Commonsense Knowledge Acquisition and Applications

Niket Tandon

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken
August, 2016

Day of Colloquium	19/08/2016
Dean of the Faculty	Uni.-Prof. Dr. Frank-Olaf Schreyer

Examination Board

Chair of the Committee	Prof. Dr. Dietrich Klakow
Supervisor and Reviewer	Prof. Dr. Gerhard Weikum
Reviewer	Henry Lieberman, HDR
Reviewer	Dr. Jilles Vreeken
Academic Assistant	Dr. Luciano Del Corro

Abstract

Computers are increasingly expected to make smart decisions based on what humans consider commonsense. This would require computers to understand their environment, including properties of objects in the environment (e.g., a wheel is round), relations between objects (e.g., two wheels are part of a bike, or a bike is slower than a car) and interactions of objects (e.g., a driver drives a car on the road).

The goal of this dissertation is to investigate automated methods for acquisition of large-scale, semantically organized commonsense knowledge. This goal poses challenges because commonsense knowledge is: (i) *implicit and sparse* as humans do not explicitly express the obvious, (ii) *multimodal* as it is spread across textual and visual contents, (iii) *affected by reporting bias* as uncommon facts are reported disproportionally, (iv) *context dependent* and thus holds merely with a certain confidence. Prior state-of-the-art methods to acquire commonsense are either not automated or based on shallow representations. Thus, they cannot produce large-scale, semantically organized commonsense knowledge.

To achieve the goal, we divide the problem space into three research directions, making up the core contributions of this dissertation:

- Properties of objects: acquisition of properties like `hasSize`, `hasShape`, etc. We develop `WebChild`, a semi-supervised method to compile semantically organized properties.
- Relationships between objects: acquisition of relations like `largerThan`, `partOf`, `memberOf`, etc. We develop `CMPKB`, a linear-programming based method to compile comparative relations, and, we develop `PWKB`, a method based on statistical and logical inference to compile part-whole relations.
- Interactions between objects: acquisition of activities like *drive a car*, *park a car*, etc., with attributes such as temporal or spatial attributes. We develop `Knowlywood`, a method based on semantic parsing and probabilistic graphical models to compile activity knowledge.

Together, these methods result in the construction of a large, clean and semantically organized Commonsense Knowledge Base that we call `WebChild KB`.

Kurzfassung

Von Computern wird immer mehr erwartet, dass sie kluge Entscheidungen treffen können, basierend auf Allgemeinwissen. Dies setzt voraus, dass Computer ihre Umgebung, einschließlich der Eigenschaften von Objekten (z. B. das Rad ist rund), Beziehungen zwischen Objekten (z. B. ein Fahrrad hat zwei Räder, ein Fahrrad ist langsamer als ein Auto) und Interaktionen von Objekten (z. B. ein Fahrer fährt ein Auto auf der Straße), verstehen können.

Das Ziel dieser Dissertation ist es, automatische Methoden für die Erfassung von großmaßstäblichem, semantisch organisiertem Allgemeinwissen zu schaffen. Dies ist schwierig aufgrund folgender Eigenschaften des Allgemeinwissens. Es ist: (i) *implizit und spärlich*, da Menschen nicht explizit das Offensichtliche ausdrücken, (ii) *multimodal*, da es über textuelle und visuelle Inhalte verteilt ist, (iii) beeinträchtigt vom *Einfluss des Berichtenden*, da ungewöhnliche Fakten disproportional häufig berichtet werden, (iv) *Kontextabhängig*, und hat aus diesem Grund eine eingeschränkte statistische Konfidenz.

Vorherige Methoden, auf diesem Gebiet sind entweder nicht automatisiert oder basieren auf flachen Repräsentationen. Daher können sie kein großmaßstäbliches, semantisch organisiertes Allgemeinwissen erzeugen.

Um unser Ziel zu erreichen, teilen wir den Problemraum in drei Forschungsrichtungen, welche den Hauptbeitrag dieser Dissertation formen:

- Eigenschaften von Objekten: Erfassung von Eigenschaften wie `hasSize`, `hasShape`, usw. Wir entwickeln `WebChild`, eine halbüberwachte Methode zum Erfassen semantisch organisierter Eigenschaften.
- Beziehungen zwischen Objekten: Erfassung von Beziehungen wie `largerThan`, `partOf`, `memberOf`, usw. Wir entwickeln `CMPKB`, eine Methode basierend auf linearer Programmierung um vergleichbare Beziehungen zu erfassen. Weiterhin entwickeln wir `PWKB`, eine Methode basierend auf statistischer und logischer Inferenz welche zugehörigkeits Beziehungen erfasst.
- Interaktionen zwischen Objekten: Erfassung von Aktivitäten, wie *drive a car*, *park a car*, usw. mit temporalen und räumlichen Attributen. Wir entwickeln `Knowlywood`, eine Methode basierend auf semantischem Parsen und probabilistischen grafischen Modellen um Aktivitätswissen zu erfassen.

Als Resultat dieser Methoden erstellen wir eine große, saubere und semantisch organisierte Allgemeinwissensbasis, welche wir `WebChild KB` nennen.

Dedication

I dedicate this dissertation to my Gurus, Shri Krishna Dutt Nagar and Ms. Meena Tandon; and to my wife, Anjali Tandon. They have been the latent driving force throughout my Ph.D.

Acknowledgements

First and foremost, I would like to thank my supervisor, Gerhard Weikum, for giving me the opportunity to carry out this research, and providing matchless guidance. I would consider my career a success if I can emulate during my entire career half of the elegance, simplicity, vision, and enthusiasm he has displayed in the course of my doctoral work.

I would like to thank the additional reviewers and examiners of my dissertation, Henry Lieberman, and Jilles Vreeken. I am grateful to Dietrich Klakow and Luciano Del Corro for being part of my defense committee.

I am very grateful to Gerard de Melo for being an excellent close-collaborator, whose humility became my platform to come forward, and whose research excellence became my propeller to explore the insights.

I would like to thank all of my additional collaborators for their stimulating discussions and ideas. This includes Fabian Suchanek, Denilson Barbosa, Marcus Rohrbach, Anna Rohrbach, Jacopo Urbani, Abir De, Aparna Varde, Ekaterina Shutova, and Sreyasi Nag Chowdhury. I would also thank my Masters students for providing me opportunities to learn more- including Charles Hariman, Ali Shah, Cuong Xuan Chu, Shilpa Garg, and Jana Kohlhase. My colleagues and friends at my department constituted a helpful, easy, and stimulating environment for research, I hope to be able cross roads with them again.

Last but not least, I would like to thank my family and relatives for their constant support throughout the years.

Contents

1 Introduction

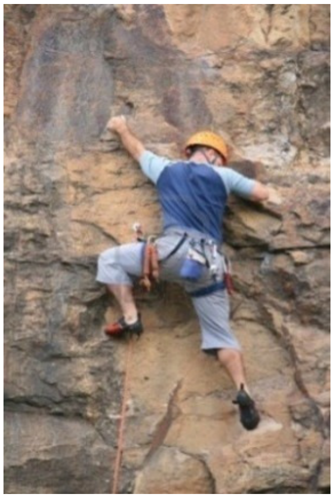


Figure 1.1: Humans possess a superior understanding of scene semantics, including the objects in the scene, their relationships and interactions, e.g., the rock and its color, the human and his body parts, and the rock climbing activity.

1.1 Motivation

Machines need human-like commonsense knowledge for natural interactions. With the advancements in science, autonomous robots are no longer a fantasy. A robot will be expected to understand the world around it and interpret novel scenes. For instance, the robot is expected to interpret a scene of a person doing rock climbing, as humans would; see Figure 1.1. The robot would need to know a variety of semantics in the scene; that the climber is a human and the human has hands, the rock is usually brown and the mountain is larger than the human. To go rock climbing, you must read the route. You might need a rope and water. You are probably adventure loving if you go rock climbing.

Such commonsense knowledge is different from encyclopedic knowledge. Over the last decade, we have seen the rise of large knowledge collections driven by Big

Data on the Web, most notably Wikipedia and online databases. Prominent examples include Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007), and DBpedia (Auer et al., 2007). The strength of these knowledge bases (KBs) is in taxonomic and factual knowledge: named entities grouped into semantic classes and relationships between entities. However, they are ignorant regarding commonsense knowledge because while encyclopedic knowledge concerns instances of classes (e.g., Alain Robert, a professional rock climber), commonsense knowledge concerns classes and general concepts (e.g., rock climber in general).

Commonsense knowledge concerns general concepts but English words are typically polysemous and thus have more than one meaning or *sense*. For example, the concept *plant* can have multiple senses describing *an industrial plant* or *a living organism*. We will denote words in *italics* and senses in `typewriter font`. WordNet (Fellbaum and Miller, 1998) is a lexical database of English words (nouns, adjectives, verbs, adverbs) that distinguishes and enumerates the different senses of a word. These sense enumerations reflect the frequency of usage of a sense in text (McCarthy et al., 2007). We denote a WordNet sense of a word w as $\mathtt{w}_n^s$ where the part of speech tag prefix (`n` for noun, `a` for adjective, `v` for verb, `r` for adverb) appears as subscript, and WordNet sense number s ($1 \leq s \leq 59$) as superscript. WordNet groups these senses into a unique set of synonyms, called a *synset* such that all word senses in a synset express the same meaning. For example, in the following synset: ($\mathtt{plant}_n^1$, $\mathtt{works}_n^8$, $\mathtt{industrial\ plant}_n^1$), all the senses have the same definition or *gloss* (buildings for carrying on industrial labor) [e.g.] they built a large plant to manufacture automobiles.

Commonsense concepts are not limited to WordNet concepts because WordNet is updated infrequently and new concepts or word phrases emerge continuously, e.g., *nuclear power plant*, while WordNet only contains *power plant*. We will call such emerging concepts as *extended phrases* (see Definition 1.1.1).

Definition 1.1.1 - Extended phrase.

A phrase (noun phrase, adjectival phrase, verb phrase) that is not present in WordNet, but whose head-word is present in WordNet as a sense $\mathtt{h}^{s}_{pos}$, is called an extended phrase. Extended phrases are a specialization or sub-class (called, *hyponym* in WordNet) of $\mathtt{h}^{s}_{pos}$. An extended phrase is either:

$$\begin{cases} \mathtt{pos} = n & \text{extended noun phrase, or extended adjectival phrase} \\ \quad , e.g., & \textit{nuclear power plant},\ \textit{thermal power plant} \\ \mathtt{pos} = v & \text{extended verb phrase} \\ \quad , e.g., & \textit{operate a power plant} \end{cases}$$

In the example of the extended phrase *nuclear power plant*, its head-noun *power plant* is present in WordNet ($\mathtt{power\ plant}^{1}_{n}$) and thus, *nuclear power plant* is a specialization of $\mathtt{power\ plant}^{1}_{n}$.

It is crucial to distinguish word senses from words or phrases in order to remove ambiguity. For example, if we observe the following text: "*plant is green*", and encode this information in the form of a *triple* $\langle$*plant* `is` *green*$\rangle$, where a triple holds the left argument (*subject*), right argument (*object*) and a relationship connecting them (`relation`). This triple has two very different interpretations depending on the context: $\langle\mathtt{plant}^{1}_{n}\ \mathtt{hasQuality}\ \mathtt{green}^{3}_{a}\rangle$, and, $\langle\mathtt{plant}^{2}_{n}\ \mathtt{hasColor}\ \mathtt{green}^{1}_{a}\rangle$[1]. These different interpretations can only become clear if the left and right arguments and the relation, connecting them is *disambiguated*, i.e. words are mapped to senses.

Commonsense knowledge relations can be divided into (at least) three kinds of knowledge about the world/ environment:

1. **Properties of objects** in the environment including, for instance, the shape, size, color of an object and the emotion it evokes;

[1] The glosses of these senses are:

$\mathtt{plant}^{1}_{n}$: *industrial plant . . .* ,

$\mathtt{plant}^{2}_{n}$: *a living organism . . .* ,

$\mathtt{green}^{1}_{a}$: *of the color between blue and yellow . . .* ,

$\mathtt{green}^{3}_{a}$: *not harmful to the environment. . . . .*

2. **Relationships between objects** in the environment including, for instance, class hierarchy, part-whole and comparisons;
3. **Interactions between objects** in the environment including, for instance, an activity in which an object participates.

Let us consider the running example about rock climbing to elicit these three kinds of commonsense. The fact that rock is usually brown ⟨rock `hasColor` brown⟩ and a rock climber is adventure-loving ⟨*rock climber* `evokesEmotion` *adventure*⟩ is commonsense on properties. The fact that the climber is a human ⟨*climber* `subClassOf` *human*⟩ and a human has hands ⟨*hand* `physicalPartOf` *human*⟩, and the mountain is larger than a human ⟨*human* `isSmallerThan` *mountain*⟩ is commonsense on relationships. The fact that you must read the route before you go rock climbing ⟨*rock climbing* `hasPrev` *read the route*⟩ and that you might need a rope ⟨*rock climbing* `hasParticipant` *rope*⟩ and water ⟨*rock climbing* `hasParticipant` *water*⟩ is commonsense on interactions.

We can organize this knowledge in a *Knowledge Base (KB)*; a KB stores a collection of facts, typically in a triple format ⟨subject `relation` object⟩. Consider two triples from the KB: ⟨*rock climbing* `hasPrev` *read the route*⟩ and ⟨*rock climbing* `hasPrev` *study the path*⟩. Ideally we should organize the knowledge in such a way that we can tell that these two triples are similar. This entails disambiguating the arguments and the relation. Such a KB that organizes triples semantically is called a *semantically organized KB*.

Goal: Our goal is to automatically construct a large-scale, *semantically organized* KB possessing these three kinds of commonsense. As an input source of information, we have large volumes of multimodal data including text, images and videos. We want to extract the three kinds of commonsense relations from the input sources, filter the noise, and organize this knowledge in a semantically organized KB containing disambiguated relations and arguments (concepts). .

The **objective** of this dissertation is to investigate automated methods for robust acquisition of semantically organized commonsense of the form ⟨`w1 r w2`⟩, where `w1` and `w2` are either a WordNet sense or an extended phrase; and are connected by a *refined* relation r. The types of commonsense we investigate are:

- Properties of objects: Here, `w1` is a noun sense/ extended noun phrase, `w2` is an adjective sense, and r is a refined `hasProperty` relation such as `hasShape`, `hasSize`, `hasColor`, `evokesEmotion`, e.g., ⟨`rock hasColor brown`⟩, ⟨`mountain hasSize huge`⟩.

- Relationships between objects: Here, `w1` is a noun sense/ extended noun phrase, `w2` is a noun sense/ extended noun phrase and r is a refined relation including `comparative` and `part-whole` relations, e.g., ⟨`rock climber isSmallerThan mountain`⟩, ⟨`rock physicalPartOf mountain`⟩.
- Interactions between objects: Here, `w1` is a verb sense/ extended verb phrase, `w2` is a verb sense/ extended verb phrase, or a noun sense/ extended noun phrase and r is a refined relation that characterizes a human activity such as its temporal sequencing, or the involved participants and location, e.g., ⟨`rock climbing hasAgent rock climber`⟩, ⟨`rock climbing hasLocation mountain`⟩, ⟨`rock climbing hasPrev read the route`⟩.

Challenges: Mining commonsense from data is a difficult task. Challenges like input noise arise in mining any kind of knowledge, including encyclopedic knowledge. However, commonsense knowledge extraction has its unique set of challenges:

- Implicit and sparse: Humans do not explicitly express the obvious, e.g., the information that a rock is hard, is possibly only implicitly available.
- Multimodal: Commonsense is spread across textual and visual contents. For example, the information that a rock is brown can be mined directly from images using image-processing techniques.
- Affected by reporting bias: Uncommon facts are often reported disproportionally more than common facts; therefore, frequencies are not an indicator of validity.
- Context dependent: Commonsense knowledge is culture and location specific and thus holds true merely with a certain confidence, e.g., green apples are uncommon in India or that Indians wear white dress in a death funeral as opposed to a wedding.
- Evolving with new concepts: The environment is dynamic and new objects and new ways of interactions continuously coming into existence.

Prior work and its limitations: Commonsense knowledge acquisition has been a long-standing goal in AI and the problem has received a lot of attention. Prior work includes the seminal projects Cyc (Lenat, 1995) and WordNet (Fellbaum and Miller, 1998) that rely on ontologists, linguists and domain experts. As the knowledge is manually compiled, it is high quality but costly (Cyc is an effort of 15 years), small in size (Cyc contains less than a million triples), and not updated (the last version of Cyc is more than a decade old).

There are several automated or semi-automated systems for commonsense acquisition including ConceptNet (Speer and Havasi, 2012), and the work by Tandon et al. (2011) and Lebani and Pianta (2012). ConceptNet is a huge collection of commonsense triples, but the vast majorities are instances of generic relations like `isA`, `conceptuallyRelatedTo`, `partOf`, or `derivedFrom`. The more specific relations like `adjectivePertainsTo` or `usedFor` have only few instances. Tandon et al. (2011) (referred as SR for their Specificity Ranking method) automatically compiled millions of triples of the form ⟨*noun* `relation` *adjective*⟩ by mining N-gram corpora, but the relations are still fairly generic such as `hasA`, `hasProperty`, or `capableOf`. Lebani and Pianta (2012) proposed encoding additional lexical relations for commonsense knowledge into WordNet, but their approach is inherently limited by relying on human input and also focuses on simple relations like `usedFor` and `partOf`.

The very recent work on commonsense acquisition from visual data has been limited in scale due to the processing time and limited in accuracy due to the challenges in automated image processing. NEIL (Chen et al., 2013) analyzes images on the Web to acquire commonsense knowledge relations like `partOf` and visual attributes of concepts like `isVisuallySimilarTo`. However, extracting commonsense from visual content requires automatic and accurate detection of objects, their attributes, poses, and interactions, which cannot be solved robustly. These visual analysis systems do not fully leverage the power of text jointly with the image. A recent system, LEVAN (Divvala et al., 2014), mines commonsense knowledge from images and text jointly. Given a concept (e.g., *hill*), LEVAN trains detectors for a wide variety of actions, interactions and attributes involving the concept (e.g., *hill walking*). LEVAN mines relevant n-grams in text that are associated with the given concept. This enables it to capture intra-concept variance. To avoid training detectors for arbitrary abstract bigrams, LEVAN assumes that only visually salient bigrams will provide any meaningful object detection accuracy.

None of these knowledge resources has *refined* relations like `hasShape`, `hasSize`, `hasTaste`, `evokesEmotion`, or `physicalPartOf`, `memberOf`, `comparatives`, or large-scale knowledge about human `activities`. None has produced large amounts of semantically organized knowledge. Thus, state-of-the-art commonsense KBs still have severe limitations:

- The prominent approaches are not automated and hence costly and limited in scale.
- Semantically different kinds of commonsense relations are conflated into a single generic relation, e.g., `hasProperty`, instead of refined relations like

`hasShape`, `hasSize`, `hasTaste`, `evokesEmotion`, or `part-whole` relation, instead of refined relationships like `physicalPartOf`, `memberOf`, `substanceOf`.

- The arguments of the triples are merely words with ambiguous meaning. There is no distinction between words and their different senses, e.g., ambiguous properties such as *hot* can refer to temperature, taste, or emotion but state-of-the-art approaches would conflate this.

To summarize, prior work cannot address our research objectives because prior work has largely followed small-scale manual approaches. The resulting Commonsense KBs (CKBs) are coarse-grained with ambiguous arguments. Further, prior work has largely been limited to textual data. Acquisition of commonsense knowledge from visual data leveraging both text and visuals is very recent and small-scale only. Table 1.1 positions this dissertation (our resulting CKB is called `WebChild KB`) against related work.

Table 1.1: Positioning the dissertation against related work

CKB	**Triples**	**Arguments**	**Relations**	**Method**	**Source**
Cyc	< 1M	Unambiguous	>100, fine	Curated	-
WordNet	< 10K	Unambiguous	<10, coarse	Curated	-
Verbosity	< 100K	Ambiguous	<100, coarse	Crowdsrc	Images
ConceptNet	< 1M	Ambiguous	<100, coarse	Semi-auto	Text
SR	> 20M	Ambiguous	<100, coarse	Automated	Text
NELL	< 10K	Ambiguous	<10, coarse	Automated	Text
ReVerb	< 10K	Ambiguous	Open, coarse	Automated	Text
NEIL	< 10K	Ambiguous	<10, coarse	Automated	Images
Levan	< 100K	Ambiguous	<10, coarse	Automated	Text, Img
WebChild KB	> 18M	Unambiguous	>1000, fine	Automated	Text, Img

1.2 Contributions

We overcome the limitations of the state-of-the-art and provide new research directions for the construction of commonsense KBs. Our approaches are scalable and automated, and rely on text by building robust methods that can capture implicit signals in text. Our approach can leverage visual data due to the recent advances in image processing, and handle the noise coming from these computer vision systems. We provide robust triple disambiguation methods that simultaneously disambiguate the arguments and classify the triple to a refined relation.

Our overriding approach is to automatically extract triples from Web-scale data using Information Extraction techniques. Starting with these noisy, ambiguous and unorganized triples, our task is to clean, disambiguate and organize them. We propose methods for joint disambiguation of the arguments of the triple and classification of the triple to a refined relation. Finally, we organize these disambiguated triples, resulting in a large-scale commonsense KB. Our methods can make use of data of multiple modalities.

The **contributions** of this dissertation are:

- Bigger commonsense KB: we propose scalable methods that lead to a large-scale, high accuracy (more than 80%) Commonsense KB (WebChild KB) containing hundreds of thousands of concepts and thousands of refined relations between these concepts. There are ca. 18 million triples in WebChild KB.
- Cleaner commonsense KB: we propose disambiguation methods capable of disambiguating both arguments of a triple.
- Richer commonsense KB: our methods classify the triples to a refined relation, thereby not conflating semantically different relations.
- From multimodal sources: our methods leverage both textual and visual contents. We mine knowledge from textual data like Web pages, as well as from visual contents like Flickr images, and movies.
- Using automated techniques: all of our methods are automated with no human intervention. Our methods are scalable, allowing scalability of up to a billion Web pages.

These contributions are reflected in various publications during the course of this doctoral work:

1. AAAI 2016 (Tandon et al., 2016): *Introduces semantic part-whole commonsense*

2. WI 2015 (Chen et al., 2015): *Introduces guided training of word2vec CBOW model with commonsense*

3. CIKM 2015 (Tandon et al., 2015a): *Introduces semantically organized activity commonsense*

4. CMU LTI-SRS Symposium 2015 (Rajagopal and Tandon, 2015): *Proposes the integration of encyclopedic and commonsense knowledge.*

5. ACL 2015 (Shutova et al., 2015): *Introduces visual tags for selectional preferences*

6. CVPR 2015 (Rohrbach et al., 2015): *Introduces audio descriptions as a source of rich visual semantics*

7. WWW 2015 (Tandon et al., 2015b): *Introduces activity commonsense as semantic frames from movie scripts*

8. AAAI 2014 (Tandon et al., 2014b): *Performs Open Information Extraction and semantic organization over 850 million web pages*

9. WSDM 2014 (Tandon et al., 2014a): *Semi-supervised modeling for semantically organized commonsense acquisition*

10. COLING 2012 (Tandon et al., 2012): *Better scoring model of commonsense facts using random walks*

1.3 Outline

This dissertation is organized based on the exploration of the three research questions, as follows:

- Chapter 2 gives background and reviews general related work on commonsense knowledge and its acquisition.
- Chapter 3 discusses the first question about property commonsense.
- Chapters 4, 5 refer to the second question about commonsense on relationships.
- Chapter 6 refers to the third question about activity commonsense.
- Chapter 7 discusses the resulting commonsense KB, called `WebChild KB` and its various use-cases.
- Chapter 8 presents conclusions and suggests new research directions.

