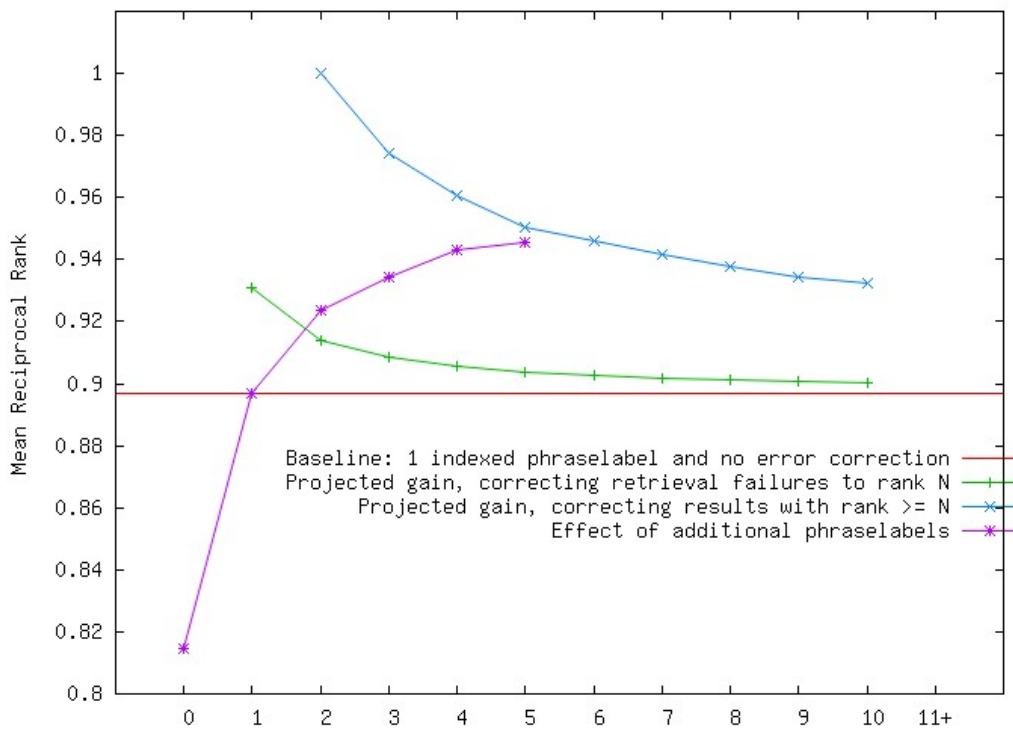


Figure 6.5: Comparison of expected gains from correction vs. additional data.

on section 3A of the Phetch data set.

When the system assigns the relevant image a rank of $N > 1$, at least two errors have occurred (a visual representation of this case is shown on the bottom half of Figure 6.1). First, the system compared a non-relevant image description to the query and determined that they describe the same image, when in fact they do not. This error is a type of false-positive judgment, which we refer to as a precision error, because it dilutes the result list with a non-relevant image at a high rank. Second, the system compared the relevant image description to the query and determined that they did not describe the same image, at least not confidently enough to rank the document first in the result list. This is a type of false-negative judgment, which we refer to as a recall error, since it implies that the system failed to recognize the relevant image when it appeared.

6.3.1 Classes of Precision Error

Because image retrieval in this setting relies on the comparison of passages of text, we will characterize precision errors according to the triggers that cause two such passages to seem more similar than they really are. These triggers make up the error types that we have annotated in our baseline retrieval run; they are summarized in Table 6.2.

The vocabulary of error classes is motivated by the research goals of this thesis. We have hypothesized that the bag-of-words representation for image descriptions leads to errors because it fails to recognize certain types of textual similarity. Specifically, the bag-of-words model fails to capture semantic similarities that are obscured by surface features like word choice, and it fails to follow the inferential chains of reasoning that human annotators envision between their descriptions and the content of an image. As a result, our error classes are composed of specialized cases of semantic and inferential mismatch that we expect to see in the errorful retrieval runs.

We arrived at this vocabulary of error types in an iterative fashion, based on observations in a sample of retrieval results from the Phetch 3A data set. In a first pass, we annotated this development sample with free-text descriptions of the evidence that a human might use to correct the errors made by the baseline system. In a second pass, these annotations were distilled by hand into a set of 14 phenomena that result in retrieval error. These 14 classes were used to re-annotate the sample. After this pass, a final revision of the annotation classes was made to focus on the most frequent

Error Type	Description and Examples
Ontology	The wrong word meaning triggered a false match “on the bank” matched “at the bank”
Faulty inference	Match based on faulty inference; also includes false match triggered by matching verbs with mismatched arguments “green bandana” matched “green shirt”, “man skating” matched “girl skating”
Contradiction	Failed to recognize contradictions; could catch with a model for contradiction “black background” matched “blue background”
Missing elements	Failed to penalize for missing major elements “globe on a stand” matched “globe”
Quantification	Failed to recognize quantification mismatches, in particular mismatched quantities of people “a color photo” matched “3 photos”, “1 guy eating fries” matched “a guy and a girl eating fries”
Negation	Failed to recognize negation “not smiling” matched “smiling”
Analogy	Failed to penalize for a hedge or analogy “tree shaped like a hat” matched “hat”
Media	Failed to recognize mismatching media types “cartoon” matched a photograph, “black and white photo” matched a color image

Table 6.2: Classes of precision error.

and clearly-defined classes. The resulting vocabulary contains 8 classes with precise definitions in the precision-error and recall-error contexts. These classes are not mutually exclusive; rather, a given retrieval error can be annotated with all classifications that apply.

6.3.2 Classes of Recall Error

The eight error classes defined above can be applied to recall errors, as well, but with slightly adapted definitions. In the case of recall errors, we are interested in the triggers that cause two such passages to seem more *dissimilar* than they really are. These triggers are summarized in Table 6.3. When these types of mismatch occur, the indexed description for an image may be incorrectly ruled out as a match for the query description of the same image.

Table 6.4 shows the frequency of these error types in an annotated sample of the 3A data set. This sample contains 50 queries. For each query we make two comparisons: we compare the query description to the indexed description of the same image in order to annotate the recall errors. We also compare the query description to the indexed description that was retrieved at rank 1 for this query, in order to annotate the precision errors.

6.3.3 Frequency of Errors by Class

Although not all of the errorful results returned by the baseline system involve errors from these classes, most of them do (90% of precision errors and 86% of recall errors). Recall errors were annotated with 3.3 of these classes, on average, and precision errors were annotated with an average of 2.5 classes.

The most frequently-appearing class in the case of recall errors were Ontology-related; that is, surface-level mismatch between concepts that would be identical or closely linked in an ontological representation of background knowledge. This is an interesting result from the point of view of error-correction strategies. When we move to an image representation that makes use of a domain-specific ontology, we aim to address these errors. The effect will be less pronounced in the case of Precision errors, where only 14% exhibited Ontology-related mismatches. This corresponds to our intuition that by referencing an ontology, we could recognize synonyms that prevent the baseline system from retrieving the correct image, increasing recall of the retrieval system.

Error Type	Description and Examples
Ontology	Failed to match words that would be the same or similar classes in an Ontology “shawl” failed to match “wrap”
Faulty inference	Failed to make a relevant inference; could be fixed with inference rules “lips are puckered” failed to match “getting ready to kiss”
Contradiction	Failed to match due to faulty contradiction “black or blue background” failed to match “black background”
Missing elements	Failed to match on major elements and ignore minor missing elements “guy smiling with glasses” failed to match “a guy smiling”
Quantification	Failed to recognize matching quantification, especially matching quantities of people “two guys” failed to match “a guy with another guy”
Negation	Failed to match due to negation “not smiling” failed to match “frowning”
Analogy	Failed to recognize and match a hedge or analogy “looks like a hat” failed to match “hat”
Media	Failed to recognize matching media types “photo” failed to match a photograph, “black and white” failed to match “black line drawing”

Table 6.3: Classes of recall error.

Error Type	Frequency in Recall Errors	Frequency in Precision Errors
Ontology	36 (72%)	7 (14%)
Quantification	34 (68%)	26 (52%)
Faulty inference	29 (58%)	20(40%)
Missing elements	23 (46%)	27(54%)
Contradiction	20(40%)	29 (58%)
Media	15 (30%)	14 (28%)
Analogy	4 (8%)	3 (6%)
Negation	4 (8%)	0 (0%)
Any	43 (86%)	45 (90%)
Total	165	126
Average	3.3	2.52

Table 6.4: Frequency of error classes.

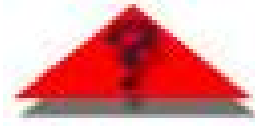
The most frequently-occurring classification of precision errors relates to contradiction. In these cases, the baseline system failed to recognize an explicit contradiction between the query and the image description that was retrieved at rank 1. A system with a model for recognizing contradiction might be able to correct the baseline system in nearly 60% of the cases where it currently makes Precision errors. This result supports recent work on other textual inference tasks, where such models are already being applied (e.g. Recognizing Textual Entailment, Question Answering).

6.3.4 Linguistic Features Contributing to Error

The precision errors and recall errors described above categorize sources of mismatch that can occur between one person’s description of an image and another’s. But what if we want to determine, for a given corpus of descriptions, whether these errors are likely? To do this we need a new set of intrinsic features of the image descriptions. In this section we describe on possible set of such features and establish how they correspond in our data set to precision and recall errors during baseline retrieval. Our features are summarized in Table 6.5. Some of the classes are self-explanatory; for others we give a detailed description in the sections below.

Feature	Description and Examples
Color	Describes visual color of image contents “yellow shirt”
Size	Describes visual size of image contents “big house”
Shape	Describes visual shape of image contents “three triangles”
Texture, Pattern	Describes visual texture of image contents “wood panels”
Spatial Arrangement	Describes visual layout of image contents “side by side”
Frame of Reference	Description uses image as a frame of reference “looking at camera”
Image as an Object	Refers to the image as an object “logo”, “cartoon”, “portrait”
Nonrelevant Discourse	Describes nonrelevant commentary “I can’t make out the text why is this image in here?”
Abstractions	Uses abstract or under-specified visual elements, including image text “cherry-shaped thing”, “letters spelling BACK”
Misleading Orthography	Spelling errors, abbreviations, or unusual word use “2” for “two”, “thingy”
Inaccuracy	Gives inaccurate information with respect to the image “shirt” for a dress, “house” for a hospital

Table 6.5: Error-inducing features in the Phetch 3A data section.



Description: “small red triangle; blue question mark in the middle; triangle has a shadow”

Figure 6.6: Sample image with *visual features* in the description.



Description: “two women; miss something; some sort of pageant; both smiling”

Figure 6.7: Sample image without *visual features* in the description.

Visual features

Visual attributes describing the subject matter of an image occur in most of the image descriptions we observed. These attributes include five subcategories: color, size, shape, texture (including patterns) and spatial arrangement. Example phrases that correspond to this type of feature are shown in Table 6.5.

An example of an image that is described almost entirely with visual attributes is shown in Figure 6.6. However not every image description uses this type of feature. An example image description that uses none of the visual features is given in Figure 6.7.

Image contents and image objects

Because an image is a representation of another object, most of the text of an image description refers to the subject captured in the image. For example, “black and white dog” in this context usually means that the annotator is looking for an image where a dog with black and white fur appears. In some cases, the image itself is the object of description, as in “black and white photo”. This second type of language can be problematic for a bag-of-words retrieval model since there is no feature that allows the system to know the difference between “black and white” in these two contexts.

Examples of this type of language include explicit references to the image-object, like “photo”, “cartoon”, or “image”. They also include phrases that describe visual attributes of the image-object, as in “grainy”, “in color”, or “high-resolution”. Finally, they can be embedded in a sentence that has mixed references to both the image contents and the image-object, as in “photo of a girl”. Several examples from the Phetch 3A data set are shown in Table 6.5.

In addition to identifying phrases that refer to the image as an object of description, we also distinguish phrases that use the image as a frame of reference. Phrases that describe image contents “in the background”, “in front”, or “on the left” without any explicit point of reference typically imply that the frame of reference is the entire image. More examples are given in Table 6.5.

Nonrelevant, abstract, and misleading descriptions

In our data set, some content of the text descriptions is not relevant to the task of retrieving the image in a collection. This type of off-task annotation is likely to occur in any uncurated data set; as a result we would like to track its frequency along with the other features listed here. A frequent source of off-task commentary is hedging, as in the example “I think maybe it’s a plant of some kind.” For the purpose of retrieving images based on this description, the only truly relevant term is “plant”. One could imagine using the rest of the description to deduce an uncertainty value for the query term “plant”, but it seems counterintuitive to include the term “maybe” in the query itself. More examples are shown in Table 6.5.

Like many types of human-generated text, the descriptions in our data set are subject to spelling errors, poor word choices, and other idiosyncratic



Description: “man; man in glasses; in suit; in suit; little bar-code thingy”

Figure 6.8: Sample image with misleading orthography and non-relevant repetition in the description.

uses of language that are difficult for humans, as well as retrieval algorithms, to handle. We annotate this type of feature in a description as Misleading orthography. The example in Figure 6.8 shows a description with a nonstandard word, “thingy”. Such words are unlikely to appear in the descriptions from more than one image describer.

Finally, in addition to misleading uses of language, some image describers simply provide inaccurate information. Common mistakes include references to a “man” in a picture of a woman, or “boy” in a picture of a girl.

Frequency of intrinsic features

Table 6.6 shows the frequency of these features in our annotated sample. For every query in the sample, we annotate three image descriptions with these features: the query description, the indexed description of the relevant image, and the indexed description of the non-relevant image returned by the baseline system at rank 1. These are the same images that form the pairwise comparisons we annotated in the section above, but this time annotation is performed on each image description in isolation.

The features from our annotation vocabulary appeared in nearly every image description that we annotated, but some were more prominent than

others. Cells shown in bold in Table 6.6 correspond to features that appeared in more than half of all of the image descriptions. These include color, spatial arrangement, and image-as-frame-of-reference.

Most of the image descriptions mention color, however this feature can be captured roughly at the single-word level. Spatial arrangement is more interesting because it is inherently a relational feature of the description: some element of the image occurs “on the left” of another, for example. While the bag-of-words model is unable to distinguish between “chair on the left of the table” from “table on the left of the chair”, a more sophisticated model that takes some syntactic features into account might be better able to handle descriptions with this feature, which occurs in well over half of the descriptions in our sample.

The frame-of-reference category is also of particular interest in the context of bringing better text representations to bear on the problem. To handle image descriptions with this feature appropriately, a system must distinguish between sentences like the two above, and sentences like “chair on the left” (i.e. of the image frame). A symbolic model of image descriptions that has separate representations for image content and the image object is a solution that we will try for this task.

6.4 Conclusions

Given this background knowledge about the nature of retrieval errors in a baseline retrieval run, we can identify some specific strategies for improving retrieval performance. We have established a set of error classes and textual features that contribute to retrieval error under the bag-of-words model. In the next chapter we will establish a more knowledge-rich model for representing image descriptions and use that model to implement handlers for the error classes described here: Ontology-based matching and inference, handling of quantification over major content elements, negation, analogy, and a model of media types.