2 Background and Related Work

This chapter provides a definition of commonsense knowledge, contrasting it with other types of knowledge most notably encyclopedic knowledge. We discuss an overview of commonsense knowledge bases (KBs). We then discuss the state-of-the-art commonsense KB acquisition methods.

2.1 Commonsense Knowledge

Commonsense knowledge has been vaguely defined as a collection of facts that even a child possesses. The original definition of commonsense knowledge by John McCarthy is that "a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows". While these definitions are informative, they do not highlight the kinds of concepts that make up commonsense knowledge.

More concretely, commonsense knowledge is the knowledge about the generic class of objects in the world rather than the instances of a class. The world is made up of physical objects and abstract concepts. These objects interact in the environment, physically (e.g., hydrogen and oxygen make up water) or abstractly (e.g., some objects and interactions evoke emotions).

Commonsense knowledge is location and culture dependent (Anacleto et al., 2006) and can be opinionated with varying modalities of frequency (Trummer et al., 2015). Thus, commonsense facts can be associated with a context metadata and a confidence score. Definition 2.1.1 provides a broad yet concrete definition of commonsense knowledge. The specific definitions of domain and range depend on the nature of the relation and will be introduced in the appropriate chapters, e.g., Definition 3.1.2 for `hasProperty` relations.

Definition 2.1.1 - Commonsense knowledge.

Commonsense knowledge is a collection of relations r such as `hasProperty`, `hasPart`, `usedFor`, `evokesEmotion`. r is a subset of the cartesian product of the domain $dom(r)$ and $rng(r)$.

The domain $dom(r)$, of r, is the set of concepts comprising of either noun phrases or verbal phrases (depending on the specific nature of r) but excludes instances, e.g., for the relation r=`evokesEmotion`, *singer* is a valid domain element whereas the *instance* `Beyonce Knowles` is invalid.

The $rng(r)$, of r, is the set of concepts comprising of either noun phrases, adjectival phrases or verbal phrases (depending on the specific nature of r) but excludes instances.
Often, commonsense does not hold universally; thus, every commonsense fact is accompanied by a confidence score $0 \leq \Theta \leq 1$ and optional metadata.

2.1.1 Commonsense knowledge vs. encyclopedic knowledge

Another way to understand commonsense knowledge is by contrasting it against encyclopedic knowledge. These two distinct types of knowledge are often conflated incorrectly.

Encyclopedic knowledge embodies facts about instances of classes, e.g., specific person, specific location. These facts span across various dimensions like encyclopedic knowledge of instances: ⟨*Albert Einstein* `isA` *physicist*⟩, relationships across instances: ⟨*Albert Einstein* `wasMarriedTo` *Elsa Einstein*⟩ and interactions or events involving these instances: ⟨*Albert Einstein* `marriedIn` *January 1903*⟩. Thus, the domain of encyclopedic knowledge relations is an instance (or Named Entity), while the range can be a Named Entity, concept or a literal.

Commonsense knowledge embodies facts about classes and concepts (not instances of classes). These facts span across various dimensions like properties of concepts ⟨book `hasProperty` solid⟩, relationships across concepts ⟨hand `partOf` a researcher⟩ and interactions or activities involving these concepts ⟨researcher `publishes` research papers⟩. Thus, the domain of commonsense knowledge relations is a subset of all concepts and extended phrases, while the range is a subset of all concepts and extended phrases.

Encyclopedic relations are functional i.e. $\forall a, b, c : f(a,b) \wedge f(a,c) \implies b = c$ as well as non-functional (the right argument admits a set of values for the same left argument). On the contrary, commonsense relations are nearly always non-functional because they generalize the properties of the instances, e.g., while the encyclopedic fact might be ⟨Nicolas Sarkozy `hasHeight` short⟩ or ⟨Abraham Lincoln `hasHeight` tall⟩; the commonsense counterpart would be ⟨president `canHaveHeight` short,tall⟩ thus making it set-valued. In general, as one moves towards the root of the concept hierarchy, the concepts become more generic and their attributes can admit many possible values while the nodes towards the leaf are more specific, with specific attributes.

Typically, encyclopedic knowledge can be obtained explicitly from Web text like Wikipedia and news corpora. However, commonsense knowledge is present across multimodal documents, including text, images, and video. Table 2.1 summarizes the key differences between commonsense and encyclopedic knowledge.

Table 2.1: Contrasting commonsense and encyclopedic knowledge

	domain	**relation type**	**source**
Encyclopedic	instances	majority functional	text
Commonsense	concepts	majority non-functional	multimodal

2.2 Commonsense Knowledge Bases (KBs)

In order to build intelligent applications, knowledge must be represented and stored in a formal machine-readable representation. Such a representation of knowledge enables the intelligent applications to reason, i.e. find implicit consequences of its explicitly represented knowledge. We discuss the different types of representations of knowledge followed by a discussion on storing this knowledge and pre-requisites for reasoning upon it.

2.2.1 Commonsense knowledge representation

Knowledge representation is concerned with how knowledge can be represented symbolically in a formalized machine-readable format and reasoned. The primary challenge in commonsense knowledge representation is the tradeoff between expressive power, compactness and efficiency. Due to this trade off, commonsense knowledge representation has received a lot of attention in the AI community even prior to the 90s, see Dahlgren et al. (1989).

Knowledge representation approaches are broadly classified into two categories. The first category is based on the hypothesis that predicate calculus can unambiguously capture commonsense facts. Reasoning over these predicates amounts to verifying logical consequences. The second category is derived from human memory and human execution of tasks like puzzle solving. Semantic networks and frames are two popular representations. Semantic networks are based on network shaped cognitive structures while a Frame is an abstract description of a category (e.g., an event), organized in a hierarchy with the slots containing data values for the semantics. Reasoning over these is performed using methods that process hierarchical structures.

Both representations have pros and cons but it is possible to arrive at the best of both worlds. The first category leads to precise reasoning but can be very complex to design. The second category lacks precise semantic characterization but given their human-centered origins, they are more appealing and effective from a practical viewpoint. Hayes (1979) showed that frames could be given semantics by relying on first-order logic. Consequently, this research led to the development of Description Logic (DL) based representation. A characteristic feature of DL is their ability to represent relationships beyond `isA`, these relations are called roles (Baader et al., 2008). DL models concepts, roles and individuals, and their relationships. The fundamental modeling concept of a DL is the axiom - a logical statement relating roles and concepts.

2.2.2 Storing commonsense knowledge

Knowledge base: A knowledge base (KB) is defined as a collection of facts, typically in a triple format ⟨subject `predicate` object⟩, representable as a graph. Commonsense knowledge can be stored and indexed in a commonsense KB in this format. A triple in such a commonsense KB holds the left argument (*subject*), right argument (*object*) and a relationship connecting them (`predicate`). Optionally, metadata like triple confidence and context can be stored using reification techniques that assign a unique ID to a triple and use the triple as the left argument and `hasConfidence` as the relation and the confidence value as the right argument.

There are two different classifications of KBs, schema-based and schema-free, based on whether the KB has a restricted and canonicalized, or, unrestricted and not canonicalized relations. In schema-based KBs, relations (and typically the arguments) are canonicalized and restricted, and hence uniquely identifiable. In schema-free KBs, relations (and typically arguments) are not canonicalized and not restricted. There has been recent work on canonicalizing schema-free KBs (Galárraga et al., 2014). Schema-based KBs allow reasoning over the relations (as they are canonicalized) while reasoning is typically not well-defined in schema-free KBs.

Schema-free KBs: Schema-free approaches are well suited for large coverage but lack semantics. It is very difficult to reason over schema-free KBs due to the ambiguity in the meaning of the concepts and relations. There are well-defined reasoning algorithms and architectures for commonsense knowledge. To leverage these reasoners, it is essential to have a schema-based commonsense KB.

Schema-based KBs: The ideal representation for a (schema-based) commonsense KB is an object model (often called an ontology in the AI community) with concepts and relations. The predicates or relations are canonicalized, and the left and right arguments of the triples are mapped to an ontology. Such an ontology enables reasoning over the commonsense KB (Davis and Marcus, 2015).

Commonsense KBs can be connected to ontologies containing generic concepts that describe very general concepts that are same across all domains. Such an ontology is also called an upper level ontology. There are several upper level ontologies (refer Mascardi et al. (2007)), most notably Cyc ontology (300K concepts) with 12K concepts mapped to WordNet, and, SUMO (20K terms and 60K axioms) with all the terms mapped to WordNet.

Reasoning over schema-based commonsense KBs: For logical reasoning, `isA` relation, organizing concepts in a hierarchy, is very valuable. WordNet is a candidate as a commonsense ontology. WordNet is an unparalleled lexical database, the most used lexical resource in computational linguistics. WordNet is connected to all prominent upper level ontologies, making it a preferred choice as an ontology.

Is WordNet truly an ontology? Although WordNet was not intended to be an ontology initially, it was eventually transformed as the organization of nouns in WordNet bore many similarities to an ontology (Miller and Hristea, 2006). Every noun begins with a single unique beginner: `entity`. In a reasonable ontology, all terms are expected to conform to the membership relation of set theory and would not contain instances. The confounding of classes and instances in WordNet was resolved manually in WordNet version 2.1. This made WordNet a suitable ontology that can be used for commonsense reasoning.

WordNet is a closed vocabulary lexical database with infrequent updates and thus limited concepts. Nevertheless, a large majority of upper level ontology concepts are already present in WordNet. Although WordNet is rich in `isA` relations, it contains a very small number of sparsely populated commonsense relations so it cannot be a substitute of a commonsense KB but could function as a reasonable ontology for the commonsense KB.

2.3 Commonsense KB Construction

The larger task of commonsense KB construction involves commonsense KB acquisition, completion and reasoning. Commonsense KB acquisition task involves populating a schema-free or schema-based commonsense KB with commonsense knowledge triples using curated, semi-automated or automated techniques over unimodal or multimodal data. A usual follow-up task after commonsense KB acquisition is commonsense KB completion. The populated commonsense KB can then be used to mine commonsense rules using inductive logic programming (Blanco et al., 2011).

We can classify commonsense KB acquisition methods into four classes, irrespective of the modality of the input data or whether the commonsense KB is schema-free or schema-based:

(i) In curated approaches, commonsense knowledge triples are created manually by experts or even non-experts using knowledge authoring tools.

(ii) In collaborative approaches, triples are created manually by an open group of volunteers. Typically, these are game-based acquisition methods.

(iii) In semi-automated approaches, either the method for extraction is semi-automated or an automated extraction method is used over a fixed-schema dataset.

(iv) In automated approaches, triples are extracted automatically from unstructured text via machine learning and natural language processing techniques.

2.3.1 Curated

The goal of curated commonsense KB is to codify millions of pieces of commonsense knowledge in machine-readable form. Typically, the assertions in a curated commonsense KB do not have a confidence score.

Cyc: The pioneering work in this direction is Cyc. The Cyc project started in early 1980s up until 2000s. Cyc contains a KB (Cyc KB), and a collection of Cyc inference engines. Cyc KB is coded in a formal predicate calculus like syntax language (CycL). Cyc assumes that each assertion should be considered true in only certain contexts. Thus, all assertions are organized into more than 20,000 micro-theories whose assertions share the same set of assumptions.

The acquisition process of Cyc is that hundreds of thousands of facts and rules have been formally codified by ontologists skilled in CycL. Additionally, Cyc contains domain specific knowledge, e.g., defense domain, for which Cyc relies on subject matter experts. According to Cyc's documentation, the number of facts entered by a subject matter expert is about 25 facts per hour. To overcome the size limitations, we will later see an extension of Cyc that uses a game-based interface and Wikipedia to automatically acquire more knowledge.

While Cyc has been ahead of its times, it is not clear whether the 15 years of curated knowledge acquisition process was successful. It is not clear what fraction of commonsense knowledge (concepts and not instances) does Cyc contain, and what fraction deals with specialized applications such as defense, terrorism or medical records. There have been conflicting reports about the usability of Cyc from practitioners. Reports suggest that a large collection of Cyc has usability problems including problems in understandability and portability to other systems. Conesa et al. (2008) further report that the Microtheory Taxonomy (MT) in ResearchCyc is very difficult to use due to its organization. Some reasons provided by them are:

- “There are over 20,000 MTs in Cyc with the taxonomical structure of MTs being as deep as 50 levels in some domains.
- Many redundant sub-type relationships make it difficult to determine its taxonomical structure.
- Some MTs are very sparse but it is very difficult to discard them.
- Some MTs do not follow a standard representation of knowledge.”

WordNet: A much smaller and less formally represented commonsense KB is WordNet. WordNet is aligned to several external knowledge sources due to its popularity. Cyc is partially aligned to WordNet while other ontologies like SUMO are mapped to WordNet. WordNet is carefully handcrafted, containing more than 155,000 words organized in over 117,000 synsets, e.g., biologist, scientist, animal, dog.

The WordNet synsets are connected by different relations, primarily either linguistic or commonsense relationships. These relations include `isA` i.e. *(hyper/hypo)nyms* that connect generic nouns and verbs (*hyper* to more specific respective nouns and verbs (*hypo*), `part-whole` relations `substance meronym`, `member meronym`, and `part meronym`. Nouns and adjective synsets are sparsely connected by `attribute` relation.

WordNet is limited in the coverage of concepts, number of relations and number of assertions. Concepts are ever emerging, e.g., *hybrid cars*, but WordNet would not keep up with such emergence. The number of relations in WordNet is limited, and the small set of relations present, are very sparse. For instance, there are only 1200 `part-whole` relation triples.

VerbNet: VerbNet (Kipper et al., 2006) is a manually curated linguistic resource for English verbs. It provides complimentary linguistic and potentially commonsense semantics to WordNet verbs. For each verb class, VerbNet lists relevant thematic roles, semantic restrictions on the arguments, and syntactic frames. For example, for the verb *shoot*, VerbNet lists multiple candidate senses, and for the first of these, $\texttt{shoot}^1_{vn}$, it provides, among others, the following syntactic frame:

Agent.animate V Patient.animate PP Instrument.solid

This would match "*He shot the man with a gun*". Here, several roles are accompanied by a semantic constraint, known as a *selectional restriction.* A selectional restriction such as `animate` for the patient requires that this patient be a living being when used in the given syntactic frame.

ImageNet (visual contents based): ImageNet (Deng et al., 2009) is a manually curated image database organized according to the WordNet noun synset hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images. ImageNet has an average of over five hundred images per WordNet noun synset. ImageNet is manually curated.

ImageNet can be directly used to populate the commonsense relation `appearsLike`. This is a useful relation in multimodal analysis for applications like robotic engines (Saxena et al., 2014).

Manually curated approaches produce a very high quality KB. However, they can be very costly especially if the task requires experts. These approaches tend to have a low coverage because humans are known to be bad to recalling facts, especially when they are as obvious as commonsense. These approaches take a very long time to develop, e.g., Cyc could gather only hundreds of thousands of facts over more than 15 years.

2.3.2 Collaborative

In collaborative approaches, users individually or collectively play games and produce useful computation as a side effect (Von Ahn, 2006), including mining generic or domain-specific commonsense knowledge. In this sub-section, we review a non-game based interface (*OMCS*) and two popular games: a two player game *Verbosity* and a single player game *Virtual Pet*, all aiming to collect commonsense.

Collaborative approaches offer several advantages. Collaborative approaches have a faster acquisition rate and higher coverage than curated approaches. A specialized audience is not required for commonsense elicitation. The quality of aggregated assertions is very high because the assertions from different players have the ensemble effect.

Verbosity: Verbosity considers two players: a narrator and a guesser. Given a word (concept), the narrator can offer some clues (short assertion phrases) to help the guesser guess the secret word. For example, if the secret word is *laptop*, the narrator might prompt the guesser: "*it has a keyboard*". Such an acquisition typically brings out the most salient aspects of a concept. Verbosity also allows a single player to play the game with a "*bot*" partner. The bot uses the collected data when it is in the role of an automated narrator.

Virtual pet game: In the Virtual Pet Game (Kuo et al., 2009), players teach their pets simple facts in order to raise the intelligence of their virtual pets. Virtual Pets Game outperforms non-game based acquisition in collection speed and quantity. That is, fewer contributors used lesser time to collect more commonsense knowledge.

OMCS: The Open Mind Common Sense (OMCS) project is designed to collect commonsense knowledge statements in English with supplementary projects providing extensions to several other languages. OMCS contains nearly a million statements from more than 15000 volunteers that led to more than 100K assertions. The strength of the project lies in the ensemble effect of the statements, and the challenge is user engagement.

OMCS uses a carefully designed interface to collect statements from the users. In the first version, there were 25 different commonsense settings, each setting with its own user interface, where the volunteer is presented a short story, e.g., "*Bob had cold and Bob went to the doctor*", encouraging a volunteer to enter commonsense knowledge like "*Bob was sick*", "*the doctor prescribed him medicine*".

In another version, volunteers can select a concept and fill predefined templates associated with the concept. For example, given the template "*can be used to*", the volunteer could fill the left and right slots with "*a pen*" and "*write*" respectively.

Visual contents based: Visual Genome (Krishna et al., 2016) is a recent, crowdsourced KB that connects structured image concepts to text. A scene graph models the commonsense assertions from an image in a graph connecting concepts to attributes, their spatial relations and their interactions. The graph representation has an advantage over natural language because it abstracts object interactions of arbitrary complexity.

Collaborative approaches have greater efficiency of collecting commonsense statements than manually curated approaches. Like the curated approaches, the relations are pre-defined, leading to a schema-based KB. However, collaborative approaches face the challenges of user engagement and maintaining a large user base for redundancy, especially when the task (collecting commonsense) is so simple and underwhelming. Further, the statements tend to be noisy and are not canonicalized.

2.3.3 Semi-automated

In semi-automated approaches, we review ConceptNet, a semantic network representation of the commonsense knowledge collected primarily from OMCS projects and some other external resources like Wikipedia.

Semi-automated approaches have the advantage that they achieve high precision due to manual intervention at the beginning. Semi-automated approaches have a faster acquisition rate and higher coverage than curated approaches but with substantially lower precision.

ConceptNet: The pioneering semi-automated commonsense acquisition project is ConceptNet (Liu and Singh, 2004; Lieberman et al., 2004; Havasi et al., 2007; Speer and Havasi, 2012). It is one of the largest repositories of commonsense assertions, covering commonsense relations including properties of objects, e.g., `hasProperty`; their relationships, e.g., `madeOf`; as well as their interactions, e.g., `motivatedByGoal`. ConceptNet contains 1.6 million assertions over 30,000 concepts. In the ConceptNet graph, the nodes denote concepts and the edges (multi graph) denote commonsense relations between concepts. Syntactically, the concepts are either noun phrases, verbs, prepositional phrases, or adjectival phrases.

ConceptNet is constructed in three stages. (i) Triple extraction using semi-structured OMCS via lexico-syntactic patterns and argument type-constraints. (ii) Normalization of arguments by stripping determiners and modals, e.g., *falling off a bike* to *fall off bike*. (iii) Handcrafted relaxation rules to increase coverage, e.g., IsA (apple, red fruit) imply `propertyOf` (apple, red).

ConceptNet supports several contextual commonsense reasoning functionalities. This includes three node-level functionalities, including contextual neighbors, analogy, and projection. In addition, there are four document-level functionalities: topic-summary, disambiguation and classification, concept identification, and emotion sensing.

ConceptNet includes a rare functionality: explicit negative assertions. Every assertion has a confidence/ reliability score (multiple agreements) and a parameter of polarity. The polarity parameter has a value of either 1 or -1, where negative values indicates a negative statement, e.g., from the OMCS statement "*people do not want to be hurt*" to *people desires be hurt*, polarity -1.

The distinction between commonsense and encyclopedic knowledge is unclear in ConceptNet-5 (latest version of ConceptNet). The `partOf` relation is additionally populated with Wikipedia and DBpedia's geo-knowledge, e.g., ⟨*Berlin* `partOf` *Deutschland*⟩. ConceptNet-5 is expanded to several languages, in addition to English.

Other approaches include Lebani and Pianta (2012), who studied relations like `hasSize`, `hasShape`, etc., assigning their instances to word senses. However, it solely relied on human elicitation for 50 nouns, and is not suited for automation at large scale.

Compared to WordNet and Cyc, ConceptNet has the advantage of having a simpler, less formal and more pragmatic contextual reasoning because ConceptNet deals with knowledge that may not hold universally. On the other hand, WordNet is optimized for lexical categorization and taxonomic word similarity, Cyc is designed for formalized logical inference.

2.3.4 Semi-automated pattern-based

Another line of semi-automated approaches leverages handcrafted patterns over an open corpus, providing more automation and higher coverage. We review the work by Pasca (2014b) that employs semi-automated pattern-based approach to extract commonsense. Other related methods include DART (Clark and Harrison, 2009). These methods differ in the input corpora, manually specified extraction patterns and ranking, typically based on an information theoretic measure like mutual information.

Pasca (2014b) extract commonsense knowledge from Google query log using manually specified targeted patterns. For example, their pattern: *why [is|was|were] [a|an|the|nothing] subj rel+obj* matches the query *why are (cars)subj (made of steel)rel+obj*.

As the patterns are fairly generic and the extractions are noisy, their scoring function is an inverted document frequency like measure: $score(F, C) = LowBound(Wilson(N^+, N))$, where the fact is F, and C is a class (subject). The score measures specificity as the lower bound of the Wilson score interval (Brown et al., 2001) where N^+ is the number of supporting queries, and N^- is the number of queries where the fact is extracted for a different class (subject).

While most of these methods acquire coarse-grained relationships, there is little prior work on acquiring fine-grained commonsense relations. Almuhareb and Poesio (2004) proposed patterns like "*<object> is a/the <attribute> of <subject>*", e.g., "*brown is a color of dogs*" to find more specific properties, but even on the Web this method yields very low recall (Baroni and Zamparelli, 2010) as commonsense relations are not as explicit in text.

Pattern-based methods provide higher coverage but lower precision than curated methods as the input corpus is orders of magnitude larger and automated methods inevitably attract noise. Commonsense relations are not as explicit in text, and it is difficult to enumerate all the potential subtle patterns by hand. To overcome these limitations, typically automated methods are employed, that we discuss next.

2.3.5 Automated

In automated techniques, the resulting commonsense KBs are either schema-free or schema-based. Schema-free techniques focus on recall, while schema-based restrict on a smaller set with higher precision. Schema-free commonsense KBs have ambiguous interpretations of triple arguments and the relations are not canonicalized while Schema-based commonsense KBs have canonicalized relations and may have disambiguated arguments.

2.3.5.1 Schema-free

The most popular schema-free methods to extract knowledge are Open Information Extraction (OpenIE) based, including shallow syntactic parsing based (e.g., TextRunner (Yates et al., 2007), ReVerb (Fader et al., 2011a), R2A2 (Etzioni et al., 2011)) or deep syntactic parsing based systems (e.g., ClausIE (Corro and Gemulla, 2013), Weltmodell (Akbik and Michael, 2014), KNext (Schubert, 2002)). These systems do not discriminate between encyclopedic and commonsense knowledge. They do not have a notion of entities and concepts primarily because the arguments are ambiguous. The relations in the extracted triples are phrases and hence the relations are open. We discuss these methods assuming that it is possible to tell apart encyclopedic and commonsense triples.

Shallow syntactic parsing methods: TextRunner extracts all possible ⟨noun phrase `verb phrase` noun phrase⟩ triples from text using a three-stage architecture. TextRunner uses a chunker to identify noun phrases and subsequently identifies normalized verb phrase relations between noun phrases, heuristically discarding non-essential modifiers. A Naive Bayes classifier, trained over a small subset of extractions that satisfy handcrafted heuristic constraints, is used to retain meaningful extractions. Finally, a redundancy-based assessor evaluates each retained extraction using a probabilistic model of redundancy.

ReVerb is a follow-up architecture to TextRunner, providing robustness over the relational phrase expression errors. A syntactic constraint on relational phrases avoids meaningless relational phrases by enforcing that the relational phrase is contiguous, begins with a verb, and ends with a preposition. A lexical constraint on relational phrases is then introduced to avoid rare and meaningless relational phrases by thresholding on the number of distinct argument support. ReVerb searches for the longest sequence of words around a verb that satisfies the syntactic and lexical constraint. It then finds the nearest noun phrases to the left or right of each such relational phrase. Finally, a logistic regression classifier, trained on five features based on aforementioned heuristic constraints, estimates confidence score for the extraction.

R2A2 uses multiple supervised classifiers to identify the arguments that go beyond just noun phrases. The supervised classifiers detect the left bound and the right bound of each argument. The classifiers use several heuristic features, e.g., whether the second argument was followed by an independent clause or verb phrase. R2A2 is the best-performing shallow syntactic parsing based OpenIE extractor.

Deep syntactic parsing methods: Among the deep syntactic parsing based OpenIE systems, we discuss ClausIE, Weltmodell and KNext. ClausIE post-processes the dependency parse output to find all possible clauses in a sentence that respect the seven rules to construct clauses in an English sentence. The quality of ClausIE is bounded by the accuracy of the parser. It is significantly slower than the shallow parsing systems due to dependence on the parser but has higher quality and higher coverage including indirect triples, e.g., "*the ball and the wheel are round*" to ⟨the ball `is` round⟩ and ⟨the wheel `is` round⟩.

Weltmodell applies the dependency output based OpenIE system by Akbik and Löser (2012) over Google syntactic books n-grams. Using heuristic rules over typed dependencies, Weltmodell collect subjects, passive subjects, particles, direct and prepositional objects of the verb. The confidence on the extracted facts is computed using PMI. Weltmodell does not discriminate between classes and named entities as arguments (an important distinction for commonsense knowledge), and is limited to single word arguments.

While ClausIE and Weltmodell can elicit explicit assertions, KNext has focused on implicit general possibilistic propositions from parse trees. General implies the open relations and possibilistic implies that the assertions are possible. For example, given the sentence "*the dog got into the car through the open window*", they can infer that "*it is possible for a dog to enter a car*", "*cars probably have windows*", and "*windows can be open*". KNext starts with general phrase structure patterns to match the parse tree in bottom-up fashion. For each successfully matched sub-tree, the system first abstracts the interpretations of each essential constituent of it, e.g., "*an open window at the rare end of the car*" would be abstracted to "*a window*". Subsequently, compositional interpretive rules help combine all abstracted interpretations and finally derive a general possibilistic proposition.

The OpenIE systems do not discriminate between encyclopedic and commonsense knowledge. This is partially because the arguments and relations are not canonicalized. These systems are typically not designed to construct and organize a commonsense KB (or even a KB), rather their goal is to acquire triples for a use-case like question answering. The shallow syntactic parsing methods typically have low precision while deep syntactic parsing methods do not scale.

2.3.5.2 Schema-based

Schema-based methods populate a predefined relation. The schema-based methods to extract commonsense knowledge are either pattern-based or reasoning/learning-based. Pattern-based methods start with a small set of seeds to induce patterns that are used to extract more facts. While pattern-based methods provide a high recall, their precision is rather low due to the semantic drift and unreliability of patterns and facts. Learning and reasoning-based methods perform joint reasoning over the pattern-based methods style extraction.

Pattern-based methods: Pattern-based methods are based on the pattern-fact duality paradigm (Brin, 1998) that good seed facts lead to good patterns for a relation, and good patterns help to extract good facts. Pattern ranking and fact ranking are the two important ingredients in these methods. Typically, corpus based statistics is used to model these scoring functions. The choice of seeds also plays a role in the quality of the extracted facts (see Tandon (2011) for an overview).

While pattern-based methods have been very popular in encyclopedic knowledge harvesting, Tandon et al. (2011) extend ConceptNet using a pattern-based method. They automatically compile millions of triples of the form ⟨ *noun relation adjective* ⟩ by mining n-gram corpora. Large n-gram corpora are a good source for Information Extraction (Tandon and De Melo, 2010) despite a small context of usually maximum five words.

Their system requires a small number of high-quality seed assertions from ConceptNet to induce the extraction patterns, with a scoring model that rewards patterns with high seed support and high specificity. They limit the iterations to only one cycle from seed assertions to patterns and from patterns to new assertions because pattern-based methods are known to be prone to semantic drift. While they gathered millions of commonsense knowledge assertions, the relation arguments were surface forms and the relations were fairly generic such as `hasA`, `hasProperty`, or `capableOf`, and the precision was rather low.

Pattern-based methods suffer from low precision and several extensions are proposed, e.g., Tandon et al. (2012). The sources of these errors are either the input source (e.g., noisy Web collections) or the patterns that can be too generic for commonsense relations like `hasProperty`, e.g., "*X is Y*".

Reasoning-based methods: While Pattern-based methods elicit the explicit or subtly implicit knowledge in text; reasoning-based methods can bring out the implicit assertions. In these methods, inductive learning helps to generalize and eventually enlarge the KB by generating rules from some existing facts to generalize to more knowledge. These methods also use rules to prune out potentially false assertions that violate the constraints. For example, if we add a constraint that "*a fruit can have only one color*", and so far we accumulate several colors of a fruit, then, we may be able to eliminate the false hypotheses by means of constraint violation.

NELL (Carlson et al., 2010) is a learning agent whose task is to acquire knowledge (mostly encyclopedic but also some commonsense knowledge like `part whole` relations) from the Web in a continuous manner. NELL takes an initial ontology as input that defines categories and their binary relations, with some seed categories and relations. NELL learns extraction/inference models and uses the models to extract/infer new knowledge by expanding the domain and range of these relations automatically from the Web and inferring new relations among them to update the initial ontology. Then, a coupled morphological classifier learns a set of binary L2-regularized logistic regression models per category or relation. Beliefs from the KB are used as training instances, and mutual exclusion relationships are used to identify negative instances. Finally, a rule learner learns a set of probabilistic Horn clauses, which are used to infer new relation instances from other already-known relation instances.

While NELL has acquired large amounts of knowledge (and some commonsense knowledge), the coverage of the system is rather low. Moreover, these systems are computationally very expensive and thus do not scale to the Web.

Learning-based methods: Parallel to pattern-based and reasoning-based methods, learning-based methods are used to acquire more knowledge, starting from a smaller set of known positive and negative examples. Statistical relation learning is a family of algorithms used to acquire more knowledge and validate existing knowledge. These methods associate assertions based on their similarity (either in the latent space or in a graph) and propagate similar labels to similar assertions. While these methods consider that the correctness of all triples is conditionally independent given the graph or the latent space, another member of this family of algorithms, namely Markov Logic Network (Richardson and Domingos, 2006) drops the independence assumption. Given a set of constraints, the Markov Logic framework optimizes for a set of assignments such that the joint assignment satisfies the constraints.

AnalogySpace (Speer et al., 2008) can generate the analogical closure of a KB through dimensionality reduction. Each concept subject S in a triple ⟨S `P` O⟩ is viewed as a vector over $P + O$. The vector is full of noise as the triples are extracted using a semi-automated method. Thus, AnalogySpace reduces the dimension of the vectors and then compares these vectors to infer more knowledge. Consider an example, suppose AnalogySpace knows that a *newspaper* has information while it does not know that anything about a *magazine*, then, given the proximity of *newspaper* and a *magazine* in the dimensionality reduced space, it can infer that a *magazine* also contains *information*.

The learning-based methods are a powerful tool, but can be computationally expensive for very large graphs. Efficient optimization algorithms are an active area of research to increase scalability.

KB verification: Directly extracting commonsense knowledge from vision poses challenging problems in vision like object and attribute detection. Instead of extracting commonsense from images, visual verification of the knowledge in the KB might be more robust. That is, commonsense may be gathered from a high-level semantic understanding of a visual scene, and a low-level pixel information can be avoided.

The task of verifying relation phrases (Vedantam et al., 2015) is to validate an existing acquired assertion by analyzing the frequency of its occurrence in text. For factual knowledge, redundancy is a very reasonable assumption. Since commonsense knowledge is oftentimes implicit in text, it is not clear whether frequency based textual verification is robust. Many high frequency relations occur in text but are not true in the real world, e.g., ⟨*pelican* `pierce` *breast*⟩. Conversely, many relations occur in text with low frequency but are true in the real world, e.g., ⟨*chimpanzee* `eats` *ice-cream*⟩. However, such facts are evident in images and visuals.

Given the recent advancement in analyzing images, new methods extract commonsense knowledge from images (Sadeghi et al., 2015) directly. However, these methods suffer from coverage issues as images only contain visual attributes and are less robust due to the noise that comes along the vision analysis systems. Vedantam et al. (2015) study the plausibility of a commonsense assertion using visual cues and verifying a commonsense triple from image. They predict the plausibility of interactions or relations between a pair of concepts by grounding commonsense assertions in the visual space and evaluating the similarity between assertions using the visual features. This is an active ongoing research area.

2.3.5.3 Visual contents based methods

Commonsense knowledge finds applications in the vision community thereby instilling interest to acquire it. Image recognition benefits by modeling commonsense context in images. Commonsense knowledge also benefits action classification systems for tasks like zero-shot affordance for human-object interactions, i.e. whether a given activity/ action can be performed by an object. Typically, external knowledge or handcrafted commonsense KBs are used for this purpose (Zhu et al., 2014). However, commonsense is also implicitly present in images prompting recent progress to mine commonsense from images.

Recent systems mine commonsense knowledge either directly from images or jointly with text. Given a concept (e.g., *hill*), LEVAN (Divvala et al., 2014) trains detectors for a wide variety of actions, interactions and attributes involving the concept (e.g., hill walking). LEVAN mines relevant n-grams in text that are associated with the given concept, thereby capturing intra-concept variance. To avoid training detectors for visually non-salient activity concepts like abstract bigrams, LEVAN assumes that only visually salient bigrams will provide any meaningful object detection accuracy. In another line of work, Johnson et al. (2015) build a scene graph representation for image retrieval which models attribute and object relations. Their system is trained on mechanical turk annotated scene graphs grounded to images. In an image, the system can elicit commonsense about the concepts, their attributes and interactions.

NEIL (Chen et al., 2013) analyzes images of the web to acquire commonsense knowledge relations like `partOf` and visual attributes of concepts using object and scene detectors to infer an object's visual attributes (color, shape) and `partOf` relationships. To start, NEIL queries Google image search with some seeds (surface forms of concepts, e.g., *jaguar* or scenes like *parking lot*). The results can be noisy, and thus NEIL first clusters the image results based on their visual appearance and then trains object/scene detectors. After training these detectors, at test time NEIL detects the objects in an image and records the visual attributes, e.g., ⟨*object* `hascolor` *yellow*⟩. Thus, it infers object-attribute relations and scene-attribute relations. Further, NEIL infers ⟨*tail* `partOf` *jaguar*⟩ and ⟨*jaguar* `found near` *tree*⟩ using bounding box techniques.

Lin and Parikh (2015) acquire visual commonsense from images and use it to answer textual fill-in-the-blank and visual paraphrasing questions. They imagine a scene as the underlying context and model visual commonsense in the context of a scene.

Extracting commonsense from visual content requires automatic and accurate detection of objects, their attributes, poses, and interactions. However, these are challenging, not completely solved problems in computer vision. Oftentimes, the visual analysis systems do not fully leverage the power of text jointly with the image, e.g., NEIL does not leverage the surrounding text that could prove useful.

In summary, we reviewed several commonsense KB construction methods and while each has its own novelty, none of them is complete, semantically organized, multimodal and automated. While curated KBs are costly and small, crowdsourced KBs are small and difficult to drive. Semi-automated approaches can scale well but there is not enough structured data and manually specified extraction patterns attract extraction noise. Automated approaches are typically not semantically organized, noisy and derived from either text or images but not multimodal.

2.4 Applications of Commonsense KBs

Commonsense knowledge finds applications across a variety of domains (Lieberman et al., 2004). We provide a brief overview of applications in NLP, computer vision and robotics that leverage commonsense knowledge.

NLP: There has been phenomenal progress across various disciplines of NLP. Natural Language Understanding (NLU) is a growing field within this landscape (Winograd, 1972). Bar-Hillel (1960), in as early as 1960, outlined the importance of commonsense knowledge for NLP, especially disambiguation.

While distributional similarity provides a sparse, noisy neighborhood that has been useful for query expansion, commonsense knowledge provides low dimensional, high quality dense vectors. Expansion of concepts using relations like `isA`, `part-whole`, `hasProperty` etc. has been shown to outperform other query expansion methods (Hsu et al., 2006).

Another line of commonsense knowledge application has been in event prediction by gathering commonsense from manually written procedural scripts (Regneri et al., 2010).

Great advancements in NLP have been made through word vector statistics and linguistic analysis. However, more intelligent NLP systems would require commonsense knowledge for more informed decision-making capabilities, e.g., statistical translation engines make silly mistakes such as the English translation of “Das Bier bestellte der Vater” (“The father ordered the beer”) to “The beer ordered the father”. If such systems are equipped with commonsense knowledge about selectional preferences that a person can order beer and not vice-versa, such mistakes are avoidable.

Recently, there have been prominent proposals for alternatives to the Turing Test, such as Winograd Schema Challenges (Levesque et al., 2011) and multimodal comprehension tests (Venugopalan et al., 2014). These require large-scale commonsense knowledge and commonsense rules.

Computer vision: Recent breakthroughs in computer vision and NLP have now led to systems that are able to interpret images and automatically generate image captions. We show how commonsense knowledge can contribute to solving the building blocks in such tasks.

An object detector can detect, demarcate and classify objects in an image. These detectors require a concept taxonomy of object classes to enable more informed hierarchical decision making (Deng et al., 2009). Such an information can be derived from a commonsense KB. Even the limited `partOf` knowledge from WordNet has helped improve object detection (Rohrbach et al., 2011). For example, having the knowledge that ⟨*wheel* `partOf` *bike*⟩, provides better estimates for bounding boxes of a *bike*.

There has been a lot of interest in scene understanding (Xiao et al., 2010) and activity understanding (Kim et al., 2010). Commonsense knowledge has been used to train the activity detectors with richer commonsense context vectors.

Current Vision systems can incorporate a limited amount of commonsense while trying to learn more automatically from images. However, these systems require large amounts of commonsense background knowledge for bootstrapping, e.g., for activity detection, we need a taxonomy of *visual* activities with their semantic contexts. Emotion analysis in abstract scenes remains an elusive problem that requires large-scale commonsense knowledge.

Robotics: Autonomous robots working in an uncontrolled environment require commonsense knowledge and acceptable norms about the environment. Robotic tasks typically include perception, planning and control.

Consider an example where a home robot is asked to perform a common task like "*Bring me cold coffee*". Solving such problems has been a long-standing goal in robotics involving a multitude of information. A robot would require spatial knowledge to prune the search space (e.g., avoiding a washroom for this task), along with other capabilities like navigation.

We only focus on the commonsense knowledge to enable this task. A modern robotic engine like in Stanford's Robo Brain project (Saxena et al., 2014) contains handcrafted commonsense knowledge about the environment including the properties of the object in the environment, their relationship with respect to each other, and their interaction semantics. When the robot is asked to "*bring cold coffee*", its engine must translate the instruction into the perceived state of the environment: (i) models for object detectors for cold coffee, and a refrigerator (ii) knowing that cold coffee can be kept on a table or inside a refrigerator, and a refrigerator or table can be found inside a kitchen, (iii) knowledge that coffee can be poured in a cup. Other types of knowledge required is that a cup can be grasped in certain ways, and needs to be kept upright, and that the pouring trajectory should obey user preferences of moving slowly to pour.

Current robotic engines handcraft the knowledge while automatically learning the visual orientations. A large-scale commonsense KB would further propel more intelligent robotic engines.

3 Commonsense on Object Properties

In this chapter, we investigate the first category of commonsense relations, i.e. commonsense on object properties. We present the methods for one instance of such class of relations: refined `hasProperty` relations, with disambiguated arguments. Such knowledge has never been automatically compiled at large-scale before.

3.1 Introduction

Motivation. We all know that apples are round in shape and carrots are orange in color and have a longish shape. Computers completely lack this kind of commonsense knowledge, yet they would enormously benefit from such an asset for various use-cases of growing relevance: language understanding for translation or summarization, human-computer dialog, faceted search and search query suggestions, sentiment analytics on social media, and more.

State-of-the-art and its limitations. There has been considerable research across several disciplines to automatically acquire knowledge about attributes of objects.

Linguistics. Selectional preference approaches (McCarthy, 2001) exploit a large corpus and model the preferences of predicates (e.g., the adjective *brown* has a strong preference towards `rock`). These methods combine observed frequencies with knowledge about the semantic classes of their arguments (obtainable from corpora or dictionaries).

Approaches based on dependency parsing over text (Akbik and Michael, 2014) have been used to extract a noun and its property i.e. *amod* edge over text like "*brown rock*" or *nsubj* edge over text like "*the rock is brown*". These methods require the computationally expensive dependency parse and their accuracy is limited by the parser's accuracy.

In the context of property commonsense, the output of these linguistic methods can be interpreted as ⟨*brown* `relatedToAttribute color`⟩ or ⟨*rock* `hasProperty` *brown*⟩. These methods typically do not disambiguate the adjective or the noun, although some work does exist to disambiguate the arguments in selectional preference (McCarthy and Carroll, 2003). More importantly, they do not deal with assertion classification.

Lexical semantics based approaches like that of Almuhareb and Poesio (2004), who propose patterns such as "<*adjective*> *is a* <*attribute*> *of* <*noun*>", e.g., "*brown is a color of rocks*" to find more specific properties. Even on the Web this method yields very low recall (Baroni and Zamparelli, 2010) as commonsense relations are not explicit in text, the frequency of "*brown rocks*" in text is much higher than "*brown is a color of rocks*".

Hartung and Frank (2010, 2011) develop distributional semantic models for mapping surface form assertion candidates into a set of refined (*fine-grained*) relations. They relax the low-recall patterns of Almuhareb and Poesio (2004) by specifying two sets of patterns, the first set gathers the attribute and the noun, while the second gathers the attribute and the adjective. For example, "<*adjective*> *in* <*attribute*>" to acquire ⟨*brown* `relatedToAttribute color`⟩ and "<*attribute*> *of* <*noun*>" to acquire ⟨*rock* `relatedToAttribute color`⟩. By aggregation, adjectives and nouns have a distribution vector over the attributes. They use vector-space as well as LDA-based topic models to then select an attribute for a pair of noun and adjective.

These methods assume that the given assertions are already correct instances of at least the generic `hasProperty` relation and require explicit commonsense relations in text. Further, these works tackle the assertion classification problem, not the problem of computing assertions from raw data and do not produce disambiguated arguments.

Sentiment analysis. In product reviews, attribute classification allows for deeper sentiment understanding. For example, consider a review "*the car is too expensive*"; in this sentence, the attribute (referred to as implicit aspect in the sentiment analysis community) is *cost* and bears a negative sentiment. Fei et al. (2012) identify the attribute of adjectives by extracting all nouns in a dictionary's gloss assuming that attributes are very likely to be present in glosses (e.g., the gloss of *expensive* mentions "*marked by high prices*"). Subsequently they iteratively expand over the neighborhood using synonyms and other related adjectives. Their method classifies based on a dominant attribute for an adjective and thus cannot handle the different senses of an adjective (e.g., *hot* might always be classified as `temperature` but never `taste`).

Computer vision. Attribute-centric image representation (Farhadi et al., 2009) treats objects as a distribution over visual attributes. For example, something *brown*, *furry*, *spotty* is likely to be a *dog*. Several approaches (Lazaridou et al., 2015) have extracted visual attributes from images directly by training classifiers for visual attributes.

Attribute classification from images is a recent and attractive complement to text based linguistic approaches. Not all commonsense is expressed in text, and for such attributes, computer vision based techniques provide complementary knowledge (Shutova et al., 2015). However these approaches are limited to a small set (annotating training images per visual attribute, e.g., *furry*, is expensive) of visual-only attributes (so for example, they cannot capture `evokesEmotion` or `hasTaste`). Secondly, these approaches do not classify the attributes and thirdly, there is no explicit disambiguation of the adjectives.

Knowledge acquisition. Among the manually curated commonsense KBs, Cyc has compiled complex assertions such as *every human has exactly one father and exactly one mother*, but did not aim to gather properties of objects at large-scale. WordNet has manually organized nouns and adjectives into lexical classes, with careful distinction between words and word senses; however, nouns and adjectives are not connected by any semantic relation, except the extremely sparse `attribute` relation (with around 1,200 links). Note that it is in fact a non-trivial task even for a human to recall a long list of possible shapes or possible `weight` attributes (other than the most basic ones like *heavy*, *light*).

Among the (semi-) automated approaches, ConceptNet contains the generic relation `hasProperty` but not fine-grained properties. Lebani and Pianta (2012) proposed encoding additional lexical relations for commonsense knowledge into WordNet, but their approach is inherently limited by relying on human input and also focuses on simple relations such as `usedFor`, `partOf`, etc. Clark and Harrison (2009) create commonsense propositions like ⟨*birds* `can` *fly*⟩ or ⟨*hotel* `can be` *small*⟩ using manually defined proposition templates, but their method does not produce fine-grained properties. Tandon et al. (2011) automatically extend ConceptNet with millions of triples of the form ⟨noun `relation` adjective⟩ by mining the N-gram corpora; their `hasProperty` relation is by definition as coarse-grained as ConceptNet.

None of these knowledge resources has refined properties like shape, size, taste, emotion, etc., and none have produced large amounts of semantically organized knowledge that distinguishes the different meanings of ambiguous properties such as *hot*, which can refer to temperature, taste, or emotion. For example, the KB by Tandon et al. (2011) would merely have simple triples like ⟨*milk* `hasProperty` *hot*⟩, ⟨*chiliPepper* `hasProperty` *hot*⟩, ⟨*dress* `hasProperty` *hot*⟩. Thus, state-of-the-art commonsense KBs still have severe limitations: i) sparseness on aspects that go beyond generic relations, ii) focus on crude relations, without distinguishing different semantic properties, and iii) no distinction between words and their different senses.

Problem statement. We aim to compile a large and clean set of fine-grained commonsense properties, connecting noun senses with adjective senses by a variety of relations. In contrast to prior work that only dealt with a generic `hasProperty` relation, we use 19 different (sub-) relations like `hasShape`, `hasSize`, `hasTaste`, `hasAbility`, `evokesEmotion`, etc. This list is systematically derived from WordNet covering the hyponyms of the WordNet noun sense `attribute`.

It is important to consider semantic refinement on the components of the property relations. A KB without this distinction, would not know how to treat ambiguous assertions. For example, ⟨*plant* `hasProperty` *green*⟩, there are two different interpretations with very different meanings as discussed in Chapter 1:
⟨*industrial-plant* `hasQuality` *green-environmental*⟩
⟨*botanical-plant* `hasColor` *green-color*⟩

Definition 3.1.1 - Property assertion.