Feature	Frequency		
	in Training Queries	in Relevant Descriptions	in Top-ranked Descriptions
Color	43 (86%)	48 (96%)	46 (92%)
Size	15 (30%)	21 (42%)	19 (38%)
Shape	10 (20%)	15 (30%)	17 (34%)
Texture/Pattern	6 (12%)	10 (20%)	8 (16%)
Spatial Arrangement	25 (50%)	39 (78%)	34 (68%)
Frame of Reference	31 (62%)	33 (66%)	38 (76%)
Image as an Object	18 (36%)	29 (58%)	30 (60%)
Nonrelevant Discourse	18 (36%)	18 (36%)	18 (36%)
Abstractions	15 (30%)	21 (42%)	27 (54%)
Misleading Orthography	22 (44%)	25 (50%)	31 (62%)
Inaccuracy	10 (20%)	10 (20%)	6 (12%)
Any Feature	49 (98%)	49 (98%)	50 (100%)
Average N. Features	4.26	3.76	4.12

Table 6.6: Frequency of error-inducing features.

Chapter 7

Applied Textual Inference Methods and Results

7.1 Introduction

In Chapter 5, we presented experimental results from a strong baseline retrieval system that models image descriptions from the Phetch data set as bags-of-words. The error analysis in Chapter 6 reveals that many of the errors made by the bag-of-words model can be framed in terms of specific knowledge-based operations over the text of image descriptions. In this chapter we implement additional models of retrieval that perform some of these operations, and we evaluate the result on a subset of the Phetch data set.

In these experiments, we hypothesize the following:

- Augmenting the representation with concepts from a knowledge base will improve performance on ontology-related errors
- Augmenting the representation with relations from untyped dependency analysis will improve performance on faulty-inference errors
- We can combine these methods and features in a way that has minimal impact on the non-error cases; i.e. correcting errors but not creating new ones

This work shows how general techniques like query expansion and graph-based semantic matching, which have been used in other ATI tasks with

success, can be applied and specialized for a new problem instance (description retrieval). The hypotheses listed above connect what we have learned from our error analysis to existing strategies from the field of applied textual inference. Next, we implement a retrieval pipeline to test these hypotheses and report the result.

7.2 Annotation with Ontology Elements

To validate the first hypothesis, we augment the representation of queries and indexed image-documents by annotating them with concepts from an ontological knowledge base (kb, or ontology). The goal is to make synonyms, hypernyms, and slot-filler properties available to the relevance estimation function that is already implemented in Indri. Figure 7.1 shows how this annotation results in more features in common for two descriptions of a key example image. Although two out of four content words from the query are covered by the index description, three additional matching features are found when semantic annotations are added. The term *sash* in the index description has been annotated with the unique role-filling association that it bears to the concept *pageant*, which captures an ontological similarity between the two terms that is not available from the words alone.

7.2.1 Procedure

Inference engine

Many steps in this process are performed in the space defined by our knowledge base. New concepts and relations must be defined, and operations over these kb elements must be implemented and made available to the retrieval process. We have implemented these components as modules for the Scone knowledge-base system¹, or *Scone*. They include new ontologies, inference routines, and APIs, written in Lisp, which make use of the marker-passing inference engine available in the core distribution of Scone (Fahlman, 2006). In the remainder of this chapter we use “SconeImage” to refer to the extended software suite that uses the Scone engine, but which is largely composed of original software. Additional detail on the knowledge acquisition process is given in Chapter 8.

¹<http://www.cs.cmu.edu/~sef/scone/>



Query Description: “**stadium** with pageant girls in the **foreground**”

Semantic Features: girl **female person pageant** ceremony

Index Description: “olympic **stadium**, women in **foreground** with sashes”

Semantic Features: woman **female person sash clothing pageant**

Figure 7.1: An example of knowledge-base annotation of a topic and index description. Terms shown in bold are features shared by both descriptions.

Data

As in Chapter 5, the data set used in this chapter is a subset of the Phetch section 5A. These images have each been annotated with at least five descriptions, allowing us to isolate three sets of topic labels: one for training, one for development, and one for testing. The remaining two descriptions are used as the index representation of the image-document. In addition, we have pruned from this data set all images that contain text in the image itself². Like the main sections of Phetch, this subset and the test, training, and development queries can be reproduced by applying freely distributed scripts to the full plain-text Phetch corpus³.

Attaching concepts to descriptions

The SconeImage kb we developed for this thesis is a lexical-semantic resource, where many of the concepts and relations are attached to English *names* that could trigger them⁴. To attach concepts from the ontology to an

²see Section 5.1.3 for additional detail

³Send email requests to atribble@cs.cmu.edu

⁴see Chapter 8 for additional detail

image description, we developed a SconeImage module that finds a greedy alignment between words in the description and concept names from the currently-loaded knowledge base. In the experiments reported here, no word sense disambiguation is performed. The set of concept names that cover the description string most completely are appended to the text of the description. In addition, each concept name is expanded according to a set of rules that select hypernyms, role-fillers, and sister terms from the ontology. The names of expansion concepts are also appended to the description text. This results in the type of annotation shown in 7.1.

To test the effect on retrieval, we re-index the annotated collection so that ontology concepts become available at retrieval time. Next, we annotate each query using the same knowledge base.

7.2.2 Results

The results are shown in Table 7.1. This table includes the effect of spelling correction, which makes very little contribution to performance on its own, but which boosts the performance of the semantic annotation process and contributes to the end result.

In our error analysis exercise, we identified retrieval failures in the training set that could be attributed to lack of ontological knowledge in the baseline system. Table 7.1 shows that the augmented system reduces these errors by 25%, a difference that is statistically significant⁵ with 95% confidence ($p = 0.05$). This result supports our hypothesis that knowledge base annotation could reduce ontology-related retrieval errors.

The improvement on ontology errors from the training data is an intuitive result, since these errors are the examples that were used to inform KB development. When we turn to the full training set, which has many more examples, we see a reduction in error of 17%. This result is marginal statistically but is still an encouraging finding in support of the hypothesis that correcting ontology errors leads to better overall performance.

The effect on ontology errors in the test set must be inferred, since calculating it directly would violate the experimental assumptions we make by isolating the test set from the development and training process (annotation of ontology errors requires the developers to compare the queries to index descriptions by hand).

⁵single-factor ANOVA

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)]		
	Test Set	Training Set	Training Set Ontology Examples
KW	0.8241	0.8172	0.2801
KW+SP+KB	0.8508 ($p = 0.1$)	0.8494 ($p = 0.06$)	0.4571 ($p = 0.05$)

KW=keywords, SP=spell-correction, KB=annotation with kb concept names

Table 7.1: Results of spelling correction and knowledge-base annotation. ANOVA significance is shown for improvement over the *keywords* baseline.

However, we see trends that indicate the effect may be there, based on performance on the test set overall. Table 7.1 shows that the improvement on the full test set is similar in magnitude to the improvement on the full training set. Error was reduced by 15% compared with the baseline. Even with marginal statistical significance of $p = 0.1$, this result shows some support for the conclusion that these improvements will generalize well to new users (i.e. unseen queries).

7.3 Graph Distance with Dependency Structures

To validate the second hypothesis, we add untyped dependency structures to our representation of the image-documents and queries. Dependencies were introduced in Chapter 3; they are syntactic connections between words that represent a kind of abbreviated phrase structure tree. An example image annotated with these structures is shown in Figure 5.6.

The syntactic structure can help us to differentiate descriptions that have matching *terms* from descriptions that have matching *structure*. The image shown in Figure 5.7 was retrieved by the baseline retrieval system. Although more words from the query appear in this alternative description, the dependency annotation reveals the mismatch between “black and white photo” and “black suit white shirt”.

7.3.1 Procedure

To apply dependency structures for improved retrieval, we use the list of documents returned by the baseline system as a candidate list, then perform additional processing to rerank the results. First, we annotate each query and result description from the baseline run with a dependency structure, storing the result in a semantic graph. Such a graph links vertices in the syntactic dependency tree (i.e. words from the description) with concepts from the knowledge base. As a result, we can calculate a similarity score between the query graph and the graph for any image in the candidate list, taking both semantic and syntactic similarity into account. This score is used to re-rank the results from keyword retrieval. An overview⁶ of the retrieval system that applies this process is shown in Figure 7.2.

Build the Dependency Graph

To build the dependency graph for an image description, we apply the Charniak constituent parser (Charniak, 2000) to the text and read the result into a data object that represents the description in `SconeImage`. Once the constituent parse is read in, a custom `SconeImage` module applies head-finding rules based on Magerman (1995) to derive dependencies, which are also attached to the image object. The structure of these objects in the knowledge base mirrors the logical structure of image-documents in the Phetch corpus, but with additional fields for parsed descriptions. A simplified visual representation of this data object is shown in Figure 7.3.

Extract Features for Reranking

As we have seen in Figures 5.6 and 5.7, features from the dependency graph can help to refine the way we compare queries to index descriptions. To apply these features to retrieval, we extract them from our `SconeImage` graphs and apply a function for combining them into a similarity score. This step is labeled as “Feature Extraction and Hand-Tuned Scoring” in Figure 7.2.

Graph-based features are computed over a pair of graphs, one derived from the current query description and one derived from a result in the candidate list (pointing to an indexed document). Features correspond to

⁶Simplified to show processing steps; the full system has additional modules for caching parses and graph features.

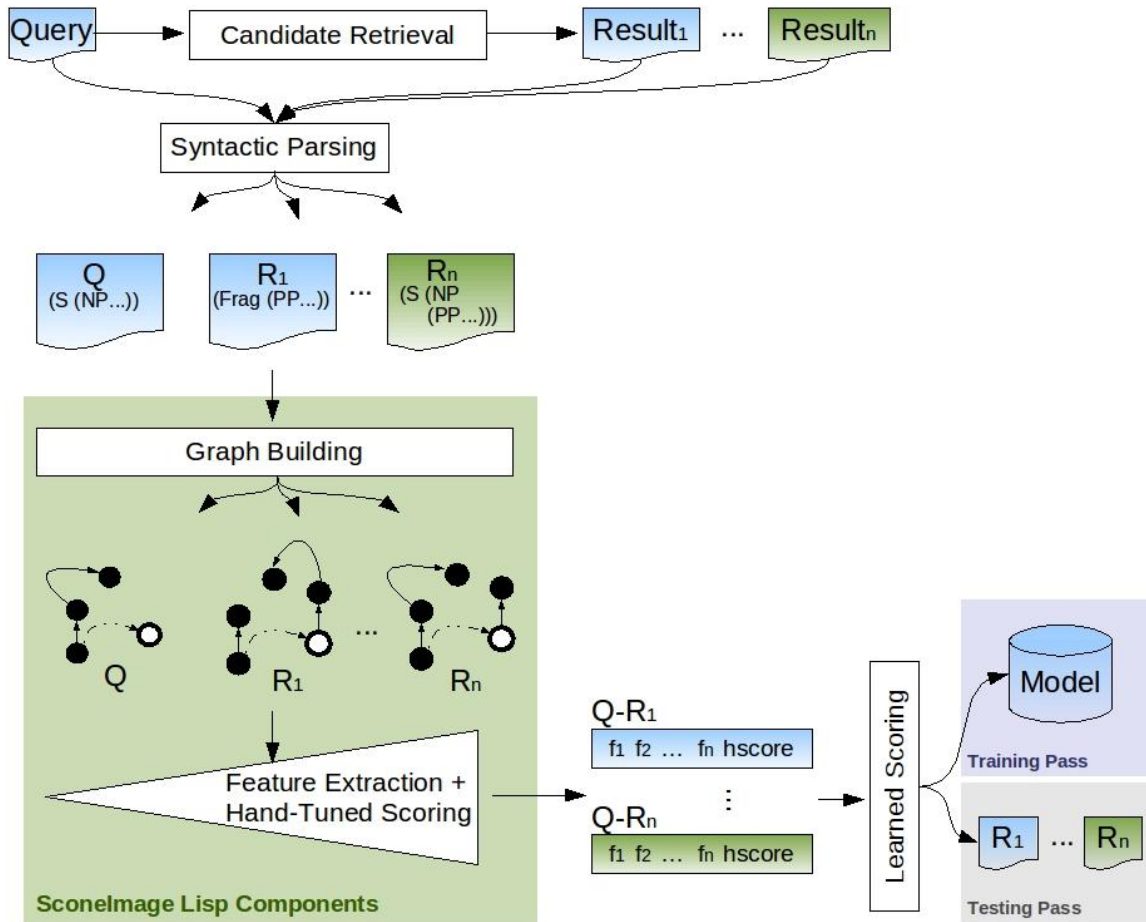


Figure 7.2: Retrieval process with graph-based reranking.

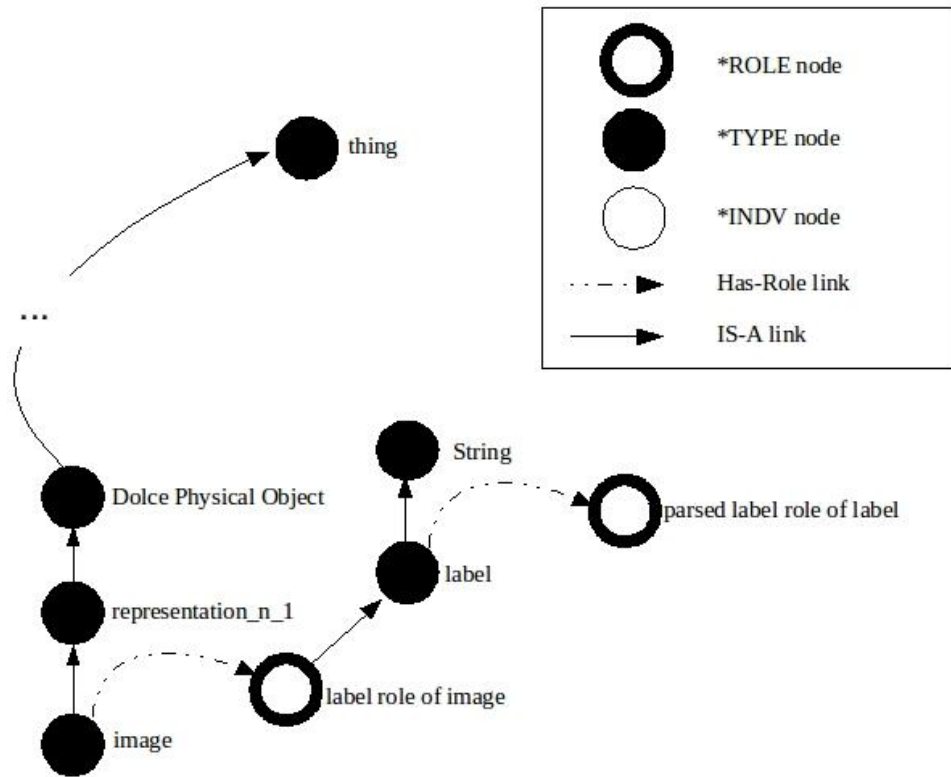


Figure 7.3: KB representation for image descriptions in SconeImage.

sub-structures that are shared by the two graphs. For example, each of the arcs shown in bold in Figure 5.6 is one positively-valued feature for the index and query descriptions shown in that figure.

The first step in calculating these features is to find an alignment between the query graph and the index-description graph. In general, these could be any source graph, G and target graph, G' . After alignment, we apply a series of tests that evaluate how closely the vertices and edges from G match their counterparts in G' . Every test generates one graph-based similarity feature. Vertex features are summarized in Table 7.2. These features are the building blocks for the similarity function.

Each vertex feature implies two corresponding edge features. For example, to calculate the string feature on a pair of edges (e, e') , we examine the vertices (v_1, v_2) of e and (v'_1, v'_2) of e' . If the value of $\text{string}(v_1, v'_1)$ is equal to 1, then the value of $\text{origin_string}(e_1, e_2)$ is 1. If the value of $\text{string}(v_2, v'_2)$ is equal to 1, then the value of $\text{terminus_string}(e_1, e_2)$ is equal to 1. An example of this calculation is shown in Figure 7.4.

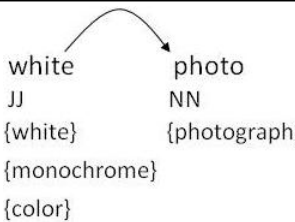
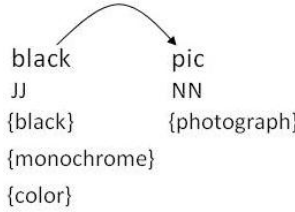
7.3.2 Reranking

The separation of feature extraction from similarity scoring is a design choice modeled on the alignment and scoring phases applied by Haghighi et al. (2005) to the problem of recognizing textual entailment.

After features have been computed according to their descriptions in Section 7.3.1, they can be combined manually or passed to a component that uses them to train a learned combination function. First, we identify the form of the similarity function, which includes several layers of free parameters. Next, we apply knowledge of the task to hand-tune the parameters, yielding an initial estimate of graph similarity. Finally, we pass this estimate along with the original similarity features into a machine learning architecture, which learns new values for the parameters, based on training data. The resulting trained similarity function is used to re-score elements from the candidate list. To evaluate, we re-order the candidate list based on this score and calculate the mean reciprocal rank for this new result file.

Feature	Definition
string =	$\begin{cases} 1 & \text{if } v_i \text{ and } v'_j \text{ have matching strings,} \\ & \text{as in ("man", "man")} \\ 0 & \text{otherwise} \end{cases}$
pos =	$\begin{cases} 1 & \text{if } v_i \text{ and } v'_j \text{ have matching parts of speech,} \\ & \text{as in ("man"+NN, "guy"+NN)} \\ 0 & \text{otherwise} \end{cases}$
syn =	$\begin{cases} 1 & \text{if } v_i \text{ and } v'_j \text{ have matching kb concepts,} \\ & \text{as in ("guy"+adult male, "man"+adult male)} \\ 0 & \text{otherwise} \end{cases}$
hyp =	$\begin{cases} 1 & \text{if } v_i \text{ is a hypernym of } v'_j, \\ & \text{as in ("someone"+person, "man"+adult male)} \\ 0 & \text{otherwise} \end{cases}$
role =	$\begin{cases} 1 & \text{if } v'_j \text{ is a role in } v_i, \text{ as in} \\ & \text{("bouquet"+flower arrangement, "flower"+flower)} \\ 0 & \text{otherwise} \end{cases}$
path =	$\begin{cases} 1 & \text{if a short path exists from } v_i \text{ to } v'_j \text{ in the kb,} \\ & \text{crossing roles and hyponyms, as in} \\ & \text{("bouquet"+flower arrangement,} \\ & \text{"petal"+part of flower)} \\ 0 & \text{otherwise} \end{cases}$

Table 7.2: Vertex features for graph-based description similarity.

Edge 1	 <pre> graph TD white[white] --> photo[photo] style white fill:none,stroke:none style photo fill:none,stroke:none </pre>
Edge 2	 <pre> graph TD black[black] --> pic[pic] style black fill:none,stroke:none style pic fill:none,stroke:none </pre>

Feature			Value
string	=	0,	since $\neg\text{string}(\text{"white"}, \text{"black"})$ and $\neg\text{string}(\text{"photo"}, \text{"pic"})$,
pos:	=	2,	since $\text{pos}(\text{"JJ"}, \text{"JJ"})$ and $\text{pos}(\text{"NN"}, \text{"NN"})$
syn:	=	1,	since $\neg\text{syn}(\{\text{white}\}, \{\text{black}\})$ and $\text{syn}(\{\text{photograph}\}, \{\text{photograph}\})$
hyp:	=	0,	since $\neg\text{hyp}(\{\text{white}\}, \{\text{black}\})$ and $\neg\text{hyp}(\{\text{photograph}\}, \{\text{photograph}\})$
role:	=	0,	since $\neg\text{role}(\{\text{white}\}, \{\text{black}\})$ and $\neg\text{role}(\{\text{photograph}\}, \{\text{photograph}\})$
path:	=	2,	since $\text{path}(\{\text{white}\} \rightarrow \{\text{monochrome}\} \rightarrow \{\text{black}\})$ and $\text{path}(\{\text{photograph}\} \rightarrow \{\text{photograph}\})$

Figure 7.4: Similarity features for a sample pair of edges.

A Function for Description Graph Similarity

Because our feature functions are directed, the similarity between two graphs G and G' may be asymmetric⁷. As a result we calculate the total similarity between a query description Q and a candidate document C as a combination of the graph-based similarity scores $\text{sim}(Q, C)$ and $\text{sim}(C, Q)$, shown in Equation 7.1.

$$\text{totalScore}(Q, I) = \alpha_1(\text{sim}(G, G')) + \alpha_2(\text{sim}(G', G)) \quad (7.1)$$

where G is the set of graphs $g_1 \cdots g_i$ associated with descriptions of Q (usually a single graph), and G' is the set of graphs $g'_1 \cdots g'_j$ associated with descriptions of I (usually 2-3 graphs). When more than one description-graph is present in either of these sets, we must combine feature scores across all descriptions, yielding a similarity score of the form shown below:

$$\text{sim}(G, G') = \frac{1}{N} \sum_{i=1}^{|G|} \sum_{j=1}^{|G'|} \text{graphSim}(g_i, g'_j) \quad (7.2)$$

where $N = |G| \times |G'|$. The graphSim function calculates semantic coverage of the graph g by the graph g' , based on coverage of vertices and coverage of edges:

$$\text{graphSim}(G, G') = \beta_1 \text{vertexSim}(g_i, g'_j) + \beta_2 \text{edgeSim}(g_i, g'_j) \quad (7.3)$$

where vertexSim and edgeSim are weighted sums of the vertex similarity features (VF) and edge similarity features (EF) described above:

$$\text{vertexSim}(g, g') = \sum_{n=1}^{|VF|} \lambda_n \times v f_n(g, g') \quad (7.4)$$

$$\text{edgeSim}(g, g') = \sum_{m=1}^{|EF|} \Lambda_m \times e f_m(g, g') \quad (7.5)$$

⁷In ad-hoc retrieval, where the user types “pets” to find instances of dogs, cats, and birds, we might leverage this asymmetry differently. In the description-retrieval task presented here, queries are designed to match a particular instance (i.e. one image). As a result there is less reason to assume that the query will have equal or greater granularity than the indexed descriptions. As a result we compensate for the asymmetry by calculating semantic distance in both directions.

A Hand-Tuned Similarity Function

To set the parameters $\lambda_1 \cdots \lambda_n$ and $\Lambda_1 \cdots \Lambda_m$ we rank the vertex and edge features according to how closely they represent matching kb elements. Synonym match, for example, implies a nearer semantic similarity than a hypernym match. The ranks are used as a guide to assign weights to each feature by hand. This type of knowledge-based parameter manipulation has been applied successfully in a variety of systems for textual inference, where knowledge of the task is considered critical for success. Examples include Haghighi et al. (2005), when tuning the relative weights of word-similarity features in a system for recognizing textual entailment, and Varelas et al. (2005), when tuning word-similarity features in a system for semantic retrieval of images and documents from the web. Although not guaranteed to find optimal weights, those systems showed that this approach can yield higher results than random or uniform weighting.

After some initial experiments to tune the parameters $\alpha_1, \alpha_2, \beta_1, \beta_2$ we found little support for weighing forward- and backward- distances unevenly, or for weighing vertex-matching more heavily than edge-matching. As a result we set each of these parameters to 1 when calculating totalScore.

In the results reported in Section 7.3.2, rather than reranking on totalScore alone, we supply the hand-tuned score to a learning mechanism, along with the feature list from which the score was derived. This mechanism is described in the next section.

A Learned Similarity Function

The hierarchical parameters described in Section 7.3.2 reflect the intuitive structure of the task, but are difficult to tune effectively by hand. As an alternative to hand-tuning, we can format SconeImage similarity features as input to a machine learning algorithm. By supplying the hand-tuned score as an additional feature, we still allow the intuitive estimate to affect the final outcome.

In addition to the hand-tuned score, the learned similarity function uses complex features where a basic feature is composed with the lexical, syntactic, or semantic context that triggered it (e.g. “pos-match and pos is NN”). Again, the hand-tuned model influences the solution by defining the structure of basic features. However the total number of complex features is too large to be weighted by hand. The learning framework allows us to include these

features in a principled way.

The results reported below were achieved by training an off-the-shelf perceptron classifier based on Collins (2002). The classifier distinguishes graph pairs that describe the same image from graph pairs that do not. To train the classifier, we generate similarity features with SconeImage for every query-candidate pair in the baseline retrieval output. When more than one description is available for an indexed image in the candidate list, we generate the features for every such description independently.

For every query in the training set, at most one candidate contains descriptions of the same image as the query. These descriptions are positive training examples. The remaining descriptions are negative training examples. The classifier learns a set of weights $\lambda_1 \cdots \lambda_N$ for a linear combination over all of the features $f_1 \cdots f_N$ that we calculate over graph pairs:

$$\text{learnedGraphSim}(g, g') = \sum_{n=1}^{|F|} \lambda_n \times f_n(g, g') \quad (7.6)$$

where F is the set of all similarity features, and λ_n is the weight of feature f_n .

At test time, we use a fresh set of queries that were never seen by the classifier during training (the test_query descriptions from Phetch Section 5A). The test queries are run through the baseline retrieval system or through the kb-augmented system described in Section 7.2. The results from this run are sent to SconeImage for graph-feature extraction and scoring using the hand-tuned similarity function. The output of the test run is one verdict and similarity measurement for every description of every candidate in the result list.

Before reranking, we combine the scores across all descriptions of a single candidate by simply taking the maximum score, resulting in a modified version of the sim function, shown in Equation 7.7. This equation also makes the simplifying assumption that the query has only one associated description, represented by the instance g rather than the set G .

$$\text{maxSim}(g, G') = \arg \max_{j \in (1 \cdots |G'|)} \text{learnedGraphSim}(g, g'_j) \quad (7.7)$$

To rerank, all candidates for a single query are ordered according to their maxSim score. The reordered list is formatted to emulate the TREC result

file format, and can be scored using the `trec_eval` program distributed by NIST⁸.

7.3.3 Results

Results from this experiment are shown in Table 7.3.

As with ontology-related errors, we annotated inference-related errors in the analysis described in Chapter 6. In comparison with the baseline, the system that applied reranking based on dependency structures reduced the error on examples annotated as inference errors by 24%. This represents a large absolute improvement on inference errors as a result of adding dependency information. Because the number of such examples is small (50), this number is only marginally significant ($p = 0.07$); however, results on the full training and test sets confirm that this reduction contributes to better performance overall, underscoring the importance of these gains.

On the full training set, including many more examples than were seen during KB development, error was reduced by 22% over the baseline, a result that is statistically significant ($p = 0.02$). This improvement carried over to the unseen test queries, where error was reduced by over 18%. This result has marginal but very encouraging significance ($p = 0.05$).

Taken together, the results on ontology-related errors and inference-related errors exhibit strong trends in support of the third hypothesis, that semantic errors can be reduced without contributing to new errors in other parts of the data set. Tables 7.1 and 7.3 show that overall performance increased when these specialized types of retrieval failure were addressed. In future work we could strengthen our confidence in these conclusions by running the full experimental pipeline (KB development, training, and testing) on larger sections of the data.

7.4 Analysis

7.4.1 Effect of Query Formulation

The Indri retrieval engine supports a rich structured query representation. We performed an additional set of experiments to compare a naive encoding of the descriptions with two types of structured query that distinguish

⁸http://trec.nist.gov/trec_eval/

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)		
	Test Set	Training Set	Training Set Inference Examples
KW	0.8241	0.8172	0.2454
KW+SP+KB	0.8508	0.8494	0.4053
KW+SP +KB+GPH	0.8567 ($p = 0.05$)	0.8575 ($p = 0.02$)	0.4251 ($p = 0.07$)

KW=keywords, SP=spell-correction, KB=annotation with kb concept names, GPH=graph features for reranking

Table 7.3: Results of re-ranking. ANOVA significance is shown for the improvement over the *keywords* baseline.

semantic annotations from the original text, allowing Indri to model them separately. An example of these encodings is shown in Figure 7.5.

Indri query syntax is described in detail in the Indri Wiki⁹. Briefly, the field-restriction syntax “*term.field*” matches *term* only if it appears in the *field* section of an sgml index document, and scores the match using a document-level language model. The field-model query “*term.(field)*” matches the term *field* in any context, but scores it using a language model trained only on text in *field* sections of indexed documents.

Retrieval results using each of these encodings are shown in Table 7.4. Both styles of query seem to bring the results down slightly, compared with naively-annotated queries.

7.4.2 Effect of Semantic Graph Features

The results in Table 7.3 show that reranking based on features from a syntactic-semantic graph can yield better retrieval performance. However, before performing reranking, we have an opportunity to test a variety of feature configurations. For example, we may test the contribution of the syntactic graph features (such as matching vertices based on part-of-speech)

⁹<http://www.lemurproject.org/lemur/IndriQueryLanguage.php>

Naive Query	<code>#combine(pageant girls girl female person pageant ceremony)</code>
Field-Restricted	<code>#combine(pageant.description girls.description girl.sem female.sem person.sem pageant.sem ceremony.sem)</code>
Field-Modeled	<code>#combine(pageant.(description) girls.(description) girl.(sem) female.(sem) person.(sem) pageant.(sem) ceremony.(sem))</code>

Figure 7.5: Samples of query encodings.

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)	
	Test Set	Training Set
Naive Queries	0.8508	0.8494
Field Restriction	0.8508	0.8416
Field Models	0.8465	0.8227

Table 7.4: Results of retrieval with naive vs. structured queries.