A property assertion $\mathscr{A}_p$ is a triple $\langle \texttt{w1}_n^s\ \ \texttt{r}\ \ \texttt{w2}_a^s \rangle$ where $\texttt{w1}_n^s$ is a noun sense in WordNet, and $\texttt{w2}_a^s$ is an adjective sense in WordNet, and r is a property relation. Every property assertion is accompanied by a confidence score $0 \leq \Theta(\mathscr{A}_p) \leq 1$.

In order to draw these distinctions, WebChild maps the arguments of the property assertion to WordNet and classifies the relations to a finer-grained taxonomy of properties.

Definition 3.1.2 - Domain and range.

A property commonsense relation r has a domain $dom(r)$ comprising of the set of noun senses that appear in r as left-hand arguments. The range $rng(r)$, of r, is the set of adjective senses that appear in r as right-hand arguments.

Thus, our goal is to populate these relations with *assertions* (see Definition 3.1.1) in the form of triples $\langle \texttt{w1}_n^s \; \texttt{r} \; \texttt{w2}_a^s \rangle$ where $\texttt{w1}_n^s$ is a noun sense in WordNet, and $\texttt{w2}_a^s$ is an adjective sense in WordNet, and r is one of the considered relations. Each relation r has a domain $dom(r)$ and range $rng(r)$, see Definition 3.1.2.

Our approach. We present `WebChild`, a large commonsense KB automatically built from Web sources by a novel method relying on semi-supervised learning. WebChild contains more than 4 million assertions for fine-grained relations such as `hasTaste`, `hasShape`, `evokesEmotion`, etc. We use a judiciously designed form of *label propagation (LP)* (see Talukdar and Crammer (2009) for an intro) for learning the domain set, the range set, and the extension of such relations, at large scale. To this end, we first construct graphs that connect nouns, adjectives, and WordNet senses as nodes, by weighted edges. The edge-weights are derived from sense relatedness, pattern statistics, and co-occurrence statistics.

We harness WordNet and Web data to obtain seeds to initialize the LP graphs, and then use LP algorithms to derive high-quality assertions for fine-grained relations between noun senses and adjective senses.

Contributions. Our system has a number of salient characteristics and results in a large commonsense KB with unique qualities:

1 **Fine-grained assertions:** WebChild is the first commonsense KB that provides refined `hasProperty` relationships between nouns and adjectives into specific and thus more informative relations. We support 19 different relations like `hasShape`, `hasSize`, `hasTaste`, `evokesEmotion`, at more than 80% accuracy. These are systematically derived from and cover the hyponyms of the WordNet noun sense `attribute` *(an abstraction belonging to or characteristic of an entity)*.

2 **Disambiguated arguments:** The arguments of all assertions in WebChild are disambiguated by mappings to WordNet senses: noun senses for the left-hand arguments of a relation, and adjective senses for the right-hand arguments.

3 **Minimal supervision:** Our method does not require any labeled assertions for training. Instead, we use bootstrapping based on Web patterns and WordNet. Our method copes with noisy input from the Web, including noisy seeds in the bootstrapping.

3.2 Methodology

We decompose the problem of finding assertions for fine-grained commonsense relations into three sub-tasks.

1. **Range population:** First, we compute adjective senses that occur in the range of each of the WebChild relations (see Table 3.1 for a list of relations). For example, for the `hasColor` relation, we obtain a list of color attributes including, e.g., $\texttt{green}_a^1$, the color sense of *green* from WordNet, but not the environmental sense of *green*. For the `hasShape` relation, we obtain a list of possible shapes, e.g., $\texttt{circular}_a^2$.
2. **Domain population:** Our second task is to compute noun senses for the domain of each relation. For example, $\texttt{war}_n^1$ for the `evokesEmotion` relation, and $\texttt{pizza}_n^2$ for the `hasTaste` relation.
3. **Computing assertions:** Finally, we aim to map generic word-level assertion candidates ⟨*noun* `hasProperty` *adjective*⟩, gathered from Web corpora, into fine-grained assertions about word senses. For example, ⟨*car* `hasProperty` *sweet*⟩ is mapped into ⟨$\texttt{car}_n^1$ `hasAppearance` $\texttt{sweet}_a^2$⟩.

Table 3.1: List of WebChild relations:

Relation r	$dom(r)$	$rng(r)$
`hasAbility`	`hasAppearance`	`hasBeauty`
`hasColor`	`evokesEmotion`	`evokesFeeling`
`hasLength`	`hasMotion`	`hasSmell`
`hasQuality`	`hasTaste`	`hasShape`
`hasSize`	`hasSound`	`hasState`
`hasStrength`	`hasSensitivity`	`hasTemperature`
`hasWeight`		

3.2.1 Candidate gathering

For all three sub-tasks we can start with a small number of *seeds* obtained from WordNet, for example, by using the *attribute* information that connects relational noun senses (e.g., $\texttt{shape}_n^1$) with adjective senses (e.g., $\texttt{straight}_a^1$ and $\texttt{crooked}_a^1$). This is very sparse data (e.g., there are only 2 adjective senses for the attribute `shape`). Our specific choice of seeds depends on which of the three sub-tasks we are dealing with. This will be discussed later in the respective sections.

To build a knowledge base of high coverage, we gather candidates for assertions from the Web. For this purpose, we harness a huge *N-gram corpus*: the Google Web 1T N-Gram Dataset Version 1 (Brants and Franz, 2006), which consists of 1.2 billion 5-grams (i.e., 5 consecutive words or other tokens) derived from the index of the Google search engine. Each of these 5-grams comes with its frequency of occurrences on the Web. Thus, we can use these frequencies to simulate a full Web corpus. However, we also face the restriction that N-grams are limited in length to 5.

To gather assertion candidates from this data, we employ surface patterns whose matches return N-grams that contain a noun and an adjective that are likely to be related, in a generic `hasProperty` sense. Note that the resulting candidates are still at the word level; there is no way of mapping them to senses at this stage. We define generic templates for lexical patterns of the form

"<noun> *linking_verb* [*adverb*] <adj>" or
"<adj> <noun>".

Linking verbs are different forms of "to be", "to become", "to smell", "to taste", etc.[1] Our templates capture many variations of assertions. Examples are

apple was really <adj>,
apple was <adj>,
<adj> apple.

Applying this family of patterns to the Google N-gram corpus results in 3.6 million noun-adjective pairs. Many of these are noise (i.e., incorrect), and none of them is disambiguated onto senses yet.

3.2.2 Semi-supervised inference on graphs

The candidates obtained by the outlined procedure are usually very noisy, not yet disambiguated, and not yet assigned to our fine-grained relations – they are just word pairs for the generic `hasProperty` relation and are still ambiguous. To distill the good pairs from the noisy pool and to map words onto proper senses

[1] see `http://en.wikipedia.org/wiki/List_of_English_copulae` for a full list

and noun-adjective sense pairs into specific relations, we use a semi-supervised classification method over judiciously constructed graphs. To this end, we employ the method of *label propagation (LP)* (Talukdar and Crammer, 2009).

For each of the three sub-tasks towards building WebChild, we construct a graph with words (or word pairs) and possible word senses (or sense pairs) as nodes. A small number of nodes encode seeds, with known relation labels. Edges reflect the relatedness of nodes; with weights derived from Web statistics and WordNet information (see Figure 3.1). The specifics of the graph depend on the sub-task that we are dealing with, and will be discussed in the following sections.

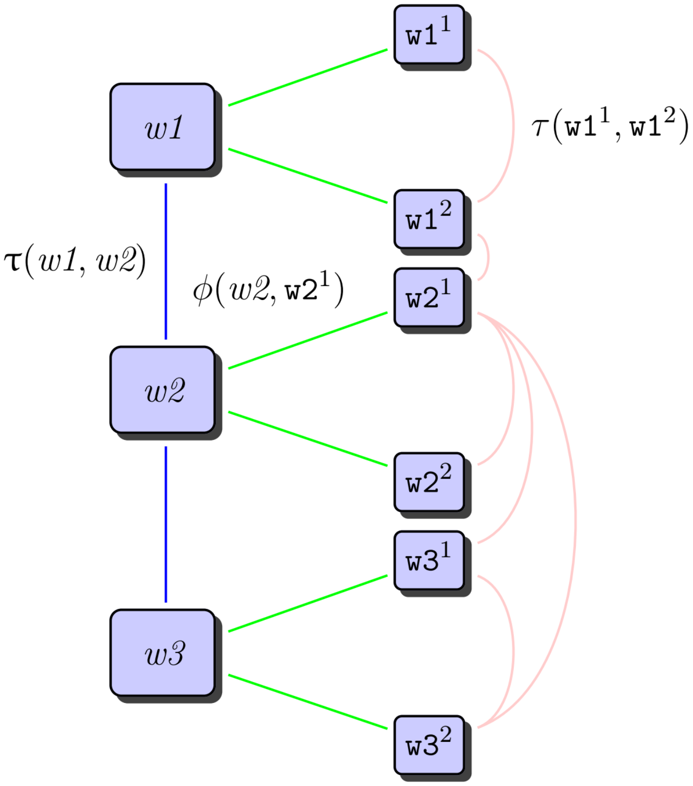


Figure 3.1: The generic graph for label propagation. The nodes represent surface forms and senses. The weighted edges are: (i) in blue: surface form-surface form similarity edges, (ii) in green: surface form - sense similarity edges, (iii) in red: sense - sense similarity edges. Computations of these edge weights appears in the respective sub-sections of Range, Domain and Assertion computations.

LP computes scores for nodes having certain labels. In our setting, these labels are used to distinguish different relation types. For inference, we use the

MAD (modified adsorption) algorithm (Talukdar and Crammer, 2009), which has shown good performance for graphs with high numbers of incident edges per node. Our graphs have this property because adjectives usually have many possible senses.

MAD propagates labels to neighboring nodes along the graph's edges; a high edge weight implies that the incident nodes are likely to have the same label. Seed nodes are expected to retain their original labels. Additionally, regularization is employed to minimize label changes within a neighborhood, which is essential to avoid overfitting. To encode this intuition, the MAD algorithm minimizes a loss function. Assume that the graph is represented as a weighted adjacency matrix W and that the label vectors of the nodes are encoded into matrices Y for the initial labeling and $\hat{Y}$ for the final predicted labeling. (Y_{*l}) and $(\hat{Y}_{*l})$ denote the l^{th} column vector of the initial matrix Y and the final label matrix $\hat{Y}$, respectively. Then the loss function is:

$$\begin{aligned} L(\hat{Y}) = \sum_{l} \Big[& \mu_1 \; (Y_{*l} - \hat{Y}_{*l})^T S^l (Y_{*l} - \hat{Y}_{*l}) \\ & + \mu_2 \; \hat{Y}_{*l}^T L \hat{Y}_{*l} \\ & + \mu_3 \left\| \hat{Y}_{*l} - R_{*l} \right\|_2 \Big], \end{aligned} \tag{3.1}$$

The first term encodes that initial and final labels for seed nodes should mostly be the same. This is enforced by the diagonal matrix S having $S_{vv} = 0$ for non-seed nodes, while for seed nodes S_{vv} is set to monotonically increase with the entropy of a node's transition probabilities (such that high degree nodes are discounted). The second term encodes that neighbor nodes obtain similar labels. This effect is realized by the unnormalized graph Laplacian L of the weighted adjacency matrix W. The third term contributes to the regularization of the estimated labels, in order to avoid over-fitting to their seed labels. This is enforced with an abandonment matrix R having a zero-valued column vector corresponding to every label, except the dummy label (the dummy label is an additional label that has a large value if the node cannot be properly labeled). A pre-defined weight is computed for the dummy label, in proportion to the nodes' degrees.

The MAD algorithm is a variant of the Jacobi method (also used for PageRank, for example), an iterative process that uses the current labels of nodes to update the label scores for neighboring nodes. When this process converges or a specified number of iterations are reached, each vertex is associated with a vector indicating the estimated labels (including the dummy label). The dummy label has a large value if the node cannot be properly labeled. In our setting, we apply this procedure for each relation separately, comparing the relation labels vs. the dummy label in the resulting output. We accept the relation label only if its final score is larger than that of the dummy label.

3.3 Relation Ranges

We now discuss how we are able to apply this same methodology to each of the three sub-tasks introduced in Section 3.2. Our first sub-task addresses the problem of identifying possible adjective senses for the range of each of the relations supported by WebChild. For example, for `hasTaste`, we expect adjectives like *delicious*, *spicy*, *hot*, *sweet*, etc., whereas these adjectives do not make much sense for the `hasShape` relation. The main difficulty that we face with this task is to move from the word level to word senses. So actually, we aim to populate the range of `hasTaste` with senses $\texttt{delicious}_a^2$, $\texttt{spicy}_a^1$, $\texttt{hot}_a^9$, $\texttt{sweet}_a^1$, etc. Some of these surface words also appear with other relations; for example, *hot* may also denote a property for the `hasAppearance` relation, however with different senses: $\texttt{hot}_a^{10}$ and $\texttt{sweet}_a^4$. The task is to carefully discriminate the senses for the ranges of different relations (although some overlap between relations may be possible).

We solve this problem in three steps:

1. **Gathering candidates** from N-grams and other sources.
2. **Constructing a graph** that encodes association strengths between adjectives and adjective senses by weighted edges.
3. **Inferring adjective senses for a relation's range** by semi-supervised Label Propagation.

Candidate gathering. We start with the raw candidates derived by extraction patterns from the Google N-gram corpus, as described in Section 3.2. For relation r, we filter the candidates by checking for the presence of the word r, any of its synonyms (e.g., *shape*, *form*, etc., or *appearance*, *look*, etc.), or any linking verb that is derivationally related to r (e.g., "tastes" for `hasTaste`). We apply these patterns also to WordNet glosses (i.e., short descriptions of word senses), to collect further candidates.

In addition, we apply the Hearst pattern "<r> such as <x>" to the N-gram data, and collect the matches for x as possible adjectives for relation r. Finally, we adopt the WebSets method (Dalvi et al., 2012) to HTML tables in specific articles of the English Wikipedia. The articles of choice are those whose article name corresponds to relation r (or its synonyms). These were manually identified for each relation r.

In total, we collected around 40,000 adjectives for all relations together.

Graph construction. So far, we have merely compiled a large set of – mostly ambiguous – words that may populate the range of a relation. We use these words as nodes in a graph and extend this node set by *all possible adjective senses* of these words. This is a simple lookup in WordNet, without any disambiguation yet. The resulting graph is called an *RPG*, see Definition 3.3.1.

Definition 3.3.1 - Range Population Graph (RPG).

The RPG of a relation r is a weighted undirected graph with nodes V_{RPG} and edges E_{RPG} as follows:

- V_{RPG} consists of all candidate adjectives for relation r, and all their corresponding adjective senses.
- E_{RPG} consists of three kinds of edges (see Table 3.2):
  - edges between two words *w1* and *w2* if they have at least one co-occurring pattern;
  - edges between two senses $\texttt{w1}^i_a$ and $\texttt{w2}^j_a$ if they are related in the WordNet structure or have WordNet glosses that suggest relatedness;
  - edges between a word w and all its senses $\texttt{w}^i_a$.

Figure 3.2 presents an example of an RPG[2].

Edge weighting. To define meaningful edge weights, we utilize statistics from the candidate gathering (see Section 3.2) and from WordNet.

- E_{RPG} ***(a1, a2)*:** For weighting the *edges among words*, we harness the co-occurrences of adjectives with nouns. We derive from the large N-gram corpus two matrices O : *noun* × *adjective* and P : *noun* × *adjective* where O_{ij} is

[2] Footnote 3.3: The WordNet senses for *hot* and *sweet* are:

- $\texttt{hot}^1_a$: used of physical heat … $\texttt{sweet}^1_a$: having taste of sugar …
- $\texttt{hot}^2_a$: violent activity … $\texttt{sweet}^2_a$: angelic nature …
- … …
- $\texttt{hot}^8_a$: wanted by police … …
- $\texttt{hot}^9_a$: spicy … $\texttt{sweet}^9_a$: unfermented …

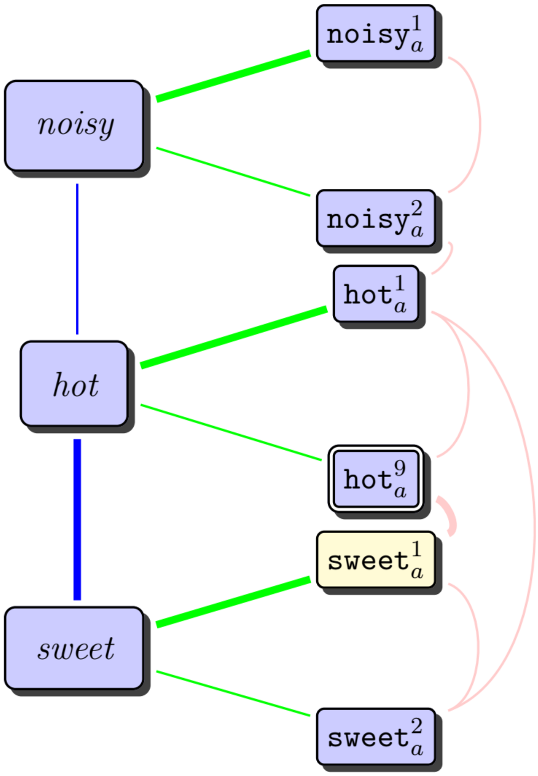


Figure 3.2: Label propagation over *RPG*, for `hasTaste` relation. The nodes represent surface forms of adjectives and adjective senses, see footnote 3.3 for the senses. Yellow nodes denote seeds provided to the label propagation algorithm, which is able to estimate the double-edged boxes as positive instances for `hasTaste`. The weighted edges are: (i) in blue: surface form- surface form similarity edges, (ii) in green: surface form- sense similarity edge, (iii) in red: sense - sense similarity edge, refer Table 3.2 for the edge weights. Darker edges denote high edge weight.

the number of occurrences of the noun-adjective pair and P_{ij} is the number of distinct extraction patterns that the noun-adjective pair occurs with. We normalize both matrices to have values in $[0, 1]$. For O, we divide all values by the maximum value. For P, we transform all values using the sigmoid function $f(x) = 1 - \frac{1}{e^{x-1}}$. The rationale here is to reward multiple patterns, but consider also the diminishing returns of observing many patterns. Finally, we combine O and P by the linear combination $\alpha O^T \times O + (1-\alpha)P^T \times P$ with hyper-parameter α (see Equation 3.2). The values of the resulting matrix are the weights for edges between two adjectives of the RPG.

- E_{RPG} **($\mathtt{w}_a^i$, $\mathtt{w}_a^j$):** For *edges between two senses* $\mathtt{w}_a^i$ and $\mathtt{w}_a^j$, we consider their taxonomic relatedness within WordNet. If there is a path between $\mathtt{u}_a^i$ and $\mathtt{w}_a^j$ using `hypernym/hyponym`, `derivationally_related`, `similar_to`, `also_see`, or `antonym` links in WordNet, then we use the Hirst measure of semantic relatedness (Hirst and St-Onge, 1998), which measures the length of the shortest path connecting two senses in the WordNet taxonomy. Unlike other path based similarity measures that limit to `hypernym` or `hyponym`, Hirst measure extends the path to all relations in WordNet by clustering them to horizontal, up, or down and penalizing changes in direction. This makes Hirst measure our only choice as the similarity measure because only Hirst measure is applicable to both adjective and noun senses, in contrast to other measures that only apply to noun and verb senses.

  We, additionally, resort to the glosses of $\mathtt{w}_a^i$ and $\mathtt{w}_a^j$, expanded by glosses of their respective hyponyms and hypernyms. We compute the number of overlapping words shared by these contexts. This is essentially the concept similarity measure by Lesk (1986). All these measures are normalized to fall between 0 and 1, and we use a down-weighting coefficient for the gloss-based values, to ensure that path-related sense pairs have higher edge weights (see Equation 3.2).

- E_{RPG} **(a, $\mathtt{w}_a^i$):** For *edges between words and senses*, we would ideally like to use the WordNet sense frequencies as a basis for edge weights. However, such information is hardly available. WordNet provides senses frequencies only for a small set of words, mostly nouns, and not nearly for all of their senses. Moreover, this information corresponds to the WordNet sense annotated documents that reflect usage within a domain. For example, for the word *tiger*, the most frequent sense according to frequency is $\mathtt{tiger}_n^1$, which stands for an audacious person, while the more general usage of animal is $\mathtt{tiger}_n^2$.

  We thus resort to the following statistics-based score (adopted from Lesk (1986); McCarthy et al. (2007)). For each word w, we take the corresponding column from matrix O (i.e., the frequencies of co-occurring nouns) as a distributional-semantics vector. For each possible sense $\mathtt{w}_a^i$, we compute a context vector from its gloss and the glosses of neighboring senses, giving us another noun distribution. The normalized scalar product between the vector of w and the vector of $\mathtt{w}_a^i$ is the weight of the edge between w and $\mathtt{w}_a^i$ (see Equation 3.2).

Label propagation. The final step is to run the MAD algorithm for label

Table 3.2: Edge weight formulae for *RPG*, for a relation r

Edge between a sense $\mathtt{a}^i$ and its observation a	
$\phi[a\ ,\ \mathtt{a}^i\ ] = \vec{a}_{nouns} \cdot \vec{\mathtt{a}^i}_{glosses}$	(3.2)
Edge between two adjective observations $a1$ **and** $a2$	
$\tau_{\mathrm{AA}}[a1\ ,\ a2] = \alpha\ O^T O + (1-\alpha) P^T P$	(3.3)
$O^T \times O\ :\ \vec{a1}_{nouns} \cdot \vec{a2}_{nouns}$	(3.4)
$P^T \times P\ :\ \vec{a1}_{patternsfreq} \cdot \vec{a2}_{patternsfreq}$	(3.5)
Edge between two senses $\mathtt{a}^i$ **and** $\mathtt{a}^j$	
$\tau_{\mathrm{AA}}[\mathtt{a}^i\ ,\ \mathtt{a}^j] = \beta\ \mathrm{hirst}[\mathtt{a}^i, \mathtt{a}^j] + (1-\beta)\mathrm{lesk}[\mathtt{a}^i, \mathtt{a}^j]$	(3.6)
$\mathrm{hirst}[\mathtt{a}^i\ ,\ \mathtt{a}^j]$: WordNet graph Hirst similarity	(3.7)
$\mathrm{lesk}[\mathtt{a}^i\ ,\ \mathtt{a}^j]\ :\ \vec{\mathtt{a}^i}_{glosses} \cdot \vec{\mathtt{a}^j}_{glosses}$	(3.8)

propagation on the constructed graph – one graph for each relation. We consider only two labels for each graph: the relation of interest and the dummy label (encoding *no relation* or *other relation*). We obtain seeds automatically by observing that the intersection of adjectives found in WordNet and on the Web, i.e. in more than one source, are more likely to be accurate. The sense of the WordNet adjective is considered for this. 30% of the remaining seeds were used as held-out test data to tune the parameters μ_1, μ_2 and μ_3 of the MAD algorithm (see Section 3.2.2). The MAD algorithm then infers which adjective senses belong to the range of the relation.

3.4 Relation Domains

After populating the ranges of WebChild's relations, we turn to the relation domains. For each relation, such as `hasTaste`, we aim to compute the noun senses that can appear as left-hand arguments of the relation, for example, $\mathtt{apple}_n^1$, $\mathtt{pizza}_n^1$, $\mathtt{plant}_n^2$, $\mathtt{beef}_n^2$, but not $\mathtt{car}_n^1$, $\mathtt{cow}_n^2$, or a different sense of *plant*: $\mathtt{plant}_n^1$ (the industrial plant). Analogously to the previous section, we solve this task in a three-step process: gathering candidates, constructing a graph, and LP inference for cleaning and disambiguation. We will see that we can harness the knowledge that we already acquired about adjective senses that appear in relation ranges.