Ranking Features					MRR (Test)	MRR (Train)
Kwds+ spell	Synt- graph	KB annot.	Sem- graph	gphSim score		
✓					0.8295	0.8216
✓	✓				0.8404	0.8383
✓		✓			0.8508	0.8494
✓	✓	✓			0.8535	0.8559
✓	✓	✓	✓		0.8531	0.8584
✓	✓	✓		✓	0.8528	0.8543
✓	✓	✓	✓	✓	0.8567 ($p = 0.05$)	0.8575 ($p = 0.02$)

Kwds=keywords, spell=spell-correction, Synt-graph=syntactic graph features for reranking, KB-annot=query expansion with KB concepts (before reranking), Sem-graph=semantic graph features for reranking, graphSim score=value of the graph-Sim hand-tuned distance function

Table 7.5: Results of retrieval using combinations of syntactic and semantic features for graph-based reranking. ANOVA significance is shown for improvement over the *keywords* baseline.

independently from semantic features (such as matching vertices based on hypernym association). In addition, we can assess the contribution of the hand-weighted graph-matching score by testing results with and without this score as a feature. Retrieval results using each of these feature configurations are shown in Table 7.5.

7.4.3 Effect of Text within Images

In these experiments we focus on images without textual content, making the assumption that images without text are the most interesting from a retrieval perspective, and that their descriptions are the most interesting from a textual inference perspective. These assumptions are supported by experiments comparing performance of the baseline and augmented systems across both types of images: images with and without textual content.

Table 7.6 addresses the hypothesis that applied textual inference techniques can yield an improvement over all images, and that the contribution is greatest on the images where the baseline system performs most poorly:

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)	
	Test Set	Training Set
keywords - All Images	0.9304	0.9259
keywords - No Text	0.8241	0.8172
augmented - All Images	0.9397 ($p = 0.09$)	0.9387 ($p = 0.02$)
augmented - No Text	0.8567 ($p = 0.05$)	0.8575 ($p = 0.02$)

Table 7.6: Results of retrieval on all images from the 5A data set, vs. the subset without textual content.

images without textual content. This improvement is marginally statistically significant ($p = 0.09$), but it represents a strong trend that gives us additional confidence in our positive results.

7.5 Conclusions

The experiments described in this chapter support the hypotheses that we formed as a result of detailed error analysis in Chapter 6. Specifically, textual inference techniques can lead to improved retrieval performance, in particular on the most interesting types of images: images with more visual content and less text than web buttons or logos, and images whose descriptions can only be interpreted with the application of ontological knowledge and inferential knowledge. In addition, these improvements can complement the strengths of a strong bag-of-words baseline to achieve better overall performance on all image types.

In these experiments we address two of the error types identified in Chapter 6. It would be an interesting extension of this work to test specific techniques that could reduce the remaining error types. For example, co-reference resolution might be beneficial in reducing errors associated with quantification mismatch, in particular as it relates to the number of people in an image description. Techniques for contradiction detection have been developed and tested for other textual inference problems, including recognizing textual entailment and question answering. These techniques could also apply to

retrieval errors caused by real or perceived contradictions between a query and an index description.

Chapter 8

Knowledge Sources for Labeled Image Retrieval

In Chapter 7 we presented results for a knowledge-based approach to retrieving descriptive image labels. This approach depends on a knowledge base that supplies the vocabulary of semantic concepts, which are used for annotation, and the relationships between concepts, which are used to calculate description similarity. In this chapter we discuss how such a knowledge base should be constructed, and we compare two different knowledge bases in terms of their effect on retrieval performance.

8.1 Scone Knowledge Base System

The purpose of the kb is to make background knowledge available to the retrieval system. To achieve this, it must be supported by a software framework and Application Programming Interfaces (APIs). In this thesis we have selected the Scone Knowledge Base System¹ (Scone) as our kb framework. The system includes the Scone engine, implemented in Common Lisp, and knowledge bases. Knowledge bases are text files that encode statements in the Scone representation language, which can be loaded into a running Scone engine process.

The Scone engine supports adding, searching, and evaluating logical statements based on marker-passing inference (Fahlman, 2006). To support the

¹<http://www.cs.cmu.edu/~sef/scone/>

experiments described in Chapter 7, we have implemented additional Common Lisp components that extend Scone engine functions. These include new ontologies, inference routines, and APIs that use Scone to annotate text and measure the semantic distance between concepts. In the remainder of this chapter we use “SconeImage” to refer to this extended software suite.

The organization of Scone and SconeImage is shown in Figure 8.1. Each component listed in Figure 8.1 is implemented as a Lisp file. Files ending in “-kb” are knowledge base files. Non-kb files contain subroutines that operate on Scone structures but do not contain declarative knowledge. Code that is original to the Scone Knowledge Base System is encapsulated in the engine module. SconeImage code includes knowledge at the world, task, and domain level, along with task-specific subroutines that read in parsed text and calculate the distance between semantic graphs using the models described in Section 7.3.2.

Knowledge bases in Scone use a frame-semantics formalism to represent a network of concepts, or Scone *elements*. Scone supports taxonomic relationships, like “a *flower* is a *plant*” as well as role-filling relationships, like “a *flower* has *scent*”. New non-taxonomic relations can be defined as well, with instances of such relations being encoded as statements, like “a *bird* *flies*”. Exceptions can be marked to handle relations that apply to most, but not all, instances of a class, as in “a *penguin* is a *bird* that *does not fly*”. Scone also includes a lexical lookup function that allows multiple strings to be attached to any element. This feature supports the annotation process described in Chapter 7.

Given such a knowledge base as input, routines defined in the Scone engine calculate the answer to queries like “Is a *rose* a *flower*?”. Extensions in SconeImage make higher-level calculations that depend on these answers, like “what is the relationship between *bouquet* and *rose*?”. In all cases, the answers returned by these calculations depend on the knowledge bases that are currently loaded. In the following sections we explore how these knowledge bases can be populated.

8.2 Retrieval with WordNet

WordNet (Fellbaum, 1998) is a free broad-coverage resource for lexical-semantic knowledge that has been applied in many textual inference systems. It is a practical starting point for testing knowledge-based algorithms, in particular

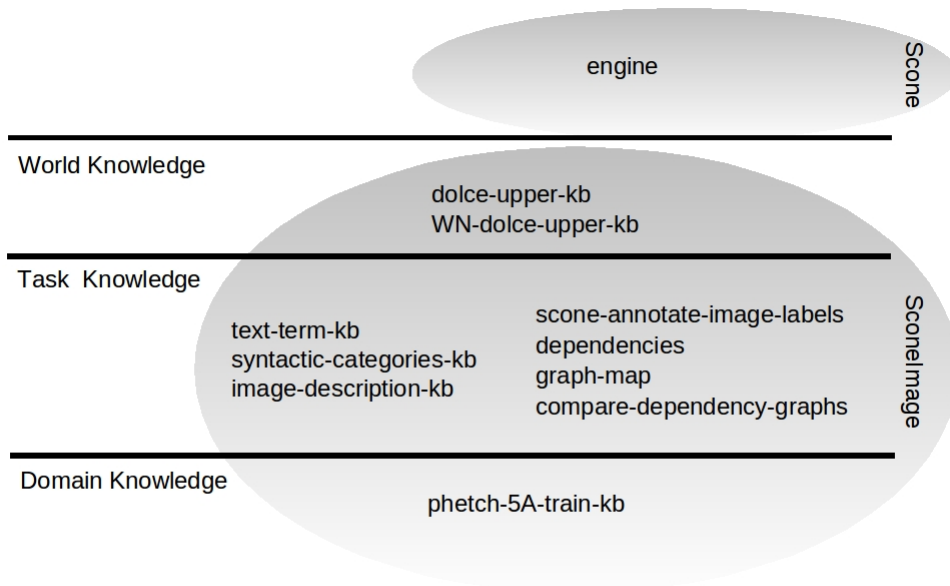


Figure 8.1: The file organization of SconeImage knowledge bases and reasoning modules.

when the knowledge acquisition is outside the problem scope.

The structure of WordNet is a semantic network system where nodes in the network capture *synsets*, or groups of cognitive synonyms with matching parts of speech². The semantic relations available in WordNet comprise a subset of the structures available in Scone. They include taxonomic relations among synsets (hyper- and hyponyms), part-of relations (meronyms) and member-of relations (holonym), as well as more strictly lexical relations such as verb-group membership and derivationally-related forms.

The current version of WordNet is Wordnet 3.0, with over 15,000 synsets covering more than 200,000 word-synset pairs.

8.2.1 Procedure

To apply WordNet knowledge in SconeImage, we replicate the experiment described in Section 7.2 using WordNet as the resource for annotation. As in Chapter 7, we apply a greedy left-to-right annotation strategy. For each token in an image description, we search WordNet for the first synset associated with the token, preferring nominal interpretations. No word sense disambiguation is performed. A list of expansion synsets is found by searching WordNet for hypernyms of the target synset. The target and expansion synsets are added to an unordered set of annotations for the entire description, from which duplicates are removed. An example description with its WordNet annotations is shown in Figure 8.2.

8.2.2 Results

When added to the retrieval pipeline described in Section 7.2, these annotations result in a small improvement over the non-annotated baseline and the spell-corrected baseline. Although statistical significance is low ($p \geq .4$ using single-factor ANOVA analysis), this improvement is consistent across the training and test queries from the data set used in Chapter 7³. The effect on Mean Reciprocal Rank is shown in Table 8.1.

²<http://wordnet.princeton.edu/>

³Phetch 5A section



Image Description: “stadium with pageant girls in the foreground”

WordNet Features: stadium structure pageant ceremony girl woman foreground view

Figure 8.2: WordNet annotation of an image description.

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)	
	Test Set	Training Set
KW	0.8241	0.8172
KW+SP	0.8295	0.8216
KW+SP+WN	0.8349 ($p = 0.5$)	0.8306 ($p = 0.4$)

KW=keywords, SP=spell-correction, WN=annotation with WordNet concept names

Table 8.1: Effect of annotation with wordnet synsets on Phetch section 5A, non-textual images. ANOVA significance is shown for improvement over the *keywords* baseline.

8.3 Improved Knowledge Base Structure

The results in Table 8.1 support the hypothesis that knowledge may help on this task, even at a very low development cost. The WordNet annotation process in Section 8.2 is entirely automatic and reproducible for new data sets. However the analysis in Chapter 6 indicates that some classes of error can only be corrected by a system that applies reasoning and inference over its knowledge base representations. The ad-hoc network structure of WordNet was not intended to support this type of reasoning.

We hypothesize that a task-specific knowledge base, structured with the challenges from Chapter 6 in mind, can improve performance even more. The intuition that kb structure, particularly at the upper levels of the ontology, plays an important role in its usability and effectiveness is supported by related work on textual inference and knowledge engineering. Fan et al. (2003), for example, describe the effect of ablating layers of the kb in a system for resolving noun-noun compounds. Their finding was that concepts from the upper levels of the ontology were critical to performance on that task, and that they had a larger impact on performance than concepts near the frontier.

8.3.1 Upper Levels: WordNet + DOLCE

The upper levels of WordNet were examined from an ontology-engineering perspective in Gangemi et al. (2003). The authors identify classes of structural inconsistency in WordNet that make it difficult to interpret the network connections as logical relations. These include conflation of *subsumption* and *instantiation* relations, confusing object-level (taxonomy of objects) and met-level (taxonomy of properties that objects may take), and heterogeneous levels of generality, among others. To correct these errors while still leveraging the broad coverage of the WordNet database, Gangemi et al. introduce the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), an open-source ontology published through the WonderWeb project⁴.

We apply these findings to SconeImage using a knowledge-development strategy similar to the work described in Fan et al. (2003). In that work, the authors select an existing upper-level ontology and re-connect a subset of WordNet concepts to that ontology, while also adding several task-specific

⁴<http://wonderweb.semanticweb.org>

concepts. In the SconeImage kb, we use the DOLCE upper ontology (Masolo et al., 2003) for the top levels, and then apply a mapping from DOLCE to WordNet following Gangemi et al. (2003), with some task-specific modifications.

An abbreviated description of the resulting kb is shown in Figure 8.3. The Scone kb files are provided in Appendices B and C.

8.3.2 Acquiring Knowledge from Training Data

After the new upper-level SconeImage kb has been constructed, we perform a round of knowledge acquisition based on the training queries from the Phetch 5A data set. The baseline retrieval run on this data set resulted in 50 retrieval failures, cases where no relevant image was found in the result list. These failures were classified according to the procedure described in Chapter 6. To expand the knowledge base, a developer examined each of the retrieval failures annotated as *Ontology* or *Faulty Inference* errors. The training query and collection document were compared, and terms were added to the ontology to compensate for the error.

An example is shown in Figure 8.4. The training query uses the term “girls” where the index description uses “women”. To compensate, the developer adds new elements *woman_n.1* and *girl_n.1* to the SconeImage ontology, connecting them to the common ancestor *female_n.1*. Naming conventions for these elements reflect the fact that they must connect to the WordNet concepts on the frontier of the upper-level ontology described in Section 8.3.1. As a result, the new elements are selected to correspond with an appropriate term from WordNet, but their arrangement in the ontology may differ significantly from their placement in WordNet.

This development strategy is consistent with the analysis that WordNet is most useful for associating strings with lexical-semantic concepts, while the arrangement of concepts into logical structures can be improved through connection to an ontology like DOLCE and an inference platform like Scone. Figures 8.5 and 8.6 provide an example of this effect. While the terms *man*, *boy*, *woman*, and *child* all appear in the WordNet hierarchy, the structure connecting “man” and “boy” is different from the structure connecting “woman” and “girl”. This type of inconsistency means that an inference rule of the form *if the query mentions a subtype of person, expand with sister terms* would correct one of these lexical mismatches but not the other. We prefer an ontology structure that uses parallel structures for conceptually parallel

abstract (*regions and quality spaces ...*)

spatio-temporal-particular

- **endurant**
 - non-physical **endurant**
 - non-physical object
 - mental-object
 - social-object
 - agentive-social-object (*people, institutions ...*)
 - **person_n_1**
 - non-agentive-social-object
 - collection (*organized and non-organized collections ...*)
 - concept (*parameters, roles ...*)
 - description (*social descriptions, theories, narratives ...*)
 - information-object (*linguistic, diagrammatic, formal objects ...*)
 - **language_unit_n_1, signal_n_1, message_n_1**
 - situation (*systems and plans ...*)
 - figure (*agentive [people], non-agentive [places] ...*)
 - **destiny_n_1, organization_n_2, vital_principle_n_1**
- physical **endurant**
 - **entity_something_n_1**
 - amount-of-matter
 - **mass_n_5, substance_n_1, fluid_n_1, atmosphere_n_1**
 - physical-object
 - **object_n_1, artifact_n_1, natural_object_n_1, representation_n_1**
 - agentive-physical-object
 - **organism_n_1**
 - rational-physical-object
 - agent
 - non-agentive-physical-object (*physical places, chemical objects ...*)
 - feature (*physical features like sides, surfaces ...*)
 - **back_n_3, front_n_3, top_n_4, surface_n_1, centerline_n_1**
- arbitrary-sum
- agent
- perdurant (*events, states, processess ...*)
- quality
 - physical-quality (*locations, shapes, colors ...*)
 - **visual_property_n_1, color_n_1, texture_n_2, shape_n_1**
 - temporal-quality (*times and durations ...*)
 - abstract-quality
- physical-realization (*depiction, motion, voicing ...*)
 - **visual_communication_n_1, print_n_1, writing_n_4**

Figure 8.3: Top level of the SconeImage Ontology, adapted from DOLCE + WordNet. (Gangemi et al., 2003)



Training	Query	“stadium with pageant girls in the foreground”
	Description:	
Index	Description:	“olympic stadium, women in foreground with sashes”

New	Ontology	woman_n_1 <i>woman</i> , female_n_1, girl_n_1 <i>girl</i>
Entries:		

Figure 8.4: Example of knowledge acquisition from training data. KB elements for “woman” “girl” and “female” are added, along with English strings that trigger “woman” and “girl”. Stemming at retrieval time compensates for the singular-plural variation.

relationships among concepts.

8.4 Retrieval with SconeImage Ontologies

8.4.1 Procedure

This knowledge development process results in the SconeImage knowledge architecture shown in Figure 8.1. This is the architecture used for experiments in Chapter 7, and the procedure for applying it is similar to the WordNet case. A SconeImage module implemented in Lisp performs a single-pass search over the words in an image description for elements in the SconeImage ontologies that are triggered by each word. All elements are added without performing word sense disambiguation. The resulting list of element identifiers⁵ is concatenated with the original description to form the annotated query or collection description. Indexing and retrieval are performed using Indri under the same model described in Section 7.2.

⁵element names are mapped to a list of abbreviated integer ids of the form *scone1*, *scone2*, .. *sconeN*.

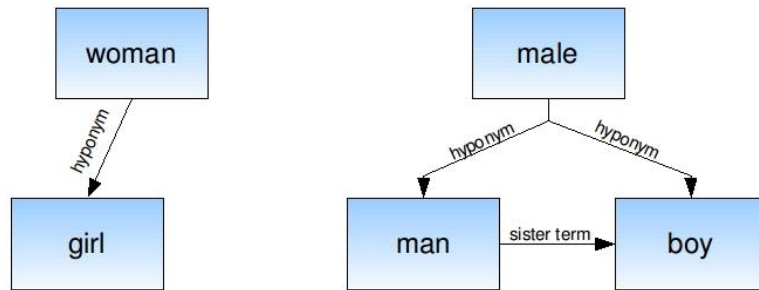


Figure 8.5: WordNet type hierarchy for 'woman' and 'man'.

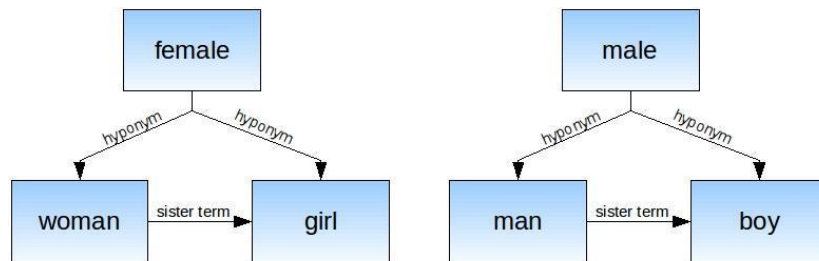


Figure 8.6: SconeImage type hierarchy for 'woman' and 'man'.

Retrieval Settings	Retrieval Performance (Mean Reciprocal Rank)	
	Test Set	Training Set
KW	0.8241	0.8172
KW+SP	0.8295	0.8216
KW+SP+WN	0.8349 ($p = 0.5$)	0.8306 ($p = 0.4$)
KW+SP+KB	0.8508 ($p = 0.1$)	0.8494 ($p = 0.06$)

KW=keywords, SP=spell-correction, KB=annotation with kb concept names, WN=annotation with WordNet concept names

Table 8.2: Comparison of knowledge resources for annotated description retrieval. ANOVA significance is shown for improvement over the *keywords* baseline.

8.4.2 Results

The results of this experiment are shown in Table 8.2. The results from Table 8.1 are repeated in this table for comparison. Under the same experimental conditions, the SconeImage ontology results in better performance than the WordNet ontology. This improvement is consistent across training queries, on which kb development was performed, and test queries, which were not seen by the developer.

The developer did observe examples from the indexed collection at kb development time. As a result this experiment does not establish portability of the kb across a new collection of images. However the consistency of improvements from training to test queries does support the hypothesis that knowledge development is portable across users of the image retrieval system.

The statistical significance of the result using Scone is marginal ($p = 0.1$), but still supportive of the improvement trend that we see across the training and test sets. This trend is favorable toward the analysis that Once an ontology for a new collection has been developed, any number of unseen users should expect improved results when they pose queries to the knowledge-augmented system.

8.5 Conclusions

In this chapter we described two alternatives for knowledge base development. First, we used the open-source semantic lexicon WordNet to reproduce the semantic annotation experiments from 7. The result was an improvement over the baseline results given in Chapter 7 that was consistent across training and test queries, but that did not have strong statistical significance.

Next, we introduced ontology engineering strategies that have appeared in related literature for knowledge-based textual inference tasks. We developed a second knowledge base with an improved upper-level structure based on the Dolce ontology. To this structure we added hand-engineered concepts and relations that were based on annotated errors in the training queries from the Phetch 5A data set. This knowledge base yielded improvements over the WordNet-based system and over the baseline, which is an encouraging result although the statistical significance is marginal ($p \leq 0.1$).

These results give some support to the hypothesis that knowledge helps in this application, and that better knowledge helps even more. As the SconeImage knowledge base continues to grow in breadth, while retaining its well-structured upper-level ontology, we would expect to see some additional improvement in retrieval performance, within the limits established by the corrective what-if analyses of Chapter 6.

Chapter 9

Conclusions

9.1 Experiments and Findings

The main claims of this work are as follows:

1. Detailed error analysis of a strong baseline system for description retrieval shows that human language users rely on specific classes of world knowledge and inferential operations to encode features of image descriptions. This analysis reveals the nature of description retrieval as an instance of textual inference.
2. We can improve the system's ability to decode and use these features by adding semantic, syntactic, and hybrid features to our representation of descriptions, and by using these features in new functions for estimating description similarity.
3. These corrections improve performance not only on our errors of interest, but on overall retrieval quality.

In the process of evaluating these claims, we have presented a series of experiments. In Chapter 4, we counted the most frequently-occurring syntactic patterns in the Phetch corpus and in a corpus extracted from the website Flickr.com. We found that descriptive text of the type found in Phetch is also common in Flickr, and that syntactic, semantic, and rhetorical structures that appear in Phetch are also relevant for Flickr. Curated, annotated data sets like Phetch are critical for allowing diverse research groups to compare results across comparable experimental conditions. The findings presented in

Chapter 4 are a positive indication that such curated data sets, in particular Phetch, accurately represent the problem of description retrieval in general. To our knowledge, this experiment and its result are novel.

In Chapter 6, we classified retrieval failures that occurred in a baseline retrieval run, according to the cause of the error. We calculated the frequency of each class of error, finding that over 90% of retrieval failures were attributed to one or more classes from this set. The depth of analysis and vocabulary of error classes are novel contributions that increase our understanding of the problem of description retrieval.

In Chapter 7, we implemented and evaluated new retrieval pipelines that were aimed at reducing the errors identified in Chapter 6, specifically ontology-related errors and errors attributed to faulty inference. We hypothesized that annotation with concepts from a well-structured knowledge base would improve performance on queries that triggered baseline-system errors due to ontological mismatch. Experimental results on the training set revealed a 25% reduction in error for these queries. Overall reduction in error was over 17% on the training set. On test queries, overall error was reduced by over 15%, compared with the baseline. Although the improvement has marginal statistical significance ($p = 0.1$), it indicates that this approach shows promise for impacting not only interesting test cases, but overall retrieval performance.

We further hypothesized that we could reduce faulty-inference errors by augmenting the retrieval system with features from a hybrid syntactic-semantic graph representation of image descriptions. We presented a function that combines these features into a similarity score for a query description and an indexed description, along with a machine-learning architecture for tuning the free parameters of this function. By applying this function to rerank the results from annotated retrieval (described above), we reduced the error on training queries classified as inference-dependent by 24%. This contributed to an overall reduction of error of 22% on training queries, compared with the baseline. On test queries, we see a similar reduction of over 18% overall, which is statistically significant¹ with 95% confidence ($p = 0.05$). Follow-on experiments in Chapter 7 analyzed the effect of query formulation, contribution of syntactic versus semantic graph features, and evaluation on larger data sets.

¹single-factor ANOVA analysis

In Chapter 8, we compared the retrieval benefit that we achieved with the task-specific knowledge base to the results of annotation with all of WordNet, a lexical-semantic resource that has much broader coverage, but less reliable support for inference. We found that annotation with WordNet yielded performance above the baseline and below the level achieved with the custom KB. This result was replicated on a training set of queries, for which the custom knowledge base had been hand-engineered, as well as on a test set of queries that were never seen by the author of the knowledge base. As with the results from Chapter 7, the trend we observe when moving from training to testing queries gives us confidence that the advantage we gain from the custom KB does not depend on the lexical content of the descriptions, but rather on the quality of the knowledge base and the appropriate encoding of semantic relations that are relevant to this task.

9.2 Contributions

9.2.1 Summary of Contributions

This work establishes the result that structure of the knowledge resources affects results in “knowledge-based” ATI systems. Related work on knowledge-based approaches to applied textual inference tasks often rely WordNet or similar general-purpose resources that are essentially lexical in nature. While the value of these resources is unquestionable, they were not developed to support reasoning. Our error analysis supports the claim that humans do apply background knowledge for reasoning about matching features in multiple descriptions of a single image. In addition, our experimental comparison in Chapter 8 shows that a well-structured knowledge resource that applies ontology engineering principles at the highest levels can lead to better performance. This approach has costs as well as benefits, but our work provides evidence for the claim that *when you have good knowledge, it helps*.

This work establishes benchmark results for retrieving labeled object descriptions. The experiments described in Section 9.1 establish a battery of results that can be used as baselines for subsequent work on this topic.

This work establishes a vocabulary for sources of error in description retrieval systems and established the frequency of those

errors in a sample corpus. Information retrieval has become an intensely empirical field, with detailed analysis techniques to explain the *amount* of error in a given experiment. However to our knowledge, our work is rare in its analysis of *why* errors occur. As we have shown, this level of analysis leads directly to system improvements that address interesting, compelling, and significant retrieval errors.

This work produced reproducible experimental sections in an evaluation corpus for applied textual inference. Shared evaluations have become a crucial tool for moving the state of the art forward in language technologies. We have performed analysis and refinement of the raw data in the Phetch corpus that makes appropriate for use in such a shared evaluation.

In particular, we have identified subsets for training and evaluation that allow researchers from multiple sites to compare their retrieval results in experimentally clean conditions (by testing on queries that have been isolated from the development and training queries). We have authored tools for replicating these evaluation subsets and provide those tools without restriction upon request. We also established experimentally the structural similarities between the Phetch corpus, which was collected in controlled conditions, and uncurated image collections. Finally, we performed a principled comparison between the Phetch corpus and existing corpora for image retrieval.

9.2.2 Refined Vocabulary for Sources of Error

In connecting this work to other problems in the class of applied textual inference and the field of language technologies in general, one useful exercise is to generalize the error classes we identified for the problem of description retrieval onto the general linguistic phenomena they may represent. An example of such a generalization is proposed in Table 9.1. Some classes from the original vocabulary have been removed. Deleted classes are described in Table 9.2.

To fully explore the significance of the general classes is outside the scope of this thesis. However we can propose a sample methodology for this exploration, similar to the one we applied in Chapter 6. The first step would be to perform 1-2 rounds of sample annotation with these candidate categories, refining the category descriptions until they are consistent and applicable as simple “yes/no” tests for membership, given a pair of texts. During this phase additional categories might also be added.

After the definitions are refined, a list of examples should be associated

Error Type	Updated Description and Examples
Lexical Ambiguity	Term-level similarity that obscures meaning-level differences “at the <i>bank</i> counter” vs. “on the river <i>bank</i> ”
Accommodation	Term-level mismatch that obscures or falsifies pragmatic similarity “shawl” vs. “wrap”; “rectangular” vs. “rounded rectangle”
Accommodation- <i>Analogy</i>	Hedges or analogies obscure or falsify pragmatic connection between phrases “tree shaped like a hat” vs. “hat”
Accommodation- <i>Contradiction</i>	Elements that reflect pragmatic conflict under real-world constraints, even after interpretation “black background” vs. “blue background”
Incorrect Attachment	Matching semantic predicates with mismatched arguments, independent of the surface form “green bandana” vs. “green shirt”, “man skating” vs. “girl skating”
Quantification	Quantification mismatches between entities or sets that should be co-referent “a color photo” vs. “3 photos”, “1 guy eating fries” vs. “a guy and a girl eating fries”
Negation	Negated semantic predicates, independent of the surface form “not smiling” vs. “smiling”; “examples of dogs, no corgis”

Table 9.1: Candidates for general linguistic causes of mismatch between semantically similar texts. Accommodation has two sub-classes, *Analogy* and *Contradiction*.

Original Error Type	Mapped Error Type	Description and Examples
Missing elements	<i>None</i>	Treat as partial matching, apply other classes as appropriate “globe on a stand” vs. “globe”
Media	<i>None</i>	Treat as a specialization of the Contradiction class “cartoon” vs. a photograph, “black and white photo” vs. a color image

Table 9.2: Error classes from the original vocabulary that were removed in the general vocabulary.

with each to help annotators make a concrete connection to what each category means. These definitions could then be dropped into an annotation interface, similar to the web-based tool we developed for our exercise in Chapter 6. A screenshot of this tool is shown in Figure 9.1.

With new category definitions and an annotation tool in place, an informative exercise would be to annotate several data sets from different example problems within the ATI class (entailment, description retrieval, paraphrasing, summarization, question answering) in order to compare the distribution of these phenomena across those problems.

9.3 Future Work

9.3.1 Summary of Future Work

Address more error types from Chapter 6

In this work we identified eight classes of error that contribute to retrieval failures in the baseline system. We focused on two of these errors by hypothesizing system improvements that would address them, and then measuring the effect both on overall performance on errors of our two target types (errors attributed to ontological mismatch and faulty inference). A natural extension of this work would be to address additional classes of error identified in Chapter 6. Media-related errors, where a query description refers

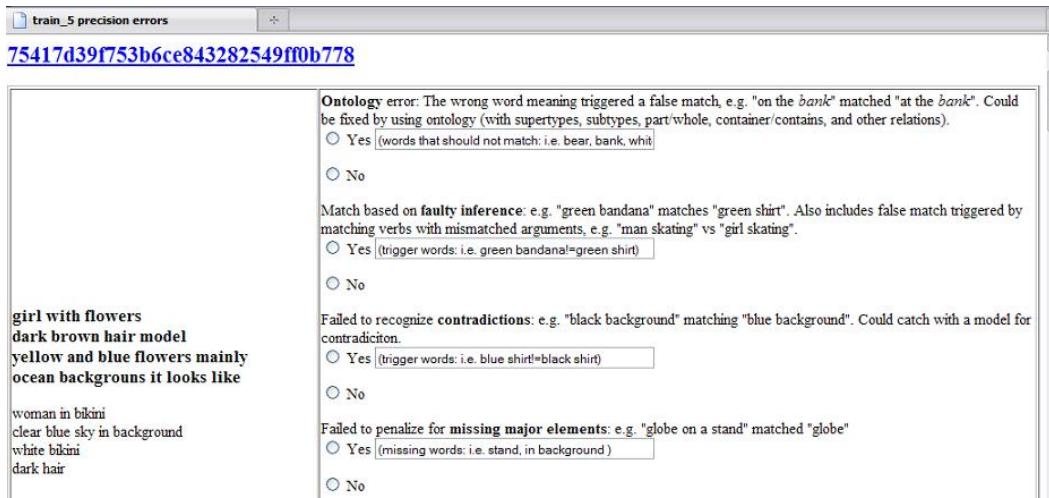


Figure 9.1: Screenshot from the web-based annotation tool used for the analysis in Chapter 6.

explicitly to a “drawing” while the index description returned by the baseline says “photograph”, could also be amenable to techniques presented here. One could imagine that a layer of the ontology could distinguish media types based on research on common subclasses of image, particularly on the web. It would be interesting to discover whether ontology development alone, integrated into the existing annotated retrieval system, could reduce these errors, or whether we would require additional modification of the retrieval pipeline as well, for example in a separate binary classifier that returns an estimate of whether the two image descriptions have matching media types.