Candidate gathering. We use the coarse-grained generic `hasProperty` noun-adjective pairs (n, a) gathered by the method of Section 3.2.1. Given a pair (n, a), if the adjective a has at least one sense that appears in the relation's range computed in Section 3.3, then the noun n becomes a domain candidate. For example, given word pair (*beef*, *salty*) and having the knowledge that *salty* occurs in the range of `hasTaste`, we infer that *beef* is a noun candidate for `hasTaste`.

Few n have only one sense in WordNet and we can directly use this noun sense because it is not ambiguous. A more typical candidate would be *java*, with co-occurrence pairs such as (*java*,*tasty*), (*java*,*easy*), (*java*,*hilly*), etc. In such situations, to move from words to senses at least in some of the cases, we harness the glosses of adjectives in WordNet to derive *semi-disambiguated* assertions where either the noun or the adjective is mapped to a WordNet sense.

Gathering *semi-disambiguated* assertions: For each adjective word in our candidate set, we find all noun-sense glosses where the adjective occurs (as a surface word). Whenever a matching noun sense gloss is found, the specific noun sense is used to replace the ambiguous surface noun in the candidate pair. We perform this analogously for nouns in adjective-sense glosses. For instance, the gloss for $\texttt{sour}_a^2$ reads "*the taste experience when vinegar or lemon juice is taken ...*". We generate two assertions from this: (*vinegar*, $\texttt{sour}_a^2$) and (*lemon juice*, $\texttt{sour}_a^2$). Note that although this technique goes further than the method for relation ranges, we still face a large amount of noisy candidates. An adjective such as *large* has seven word senses in WordNet, and we can obtain numerous noun-sense candidates whose glosses contain "large".

Graph construction. Next, we construct a graph for subsequent Label Propagation similarly as in the method for range population (Section 3.3). The resulting graph is called a *DPG*, see Definition 3.4.1.

Definition 3.4.1 - Domain Population Graph (DPG).

The DPG for a relation r is a weighted undirected graph with nodes V_{DPG} and edges E_{DPG} as follows:

- V_{DPG} consists of all candidate nouns for r and their possible noun senses from WordNet.
- E_{DPG} edges and their weights are computed exactly as described for RPGs (see Table 3.3), with nouns taking the place of adjectives.

Table 3.3: Edge weight formulae for DPG, for a relation r

Edge between a sense $\mathtt{n}^i$ and its observation n	
$\phi[n\ ,\ \mathtt{n}^i\ ] = \vec{n}_{adjectives} \cdot \vec{\mathtt{a}^i}_{glosses}$	(3.9)
Edge between two noun observations *n1* and *n2*	
$\tau_{\text{NN}}[n1\ ,\ n2] = \alpha\ OO^T + (1-\alpha)PP^T$	(3.10)
$O \times O^T\ :\ \vec{n1}_{adjectives} \cdot \vec{n2}_{adjectives}$	(3.11)
$P^T \times P\ :\ \vec{n1}_{patternsfreq} \cdot \vec{n2}_{patternsfreq}$	(3.12)
Edge between two senses $\mathtt{n}^i$ and $\mathtt{n}^j$	
$\tau_{\text{NN}}[\mathtt{n}^i\ ,\ \mathtt{n}^j] = \beta\ \text{hirst}[\mathtt{n}^i, \mathtt{n}^j] + (1-\beta)\text{lesk}[\mathtt{n}^i, \mathtt{n}^j]$	(3.13)
$\text{hirst}[\mathtt{n}^i\ ,\ \mathtt{n}^j]\ :\ \text{WordNet graph Hirst similarity}$	(3.14)
$\text{lesk}[\mathtt{n}^i\ ,\ \mathtt{n}^j]\ :\ \vec{\mathtt{n}^i}_{adjectives} \cdot \vec{\mathtt{n}^j}_{glosses}$	(3.15)

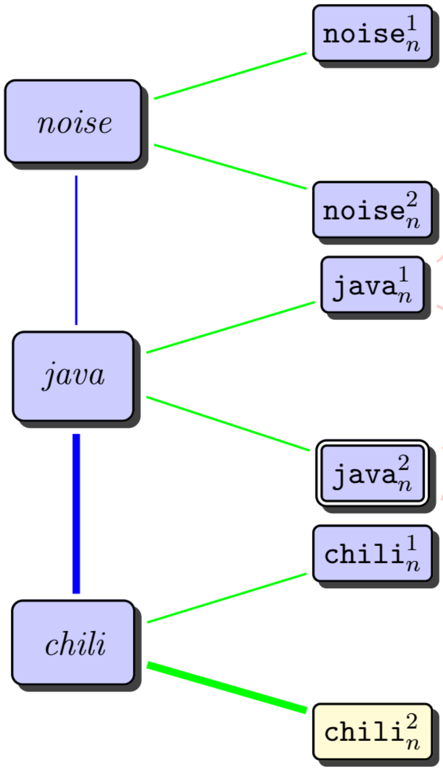


Figure 3.3: Label propagation over *DPG*, for `hasTaste` relation. The nodes represent surface forms of nouns and noun senses, see footnote 3.4 for the senses. Yellow nodes denote seeds provided to the label propagation algorithm, which is able to estimate the double-edged boxes as positive instances for `hasTaste`. The weighted edges are: (i) in blue: surface form- surface form similarity edges, (ii) in green: surface form- sense similarity edge, (iii) in red: sense - sense similarity edge, refer Table 3.3 for the edge weights.

Figure 3.3 shows an example over label propagation over a DPG[3].

Label propagation. Again, we run the MAD algorithm for Label Propagation on the DPG for each relation. To generate seeds, we use `hasProperty` triples

[3] Footnote 3.4: The WordNet senses for *java* and *chili* are:

- $\texttt{java}_n^1$: Indonesian island . . . $\texttt{chili}_n^1$: ground beef . . .
- $\texttt{java}_n^2$: coffee beverage . . . $\texttt{chili}_n^2$: hot tapering pepper . . .
- $\texttt{java}_n^3$: programming language . . .

that have unambiguous nouns and adjectives that have been assigned to only a single relation r by the Range Population method. Some of these nouns are genuinely unambiguous while others are unambiguous for us because we have previously identified the correct sense using the WordNet gloss heuristics mentioned previously in *semi-disambiguated* assertion gathering. In either case, the single noun senses of such unambiguous nouns serve as *seeds*. The parameters of the MAD algorithm were tuned as described previously using held-out data. The MAD output provides us with scores for the noun senses of the ambiguous nouns. For the domain of relation r, we accept the noun senses whose score exceeds the dummy label score.

3.5 Computing Assertions

Finally, we leverage the domain and range knowledge for distilling the raw and ambiguous assertion candidates for the generic `hasProperty` relation, gathered as explained in Section 3.2, into word-sense pairs for fine-grained relations. Thus, we need to disambiguate the left and right arguments of the candidate assertions and determine the respective relations. Again, we build a graph representation for each relation r, and apply Label Propagation on these graphs.

For a candidate assertion with an ambiguous noun and adjective, we would often generate a large set of nodes when each of the two words has many different senses. To prevent the graph size from becoming intractable, we harness the already acquired knowledge about the domain and range of each r and consider only those sense pairs that fall into the previously computed domain and range population, respectively. This yields an enormous pruning effect, and makes the difference between a hopelessly intractable graph and a practically viable method.

Graph construction. We construct a graph for subsequent Label Propagation analogously to the method for range population (Section 3.3). The resulting graph is called an *APG*, see Definition 3.5.1. There are two differences from the previous graphs. First, every node is a word pair instead of a word. Second, there is an additional candidate node pruning step based on domain and range as described earlier. We capture these by constructing graphs of the following form for each relation.

Definition 3.5.1 - Assertion Graph (AG).

The AG of a relation r is a weighted undirected graph with nodes V_{AG} and edges E_{AG} as follows:

- V_{AG} consists of all word-level assertion candidates and all sense-level pairs that are not pruned by testing against the domain and range of r (see above).
- E_{AG} consists of three kinds of edges (see Table 3.4):
  - edges between two word-level assertions,
  - edges between a word-level assertion and a sense-level assertion,
  - edges between two sense-level assertions.

Figure 3.4 shows an example of an AG.

Edge weighting. For all three types of edges, we compute edge weights between two assertions $\langle n_1\ r\ a_1 \rangle$ and $\langle n_2\ r\ a_2 \rangle$ by considering the similarity between n_1 and n_2 and the similarity between a_1 and a_2. Here, n_1 and n_2 may be either nouns or noun senses, and similarly a_1 and a_2 may be either adjectives or adjective senses. In all cases, we use the multiplicative score

$$\text{sim}(n_1, n_2) \cdot \text{sim}(a_1, a_2)$$

as the edge weight. This similarity yields, for instance, that $\langle \texttt{car}^1_n\ \texttt{red}^1_a \rangle$ is similar to $\langle \texttt{car}^1_n\ \texttt{pink}^1_a \rangle$, $\langle \texttt{vehicle}^1_n\ \texttt{red}^1_a \rangle$, $\langle \texttt{bus}^1_n\ \texttt{colorful}^1_a \rangle$.

The individual noun-noun, noun-noun sense, and noun sense-noun sense similarities (all denoted by $\text{sim}(n_1, n_2)$ here) are computed just as for the different types of edge weights earlier in Section 3.4. Similarly, the adjective-adjective, adjective-adjective sense, and adjective sense-adjective sense similarities (all denoted by $\text{sim}(a_1, a_2)$) are computed just as for the Range Population Graph's edge weights, described earlier in Section 3.3. See Table 3.4 for formulae to compute the edge weights of AG.

Since there are $O(|E_{\text{AG}}|^2)$ possible assertion edges, we use top-k retrieval methods to efficiently aggregate scores from multiple ranked lists and avoid computing similarities below a threshold.

Label propagation. For seeds, we consider all assertions where both the noun and the adjective are unambiguous, either because they have only one sense each

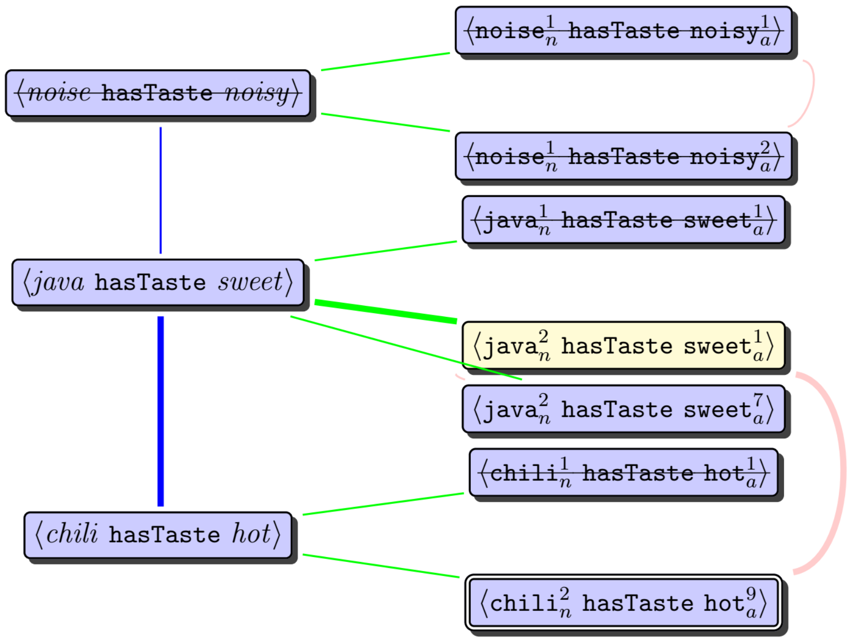

~~$\langle$noise^1_n hasTaste noisy$^1_a\rangle$~~
~~$\langle$*noise* hasTaste *noisy*$\rangle$~~
~~$\langle$noise^1_n hasTaste noisy$^2_a\rangle$~~
~~$\langle$java^1_n hasTaste sweet$^1_a\rangle$~~
$\langle$*java* hasTaste *sweet*$\rangle$
$\langle$java^2_n hasTaste sweet$^1_a\rangle$
$\langle$java^2_n hasTaste sweet$^7_a\rangle$
~~$\langle$chili^1_n hasTaste hot$^1_a\rangle$~~
$\langle$*chili* hasTaste *hot*$\rangle$
$\langle$chili^2_n hasTaste hot$^9_a\rangle$

Figure 3.4: Label propagation over AG, for hasTaste relation. The nodes represent surface forms of triples and their possible senses, see footnotes 3.3 and 3.4 for the senses. Yellow nodes denote seeds provided to the label propagation algorithm, which is able to estimate the double-edged boxes as positive instances for hasTaste. Some nodes are struck because their arguments are not in the domain $dom($hasTaste$)$ or range $rng($hasTaste$)$. The weighted edges are: (i) in blue: surface form-surface form similarity edges, (ii) in green: surface form- sense similarity edge, (iii) in red: sense - sense similarity edge, refer Table 3.4 for the edge weights.

in WordNet or because our domain- and range-based pruning left us with only one sense pair for the two words. Again, 30 % of the remaining seeds are used for tuning the parameters of the MAD algorithm. Based on these seeds, MAD computes scores for candidate assertions. We accept all assertions for r whose score exceed the dummy label score.

3.6 Results

Table 3.5 summarizes the size of the WebChild knowledge base: the number of distinct senses and assertions, the number of instances of noun and adjective

Table 3.4: Edge weight formulae for *APG*, for a relation r

Edge between a triple $\mathscr{A}$ and its observation $\mathscr{A}^*$	
$\phi[\langle n \;\; a\rangle \;,\; \langle \mathtt{n}^i \;\; \mathtt{a}^j\rangle \;] = \phi[n \;,\; \mathtt{n}^i] \times \phi[a \;,\; \mathtt{a}^j]$	(3.16)
Edge between two observations $\mathscr{A}^*$ and $\mathscr{A}^*$	
$\tau[\langle n1 \;\; a1\rangle \;,\; \langle n2 \;\; a2\rangle] = \tau_{\mathrm{NN}}[n1 \;,\; n2] \times \tau_{\mathrm{AA}}[a1 \;,\; a2]$	(3.17)
Edge between two triples $\mathscr{A}$ and $\mathscr{A}$	
$\tau[\langle \mathtt{n}^i \;\; \mathtt{a}^j\rangle \;,\; \langle \mathtt{n}^k \;\; \mathtt{a}^l\rangle] = \tau_{\mathrm{NN}}[\mathtt{n}^i \;,\; \mathtt{n}^k] \times \tau_{\mathrm{AA}}[\mathtt{a}^j \;,\; \mathtt{a}^l]$	(3.18)

senses (where a noun or adjective sense that occurs in k different relations counts k times), and the precision is estimated by extensive sampling (see below). Table 3.6 illustrates WebChild by anecdotal examples for range, domain, and assertions. These are top-ranked results, based on a simple scoring function that rewards many occurrences as well as occurrences with multiple distinct patterns.

Table 3.5: WebChild statistics

	#distinct	#instances	Precision
Noun senses	78,077	221,450	0.80
Adj. senses	5,588	7,783	0.90
Assertions	4,649,471	4,649,471	0.82

We conducted extensive experiments to assess the viability of our approach and the quality of the resulting commonsense relations. Our experiments cover the three tasks addressed in this chapter: Subsection 3.6.1 reports on the quality of relation ranges, Subsection 3.6.2 presents results on relation domains, and Subsection 3.6.3 discusses the quality of sense-disambiguated assertions. For each task, we compare against various baseline competitors.

Table 3.6: Anecdotal example results for `hasTaste`, and `hasShape`

Relation	Range	Domain	Assertions
`hasTaste`	$\texttt{sweet}^1_a$	$\texttt{strawberry}^1_n$	$(\texttt{chocolate}^1_n,\texttt{creamy}^2_a)$
	$\texttt{hot}^9_a$	$\texttt{chili}^1_n$	$(\texttt{pizza}^1_n,\texttt{delectable}^1_a)$
	$\texttt{sour}^2_a$	$\texttt{salsa}^1_n$	$(\texttt{salsa}^1_n,\texttt{spicy}^2_a)$
	$\texttt{salty}^3_a$	$\texttt{sushi}^1_n$	$(\texttt{burger}^1_n,\texttt{tasty}^1_a)$
	$\texttt{lemony}^1_a$	$\texttt{java}^2_n$	$(\texttt{biscuit}^2_n,\texttt{sweet}^1_a)$
`hasShape`	$\texttt{triangular}^1_a$	$\texttt{leaf}^1_n$	$(\texttt{palace}^1_n,\texttt{domed}^1_a)$
	$\texttt{meandering}^1_a$	$\texttt{circle}^1_n$	$(\texttt{table}^2_n,\texttt{flat}^1_a)$
	$\texttt{crescent}^1_a$	$\texttt{ring}^8_n$	$(\texttt{jeans}^2_n,\texttt{tapered}^1_a)$
	$\texttt{obtuse}^2_a$	$\texttt{egg}^1_n$	$(\texttt{tv}^2_n,\texttt{flat}^1_a)$
	$\texttt{tapered}^1_a$	$\texttt{face}^1_n$	$(\texttt{lens}^1_n,\texttt{spherical}^2_a)$

3.6.1 Relation ranges

Baselines. Since there is no direct competitor, we designed several baselines as follows.

WordNet attributes: For some relations, WordNet provides the range directly by its `attribute` relation (e.g., `size` contains the adjective senses $\texttt{small}^1_a$ and $\texttt{big}^1_a$).

WordNet attributes expanded: We expanded the above data by including related word senses using `synonyms`, `antonyms`, `similarTo`, and `derivationally Related` links. We then iterated this step once more to enlarge the set, but stopped at this point to curb the inevitable topic drift.

WordNet glosses: WordNet provides a short gloss for each adjective. If the gloss mentions a relation, we include the adjective sense in the relation's range. For example, the gloss of $\texttt{red}^1_a$ mentions the word *color*.

Controlled LDA MFS: Hartung and Frank (2011) developed a method for creating pseudo-documents per relation r (e.g., color) using nouns and adjectives that appear in the relation. An LDA model estimates the probability $P[a|d]$ for an adjective a given a pseudo-document d, thereby approximating $P[a|r]$ to $P[a|d]$. All adjectives above a threshold for this probability form the range of the relation. We map these adjectives to their most frequent sense (MFS) according to WordNet.

Google Sets MFS: This service, now part of the spreadsheet processor of `docs.google.com`, expands sets of similar words given a few seeds to start with. We use it to find, for each relation, large candidate sets, using five WordNet adjectives as seeds. The resulting adjectives are mapped to the most frequent sense according to WordNet.

Results. We constructed a random sample of 30 adjectives for each relation from the output of WebChild (a total of 570 samples). These were manually evaluated by three people. The Kappa value for inter-annotator agreement was 0.869. We likewise drew samples from the outputs of the baseline competitors (or used the entire output when less than 30), and manually assessed them, too. For statistical significance, we computed Wilson score intervals for $\alpha = 95\%$ (Brown et al., 2001).

The results of this evaluation are shown in Table 3.7, reporting the macro-averaged precision and the total number of results (coverage). WebChild stands out in this comparison: It discovers far more (sense-mapped) adjectives than any other method, and achieves a very good precision of 90%. WebChild's coverage is three times larger than that of the best prior method (Hartung and Frank, 2011).

Table 3.7: Results for range population

Method	Precision	Coverage
WordNet attributes	1.00	40
WordNet attributes expanded	0.61 ± 0.03	5,145
WordNet glosses	0.70 ± 0.06	3,698
Controlled LDA MFS	0.30 ± 0.06	2,775
Google Sets MFS	0.27 ± 0.04	426
WebChild	0.90 ± 0.03	7,783

3.6.2 Relation domains

Baselines. We compare WebChild against the following competitors.

Extraction unambiguous: Almuhareb and Poesio (2004) and Hartung and Frank (2010) manually defined eight patterns (e.g., "the <adj> of <noun> was") to populate the domain of a relation. We applied these patterns to the N-gram corpus and to WordNet glosses. As this technique yields nouns rather than noun senses, we consider only unambiguous nouns with a single sense in WordNet.

Extraction MFS: For higher coverage, we considered all nouns obtained by the previous method and mapped them to their most frequent sense according to WordNet.

Controlled LDA MFS: Using the method of (Hartung and Frank, 2011) (see baselines on range population) we collect nouns n with a probability $P[n|d]$ above a threshold. We map the nouns to their most frequent senses in WordNet.

WordNet glosses: If a relation name (e.g., "*color*") appears in the WordNet gloss of a noun sense, we capture the noun sense as an instance of the relation's domain.

WebChild adj. projections: Our candidate gathering step extracted a large set of noun-adjective pairs from the Web. Since WebChild already has mapped adjectives to specific relations' ranges, a heuristic technique is to assign the co-occurring nouns to the domains of the same relations. Since these nouns are not yet disambiguated, we map them to the most frequent sense in WordNet.

Google sets: For domain population, this technique performed very poorly due to heterogeneity of seeds; so we do not show any results below.

Results. Table 3.8 shows the results of this comparison. Again, WebChild stands out, especially by its high coverage. At the same time, its precision of 83% is still fairly high. The method based on WordNet glosses performed slightly better in terms of precision, but yields an order of magnitude lower coverage.

Table 3.8: Results for domain population

Method	Precision	Coverage
Extraction Unambiguous	0.76 ± 0.06	6,190
Extraction MFS	0.75 ± 0.05	30,445
Controlled LDA (Hartung and Frank, 2011) MFS	0.71 ± 0.06	9,632
WordNet glosses	0.86 ± 0.03	14,328
WebChild adj. projections	0.71 ± 0.03	175,480
WebChild	0.83 ± 0.03	221,450

Table 3.9: Results for assertions on data of Hartung and Frank (2011)

	Precision	Recall
Controlled LDA (Hartung and Frank, 2011)	0.33	0.23
WebChild	0.93	0.50

3.6.3 Assertions

As for the main task on commonsense knowledge acquisition, computing fine-grained assertions between noun senses and adjective senses, we can compare WebChild's performance directly with the prior method Controlled LDA (C-LDA) of (Hartung and Frank, 2011). C-LDA treats the task as a classification problem, with relations as the classifier's labels. We use the same data that the experiments of Hartung and Frank (2011) were based on. As C-LDA works at the word rather than word-sense level, for the results of their system, a noun-adjective pair is counted as correct if there is at least one sense pair for which the relation label is meaningful. In contrast, we give WebChild the significant disadvantage of considering an assertion as correct only if the senses are correctly chosen, too. Table 3.9 shows the results of this experiment, demonstrating the clear superiority of WebChild over the prior state-of-the-art.

Baselines. For more comprehensive studies, on our Web-scale candidates, we again designed a variety of baseline competitors.

Controlled LDA MFS: We use C-LDA (Hartung and Frank, 2011) to map a `hasProperty` candidate pair onto fine-grained relations. Nouns and adjectives are mapped to their most frequent senses in WordNet.

Vector space MFS: This is analogous to the previous baseline, except that we use the vector space model of Hartung and Frank (2010) rather than LDA.

Web unambiguous adjective: We consider only those noun-adjective pairs where the adjective has a single sense. We use WebChild's range knowledge to map the adjective to one or more relations. The noun is mapped to its most frequent sense in WordNet.

WebChild independence: For a given relation, we consider all combinations of noun senses from the domain and adjective senses from the range as assertions for the relation.

Results. Table 3.10 shows the results of the comparison. WebChild yields more than 4 million assertions at a good precision of 82%. It outperforms all competitors by a large margin, with ten times higher coverage and twice better precision than the best of the prior method. Interestingly, even the relatively crude WebChild Independence technique performs better than the other baselines. However, its precision is far behind that of the full WebChild method.