This type of future work could also incorporate additional Natural Language Processing tools. Co-reference resolution might be beneficial in reducing errors associated with quantification mismatch, in particular as it relates to the number of people in an image description. Techniques for contradiction detection have been developed and tested for other textual inference problems, including recognizing textual entailment and question answering. These techniques could also apply to retrieval errors caused by real or perceived contradictions between a query and an index description.

Extend the Knowledge Framework with transformation rules in addition to KBs

The solution described here relies on the knowledge base to ground concepts and encode important relations, including paths (e.g. *sash* is a garment worn at a *pageant*). However, the features we extracted from this encoding only leverage a small part of the power of the knowledge base. A natural extension of the system described here would be to add knowledge-based interpretation *rules* that could work to overcome more of the structural differences that occur between descriptions at the syntactic level.

Some examples along these lines include the work by Fan and Porter to resolve Loose Speak in question encodings (Fan and Porter, 2004) and the work by Barnett et al. (1990) to perform semantic transformations in the context of knowledge-based machine translation.

Leverage structured retrieval architectures

As we describe in Chapter 2, the work presented in this thesis builds on contemporary research in information retrieval and in textual inference. Natural extensions of the experiments we describe here could re-connect our results with threads in these two areas. For example, this thesis focuses on examining layers of world knowledge that are required to solve the most interesting cases of description retrieval. Contemporary research on IR has generated alternative architectures for structured retrieval, but has been agnostic with respect to what features should be included in these structures and why (Bilotti et al., 2008). A promising experiment could combine the knowledge base developed in this thesis with one of these architectures, possibly yielding higher accuracy than either system could achieve alone.

9.3.2 Extension to Other ATI Tasks

Figure 9.2 shows a diagrammatic description of the relationship between Language Understanding and several examples of applied textual inference problems. This diagram captures the common features of ATI tasks as we described them in Chapter 1. Given a pair of texts, *Text A* and *Text B*, an ATI system should perform some processing over both, then calculate a function F over the result of this processing in order to arrive at a decision regarding the relationship between *Text A* and *Text B*. This relationship

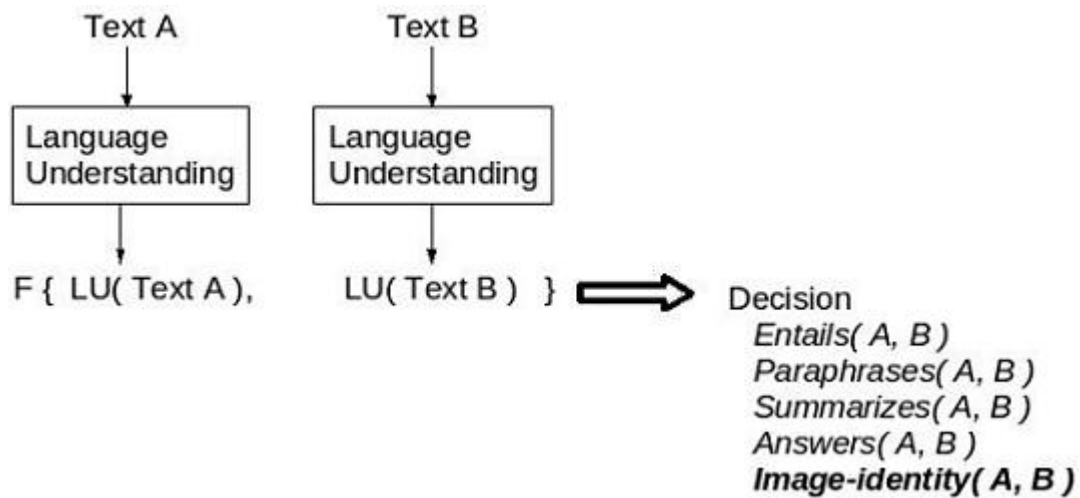


Figure 9.2: Diagrammatic description of applied textual inference Tasks. They are recognized to be hard enough to require some amount of language understanding for success, but they are evaluated based on the accuracy of a decision output. Entailment, Paraphrasing, and other well-known tasks are shown here as examples that meet this description. Image-identity is shown as a new example.

may be a question of Entailment: does A entail B ; or of another relation: does A paraphrase B ; is A an answer for B . For each of these problems, the relation is assumed to be dependent on some, though perhaps not full, language understanding.

In this work we frame the retrieval of image descriptions as a problem that fits the ATI mold. Error analysis showed that recognizing the relation *image-identity*(A, B) does sometimes depend on language understanding at the semantic and world-knowledge level. In addition, we built and evaluated a system that returns descriptions where this relation is most likely to hold, according to our model of the problem.

As a result, we might hypothesize that techniques we developed and applied here would be relevant for other ATI problems as well. For example, could the text similarity function that we use for reranking in Chapter 7 be re-tuned for the problem of recognizing textual entailment, or for passage retrieval in question answering?

To answer these questions concretely in future work, we would follow the steps outlined below.

1. Analyze differences in the data.

The task will now be to read in texts A and B and produce a decision about them, where the decision will be determined by the training data that we select for the new problem. In the architecture shown in Figure 7.2, pairwise decisions about the relationship between two descriptions are made in the lower two-thirds of the diagram. The SconeImage extension to Scone expects input in the form of a *query* and a *candidate_list*, and produces pairwise *query_description*, *candidate_description* features. As a result, the data format for a new problem could be massaged into a structure where there is only one text, B , in the candidate list, while text A is formatted as the query. Some massaging of the data into this format will be required. We must also identify how ground-truth labels may be captured and attached for training purposes.

2. Determine an appropriate baseline solution.

In Chapter 5 we described a baseline solution for image description retrieval that relies on keyword features and was implemented using the Lemur Information Retrieval Toolkit with the Indri Search Engine. In the case of a new ATI task, a comparable baseline might be the

combination of keyword-only features in a linear classifier, similar to the solution that we implemented for the image retrieval problem.

3. Perform error analysis.

In Section 9.2.2, we described a candidate set of refined error classes. At this step, we would depend on future work as outlined in that section to establish a final error vocabulary. We would also have to modify the method for selecting items to annotate, since our current description of “most severe” errors depends on a ranked list as output, rather than a binary decision. However the confidence score returned by the classifier for miss-classified examples could be one factor in choosing items for annotation.

4. Extend the KB and perform classifier training.

After annotating errors, we would use the annotation results to populate a domain-specific KB, using the methods developed in Chapter 8. This process could help us answer some interesting questions empirically: is it better to throw away the old domain-specific knowledge base, or simply add to it? Does this extension require changes to the upper-level knowledge bases? Does this method continue to out-perform WordNet on the new task?

5. Test on held-out examples.

Finally, we would test on a development or testing set that appears in a similar format to the training data. Because the decisions in this general case are likely to be binary, MRR is inappropriate as an evaluation metric, but accuracy, precision, recall, and combinations of these would apply. For data sets where results have been published, we would apply the metrics that are in current use, for comparable results.

While execution of these steps is reserved for future work, the exercise of describing how to apply our system to other ATI data sets strengthens the relevance of our work to concurrent research on tasks like recognizing entailment.

9.4 Conclusions

The experiments described in this thesis yield encouraging results with respect to the hypotheses that we formed as a result of detailed error analysis in Chapter 6. Specifically, textual inference techniques can lead to improved retrieval performance, in particular on the most interesting types of images: images with more visual content and less text than web buttons or logos, and images whose descriptions can only be interpreted with the application of ontological knowledge and inferential knowledge.

Statistically significant improvement was shown not only on the examples that were used for KB development, but also on the full set of training queries from which these examples were drawn. Strong trends between this result and a similar result on an unseen set of test queries encourage us to continue exploring the effect on larger data sets, in the future. These trends are a positive indication that knowledge-based techniques can complement the strengths of a strong bag-of-words baseline to achieve better overall performance on all image types.

The success of shared tasks for ATI in the last decade indicates growth in the field of NLU, and in particular a growing interest in deep text representations that can be leveraged by modern machine learning frameworks. This observation was made after the first Recognizing Textual Entailment Challenge, in 2005:

The fact that quite sophisticated inference levels were applied by some groups, with 5 systems using logical provers, provide an additional indication that applied NLP research is progressing towards deeper semantic analyses. Further refinements are needed though to obtain sufficient robustness for the Challenge types of data. (Dagan et al., 2006)

The work of this thesis contributes to better understanding of why deep representations are necessary, and how they may be effectively applied.

Bibliography

- Rod Adams, Gabriel Nicolae, Cristina Nicolae, and Sanda Harabagiu. Textual entailment through extended lexical overlap and lexico-semantic matching. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, 2007.
- Eneko Agirre, Bernardo Magnini, Oier Lopez de Lacalle, Arantxa Otegi, German Rigau, and Piek Vossen. Semeval-2007 task 01: Evaluating wsd on cross-language information retrieval. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, 2007.
- Y. Alp Aslandogan and Clement T. Yu. Multiple evidence combination in image retrieval: Diogenes searches for people on the web. In *ACM SIGIR*, Athens, Greece, 2000.
- Y. Alp Aslandogan, Chuck Their, Clement T. Yu, Jon Zou, and Naphtali Rishe. Using semantic contents and wordnet in image retrieval. In *20th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, 1997. ACM.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007.
- Jim Barnett, Kevin Knight, Inderjeet Mani, and Elaine Rich. Knowledge and natural language processing. *Communications of the ACM*, 33(8), 1990.
- Matthew W. Bilotti, Le Zhao, Jamie Callan, and Eric Nyberg. Focused retrieval over richly-annotated collections. In *Proceedings of the SIGIR 2008 Workshop on Focused Retrieval*, 2008.

- Catherine Blake. The role of sentence structure in recognizing textual entailment. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 101–106, 2007.
- D. Blei and M. Jordan. Modeling annotated data. In *the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 127134. ACM Press, 2003.
- A. Budanitsky and G. Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of NAACL-2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, 2001.
- James P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY Retrieval System. In *the Third International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.
- Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D. Manning. Learning alignments and leveraging natural logic. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170, Prague, 2007.
- Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139. Morgan Kaufmann Publishers Inc., Seattle, Washington, 2000.
- Francois-Regis Chaumartin. Upar7: A knowledge-based system for headline sentiment tagging. In *the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 422–425, 2007.
- Peter Clark and Bruce Porter. Building concept representations from reusable components. In *Fourteenth National Conference on Artificial Intelligence (AAAI'97)*, pages 369–376. AAAI Press, 1997.
- Peter Clark, John Tompson, and Bruce Porter. A knowledge-based approach to question-answering. In *AAAI'99 Fall Symposium on Question Answering Systems*, 1999.

- Paul Clough and Mark Sanderson. The clef 2003 cross language image retrieval track. In *Cross Language Evaluation Forum (CLEF 2003)*, Trondheim, Norway., 2003.
- William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *NIPS*. MIT Press, 1998.
- Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: <http://dx.doi.org/10.3115/1118693.1118694>.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. *Springer Lecture Notes in Computer Science*, 3944:177 – 190, 2006.
- Scott E. Fahlman. Marker-passing inference in the scone knowledge-base system. In *the First Annual International Conference on Knowledge Science, Engineering, and Management (KSEM 2006)*, Guilin, China, 2006.
- Scott E. Fahlman. *NETL: A System for Representing and Using Real-World Knowledge*. MIT Press, Cambridge, MA, 1979.
- James Fan and Bruce Porter. Interpreting loosely encoded questions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI '04)*, 2004.
- James Fan, Ken Barker, and Bruce Porter. The knowledge required to interpret noun compounds. Technical Report UT-AI-TR-03-301, University of Texas at Austin, 2003.
- C. Fellbaum. *WordNet An Electronic Lexical Database*. Bradford Books, 1998.
- Norbert Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *Computer Journal*, 35(3):243–255, 1989.
- Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. Sweetening wordnet with dolce. *AI Magazine*, 24(3):13 – 24, 2003.

- Ruifang Ge and Raymond J. Mooney. Discriminative reranking for semantic parsing. In *Coling/ACL 2006*, Sydney, Australia, 2006. ACL.
- Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *RTE '07: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Morristown, NJ, USA, 2007. Association for Computational Linguistics.
- Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio, and William Dolan. The fourth pascal recognizing textual entailment challenge. In *Text Analysis Conference (TAC 2008)*, 2008.
- Yolanda Gil and Marcelo Tallis. A script-based approach to modifying knowledge bases. In *Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, , Providence, RI., 1997.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. Semeval-2007 task 04: Classification of semantic relations between nominals. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, Prague, 2007.
- Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Sixteenth International Conference on Computational Linguistics (COLING)*, pages 466–471, Kopenhagen, 1996.
- Michael Grübinger, Paul Clough, Henning Mller, and Thomas Deselaers. The IAPR TC-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage2006 Language Resources for Content-Based Image Retrieval, held in conjunction with LREC'06*, pages 13–23, Genoa, Italy, 2006.
- J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In *In Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pages 379–386, Singapore, 2008. ACM, New York, NY.
- Aria Haghighi, Andrew Ng, and Christopher Manning. Robust textual inference via graph matching. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 387–394, Vancouver, British Columbia, Canada, 2005.

- Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girji, Vasile Rus, and Paul Morarescu. Falcon: Boosting knowledge for answer engines. In *Ninth Text Retrieval Conference (TREC-9)*, 2000.
- Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. Negation, contrast and contradiction in text processing. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 755–762. AAAI Press, 2006.
- Donna Harman. Overview of the first text retrieval conference (TREC-1). In *the First Text REtrieval Conference (TREC-1)*, pages 1–20, 1992.
- Stefan Harmeling. An extensible probabilistic transformation-based approach to the third recognizing textual entailment challenge. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 137–142, Prague, 2007.
- Andrew Hickl and Jeremy Bensley. A discourse commitment-based framework for recognizing textual entailment. In *ACL 2007 Workshop on Textual Entailment and Paraphrasing*, Prague, 2007. ACL.
- Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. In *TREC-9*. National Institutes of Standards and Technology (NIST), 2001.
- Peng Jin, Yunfang Wu, and Shiwen Yu. Semeval-2007 task 05: Multilingual chinese-english lexical sample. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, Prague, 2007.
- C. Jørgensen. Towards an image test bed for benchmarking image indexing and retrieval systems. In *the International Workshop on Multimedia ContentBased Indexing and Retrieval*, Rocquencourt, France, 2001.
- Jason S. Kessler. Polling the blogosphere: a rule-based approach to belief classification. In *International Conference on Weblogs and Social Media*, 2008.
- David M. Magerman. Statistical decision-tree models for parsing. In *Meeting of the Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics, 1995.

- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC 2006*, 2006.
- Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc Schneider. The wonderweb library of foundational ontologies. Technical Report WonderWeb Deliverable D17, National Research Council, Institute of Cognitive Sciences and Technology (ISTC-CNR), 2003.
- Diana McCarthy and Roberto Navigli. Semeval-2007 task 10: English lexical substitution task. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, 2007.
- David B. McDonald. Understanding noun compounds. Technical Report CMU-CS-82-102, Department of Computer Science, Carnegie Mellon University, 1982.
- Arun Meena and T. V. Prabhakar. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *Advances in Information Retrieval*, volume 4425 of *Lecture Notes in Computer Science*. Springer, 2007.
- Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *21st National Conference on Artificial Intelligence (AAAI)*, Boston, 2006.
- Henning Müller, Stephane Marchand-Maillet, and Thierry Pun. The truth about corel - evaluation in image retrieval. In M. S. Lew, N. Sebe, and J. P. Eakins, editors, *Lecture Notes In Computer Science*, volume 2383, pages 38–49. Springer-Verlag, London, 2002.
- Paul Ogilvie and Jamie Callan. Combining document representations for known item search. In *the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 143–150, 2003.

- Zeynep Orhan, Emine elik, and Demirg Neslihan. Semeval-2007 task 12: Turkish lexical sample task. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, Prague, 2007.
- Monica Lestari Paramita, Mark Sanderson, and Paul Clough. Diversity in Photo Retrieval: Overview of the ImageCLEFPhoto Task 2009. In *Proceedings of the Ninth Cross Language Evaluation Forum (CLEF 2009)*, Corfu, Greece, 2009.
- S. Patwardhan, S. Banerjee, and T. Pedersen. Umnd1: Unsupervised word sense disambiguation using contextual semantic relatedness. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, pages 390–393, Prague, Czech Republic, 2007.
- Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR 1998*, pages 275–281, 1998.
- Fabio Rinaldi, James Dowdall, Michael Hess, Diego Molla, Rolf Schwitter, and Kaarel Kaljurand. Knowledge-based question answering. In V. Palade, R.J. Howlett, and L.C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 785–792. Springer-Verlag, 2003.
- Patrick Ruch. Information retrieval and spelling errors: Improving effectiveness by lexical disambiguation. In *ACM-SAC Information Access and Retrieval Track, 2002*, 2002.
- Roger C. Shank and Larry Tesler. A conceptual dependency parser for natural language. In *COLING 1969*, 1969.
- Nikhil V Shirahatti and Kobus Barnard. Evaluating image retrieval. *Computer Vision and Pattern Recognition*, 1:955–961, 2005.
- Trevor Strohman, Dan Metzler, Howard Turtle, and W. Bruce Croft. Indri: A language model-based search engine for complex queries. In *International Conference on Intelligence Analysis (ICIA) (poster)*, McLean, VA, 2005.
- Alicia Tribble and Scott E. Fahlman. Resolving noun compounds with multi-use domain knowledge. In *Proceedings of FLAIRS-2006*, Melbourne Beach, FL, 2006.

- Alicia Tribble and Scott E. Fahlman. CMU-AT: Semantic Distance and Background Knowledge for Identifying Semantic Relations. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 121–124, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- Alicia Tribble, Benjamin Lambert, and Scott E. Fahlman. SconeEdit: A Text-guided Domain Knowledge Editor. In *Demonstration Sessions of HLT/NAACL-2006*, New York, 2006.
- Howard Turtle and W. Bruce Croft. Evaluation of an inference network based retrieval model. *Trans. Inf. Syst.*, 9(3):187–222, 1991.
- Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides G.M. Petrakis, and Evangelos E. Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *the 7th Annual ACM International Workshop on Web Information and Data Management*, pages 10–16, Bremen, Germany, 2005. ACM.
- Remco C. Veltkamp and Mirela Tanase. A survey of content-based image retrieval systems. In O. Marques and B. Furht, editors, *Content-Based Image and Video Retrieval*, pages 47–101. Kluwer, 2002.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. Semeval-2007 task 15: Tempeval temporal relation identification. In *SemEval-2007, Workshop of the Association for Computational Linguistics (ACL 2007)*, 2007.
- Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM Press, Vienna, Austria, 2004.
- Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 1994. ACM.
- Huan Wang, Xing Jiang, Liang-Tien Chia, and Ah-Hwee Tan. Ontology enhanced web image retrieval: Aided by wikipedia & spreading activation theory. In *MIR 2008*. ACM, 2008.

- Rui Wang and Guenter Neumann. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 36–41, Prague, 2007.
- Li Zhuang, Feng Jing, Xiao yan Zhu, and Lei Zhang. Movie review mining and summarization. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 43–50, 2006.
- Amal Zouaq, Roger Nkambou, and Claude Frasson. The knowledge puzzle: An integrated approach of intelligent tutoring systems and knowledge management. In *the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006)*, pages 575–582, 2006.

Appendix A

Sample Parameter Files

```
<!--
    Sample build_param file for the phetch 5A section, formatted as a
    SconeImage sgml document (tags, descriptions, phrases) with
--!>
<parameters>
<index>/home/Projects/IR/indri_index/phetch-5A</index>
<corpus>
<path>/home/Projects/IR/indri_data/phetch-5A.sgml</path>
<class>trectext</class>
</corpus>
<memory>1G</memory>
<stemmer><name>krovetz</name></stemmer>
<field><name>tags</name></field>
<field><name>descriptions</name></field>
<field><name>description</name></field>
<field><name>phrase</name></field>
</parameters>
```


Appendix B

Upper-level Ontology: DOLCE

```
;;; -*- Mode:Lisp -*-
;;; *****
;;; Scone Knowledge Representation System
;;;
;;;
;;; dolce-upper-kb.lisp:
;;; Dolce Upper Ontology, in Scone
;;;
;;;
;;; Adapted from the work of:
;;;
;;; Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari.
;;; Sweetening wordnet with dolce. AI Magazine, 24(3):13-24, 2003.
;;;
;;; Author: Alicia Tribble (atribble@cs.cmu.edu)
;;; *****
;;;

(in-namespace "dolce" :include "common")

;; Create a new root node for a parallel
;; hierarchy to the Scone Native types
(new-type {DOLCE-ROOT} {thing})
```



```

(new-complete-split-subtypes {DOLCE-ROOT}
  '({abstract} :adj-noun)
  ({spatio-temporal-particular} :adj-noun)))

(new-complete-split-subtypes {spatio-temporal-particular}
  '({endurant} :adj-noun "endurant" "continuant")
  ({perdurant} :adj-noun "perdurant" "occurrence")
  ({quality} :adj-noun)
  ({physical-realization} :adj-noun "physical realization")))

;;; ROOT -> PARTICULAR -> ENDURANT ;;;

;; physical and non-physical endurants are disjoint
(new-split-subtypes {endurant}
  '({non-physical-endurant} "non-physical endurant" "non-physical continuant"
    ({physical-endurant} "physical endurant" "physical continuant")
    ({arbitrary-sum} :adj-noun)))

;; agent is an endurant that can have subclasses in common
;; with the other endurants
(new-type {agent} {endurant})

;; ENDURANT -> NON-PHYSICAL-ENDURANT ;;

(new-type {non-physical-object} {non-physical-endurant})

(new-complete-split-subtypes {non-physical-object}
  '({social-object}
    ({mental-object})))

;; ENDURANT -> NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT
;; -> SOCIAL-OBJECT

```

```

(new-type {agentive-social-object} {social-object})
(new-type {figure} {social-object})
(new-type {non-agentive-social-object} {social-object})

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT
;; -> FIGURE

(new-split-subtypes {figure}
  '({non-agentive-figure}
    {agentive-figure}))

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT
;; -> AGENTIVE-SOCIAL-OBJECT ;;

(new-is-a {agentive-figure} {agentive-social-object})

(new-type {socially-constructed-person} {agentive-figure})

(new-split-subtypes {socially-constructed-person}
  '({natural-person} {organization}))

(new-type {institution} {organization})

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT
;; -> NON-AGENTIVE-SOCIAL-OBJECT ;;

(new-split-subtypes {non-agentive-social-object}
  '({collection}
    {concept}
    {description}
    {information-object})

```

```

    {situation}))

(new-is-a {non-agentive-figure} {non-agentive-social-object})

(new-split-subtypes {collection}
  '({non-physical-collection}
    {organized-collection}
    {simple-collection}
    {collective}))

(new-split-subtypes {concept}
  '({course} {parameter} {role}))

(new-split-subtypes {description}
  '({constitutive-description}
    {information-encoding-system}
    {method}
    {modal-description}
    {social-description}
    {theory}
    {narrative}
    {subject}
    {system-as-description}))

(new-split-subtypes {information-object}
  '({creative-object}
    {diagrammatic-object}
    {formal-expression}
    {iconic-object}
    {linguistic-object}))

(new-type {non-physical-place} {non-agentive-figure})

(new-type {geographical-place} {non-physical-place})

(new-type {political-geographic-object} {geographical-place})

(new-type {country} {political-geographic-object})

```

```

(new-split-subtypes {situation}
  '({goal-situation}
    {plan-execution}
    {communication-situation}
    {interpretation-situation}
    {production-workflow-execution}
    {system-as-situation}
  ))

;; ENDURANT -> NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> MENTAL-OBJECT ;;

;; ENDURANT -> PHYSICAL-ENDURANT ;;

(new-split-subtypes {physical-endurant}
  '({amount-of-matter})
    ;;; Amounts of matter are Endurants with no unity
    ;;;(none of them is an essential whole).

    ({physical-object})
      ;;; Physical Objects is that they are Endurants with unity.

    ({feature})
      ;;; Typical features are parasitic entities, such as holes, boundaries,
      ;;; surfaces, or stains, ... constantly dependent on physical Objects (their
      ;;; hosts).
  ))

;; ENDURANT -> PHYSICAL-ENDURANT -> AMOUNT-OF-MATTER ;;

(new-type {functional-matter} {amount-of-matter})

```

```

;; ENDURANT -> PHYSICAL-ENDURANT -> FEATURE ;;

(new-split-subtypes {feature}
;; "A hole in a piece of cheese, a surface of a table"
'({dependent-place})

    ;; "A relevant part of a host object,
        ;; like a bump or an edge"
    ({relevant-part}))

(new-type {spatial-feature} {relevant-part})

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT ;;

;; agentive and non-agentive physical objects are disjoint
(new-split-subtypes {physical-object}
'({agentive-physical-object})
  ({non-agentive-physical-object}))

;; other subclasses are not
(new-type {physical-plurality} {physical-object})

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT
;; -> AGENTIVE-PHYSICAL-OBJECT ;;

(new-type {rational-physical-object} {agentive-physical-object})

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT
;; -> NON-AGENTIVE-PHYSICAL-OBJECT ;;

```

```

(new-split-subtypes {non-agentive-physical-object}
  '({physical-body}
    {physical-place}
    {material-artifact}
  ))

;; physical-object -> non-agentive-physical-object -> physical-body
(new-split-subtypes {physical-body}
  '({biological-object}
    {chemical-object}
  ))

;; physical-object -> non-agentive-physical-object -> physical-place
(new-split-subtypes {physical-place}
  '({geographical-object}
  ))

;; physical-object -> non-agentive-physical-object -> physical-artifact

;; ENDURANT -> AGENT ;;

;; The subtypes of edns:agent exhibit multiple inheritance
(new-is-a {agentive-physical-object} {agent})
(new-is-a {agentive-social-object} {agent})

(new-type {rational-agent} {agent})

(new-is-a {rational-physical-object} {rational-agent})

;;; ROOT -> PARTICULAR -> PERDURANT ;;;

(new-complete-split-subtypes {perdurant}
  '(({event})
    ({stative})))

```

```

;;; PERDURANT -> EVENT ;;;

;;; Eventive occurrences (events) are called achievements
;;; if they are atomic and accomplishments otherwise.
(new-type {accomplishment} {event})
(new-type {achievement} {event})
(new-type {cognitive-event} {event})

;; perdurant -> event -> accomplishment
(new-split-subtypes {accomplishment}
  '({action}
    {communication-event}
    {phenomenon}))

;;; PERDURANT -> STATIVE ;;;

(new-complete-split-subtypes {stative}
  '({state}
    ({process})))

;; Subtypes of process
(new-type {flux} {process})
(new-type {reconstructed-flux} {flux})

;; Subtypes of state
(new-type {cognitive-state} {state})
(new-type {decision-state} {state})

;;; ROOT -> PARTICULAR -> QUALITY ;;;

;;; From "Sweetening WordNet with Dolce":
;;; Qualities can be seen as the basic entities we can perceive or

```

```

;;; measure: shapes, colors, sizes, sounds, and
;;; smells as well as weights, lengths, electrical
;;; charges, and so on.
;;;
(new-split-subtypes {quality}
  ;;; Physical qualities are those that directly
  ;;; inhere to physical Endurants
  '({physical-quality})

  ;;; Temporal qualities are those that directly
  ;;; inhere to perdurants
  ({temporal-quality})

  ;;; Abstract qualities are those that directly
  ;;; inhere to nonphysical perdurants
  ({abstract-quality}))

;;; Attach the Qualitites to the Objects they describe
(new-type-role {physical-quality (role)}
  {physical-endurant}
  {physical-quality})

(new-type-role {temporal-quality (role)}
  {perdurant}
  {temporal-quality})

(new-type-role {abstract-quality (role)}
  {non-physical-endurant}
  {abstract-quality})

;; quality -> physical-quality

(new-type {spatial-location_q} {physical-quality})

;; quality -> temporal-quality

(new-type {temporal-location_q} {temporal-quality})

```



```
;;; ROOT -> PARTICULAR -> PHYSICAL-REALIZATION ;;;
```

```
;; From the OWN kb comments, a Physical Realization is:
;; Any physical particular that realizes a non-physical enduring.
;; Such physical particulars can be either physical enduring,
;; physical qualities, physical regions, perdurants with at least
;; one physical participant, or a situation with one physical entity
;; in its setting. Ultimately, a physical realization depends on at
;; least one physical enduring (each of the others physical entity
;; types depend on a physical enduring to be considered as such)
(new-type {information-realization} {physical-realization})
```

```
(new-type {bodily-motion} {information-realization})
```

```
(new-type {depiction} {information-realization})
```

```
(new-type {voicing} {information-realization})
```

```
;;; ROOT -> ABSTRACT ;;;
```

```
(new-complete-split-subtypes {abstract}
'({region})
  ({abstract-set})
  ({proposition})))
```

```
;; abstract -> region
(new-split-subtypes {region}
'({abstract-region}
  {physical-region}
  {quale}
  {quality-space}
  {temporal-region}
  ))
```