Table 3.11 shows the results of WebChild, per relation.

Table 3.10: Results for assertions

Method	Precision	Coverage
Controlled LDA MFS	0.35 ± 0.06	254,576
Vector Space MFS	0.40 ± 0.09	355,018
Web Unambiguous Adjective	0.54 ± 0.09	709,337
WebChild Independence	0.62 ± 0.06	3,399,312
WebChild	0.82 ± 0.03	4,709,149

3.7 Discussion

We presented WebChild, a comprehensive commonsense knowledge base with fine-grained relations about sense-disambiguated nouns and adjectives. Our methodology combines pattern-based candidate gathering from Web corpora with semi-supervised Label Propagation over judiciously constructed weighted graphs. Our method performs collective classification, and is robust to noise. Experiments demonstrate that this methodology can achieve high precision with good coverage. *WebChild* is publicly available at (`http://people.mpi-inf.mpg.de/~ntandon/resources/readme-property.html`).

Strengths:

- Our method requires a very small amount of training data, between 20-50 training examples for the different relations. It is robust to incoming noise.
- Our method can readily deal with any other relations whose inherited hypernym is the WordNet noun sense `attribute`. We show the effectiveness of our

Table 3.11: Quality of WebChild relations

Relation	Precision	Coverage
ability	0.80 ± 0.10	90,288
appearance	0.95 ± 0.05	365,201
beauty	0.70 ± 0.15	95,838
color	0.70 ± 0.15	494,380
emotion	0.90 ± 0.09	79,630
feeling	0.91 ± 0.08	141,453
length	0.70 ± 0.15	90,021
motion	0.80 ± 0.10	146,148
smell	0.82 ± 0.10	25,347
quality	0.82 ± 0.10	793,484
sensitivity	0.70 ± 0.15	5,727
shape	0.80 ± 0.10	359,789
size	0.82 ± 0.10	910,901
sound	0.71 ± 0.15	130,952
state	0.88 ± 0.09	563,022
strength	0.82 ± 0.10	165,412
taste	0.70 ± 0.15	19,892
temperature	0.80 ± 0.13	27,399
weight	0.70 ± 0.15	144,587
overall	0.82 ± 0.03	4,709,149

method on a set of 19 common relations but our method can deal with a set of more than 250 relations.

- Our method is not strongly coupled with WordNet. We can deal with any alternative like Wiktionary that distinguishes and provides a gloss for different senses of a word. This is because the edge weights in all our graphs are either distributional (based on glosses and text corpora) or taxonomic (based on the hypernymy structure that is also provided by alternatives such as Wiktionary). This is an important reason for not relying on external resources constructed over WordNet like GlossTag[4], which could replace the ambiguous WordNet glosses in Section 3.6.1.

[4] http://wordnet.princeton.edu/glosstag.shtml

Weaknesses:

- Our method cannot deal with multilingual data. We define a generic pattern for mining `hasProperty` assertions, however, it is limited to English language grammar. The distributional weights can still be applied to other languages. We need a multilingual taxonomic resource for the taxonomic weights.

  As a solution to this problem, we can use multilingual WordNets or resources like Universal WordNet (de Melo and Weikum, 2009) which are linked to WordNet. The WebChild KB is currently linked to WordNet, hence this provides a direct mapping of the arguments of an assertions in different languages.

- Our method does not deal with reporting bias (infrequent assertions are observed disproportionally more). For example, consider the correct assertion ⟨`elephant hasColor grey`⟩ and an incorrect assertion ⟨`elephant hasColor pink`⟩ (arising from usage like "*pink elephant*" as a toy or "*seeing pink elephants*" as a euphemism for drunken hallucination). We are also limited by WordNet sense entries because there is no WordNet sense corresponding to the elephant in abstract or toy sense. This is actually a mistake of the graph-based disambiguation because the method is expected to increase the dummy score for *elephant*. However, our method encourages global coherence across the assertions. Note that `pink` and `grey` are sibling senses and hence related so the edge weights between them is high. Our method thus increases the confidence on both of these assertions, with no knowledge that this is reporting bias.

  As a solution to this problem, we can verify these assertions in images. The main challenge would be dealing with the noisy object detections and limited training data for properties like *pink* or *large*. To overcome this challenge, it is possible to use Flickr tags and look for the co-occurring noun and adjective in images. Using our noun adjective collocation data (Shutova et al., 2015) computed over 100 Million Flickr images, we confirm that the co-occurrence frequency of *elephant* with *dusty*, *brown*, *grey* is three times more than *pink*.

  Another solution would be to consider linguistic cues like the expressive power of the adverb, e.g., "*always*" in the input. Additionally, we can assign more weight to plurals as they are more reliable cues, e.g., "*some elephant was pink*" vs "*elephants are pink*".

  One could imagine taking into account the domain authority of the input page, e.g., a page that knows about animal related topics can more reliably tell the color of an elephant rather than a shopping page. Our input sources also include Google N-grams where such provenance information is unavailable, rendering this method inapplicable.

- WebChild cannot deal with negations. Such data is sometimes useful for reasoning, e.g., ⟨`elephant`^{1} `neverHasColor pink`⟩. ConceptNet is the only prominent KB that possesses negative assertions. Suppose we assign a class label for negation, our method will not be able to handle this limitation because there could be a strong similarity edge between the positive nodes and negative nodes (`pink` and `grey` are similar to each other leading to a strong edge). This problem is related to reporting bias, except that negative assertions can never hold while assertions with biased frequency can still hold but need to be discounted.

  As a solution, if we propose usage of multimodal data, one could argue that we have never seen enough images to be able to negate the occurrence in any unseen images. In principle, KB completion techniques should be able to estimate this but such a completion requires very discriminating negative training data that is hard to obtain automatically. Therefore, like ConceptNet, we could involve a human in the loop to address this problem.

  An alternate solution would be to consider negating adverbs (*never*) and minimizers (*hardly*) (Benamara et al., 2007) in order to obtain negations.

Lessons learned:

- By intersecting noisy WordNet candidates with noisy Web candidates, we get a very good ensemble effect
- Collective classification can robustly deal with disambiguation in the absence of context. Our input text can sometimes be very short (N-grams), but our method yield good disambiguation.

Assumptions: We implicitly make the following two assumptions:

- The graph-based method that we employ (MAD) makes an assumption that high degree nodes are noisy. High degree nodes can very likely drift the random walk to nodes belonging to different classes.
- There is no ordering of assertions within a relation r. The confidence $\Theta(\mathscr{A}_p)$ reflects correctness but not salience.

Extensions:

- If we break the first assumption, we must develop a graph-based semi-supervised method that discounts only noisy hubs. The hypothesis is that a high degree node is not necessarily noisy. It is noisy only if the label variance of the first-degree neighbors have a high variance in their estimated labels in a random walk iteration. The graphs (*RPG*, *DPG*, *AG*) have many high degree nodes. Consider the case of an *RPG*; WordNet arranges adjectives as spokes therefore there are several popular adjectives with high degree. MAD would discount these heavily and they will all have a high dummy label probability, even if the discounting parameter is adjusted because the algorithm makes this assumption.

There is a recent method (TACO (Orbach and Crammer, 2012)) that performs graph-based transduction with confidence that could be useful in this direction. It is unclear whether TACO comprehensively beats MAD. More investigation is required.

- We can perform KB completion to estimate properties of previously unseen concepts or extended concepts. This entails investigating methods to obtain negative training data, further highlighting the importance of negations.
- If we break the second assumption, it leads to mining *salient* commonsense, an aspect that we did not consider currently. Salience not only refers to a distinguishing property but also the most important property that comes to a human mind if we think about the concept. For instance, a *cheetah* reminds us of a *spotted* large cat that is very *fast*. Negations are also a form of salience. This assumption is also reflected in our evaluations, where the judges mark as true anything that *can be* true. For salient commonsense, a more rigorous evaluation would need to be performed.

We touched upon negations in this discussion. As for a discriminating feature, one could employ existing dimensionality reduction techniques like principal component analysis. These techniques rely on co-occurrence and frequencies but we have a reporting bias problem. Therefore, multimodal evidence could provide additional signal but that is only limited to visual properties.

Besides using a human in the loop, we can make use of a different type of knowledge, comparative commonsense, e.g., the phrase "*x is faster than a cheetah*" elicits that `hasSpeed` is an important relation for a *cheetah*. We will present comparative commonsense in the next chapter.

4 Commonsense on Relationships: Comparisons

4.1 Introduction

In this chapter, we investigate the second category of commonsense relations: commonsense on relationships. This chapter investigates methods for one instance of such relations, fine-grained taxonomy of `comparative` relations.

Methods for large-scale extraction and organization of fine-grained comparative relations have never been explored before. In this chapter, we will present methods and results on acquisition of comparative relations.

Motivation. If your smart phone suggests *a burger joint* for lunch, and the user wants something `healthier`, the computer should know that *a seafood restaurant* is probably a better suggestion than *a burger joint*. Rather than hard coding various types of information into each relevant reasoning engine, genuine machine intelligence will require automatically drawing commonsense inferences from large amounts of data. A novel aspect of this greater endeavor of making computer systems more intelligent is *comparative knowledge*, e.g., that ⟨*juice* `is sweeter than` *water*⟩, or that ⟨*gold* `is more expensive than` *silver*⟩, is an important part of human commonsense knowledge.

State-of-the-art and its limitations. Prior work on comparative commonsense knowledge has been sparsely spread across several disciplines.

Linguistics. There has been only little prior work on comparative knowledge mining and comparative commonsense knowledge mining in particular. Jain and Pantel (2011) used query logs to identify entities that are comparable in some respect, e.g., “Nikon D80 vs. Canon Rebel XT”. (Jindal and Liu, 2006) focused on the problem of identifying potential comparative sentences and their elements using sequence mining techniques.

Jang et al. (2012) proposed graph-based methods to predict new comparable entity pairs that have not been observed in the input data. These approaches only produce sets of related entities (and sometimes commonsense concept classes) and do not aim at gathering assertions about how they compare.

Numerical attribute based: Davidov and Rappoport (2010) perform extraction using patterns bootstrapped with three types of patterns relating to value extraction ("*noun is * [width unit] wide*"), bounds ("*the widest [noun] is * [width unit]*"), and comparison ("*[noun1] is broader than [noun2]*"). Takamura and Tsujii (2015) propose a regression model for object size to combine different types of features including both value extraction ("*X is 10cm long*") and comparisons ("*X is larger than Y*"). As these methods can deal with unit normalization, if a dictionary of comparative adjectives and units is available, then these methods can be used be extract large-scale comparative commonsense.

These methods collect the comparative values by issuing queries to a search engine and parsing the returned search snippets. However, search engines limit the number of queries to a few hundred queries a day, thus severely limiting large-scale extraction and secondly these results are not geared for coverage. The attributes have been limited to size and length only and it is not clear how these methods would scale to more attributes. Finally, the arguments or the comparative adjective are ambiguous.

Visual contents based. Comparative adjectives have been used to reduce ambiguities in object detection by exploiting these relationships between objects in an image (Gupta and Davis, 2008). Previous approaches to gather comparative knowledge have typically extracted over text using comparative patterns or numerical value comparisons or both. Very recently, Bagherinezhad et al. (2016) propose a multimodal model that maximizes the joint likelihood of textual and visual observations. Thus, their method learns reliable relative size estimates, with no explicit human supervision. Like in non-comparative assertions, these approaches are complimentary to text-based approaches because some comparative commonsense may not be expressed in text. However, these approaches are limited to a small set of comparable visual attributes like `comparative size`. The arguments are still ambiguous, for example ⟨*plant* `bigger than` *plant*⟩ where *plant* refers to the industrial plant and a tree respectively.

Knowledge acquisition. Entity linking deals with the disambiguation of mentions to entities in KBs. Usually the entity is a named entity, not a concept and thus we do not explore entity-linking methods. However, Lin et al. (2012)

propose a scalable entity linking method to disambiguate the arguments x and y of a triple $\langle x\ \texttt{r}\ y\rangle$ where x and y are mostly named entities but can also be noun phrases. The relation $\texttt{r}$ is a verb phrase that they do not disambiguate. They use features from encyclopedic KBs like Freebase. Their problem is different from ours because in our case x and y are noun phrases while $\texttt{r}$ is an adjective, and their method does not provide adjective sense disambiguation.

In the realm of comparative commonsense knowledge, Cao et al. (2010) performed a small study on extracting commonsense comparative observations using manually defined patterns. Their comparable fact scoring function relies on semantic frequency based on support and oppose set of a triple, e.g., $\langle \textit{car}\ \texttt{\textit{fast}}\ \textit{bike}\rangle$ is in oppose set of $\langle \textit{bike}\ \texttt{\textit{fast}}\ \textit{car}\rangle$. However, they assume that such a similarity function would exist. Additionally, any higher order logic such as this can only be performed accurately if the triples are disambiguated. The focus of our work, in contrast, is to go beyond just a simple and small-scale extraction of natural language comparisons.

Problem statement. The goal of this work is to automatically extract and infer large amounts of comparative commonsense knowledge (see Definition 4.1.2). Given very large-scale unstructured data, our goal is to produce a large semantically disambiguated and consolidated knowledge base that recognizes semantically equivalent triples.

An example of the knowledge that we aim to compile is $\langle \texttt{steel}_n^2\ \texttt{more sharp}_a^2\ \texttt{than wood}_n^1\rangle$ that distinguishes between $\langle \texttt{steel}_n^2\ \texttt{sharp}_a^2\ \texttt{than wood}_n^1\rangle$ or $\langle \texttt{photo}_n^1\ \texttt{sharp}_a^1\ \texttt{sketch}_n^1\rangle$. Finally, we want to estimate semantically equivalent triples to our example triple like $\langle \texttt{wood}_n^1\ \texttt{more blunt}_a^1\ \texttt{than steel}_n^2\rangle$.

In particular, we consider relationships that can be expressed using the comparative forms of adjectives, e.g., *is bigger than*, *is more reliable than*. As we are aiming at commonsense, most knowledge we would expect to capture will not hold as absolute universal truths; rather, be a reflection of overall tendencies. For example, although cheetahs are generally known to be faster than lions, an individual cheetah might be too young or unhealthy to be faster than a given lion.

In an effort to cover a wider range of commonsense phenomena, we do not limit ourselves to arguments $\texttt{x}$ and $\texttt{y}$ that directly correspond to nominal concepts in WordNet. Additionally, we also aim at obtaining large amounts of information about extended concepts (see Definition 4.1.1, which is a special case of Definition 1.1.1) as given by disambiguated adjective-noun or noun-noun combinations, e.g., *cold water* (as a hyponym of $\texttt{water}_n^1$) or *vegetable dumpling* (as a hyponym of

$\texttt{dumpling}_n^1$).

Definition 4.1.1 - Extended concept.

A noun phrase that is not present in WordNet, but whose head noun surface form is present as a WordNet noun-sense $\texttt{h}_n^s$, is called an *extended concept*. Extended concepts are disambiguated as a hyponym of $\texttt{h}_n^s$. An extended concept is either an adjective-noun phrase, represented as ($\texttt{a}_a^s$ $\texttt{h}_n^s$) or noun-noun phrase, represented as ($\texttt{n}_n^s$ $\texttt{h}_n^s$). Examples of an extended concept include the noun-noun phrase *vegetable dumpling*, whose head noun $\texttt{dumpling}_n^1$ is present in WordNet, or, the adjective-noun phrase *green energy*, with the head noun $\texttt{energy}_n^1$.

Our input will be a large collection of text. Our output will be a set of $\mathscr{A}_c$ (see Definition 4.1.2). The arguments we expect to obtain at the end are not ambiguous words but sense-specific identifiers for noun and adjective concepts. For this, we assume the existence of a repository of noun and adjective concept identifiers. Specifically, we rely on WordNet, a well-known lexical knowledge base that distinguishes the different senses of ambiguous words like *bass* (music instrument or fish) or *green(er)* (color or environmental friendliness), while also grouping together near-synonyms like *fast(er)* and "*quick(er)*". For example,

- $\langle \texttt{car}_n^1\ \texttt{fast}_a^1\ \texttt{bike}_n^1 \rangle \implies \texttt{car}_n^1$ is faster ($\texttt{fast}_a^1$) than $\texttt{bike}_n^1$
- $\langle \texttt{melon}_n^1\ \texttt{big}_a^1\ \texttt{apple}_n^1 \rangle \implies \texttt{melon}_n^1$ is bigger ($\texttt{big}_a^1$) than $\texttt{apple}_n^1$
- $\langle \texttt{lemon}_n^1\ \texttt{sour}_a^1\ \texttt{apple}_n^1 \rangle \implies \texttt{lemon}_n^1$ is more sour ($\texttt{sour}_a^1$) than $\texttt{apple}_n^1$

Definition 4.1.2 - Comparative assertion.

A comparative assertion $\mathscr{A}_c$ is a triple $\langle \texttt{x}_n^s\ \texttt{adj}_a^s\ \texttt{y}_n^s \rangle$ where $\texttt{x}_n^s$ and $\texttt{y}_n^s$ are WordNet noun senses or extended concepts, and $\texttt{adj}_a^s$ is an adjective sense in WordNet. $\texttt{x}_n^s$ and $\texttt{y}_n^s$ are compared over a comparative adjective that is denoted by $\texttt{adj}_a^s$. The position of the arguments define that $\texttt{x}_n^s$ is usually comparably more $\texttt{adj}_a^s$ than $\texttt{y}_n^s$. Every comparative assertion is accompanied by a support score $1 \leq \Theta(\mathscr{A}_c)$. $\mathscr{A}_c$ is irreflexive and transitive but not necessarily asymmetric because $\mathscr{A}_c$ is not an absolute universal truth.

Our approach. Our approach involves first using Open Information Extraction (OpenIE) techniques to capture arbitrary comparative relationships expressed in text, in an open-ended manner. For example, the fact that seafood, on average, is perceived as healthier than a hamburger. While several OpenIE systems have been presented in recent years (Etzioni et al., 2011; Carlson et al., 2010), existing systems simply deliver textual extractions rather than semantically disambiguated arguments.

OpenIE leads to triples of surface phrases, e.g., $\langle \textit{steel}\ \texttt{\textit{sharper than}}\ \textit{wood} \rangle$. Our method goes much further by computing triples of disambiguated word senses, e.g., $\langle \texttt{steel}^2_n\ \texttt{sharp}^2_a\ \texttt{than wood}^1_n \rangle$ or $\langle \texttt{photo}^1_n\ \texttt{sharp}^1_a\ \texttt{sketch}^1_n \rangle$, where the numbers are the WordNet sense numbers for the ambiguous words.

In order to move from the original text strings to more semantically coherent relationships, our approach relies on word sense disambiguation and classification techniques to consolidate equivalent extractions as well as disambiguate the arguments of each relation predicate.

Our method uses clustering and linear optimization methods to clean and consolidate this knowledge, while also inferring new information. In the end, we obtain sense-disambiguated knowledge that properly distinguishes, for example, the temperature sense of *cool* from the hipness sense of *cool*. We recognize semantically equivalent triples using joint inference techniques.

Contributions. We make the following contributions.

1. We present the first OpenIE system for harvesting large amounts of *comparative knowledge* from Web contents.

2. We introduce a novel algorithm to organize such comparative knowledge with proper semantic rigor such that arguments are *sense-disambiguated*, by linking them to the lexical knowledge base, WordNet.

3. We publish a large, semantically refined knowledge base of comparative commonsense knowledge, which we call *CMPKB*.

4.2 KB Construction

In order to arrive at such a knowledge base given just a large text collection, we: (1) use information extraction to obtain observed textual facts, and subsequently, (2) develop a model to disambiguate and semantically organize the extractions.

4.2.1 Open information extraction

In the extraction phase, we run through the input corpus and collect all triples matching the template *(noun phrase) + (comparative predicate) + (noun phrase)*.

As noun phrases, we consider nouns listed in WordNet (*water*, *dark chocolate*), adjective-noun pairs (*cold water*) and noun-noun expressions (*football manager*) that are not in WordNet. The nouns phrases are stemmed after dropping any leading stop words (*the*, *a*, *our*, etc.). We heuristically identify the head noun of a noun phrase as the right-most stemmed noun (*water* in *cold water*) or the left-most stemmed noun when a preposition is present (*bucket* in *bucket of water*).

As comparative phrases, we allow inflections of the word *to be* followed by comparative forms of adjectives (e.g., *bigger than*, *more educated than*, etc.). We also allow them to contain modifying adverbs/negations, as e.g., in *are typically bigger than*, *are never bigger than*, or *is not only faster than*. We manually encode a list of negation phrases like *not*, *never* and some exceptions (*not only*). As a heuristic, we capture negations by assuming negations imply the opposite, in common-sense terms. Thus, *bikes are not faster than cars* is stored as a triple $\langle car \texttt{ faster } bike\rangle$. Comparative forms of adjectives are detected using WordNet. An exhaustive list of potential comparative forms of adjectives is generated by adding the suffix "er" and prefixes "more ", "less " to each WordNet adjective (*colder*, *more cold (than)*). WordNet additionally provides a list of irregular forms that cannot be generated in this way (e.g., *better*).

Using all of this information, we developed a fast pattern matcher to detect instances of this template. Our implementation is based on Hadoop MapReduce in order to quickly process large Web corpora in a distributed hardware cluster. The output of the extraction phase consists of i) left noun phrase (and its head noun), ii) relation (and its embedded adjective), iii) right noun phrase (and its head noun), iv) frequency, v) direction.

4.2.2 Disambiguation and semantic organization

The next step is to disambiguate and organize the knowledge. The original extractions are often ambiguous. For example, *hotter than* can refer to heat or to attractiveness, and *richer than* can refer to money or to calories. The left and right arguments are also often ambiguous. At the same time, our extractions do not group together equivalent forms. Given an original *observed triple* $\langle n_1{}^* \; \texttt{a}^* \; n_2{}^*\rangle$ from the information extraction step, our principal goal will be to choose relevant *grounded triples* $\langle n_1 \; \texttt{a} \; n_2\rangle$, where n_1, a, and n_2 are no longer simple strings from the text, but disambiguated word sense IDs with respect to a lexical knowledge base like WordNet.

We first present a simple local baseline model, which assumes independence across the triples. Then we describe a more advanced model, which makes use of integer linear programming problems (ILPs) and does not assume independence across triples.

Local model. Similar to state-of-the-art methods for word sense disambiguation on text (Navigli, 2009), the local model assumes that the most likely disambiguation is the one that has the highest internal coherence, while simultaneously also preferring more frequent senses. A grounded triple exhibits high internal coherence when the word senses within it are similar to each other. Thus, for every grounding $\langle n_1 \; \mathtt{a} \; n_2 \rangle$ of an observation $\langle {n_1}^* \; \mathtt{a}^* \; {n_2}^* \rangle$, a score is computed as follows:

$$\begin{aligned} \text{score}(n_1, a, n_2) \; &= \tau_{\text{NN}}(n_1, n_2) \\ &+ \tau_{\text{NA}}(n_1, a) + \tau_{\text{NA}}(n_2, a) \\ &+ \phi(n_1^*, n_1) + \phi(n_2^*, n_2) \\ &+ \phi(a^*, a) \end{aligned} \tag{4.1}$$

This score combines three different kinds of components:

- $\tau_{\text{NN}}(n_1,n_2)$: A taxonomic relatedness score between two noun senses n_1 and n_2, computed using a WordNet path similarity measure (Pedersen et al., 2004), identical to Table 3.3 in Chapter 3. In addition, if one of the two arguments is an extended concept like *ripe fruit*, we have separate senses for the first word and for the second word, so we compute two scores and take the average. If both n_1 and n_2 are extended concepts, we compute all four pairwise scores between involved senses for n_1 and senses for n_2, again taking the average. While doing this, any scores between two noun senses are computed as above using the WordNet path similarity measure, while any scores involving an adjective sense are computed as for $\tau_{\text{NA}}(\mathtt{n},\mathtt{a})$ below.