Appendix C

Upper-level Ontology: WN+DOLCE

```
;;; -*- Mode:Lisp -*-
;;; *****
;;; Scone Knowledge Representation System
;;;
;;;
;;; WN-dolce-upper-kb.lisp:
;;; Top-level WordNet Concepts, linked to the Scone-ified Dolce Upper Ontology
;;;
;;; Adapted from the work of:
;;;
;;; Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari.
;;; Sweetening wordnet with dolce. AI Magazine, 24(3):13-24, 2003.
;;;
;;; Author: Alicia Tribble (atribble@cs.cmu.edu)
;;; *****
;;;
```

```
(in-namespace "dolce" :include "common")
```

```
(new-type {wordnet-type} {thing})
```

```

;;; ROOT -> PARTICULAR -> ENDURANT ;;;

;; ENDURANT -> NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT;;

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT -> FIGURE ;;

;; social-object -> figure -> non-agentive-figure
(new-type {philosophers_stone_n_1} {non-agentive-figure})

;; social-object -> figure -> agentive-figure
(new-split-subtypes {agentive-figure}
  '({destiny_n_2}
    {first_cause_n_1}
    {imaginary_being_n_1}
    {nature_n_3}
    {organization_n_2}
    {organized_crime_n_1}
    {supernatural_n_1}
    {vital_principle_n_1}
  ))

;; social-object -> figure -> non-agentive-figure -> non-physical-place
(new-split-subtypes {non-physical-place}
  '({address_n_3}
    {biogeographical_region_n_1}
    {grave_n_1}
    {imaginary_place_n_1}
    {point_n_6}
    {sign_of_the_zodiac_n_1}
  ))

;; social-object -> figure -> non-agentive-figure -> non-physical-place
;; -> geographical-place

```

```

(new-split-subtypes {political-geographic-object}
  '({district_n_1}))

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT
;; -> AGENTIVE-SOCIAL-OBJECT

;; agentive-social-object -> socially-constructed-person -> person
(new-type {person_n_1} {socially-constructed-person})

(new-split-subtypes {person_n_1}
  '({adult__growable}
    {child__baby}))

(new-split-subtypes {person_n_1}
  '({female_n_1}
    {male_n_1}))

(new-type {man_n_1} {male_n_1})
(new-type {male_child__boy__child} {male_n_1})

(new-type {woman_n_1} {female_n_1})
(new-type {female_child__girl__child} {female_n_1})

;; NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT -> SOCIAL-OBJECT
;; -> NON-AGENTIVE-SOCIAL-OBJECT

;; social-object -> non-agentive-social-object -> collection
(new-split-subtypes {collection}
  '({art_collection_n_1}
    {exhibition_n_2}
    {group_n_1}
    {repertoire_n_2}
    {repertory_n_1}))

```

```

;; social-object -> non-agentive-social-object -> concept

;; social-object -> non-agentive-social-object -> description

;; social-object -> non-agentive-social-object -> information-object
(new-split-subtypes {information-object}
  '({language_unit_n_1}
    {signal_n_1}
    {message_n_1}))

;; social-object -> non-agentive-social-object -> signal_n_1
(new-type {symbol_n_1} {signal_n_1})

;; social-object -> non-agentive-social-object -> situation

;; social-object -> non-agentive-social-object -> non-agentive-figure
;;
;; duplicate parent, covered under
;; social-object -> figure -> non-agentive-figure

;; ENDURANT -> NON-PHYSICAL-ENDURANT -> NON-PHYSICAL-OBJECT
;; -> MENTAL-OBJECT

;; non-physical-object -> mental-object
(new-split-subtypes {mental-object}
  '({cognition_n_1}
    {mind_n_1}
    {psychological_feature_n_1}))

;; ENDURANT -> PHYSICAL-ENDURANT ;;

;; An under-specifiec physical endurant
(new-type {entity_something_n_1} {physical-endurant})

```

```
;; ENDURANT -> PHYSICAL-ENDURANT -> AMOUNT-OF-MATTER ;;
```

```
;; amount-of-matter
(new-split-subtypes {amount-of-matter}
  '({atmosphere_n_1}
    {chemical_element_n_1}
    {compound_n_2}
    {fluid_n_1}
    {fluid_n_2}
    {ice_n_1}
    {land_ground_soil}
    {mass_n_5}
    {substance_n_1}
    {substance_matter}
  ))
```

```
;; ENDURANT -> PHYSICAL-ENDURANT -> FEATURE ;;
```

```
;; feature -> relevant-part -> spatial-feature
(new-split-subtypes {spatial-feature}
  '({air_n_3}
    {back_n_3}
    {base_n_5}
    {bottom_n_1}
    {boundary_n_2}
    {centerline_n_1}
    {depth_n_3}
    {extremity_n_2}
    {front_n_3}
    {geological_formation_n_1}
    {perimeter_n_1}
    {side_n_7}
    {surface_n_1}
    {top_n_4}))
```



```

;; feature -> dependent-place
(new-type {abutment_n_2} {dependent-place})
(new-type {antipodes_n_1} {dependent-place})
(new-type {blind_spot_n_1} {dependent-place})
(new-type {center_n_1} {dependent-place})
(new-type {corner_n_1} {dependent-place})
(new-type {corner_n_4} {dependent-place})
(new-type {crossing_n_3} {dependent-place})
(new-type {enclosure_n_1} {dependent-place})
(new-type {enclosure_n_3} {dependent-place})
(new-type {focus_n_1} {dependent-place})
  ;; i.e. the inside of something
(new-type {inside_n_2} {dependent-place})
(new-type {junction} {dependent-place})
(new-type {layer_n_3} {dependent-place})
;; "the left" i.e. left side of something
(new-type {left_n_1} {dependent-place})
(new-type {opening_n_3} {dependent-place})
(new-type {pass_n_4} {dependent-place})
(new-type {right_n_3} {dependent-place})
(new-type {space_n_4} {dependent-place})
(new-type {trichion_n_1} {dependent-place})
(new-type {vacuum_n_1} {dependent-place})

;; feature -> relevant-part
(new-type {belt_n_3} {relevant-part})
(new-type {connection_n_6} {relevant-part})
(new-type {corner_n_5} {relevant-part})
(new-type {covering_n_5} {relevant-part})
(new-type {curvature_n_2} {relevant-part})
(new-type {feature_n_1} {relevant-part})
(new-type {flare_n_2} {relevant-part})
(new-type {fragment_n_2} {relevant-part})
(new-type {head_n_8} {relevant-part})
(new-type {navel_n_1} {relevant-part})
(new-type {nub_n_1} {relevant-part})
(new-type {part_n_2} {relevant-part})

```

```

(new-type {part_n_9} {relevant-part})
(new-type {segment_n_1} {relevant-part})
(new-type {slice_n_2} {relevant-part})
(new-type {strip_n_1} {relevant-part})

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT ;;

;; physical-object -> object-physical-object
(new-type {object_n_1} {physical-object})

(new-type {artifact_n_1} {object_n_1})
(new-type {natural_object_n_1} {object_n_1})

;; physical-object -> physical-plurality
;; a.k.a. unitary collection in D18. The physical counterpart
;; (realization) of a collection. A collection (see) is
;; characterized by a conventional or emergent property.
;; Physical pluralities have as *proper parts* only physical
;; objects that are *members* of a same collection.
(new-split-subtypes {physical-plurality}
  '({aviation_n_1}
    {content_n_2}
    {mail_n_3}
  ))

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT
;; -> AGENTIVE-PHYSICAL-OBJECT

;; physical-object -> agentive-physical-object
(new-split-subtypes {agentive-physical-object}
  '({automaton_n_1}
    {autopilot_n_1}
    {organism_n_1}
  ))

```

```

    {machine_n_1}
  ))

;; ENDURANT -> PHYSICAL-ENDURANT -> PHYSICAL-OBJECT
;; -> NON-AGENTIVE-PHYSICAL-OBJECT

;; physical-object -> non-agentive-physical-object
(new-split-subtypes {non-agentive-physical-object}
  '({block_n_1}
    {cocoon_n_1}
    {consumer_goods_n_1}
    {decoration_n_1}
    {excavation_n_3}
    {fixture_n_1}
    {float_n_1}
    {insert_n_2}
    {line_n_2}
    {marker_n_1}
    {mechanism_n_2}
    {nest_n_5}
    {square_n_2}
    {strip_n_2}
    {way_n_2}
  ))

;; physical-object -> non-agentive-physical-object -> physical-body
;; -> biological-object

(new-split-subtypes {biological-object}
  '({body_n_1}
    {body_part_n_1}
    {cell_n_1}
    {plant_part_n_1}
  ))

```

```
;; physical-object -> non-agentive-physical-object -> physical-body
;; -> chemical-object
```

```
(new-split-subtypes {chemical-object}
  '({atom_n_1}
    {group_n_2}
    {molecule_n_1}
    {unit_n_5}
  ))
```

```
;; PARTICULAR -> ENDURANT -> AGENT
;;
;; the agent instances are redundantly covered under
;; agentive-social-object and agentive-physical-object, above
;;
;;
```

```
;;; ROOT -> PARTICULAR -> PERDURANT ;;;
```

```
;;; PERDURANT -> EVENT ;;;
```

```
(new-type {contact_n_4} {event})
(new-type {crash_n_3} {event})
(new-type {creation_n_3} {event})
(new-type {discharge_n_1} {event})
(new-type {emergence_n_1} {event})
(new-type {event_n_1} {event})
(new-type {fire_n_2} {event})
(new-type {flash_n_2} {event})
(new-type {movement_n_14} {event})
(new-type {social_event_n_1} {event})
(new-type {sound_n_2} {event})
(new-type {thing_n_8} {event})
(new-type {union_n_2} {event})
```

```
;; perdurant -> event -> cognitive-event
```

```

(new-split-subtypes {cognitive-event}
  '({feeling_n_1}
    {motivation_n_1}
    {process_n_2}
    {process_n_4}
  ))

;; perdurant -> event -> accomplishment

;; perdurant -> event -> accomplishment -> action

;; perdurant -> event -> accomplishment -> communication-event

;; perdurant -> event -> accomplishment -> phenomenon

;;; PERDURANT -> STATIVE ;;;

;; perdurant -> stative -> state

(new-split-subtypes {state}
  '({hyalinization_n_1}
    {isomerism_n_1}
    {motionlessness_n_1}
    {psychological_state_n_1}
    {serration_n_3}
    {tilth_n_1}
    {turgor_n_1}
    {wetness_n_1}))

;; perdurant -> stative -> state -> cognitive-state

(new-type {cognitive_state_n_1} {cognitive-state})

;; perdurant -> stative -> process

```

```
;;; ROOT -> PARTICULAR -> QUALITY ;;;
```

```
;; quality -> property
(new-type {property_n_2} {quality})
```

```
;; quality -> quality
(new-type {quality_n_3} {quality})
```

```
(new-type {texture_n_4} {quality_n_3})
```

```
;; quality -> thing
(new-type {thing_n_4} {quality})
```

```
;;; ROOT -> PARTICULAR -> QUALITY -> ABSTRACT-QUALITY ;;;
```

```
(new-split-subtypes {abstract-quality}
  '({analyticity_n_1}
    {disposition_n_1}
    {manner_n_1}
    {personality_n_1}
    {quality_n_1}
    {selectivity_n_1}
    {trait_n_1}
  ))
```

```
;;; ROOT -> PARTICULAR -> QUALITY -> PHYSICAL-QUALITY ;;;
```

```
;;; Physical Qualities are Qualities that inhere to Physical Objects
```

```
(new-split-subtypes {physical-quality}
  '({actinism_n_1}
    {blob_n_1}
    {bodily_property_n_1}
    {consistency_n_1}
  ))
```

```

    {constitution_n_4}
    {distortion_n_2}
    {edibility_n_1}
    {heredity_n_1}
    {natural_shape_n_1}
    {newness_n_1}
    {oldness_n_1}
    {oldness_n_2}
    {olfactory_property_n_2}
    {physical_property_n_1}
    {saltiness_n_1}
    {shape_n_1}
    {spatial_property_n_1}
    {stainability_n_1}
    {strength_n_1}
    {tactile_property_n_1}
    {taste_property_n_1}
    {viability_n_1}
    {visual_property_n_1}
    {weakness_n_1}
    {youngness_n_1}
  ))

(new-split-subtypes {visual_property_n_1}
  '({color_n_1}
    {color_property_n_1}
    {colorlessness_n_1}
    {dullness_n_3}
    {lightness_n_5}
    {texture_n_2}
    ;; added by atribble, not in OWN.owl
    {softness_n_6}
  ))

(new-type {chromatic_color_n_1} {color_n_1})
(new-type {achromatic_color_n_1} {color_n_1})

```

```
(new-type {position__place} {spatial-location_q})
```

```
;;; ROOT -> PARTICULAR -> QUALITY -> TEMPORAL-QUALITY ;;;
```

```
(new-split-subtypes {temporal-quality}
  '({age_n_1}
    {sound_property_n_1}
    {temporal_property_n_1}
  ))
```

```
;;; ROOT -> PARTICULAR -> QUALITY -> PROPERTY ;;;
```

```
;;; ROOT -> PARTICULAR -> PHYSICAL-REALIZATION ;;;
```

```
;; physical-realization -> information-realization
(new-type {auditory_communication_n_1} {information-realization})
(new-type {brochure_n_1} {information-realization})
(new-type {creation_n_2} {information-realization})
(new-type {sign_n_1} {information-realization})
(new-type {visual_communication_n_1} {information-realization})
(new-type {written_communication_n_1} {information-realization})
```

```
;; physical-realization -> information-realization
```

```
;; -> visual_communication_n_1
```

```
(new-split-subtypes {visual_communication_n_1}
  '({artwork_n_1}
    {body_language_n_1}
    {demonstration_n_1}
    {display_n_2}
    {gesture_n_1}
    {projection_n_3}
    {video_n_1}
  ))
```



```

;; physical-realization -> information-realization
;; -> written_communication_n_1
(new-split-subtypes {written_communication_n_1}
  '({code_n_1}
    {correspondence_n_2}
    {prescription_n_1}
    {prescription_n_2}
    {print_n_1}
    {reading_n_2}
    {transcription_n_1}
    {writing_4}
    {writing_2}
  ))

```

```

;;; ROOT -> ABSTRACT ;;;

```

```

;; abstract -> region
(new-split-subtypes {region}
  '({attribute_n_1}
    {magnitude_n_1}
    {measure_n_4}
    {measure_n_2}
  ))

```

```

;; abstract -> region -> magnitude
(new-split-subtypes {magnitude_n_1}
  '({amount_n_1}
    {amplitude_n_1}
    {bulk_n_1}
    {degree_n_1}
    {extent_n_1}
    {muchness_n_1}
    {multiplicity_n_1}
    {order_n_2}
    {proportion_n_1}
  ))

```

```
{size_n_2}
{size_n_3}
))

;; abstract -> region -> magnitude -> size_n_2
(new-split-subtypes {size_n_2}
  '({circumference_n_1}
    {largeness_n_3}
    {smallness_n_1}
  ))

;; abstract -> set
(new-split-subtypes {abstract-set}
  '({set_n_5}))

;; abstract -> proposition
;; No subtypes as of 01/05/2010
```