- $\tau_{\text{NA}}(n, a)$: A taxonomic relatedness score between a noun sense and an adjective sense, computed by determining the overlap between their extended WordNet glosses. The extended glosses are constructed by concatenating the original sense's gloss with the glosses of related senses in the taxonomic neighborhood. The taxonomic neighborhood of a sense includes its directly related senses (e.g., similar-to, antonym senses in WordNet). For nouns, the hypernyms and hyponyms of a given sense are also considered. We then create bag-of-words feature vectors and compute the cosine similarity.

When n is an extended concept, the relatedness is the average over the two scores between its respective component senses and the adjective a.

- $\phi(w, s)$: A prior for the sense s of a word w, computed as $\frac{1}{1+r}$, given the WordNet sense rank r. Thus, the first sense obtains $\frac{1}{2}$, the second sense $\frac{1}{3}$, and so on. For extended concepts, the sense score is the average sense score of its components.

Table 4.1 lists the formulas for the different scores used in the local model.

Table 4.1: Score computations: Local model

Prior sense score (extended concepts)	
$\phi(n\ h,\ \texttt{n}^s_n\ \texttt{h}^s_n) = \dfrac{\phi(n,\ \texttt{n}^s_n) + \phi(h,\ \texttt{h}^s_n)}{2}$	(4.2)
$\phi(a\ h,\ \texttt{a}^s_a\ \texttt{h}^s_n) = \dfrac{\phi(a,\ \texttt{a}^s_a) + \phi(h,\ \texttt{h}^s_n)}{2}$	(4.3)
Two nouns phrases (extended concepts)	
$\tau_{\mathrm{NN}}(\texttt{k1}^s\ \texttt{h1}^s_n,\ \texttt{k2}^s\ \texttt{h2}^s_n) = \dfrac{\sum_{k\in\{n_1,a_1\}}\sum_{h\in\{h_1,h_2\}}\tau(k,h)}{4}$	(4.4)
Noun phrase and adjective (sense level)	
$\tau_{\mathrm{NA}}(\texttt{n1}^s_n,\ \texttt{a2}^s_a) = lesk(\texttt{n1}^s_n,\ \texttt{a2}^s_a)$	(4.5)
$\tau_{\mathrm{NA}}(\texttt{n1}^s_n\ \texttt{h1}^s_n,\ \texttt{a2}^s_a) = \dfrac{lesk(\texttt{n1}^s_n,\ \texttt{a2}^s_a) + lesk(\texttt{h1}^s_n,\ \texttt{a2}^s_a)}{2}$	(4.6)
$\tau_{\mathrm{NA}}(\texttt{a1}^s_a\ \texttt{h1}^s_n,\ \texttt{a2}^s_a) = \dfrac{\tau_{\mathrm{AA}}(\texttt{a1}^s_a,\ \texttt{a2}^s_a) + lesk(\texttt{h1}^s_n,\ \texttt{a2}^s_a)}{2}$	(4.7)
$lesk(\texttt{n1}^s_n,\ \texttt{a2}^s_a) = \texttt{n}\vec{\texttt{1}}^s{}_{glosses} \cdot \texttt{a}\vec{\texttt{2}}^s{}_{glosses}$	(4.8)

Joint model. Although all of its components are well motivated, the local model ultimately still only has a limited amount of information at its disposition. Two or more groundings can easily end up obtaining very similar scores, without a clear winner. In particular, the local model does not consider any form of dependency across grounded triples. For example, it fails in disambiguating ⟨*tiger* `faster` *auto*⟩ by incorrectly disambiguating *tiger* to the audacious-person sense $\texttt{tiger}_n^1$ ignoring other related triples like ⟨*car* `slower` *cheetah*⟩. In reality, however, the disambiguation of a triple like ⟨*car* `faster` *bike*⟩ is highly correlated with the disambiguation of related triples, e.g., ⟨*bicycle* `slower` *automobile*⟩. We thus design a more sophisticated joint model based on the following desiderata.

a) Encourage high coherence within a triple and prefer frequent senses.

b) Encourage high coherence across chosen grounded triples.

c) Prefer same senses of a word across observations.

d) Properly handle extended concepts.

We define our Joint Model using integer-linear programs (ILPs) to encode the intuition that similar grounded triples collectively aid in disambiguation. The desired properties are soft constraints and become part of the objective. We assume we are given a series of observed triples, denoted by index i. For each observed triple $(n_1^{i*}, a^{i*}, n_2^{i*})$, we have a number of candidate groundings, denoted by index j. We refer to such a grounded triple as $(n_1^{ij}, a^{ij}, n_2^{ij})$. The ILP requires pre-computing the following coherence scores for such grounded triples.

- coh_{ij}: The coherence of an individual grounded triple, computed as $\frac{1}{3}$ of $\tau_{\mathrm{NN}}(n_1, n_2) + \tau_{\mathrm{NA}}(n_1, a) + \tau_{\mathrm{NA}}(n_2, a)$ just like the local model.
- ϕ_{ij}: The average sense score of a grounded triple, computed as $\frac{1}{3}$ of $\phi(n_1^*, n_1) + \phi(n_2^*, n_2) + \phi(a^*, a)$ from the local model.
- $\mathrm{sim}_{ij,kl}$: The taxonomic relatedness between a grounded triple with index ij and another grounded triple with index kl. This is computed as

$$\begin{aligned}&\sum_{i_1\in\{1,2\}}\sum_{i_2\in\{1,2\}} \tau_{\mathrm{NN}}(n_{i_1}^{ij}, n_{i_2}^{kl})\\&+\sum_{i_1\in\{1,2\}} \tau_{\mathrm{NA}}(n_{i_1}^{ij}, a^{kl}) + \tau_{\mathrm{NA}}(n_{i_1}^{kl}, a^{ij})\\&+\tau_{\mathrm{AA}}(a^{ij}, a^{kl})\end{aligned}$$

  where, $\tau_{\mathrm{AA}}(a^{ij}, a^{kl})$ is a semantic relatedness score between two adjectives, computed as an extended gloss overlap just as for the τ_{NA} scores.

- $\mu_{ij,kl}$: Semantically equivalent triples are detected using synonymy and antonymy information, as explained later on in more detail. We set $\mu_{ij,kl} = 1$ if the two triples are semantically equivalent, and 0 otherwise.

Table 4.2 lists the formulas for the different scores used in the local model.

Table 4.2: Score computations: Global model

Prior sense score and coherence	
$\phi_{ij} = \dfrac{\phi(n_1^*, n_1) + \phi(n_2^*, n_2) + \phi(a^*, a)}{3}$	(4.9)
$\mathrm{coh}_{ij} = \dfrac{\tau_{\mathrm{NN}}(n_1, n_2) + \tau_{\mathrm{NA}}(n_1, a) + \tau_{\mathrm{NA}}(n_2, a)}{3}$	(4.10)
Two grounded triples	
$\mathrm{sim}_{ij,kl} = \sum_{i_1 \in \{1,2\}} \sum_{i_2 \in \{1,2\}} \tau_{\mathrm{NN}}(n_{i_1}^{ij}, n_{i_2}^{kl}) + \sum_{i_1 \in \{1,2\}} \tau_{\mathrm{NA}}(n_{i_1}^{ij}, a^{kl}) + \tau_{\mathrm{NA}}(n_{i_1}^{kl}, a^{ij}) + \tau_{\mathrm{AA}}(a^{ij}, a^{kl})$	(4.11)

Given these scores, our joint model relies on the objective and constraints provided in Table 4.3. In the objective function, the x_{ij} variables capture whether a given grounding is chosen and thus the first component encourages accepting groundings with high coherence and frequent senses, just like in the local model. The second component, in contrast, allows this model to go beyond the local model by encouraging that groundings are chosen that are similar to other chosen groundings. This is a major part of what allows our joint model to make joint decisions. We use $B_{ij,kl}$ variables to reflect whether two groundings were both simultaneously chosen. In practice, we prune the linear program significantly by only instantiating such variables when they are necessary. Finally, the third and fourth components encourage us to prefer fewer of the senses s of an adjective m or noun m, respectively, across the entire graph.

In order to ensure that all variables reflect their intended semantics, we need to enforce linear constraints. Constraint (1) specifies that a grounding can be either accepted or rejected. Constraint (2) ensures that at most one grounding of an

Table 4.3: Joint Model: Integer Linear Program

maximize

$$\sum_i \sum_j (\mathrm{coh}_{ij} + \phi_{ij}) x_{ij} \ + \ \sum_i \sum_j \sum_k \sum_l \mathrm{sim}_{ij,kl} B_{ij,kl} - \sum_{m \in \mathrm{adj}} \sum_s a_{ms} - \sum_{m \in \mathrm{nouns}} \sum_s n_{ms}$$

subject to

$$\begin{array}{lcll}
x_{ij} & \in & \{0,1\} & \forall i,j \qquad (1)\\
\sum_j x_{ij} & \leq & 1 & \forall i,j \qquad (2)\\
B_{ij,kl} & \in & \{0,1\} & \forall i,j,k,l \qquad (3)\\
B_{ij,kl} & \leq & x_{ij} & \forall i,j,k,l \qquad (4)\\
B_{ij,kl} & \leq & x_{kl} & \forall i,j,k,l \qquad (5)\\
a_{ms} & \in & \{0,1\} & \forall m,s \qquad (6)\\
n_{ms} & \in & \{0,1\} & \forall m,s \qquad (7)\\
\sum_s a_{ms} & \geq & 1 & \forall \text{ adjectives } m \qquad (8)\\
\sum_s n_{ms} & \geq & 1 & \forall \text{ nouns } m \qquad (9)\\
x_{ij} & \leq & a_{ms} & \forall m,s \text{ of all adjective senses for } i{,}j \qquad (10)\\
x_{ij} & \leq & n_{ms} & \forall m,s \text{ of all } n_1 \text{ senses for } i{,}j \qquad (11)\\
x_{ij} & \leq & n_{ms} & \forall m,s \text{ of all } n_2 \text{ senses for } i{,}j \qquad (12)\\
x_{ij} & = & x_{kl} & \forall i,j,k,l : \mu_{ij,kl} = 1 \qquad (13)
\end{array}$$

observed triple is accepted. Note that the model does not require a grounding to be chosen for every observed triple. Constraints (3) to (5) ensure that the $B_{ij,kl}$ variables are 1 if and only if both x_{ij} and x_{kl} are 1.

Constraints (6) to (9) enforce that at least one word sense per word (adjective or noun, respectively) is accepted. Constraints (10) to (12) ensure that if a grounding is accepted, its word senses are marked as accepted.

Finally, constraint (13) ensures that semantically equivalent triples are tied together. Thus, if one grounding is chosen, then all equivalents must be accepted as well. The model must choose either all or none of them. The details of how we determine $\mu_{ij,kl}$ are explained below in the next section.

Maximizing the objective subject to the constraints and taking those groundings for which $x_{ij} = 1$, we obtain a set of disambiguated triples that are not only highly ranked on their own but also coherent with the groundings chosen for related observations.

4.2.3 Triple organization

In an effort to obtain a more well-defined and structured knowledge base, all semantically equivalent groundings are grouped together. For example, for appropriate chosen senses, the grounded triples ⟨*car* `faster` *bike*⟩, ⟨*bicycle* `slower` *automobile*⟩, ⟨*car* `speedier` *cycle*⟩ all express the fact that cars are generally faster than bicycles.

Equivalent comparative triples To determine equivalent triples, we make use of the following heuristics:

- Synonymy: Since groundings are disambiguated to WordNet synsets, groundings with synonymous word senses become identified, e.g., ⟨*car* `faster` *bike*⟩ and ⟨*automobile* `speedier` *bicycle*⟩.
- Antonymy: WordNet marks pairs of word senses like *fast* vs. *slow* as antonymous, i.e. as expressing semantically opposite meanings. If two adjective senses have opposite meanings, we can assume that their triples are equivalent if the arguments are in reverse order but otherwise equivalent. Thus ⟨*car* `faster` *bike*⟩ is equivalent to ⟨*bike* `slower` *car*⟩. Since WordNet's coverage of antonyms is limited, we also include indirect antonyms, considering antonymy for up to two levels of indirection (e.g., the synonym of an antonym of a synonym is also considered an antonym).
- Negation: While negation does not necessarily explicitly express the opposite, we have found that we obtain good results by treating negated adjectives (e.g., not faster than) just like antonyms (slower than). We use a small manually compiled list of negation markers for this.

More specifically, $\mathscr{A}_c$ $\langle \texttt{x1}_n^s\ \texttt{a1}_a^s\ \texttt{y1}_n^s \rangle$ and $\mathscr{A}'_c$ $\langle \texttt{x2}_n^s\ \texttt{a2}_a^s\ \texttt{y2}_n^s \rangle$ are synonyms if:

- the arguments are swapped and the relation is antonymous i.e.
  $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{a1}_a^s$ `isAntonymOf` $\texttt{a2}_a^s$;
  e.g., $\langle \texttt{car}_n^1\ \texttt{fast}_a^1\ \texttt{bike}_n^1 \rangle$ `isSynonymOf` $\langle \texttt{bicycle}_n^1\ \texttt{slow}_a^1\ \texttt{car}_n^1 \rangle$
- the arguments are not swapped and the relation is synonymous i.e.
  $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{x2}_n^s$; $\texttt{y1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{a1}_a^s$ `isSynonymOf` $\texttt{a2}_a^s$;
  e.g., $\langle \texttt{car}_n^1\ \texttt{fast}_a^1\ \texttt{bike}_n^1 \rangle$ `isSynonymOf` $\langle \texttt{car}_n^1\ \texttt{speedy}_a^1\ \texttt{bicycle}_n^1 \rangle$

Grouping equivalent triples We refer to a set of equivalent groundings as a *Csynset* (see Definition 4.2.2), similar to the notion of a WordNet synset (synonym set). We use this notion of semantic equivalence for the $\mu_{ij,kl}$ scores in the

ILP, to ensure consistency and joint assignments, as well as to provide the final output of our system in a more semantically organized form.

Definition 4.2.1 - Csynset.

A Csynset is a set of semantically equivalent comparative triples $\mathscr{A}_c$. A Csynset is accompanied by a support score aggregated over the comparative triples in the Csynset and is defined as $\sum_{\mathscr{A}_c} \Theta(\mathscr{A}_c)$.

Thus, our overall output is a large number of Csynsets expressing comparative knowledge. Every Csynset is itself a small set of equivalent grounded triples chosen by our joint model.

To make our knowledge base more consistent, we check for any Csynsets whose inverses are also present. Every triple in an antonym of a Csynset is the antonym of every triple contained in the Csynset (see Definition 4.2.2). We define antonym of a comparative triple as:

$\mathscr{A}_c$ $\langle \texttt{x1}_n^s\ \texttt{a1}_a^s\ \texttt{y1}_n^s \rangle$ and $\mathscr{A'}_c$ $\langle \texttt{x2}_n^s\ \texttt{a2}_a^s\ \texttt{y2}_n^s \rangle$ are antonyms if:

- the arguments are swapped and the relation is synonymous i.e.
  $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{a1}_a^s$ `isSynonymOf` $\texttt{a2}_a^s$;
  e.g., $\langle \texttt{car}_n^1\ \texttt{fast}_a^1\ \texttt{bike}_n^1 \rangle$ `isAntonymOf` $\langle \texttt{bicycle}_n^1\ \texttt{fast}_a^1\ \texttt{car}_n^1 \rangle$
- the arguments are not swapped and the relation is antonymous i.e.
  $\texttt{x1}_n^s$ `isSynonymOf` $\texttt{x2}_n^s$; $\texttt{y1}_n^s$ `isSynonymOf` $\texttt{y2}_n^s$; $\texttt{a1}_a^s$ `isAntonymOf` $\texttt{a2}_a^s$;
  e.g., $\langle \texttt{car}_n^1\ \texttt{fast}_a^1\ \texttt{bike}_n^1 \rangle$ `isAntonymOf` $\langle \texttt{car}_n^1\ \texttt{slow}_a^1\ \texttt{bicycle}_n^1 \rangle$

Definition 4.2.2 - Antonym of a Csynset.

An antonym of a Csynset is a Csynset such that every triple $\mathscr{A}_c$ in the antonym Csynset is an antonym of a triple in the Csynset.

We insert an `isAntonymOf` relation between the Csynsets and its antonym. This gives us, *CMPKB*, our final output knowledge base, disambiguated and connected to WordNet.

4.3 Results

Corpora. We ran our extraction system on the following two very large Web corpora.

- **ClueWeb09:** The ClueWeb09 data set[1] is a large multilingual set of Web pages crawled from the Web in 2009. We used the 504 million Web pages in the English portion.
- **ClueWeb12:** The ClueWeb12 data set[2] consists of 27 TB of data from 733 million English Web pages crawled from the Web in 2012.

Evaluation dataset. To evaluate our system, we created three test sets sampling three different kinds of triples from this raw, ambiguous data:

i) **WN:** both the left and right argument of the triple are surface forms that appear as words in WordNet, e.g., *steel*, *wood*, *photo*, *sketch*.

ii) **Extended:** both the arguments are extended concepts, e.g., *math professor*, *novice student*, *digital image*, *brush sketch*.

iii) **WN/extended:** one of the two arguments is in WordNet, the other is an extended concept.

Each of these three sample sets contained 100 randomly chosen observations. For each observation triple, human annotators were asked to choose the best possible word senses, not just surface forms. When an annotator found that none of the possible senses yields a true statement, i.e. the extracted triple is noise, and none of the senses were selected. In case of an extended concept, the annotators annotated only the head word, e.g., $\texttt{professor}_n^1$ in *math professor*, $\texttt{sharp}_a^3$ in *sharper than*, $\texttt{student}_n^1$ in *novice student*.

For development and tuning, we additionally relied on a separate set of around 40 annotated observations in order to avoid experimenting with different variants of our model on the test set.

Baselines. We consider the following two baselines.

1. **Most-frequent-sense heuristic (MFS):** The standard baseline for disambiguation tasks is MFS that maps an observation triple $\langle x^* \ \mathtt{a}^* \ y^* \rangle$ to $\langle \mathtt{x}^1 \ \mathtt{a}^1 \ \mathtt{y}^1 \rangle$ using the WordNet sense rankings. In WordNet and many other

[1] http://lemurproject.org/clueweb09/

[2] http://lemurproject.org/clueweb12/

lexical resources, sense entries are ranked such that the most frequent or important senses are listed first. For example, MFS disambiguates ⟨*car* `fast` *bike*⟩ as ⟨$\texttt{car}_n^1$ $\texttt{fast}_a^1$ $\texttt{bike}_n^1$⟩. In word sense disambiguation studies, the MFS heuristic has often been mentioned as hard to beat.

2. **Local model:** Our second baseline is the local model described earlier. For every observed triple, the top-ranked grounding with respect to the score from Equation 4.1 is selected. The local model not only uses the sense rankings but also additionally incorporates the intra-grounding coherence. Unlike our joint model, however, this baseline disregards any coherence across triples.

Results. Having run our extraction code over the ClueWeb corpora, we obtained 488,055 extractions from ClueWeb09, and, 781,216 from ClueWeb12. Together, these amount to 1,064,066 distinct extractions. This is mainly because the crawling strategies for the two ClueWeb corpora differed significantly. ClueWeb12 was created as a companion for ClueWeb09 with very different content (highly popular sites and Twitter links) and better spam detection. Thus, there is little overlap between the two corpora.

In order to evaluate our joint model, we added additional related triples from the extractions to create a graph for every observed triple to be assessed. We greedily chose the most similar observed triples up to a maximal size of 10 observed triples, and then for every observed triple, possible candidate groundings were considered. We used these to instantiate the ILP, but smartly pruned out unnecessary variables (removing $B_{ij,kl}$ variables when $sim_{ij,kl}$ is zero or near-zero). For optimization, we use the Gurobi optimizer version 5.6 (`www.gurobi.com`).

The evaluation is done separately for the three kinds of triples. Table 4.4 provides accuracy scores (95% Wilson confidence intervals) for the three different categories in the test set, and for the overall test set aggregated over all the categories.

We see that the local model outperforms the MFS baseline by a small margin. Although the local model makes use of valuable sense ranking and coherence information, it does not deliver satisfactory results. For example, the local model failed on ⟨*tiger* `fast` *auto*⟩ by incorrectly disambiguating it onto ⟨$\texttt{tiger}_n^1$ (wrong sense: strong person) $\texttt{fast}_a^1$ $\texttt{auto}_n^1$⟩.

Instead, our joint ILP model is the clear winner here, as it is able to take into account additional information about other related triples (e.g., ⟨*car* `slow` *cheetah*⟩) when making decisions. As another example, given the observed triple

Table 4.4: Test Set Results (Precision)

Approach	WN	WN/extended	extended	all
MFS	0.42 ± 0.09	0.43 ± 0.09	0.46 ± 0.08	0.43 ± 0.05
Local Model	0.47 ± 0.09	0.49 ± 0.09	0.44 ± 0.08	0.47 ± 0.09
Joint Model	0.83 ± 0.06	0.85 ± 0.06	0.80 ± 0.06	0.83 ± 0.04

⟨*pork more tender beef*⟩, our model correctly infers that the highly ambiguous adjective *tender*, with eight senses in WordNet, is not used in its initial senses (sentiment-related) but in its fifth sense (easy to cut or chew). Our model simultaneously also correctly infers that *pork* is used in its first out of two listed senses, but that *beef* is not used in its first sense (cattle reared for their meat), but in its second out of three senses (meat from an adult domestic bovine).

Overall, our knowledge base provides around a million disambiguated comparative assertions. Table 4.5 lists some examples of the type of semantically organized knowledge one can find among these assertions.

Table 4.5: Example Disambiguated Assertions

Type	Argument 1	Relation/Adjective	Argument 2
WN	$\texttt{snow}_n^2$	less $\texttt{dense}_a^3$	$\texttt{rain}_n^2$
	$\texttt{marijuana}_n^2$	more $\texttt{dangerous}_a^1$	$\texttt{alcohol}_n^1$
	$\texttt{diamond}_n^1$	sharper ($\texttt{sharp}_a^3$)	$\texttt{steel}_n^2$
WN/extended	little $\texttt{child}_n^1$	happier ($\texttt{happy}_a^1$)	$\texttt{adult}_n^1$
	$\texttt{private\_school}_n^1$	more $\texttt{expensive}_a^1$	public $\texttt{institute}_n^1$
	pot $\texttt{soil}_n^3$	heavier ($\texttt{heavy}_a^1$)	$\texttt{peat}_n^1$
extended	peaceful $\texttt{resistance}_n^1$	more $\texttt{effective}_a^1$	violent $\texttt{resistance}_n^1$
	hot $\texttt{food}_n^2$	more $\texttt{delicious}_a^2$	cold $\texttt{dish}_n^2$
	wet $\texttt{wood}_n^1$	softer ($\texttt{soft}_a^1$)	dry $\texttt{wood}_n^1$

Use-case. As an example use-case, we consider computational advertisement, following (Xiao and Blat, 2013). Advertising frequently relies on metaphors to convey attributes of products and services. The *salience* of an attribute is typically manifested very well in comparative statements. For example, with smartness as the target attribute, we can query our knowledge base for triples with `smarter` as the relationship and obtain $\texttt{dog}^1_n$, $\texttt{dolphin}^1_n$, $\texttt{pundit}^1_n$ as the most frequent left or right arguments. Similarly, with *heavier* as the relationship, the top three arguments are $\texttt{iron}^1_n$, $\texttt{air}^1_n$, $\texttt{steel}^1_n$.

To score potential representatives for a given query attribute, we computed the frequency of objects listed as left or as right arguments. Top-ranked results for three example queries that are often relevant in advertising are given in Table 4.6.

Table 4.6: Advertising Suggestions

smarter	more expensive	heavier
dog	gold	iron
dolphin	oil	air
pundit	organic food	steel

4.4 Discussion

We have presented the first approach for mining and consolidating large amounts of comparative commonsense knowledge from Big Data on the Web. Our algorithm successfully exploits dependencies between triples to connect and disambiguate the data, outperforming strong baselines by a large margin. The resulting knowledge base, *CMPKB*, is freely available at (`http://people.mpi-inf.mpg.de/~ntandon/resources/readme-comparative.html`).

Strengths:

- In this work, r in a triple $\langle x \texttt{ r } y\rangle$ is an adjectival phrase. Our methods are generic i.e. r can also be a verbal phrase. We only need to compute the verb-noun and verb-verb sense similarity analogously to adjective similarity to enable our method for verbal phrases. Our method can operate in a limited context enabling us to add semantic refinement over any existing triples extracted via OpenIE.
- Our method is scalable and does not require any labeled data. It is not tightly coupled with WordNet. We can compute the taxonomic similarities from any

lexical resource like Wiktionary that contains senses along with their glosses.

Weaknesses:

- Our global model can be computationally expensive when the graph is very large as the number of variables in the ILP will be of the order $O(N^2)$ where N is the number of triples in the graph.

  We overcome this weakness partially, by pruning out the input graph candidates (smaller graph = fewer variables). We consider only those triples that might potentially help in disambiguation of the candidate triple, e.g., to disambiguate ⟨`car fast bike`⟩, only triples like ⟨`auto speedy bike`⟩ might help but not ⟨`juice sweet water`⟩. We maintain top-k similarity lists for each word sense.

- In our setup, the extraction phase is not a major focus and thus we currently rely on simple heuristics that can easily be applied to terabyte-scale Web corpora.

  Existing techniques such as (Jindal and Liu, 2006), could potentially be used to improve our extraction phase by identifying potential comparative sentences and their elements using sequence mining techniques.

- Currently it is difficult to perform reasoning such as consistency rules and transitivity rules over Csynsets. The antonym of the Csynsets conveys exactly the opposite information and thus consistency rules like anti-symmetricity fail.

  As a solution, we could define a function that takes a Csynset and its antonym Csynset as input and selects exactly one of these. This function could return the Csynset with the larger support score.

Lessons learned:

- Global coherence is simple, robust to noise and can perform joint disambiguation.
- Comparative commonsense is not affected by reporting bias. Comparatives can be seen as an alternative view of non-comparative relations i.e. property relations and thus help in estimating salience and overcoming reporting bias to some level.

Assumptions:

- There is a strong coherence within all pair of components of a triple.

Extensions:

- Methods that mine comparable commonsense from numerical data are a complementary source; see related work in Section 4.1. This would require a KB of units (for conversion , e.g., from ft to m)
- The triple ⟨`car faster bike`⟩ increases the likelihood that ⟨`bike hasSpeed slow`⟩ and ⟨`car hasSpeed fast`⟩. Two different views can help to overcome reporting bias to a certain extent. Secondly, salience can be estimated because `fast` (`speed`) is a typical attribute associated with a `car`.

  In the next chapter, we will see that such interplay or multiple views of the same knowledge is also possible via multimodal data (i.e. text and images).

5 Commonsense on Relationships: Part-Whole

5.1 Introduction

In this chapter, we investigate the second category of commonsense relations, i.e. commonsense of relationships. This chapter investigates methods for an instance of such relations, fine-grained taxonomy of `part-whole` relations.

Part-whole relations are well explored but fine-grained semantic distinction between the sub-relations of part-whole has never been explored. In this chapter, we will present methods and results on acquisition of `part-whole` relations.

Motivation. We all know that a thumb is part of a hand, and that a goalkeeper is part of a soccer or hockey team. For machines this kind of commonsense is not obvious at all, yet many modern computer tasks – like computer vision, Web search, question answering, or ads placement – require this kind of background knowledge to simulate human-like behavior and quality. For example, suppose a visual object detection algorithm has recognized two wheels, pedals and a chain in an image or video; a smart interpretation could then harness knowledge to infer that there is a bike in this scene. This would be a novel element and potential performance booster in computer vision (Rohrbach et al., 2011). However, there is no comprehensive part-whole knowledge base available today.

State-of-the-art and its limitations. There has been considerable research to automatically acquire part-whole knowledge across several disciplines.

Philosophy. In *mereology*, there is wide consensus that the part-whole relation should be modeled as a weak partial ordering, i.e., a property that is *reflexive*, *transitive*, and *anti-symmetric* (Varzi, 2010). Winston et al. (1987) and Keet and Artale (2008) discuss semantic variants of part-whole relations in natural languages. Smith et al. (2005) discusses the specific setting of biomedical ontologies. Our work, just like WordNet, follows the conceptual framework of Winston et al. (1987).

Computational linguistics. In contrast to the ample work on lexico-syntactic patterns for hyponymy/hypernymy and taxonomy induction, there is relatively little work on extracting meronymy/holonymy concept pairs. Berland and Charniak (1999) used two Hearst patterns, on genitive forms, to extract candidate pairs and used statistical measures for ranking. However, the high ambiguity of genitive forms ('s, of) led to very limited results.

Girju et al. (2003, 2006) extended and generalized this approach by using additional, still handcrafted, patterns and adding constraints about the lexical hypernyms (in WordNet) that concept pairs need to be in a meaningful part-whole relation. For example, two concepts that belong to the WordNet senses `location` and `people`, respectively, would be disallowed for part-whole. These constraints were automatically learned by a decision-tree classifier, but required a substantial amount of training samples. For mapping words to concepts, standard word sense disambiguation techniques were used. The method achieved a precision of ca. 80% on a few 10,000 sampled sentences from news corpora.

Pantel and Pennacchiotti (2006) developed the Espresso algorithm that extended prior work on seed-based pattern induction (such as Ravichandran and Hovy (2002)) by introducing PMI-based pattern rankings. Here, seeds were concept pairs, and patterns were automatically learned. This resulted in a precision of 80% for part-whole extractions from benchmark corpora. The output pairs were not sense-disambiguated and the output size was small. Ruiz-Casado et al. (2007) harnessed Wikipedia and patterns near hyperlinks, and achieved a precision for meronymy/holonymy $\leq 70\%$ in small-scale experiments.

Recent works on acquisition of lexical relations include Tandon et al. (2011) and Ling et al. (2013). The former addressed a wide variety of commonsense relations without specific concern for part-whole, whereas the latter was geared for meronyms among biological concepts.

Ittoo and Bouma (2010, 2013) studied refined classes of part-whole relations, based on the taxonomy of Keet and Artale (2008). They extended prior work by using different seed sets for different part-whole relations extracted from Wikipedia texts, and achieved an overall precision of ca. 80% for an output of ca. 10,000 concept pairs.

None of these prior works was designed for constructing a large, fine-grained and disambiguated part-whole KB.

Knowledge acquisition. Commonsense acquisition projects like Cyc (Lenat, 1995; Matuszek et al., 2005), ConceptNet (Havasi et al., 2007; Speer and Havasi, 2012), NELL (Carlson et al., 2010; Mitchell et al., 2015), and WebChild (Tandon et al., 2014a) have compiled large amounts of commonsense knowledge.

Among these, only Cyc and ConceptNet contain a sizable number of high-quality instances of part-whole relations. Cyc has relied on manual expert input, which is expensive and does not scale. ConceptNet is based on crowdsourcing, but lacks argument disambiguation and semantic refinement. Their coverage is far from anywhere near being complete.

Automated efforts like Tandon et al. (2011) or NEIL (Chen et al., 2013) had to cope with fairly noisy inputs, like n-gram corpora or images; so their outputs for part-whole relations are quite inferior in quality compared to WordNet or ConceptNet. The NEIL project (Chen et al., 2013) has embarked on discovering part-whole and other commonsense relations about scenes by analyzing a large number of images. So far the project has acquired around a hundred instances of a generic part-whole relation.

Thus, prior part-whole KBs have major limitations:

i) The automated efforts to compile part-whole knowledge, such as NEIL or Tandon et al. (2011) conflate different kinds of part-whole relations into a single generic relation `partOf` and miss out on the semantic differences between `physicalPartOf` (e.g., ⟨wheel `physicalPartOf` bike⟩), `memberOf` (e.g., ⟨cyclist `memberOf` team⟩), or `substanceOf` (e.g., ⟨rubber `substanceOf` wheel⟩).

ii) In all part-whole KBs except WordNet, the arguments of the relations (e.g., *screen*, *notebook*) are merely words with ambiguous meaning, whereas they should ideally be unique word senses, for example, by disambiguating them onto WordNet synsets.

iii) In all part-whole KBs, the assertions are merely qualitative; there is no information about either visibility or cardinality. Existing KBs lack the distinction between *visible* and *invisible* physicalPartOf (e.g., for an ordinary human, ⟨`nose physicalPartOf human`⟩ is visible, while ⟨`kidney physicalPartOf human`⟩ is invisible). Further, it could be important to know that a bike has two wheels rather than three, and that a car has one steering wheel rather than two. These distinctions are crucial for visual applications.

iv) The coverage of part-whole knowledge is very limited. For example, ConceptNet contains only 1,086 instances of various part-whole relations in total. It has the notion of a `memberOf` relation and knows the concepts of a *cyclist* and *sport team*, yet does not have any `memberOf` information for these concepts.

Problem statement. Our goal is to automatically mine assertions for fine-grained part-whole relations. These fine-grained relations include: `physicalPartOf`, e.g., $\langle$`wheel physicalPartOf bike`$\rangle$; `memberOf`, e.g., $\langle$`cyclist memberOf team`$\rangle$; and, `substanceOf`, e.g., $\langle$`rubber substanceOf wheel`$\rangle$. Further, we aim to enrich these assertions with two new attributes: *visibility* and *cardinality*. The first indicates whether the part can be visually perceived. The second attribute defines the number of parts in the whole.

Definition 5.1.1 - Part-whole assertion.

A part-whole assertion $\mathscr{A}_{pw}$ is a triple $\langle \texttt{x}_n^s\ \texttt{r}\ \texttt{y}_n^s \rangle$ where $\texttt{x}_n^s$ and $\texttt{y}_n^s$ are WordNet noun senses or extended concepts, and $\texttt{r} \in \prec_P, \prec_M, \prec_S$ where, $\prec_P$ denotes `physicalPartOf`, $\prec_M$ denotes `memberOf` and $\prec_S$ denotes `substanceOf`. Every part-whole assertion is accompanied by a confidence score $0 \leq \Theta(\mathscr{A}_{pw}) \leq 1$.

Examples of the kind of assertions we aim to mine are: $\langle$*atleast three* $\texttt{sheep}_n^1$ `memberOf` $\texttt{herd}_n^1\rangle$ or that $\langle$*two* `license` $\texttt{plate}_n^2$ `physicalPartOf` $\texttt{car}_n^1\rangle$ or that $\langle\texttt{salt}_n^1$ `substanceOf` $\texttt{sea}_n^1\rangle$.

We use WordNet to disambiguate concepts extracted from Web and image tags. WordNet connects synsets by various relations. Relevant for us are hypernymy/hyponymy (type, subclass), which relate broader concepts to more specific ones, and three kinds of part-whole relationships: *(physical) partOf*, *memberOf*, and *substanceOf*.

Due to the nature of the part-whole relations, not every synset can be accepted as left argument (i.e., part – domain of the relation) or right argument (i.e., whole – range of the relation). For instance, `physicalPartOf` restricts both domain and range to be physical, `memberOf` restricts the range to be abstract, while `substanceOf` restricts the domain to be substance. Therefore, we first consider the synsets that are hyponyms of *Abstract Entity* (V_a) or *Physical Entity* (V_p). We assume $V_a \cap V_p = \emptyset$. WordNet has exceptions to this disjointness: around 1,000 synsets have hypernyms in both V_a and V_p (McCarthy, 2001), e.g., `roller coaster`. For these we only use hypernyms in V_p.

Abstract entities include, for example, *teams*, *organizations*, *music*, *poems*, etc. Physical entities include everything that one can possibly touch, such as *bikes*, *cars*, *fingers*, *bones*, etc. Furthermore we distinguish the synsets under *Substance*, denoted as V_s, which is a hyponym of the physical entity, so that $V_s \subset V_p$. *Substance* synsets include for examples *iron*, *oxygen*, *clay*, *oil*, etc.

Table 5.1 summarizes the part-whole relations with our type restrictions.

Table 5.1: Part-whole relations with type restriction

r	**domain**(r)	**range**(r)	**example**
$\prec_P$	V_p	V_p	wheel $\prec_P$ bike
$\prec_M$	$V_p \cup V_a$	V_a	cyclist $\prec_M$ team
$\prec_S$	V_s	V_p	rubber $\prec_S$ wheel

Approach. Our method includes an extension of pattern-based extraction that substantially improves the extraction quality on large and noisy text corpora like the Wikipedia full text and the Google n-gram collection. Subsequently, we further eliminate false positives by devising rules for constraint checking, and we also infer additional assertions by logical deduction rules.

For high coverage, these rules need to consider candidate assertions over multi-word noun phrases. To properly handle these, we have devised a new technique to integrate such phrases into WordNet. Finally, we developed novel techniques to enhance the assertions with visibility attribute values by tapping into image tags obtained from 100 Million Flickr images, and cardinality attribute values by tapping into Google-books n-grams from multiple languages.

Our method proceeds in three phases:

- *Phase 1 – KB Construction:* We extend statistical techniques for pattern-based extraction by introducing weighted seeds in candidate scoring to improve the output quality on large and noisy text corpora. This phase gives us candidate assertions for part-whole relations, and we map the arguments of the candidate assertions to WordNet senses.
- *Phase 2 – KB Enrichment:* The candidate assertions from Phase 1 still contain many false positives. We devise logical inference rules to enforce consistency and obtain cleaner assertions. Additionally, we propose deduction rules for deriving additional assertions, enlarging the PWKB. A key novelty here is that these rules apply to assertions over multi-word noun phrases. We have developed techniques to handle these by carefully extending the WordNet taxonomy.
- *Phase 3 – KB Enhancement:* We enhance the part-whole relations by two new attributes *visibility* and *cardinality*. We develop a novel technique that

exploits image tags to detect the visibility of the part in the whole. For cardinality, we exploit the grammatical structure of German and Italian to handle cases which cannot be easily dealt with in English.

Contributions. The contribution of this work is to overcome the limitations of the state-of-the-art and build a comprehensive, semantically organized, high-quality part-whole KB, *PWKB* for short.

We overcome all four aforementioned limitations of the state-of-the-art:

i) Distinguishing `physicalPartOf`, `memberOf`, and `substanceOf`,

ii) Mapping all arguments of our assertions to WordNet senses, thus eliminating ambiguity and redundancy,

iii) Inferring visibility and cardinality information for many instances of the various part-whole relations, and

iv) Building a large PWKB with about 6.75 million assertions – orders of magnitude larger than WordNet or ConceptNet while having similar of better quality.

5.2 KB Construction

We construct our PWKB (Part-Whole Knowledge Base) by introducing novel extensions of the state-of-the-art pattern-based extraction techniques (with a new scoring model) and disambiguation techniques (extending from words to phrases).

Extraction of $\prec_P$,$\prec_M$,$\prec_S$. We use a pattern-based information extraction approach, following (Tandon et al., 2011), to obtain candidate patterns from text. This method requires a small number of high-quality seed assertions to bootstrap the identification of extraction patterns. As the text source, we use the full text of Wikipedia and the Google-Web n-grams. As seeds, we pick 1,200 instances of the physicalPartOf, memberOf, and substanceOf relations of WordNet. Patterns are automatically obtained by matching the seed pairs in our input corpora, and extracting the essential words between the two concepts (i.e., considering only words with certain part-of-speech tags). For example, the seed `goalkeeper` $\prec_M$ `team` leads to the extraction pattern <Noun> *of the* <Noun>.

Scoring model for candidate ranking. The quality of patterns varies widely. We identify good patterns regarding two aspects: i) patterns should co-occur with many distinct seeds (not just very frequently with some seeds), and ii) patterns should discriminate between the three part-whole relations that we aim to populate. The Specificity Ranker (SR) of (Tandon et al., 2011) already considers the first aspect. However, we improve this prior model by introducing a notion of weighted support and by considering the second aspect.

Let $\sigma_{SR}(p_i)$ denote the score that SR assigns to pattern p_i, using all seeds for all relations, and let $\sigma_{SR}(p_i|R_j)$ be the score if only seeds for relation R_j (e.g., $\prec_M$) are used. We leverage these SR scores as weights for scoring the candidate assertions that result from the obtained patterns. The *weighted support* of candidate assertion a_k is

$$supp(a_k) = \sum_{p_i} \sigma_{SR}(p_i)\delta(p_i, a_k)$$

where, $\delta(p_i, a_k)$ is 1 if p_i co-occurs with a_k and 0 otherwise. Analogously, we define the *R_j-specific weighted support* for a_k as

$$supp(a_k|R_j) = \sum_{p_i} \sigma_{SR}(p_i|R_j)\delta(p_i, a_k)$$

This is the basis for defining the *discriminative strength* of a_k for R_j:

$$str(a_k|R_j) = \sum_{\nu \neq j} \left(\frac{supp(a_k|R_j)}{supp(R_j)} - \frac{supp(a_k|R_\nu)}{supp(R_\nu)} \right)$$

where, $supp(R_j) = \sum_{a_k} supp(a_k|R_j)$.

Finally, we normalize both support and strength, to yield values between 0 and 1, and combine them into the overall score of assertion candidate a_k:

$$\sigma(a_k) = \frac{e^{supp(a_k)}}{1 + e^{supp(a_k)}} \frac{e^{str(a_k)}}{1 + e^{str(a_k)}}$$

Thus, we can rank candidates and apply thresholding to reduce false positives.

Mapping words and phrases to senses. The selected assertions are word pairs and hence ambiguous. We extend the IMS (ItMakesSense) tool (Zhong and Ng, 2010) to disambiguate words onto WordNet senses. IMS operates at a word level and can thus not handle multi-word noun phrases. Our novel contribution is to add a new layer on top of IMS to additionally disambiguate multi-word noun phrases including extended concepts, e.g., *mountain bike*, *lightweight racing bike* (see Definition 4.1.1 of extended concepts introduced in Chapter 4). First, we perform noun phrase chunking on the input sentence where the assertion occurs.

We use the widely used OpenNLP Chunker (`opennlp.apache.org`). Next, for every noun phrase, we identify and disambiguate its lexical head using IMS (e.g., the out of WordNet phrase `the electrical plant` to `plant`[1]). This yields canonicalized assertions for our part-whole relations, with unique senses and free of redundancy. This also enables us to apply type-restrictions based on the domain and range of the relations (see Table 5.1) to further filter the assertions.

5.2.1 KB Enrichment

We enrich the PWKB by proposing logical inference rules for deduction and consistency.

5.2.1.1 Increasing coverage

We improve the PWKB coverage by applying the following two deduction rules (see Table 5.2).

Table 5.2: Deduction rules for increasing coverage

C1.	Transitivity:	$(a \prec b \wedge b \prec c) \Rightarrow a \prec c$
C2.	Inheritance:	$(a \prec b \wedge c \ \texttt{hyponymOf} \ b) \Rightarrow a \prec c$

We exploit the fact that physicalPartOf and substanceOf are transitive (Keet and Artale, 2008) and perform a 2-step transitive closure. We do not consider the full transitive closure as it tends to produce too many trivial assertions (e.g., `atom` $\prec_P$ `matter`). We propose $C2$ to propagate part-whole relations to hyponyms of the whole. For example, having the knowledge: `wheel` $\prec_P$ `bike` and `mountain bike hyponymOf bike`, we infer: `wheel` $\prec_P$ `mountain bike`. Here, `mountain bike` is an extended concept, thus, $C2$ also applies to extended concepts.

While $C2$ is useful in many cases (e.g., deducing that mountain bikes have wheels, too), it also comes with the risk of generating false assertions, e.g., that mountain bikes have headlights. Here we rely on the pragmatic assumption that WordNet's hyponymy links induce subsumptions between the sets of instances for the respective synsets/classes. Our experimental evaluation reports on the benefits and risks of the deduction rules.

Table 5.3: Consistency check rules for increasing quality

Q1.	Irreflexivity:	$\neg(a \prec a)$
Q2.	Acyclicity:	$\neg(a \prec b \wedge b \prec a)$

5.2.1.2 Improving quality

We improve the PWKB quality by checking for inconsistencies and eliminating false assertions, using the constraints listed in Table 5.3.

We drop assertions that violate the first type of inconsistency. For the second type, we detect all cycles of length ≤ 3 and break each cycle by dropping the assertion with the lowest score computed in Phase 1.

5.2.2 KB enhancement

We enhance the PWKB assertions by introducing two new attributes: visibility and cardinality.

5.2.2.1 Visibility attribute

Our goal is to determine which physical parts of a whole are visible (for an ordinary human, e.g., not a mechanic or surgeon). If a and b co-occur in an image and we have the knowledge that $a \prec_P b$, then a is visible. The superscript V is used for $\prec_P$ to denote visibility (e.g., `license plate` $\prec_P^V$ `car`) while NV denotes non-visibility (e.g., `automatic brake system` $\prec_P^{NV}$ `car`).

We could consider obtaining visibility information directly from images, or alternatively, from annotations of images like captions and tags. To compare these two approaches, we computed co-occurrence statistics from i) running a visual object detector (LSDA (Hoffman et al., 2014)) versus ii) user-provided tags that annotate Yahoo! Flickr images (Shamma, 2014). We compared both results against the already compiled $\prec_P$ assertions for a sample of 100K Flickr images. About 20% of the images show overlap between object class names and tags, on average 1.5 words. This suggests that the two approaches are complementary. Consider an image depicting a man playing golf. The Flickr tags are `golfer, club, field`, while the detections are `button, cap`. We obtained ca. 12,000 positive matches with LSDA object detections versus ca. 26,000 with tags. Thus, image annotations give better coverage.

We thus used Flickr tags to compute $\prec_P^V$ at large scale. We set the visibility of $a \prec_P b$ to true, if a and b co-occur as tags of at least a certain number of Flickr images. In the experiments, we set this co-occurrence threshold to two.

5.2.2.2 Cardinality attribute

Consider the computer vision task of recognizing different types of cycles (unicycle, bicycle, tricyle). Knowing that a unicycle has one wheel, bicycle has two, whereas tricycle has three wheels, will help the object detector. This motivates us to further extend the PWKB by cardinality information, where we distinguish the cases *1,2,3+* and *uncountable* denoted as ω. The uncountable case applies, for example, to the fur of a dog or pebbles of a beach.

Definition 5.2.1 - Cardinality.

We represent the cardinality as an attribute that we add to the $\prec_P$ and $\prec_M$ relations, and denote it by a superscript c; e.g., `wheels`$\prec_P^{\{2,V\}}$ `bike` denoting that a bike has two visible wheels.

The method to infer c in `a` $\prec_{r\in\{P,M\}}^c$ `b` has three steps:

1) Determine whether a and b are countable. We use `wiktionary.org` to look up if a word is countable.
2) If the dictionary does not have that information for a, then we compute the frequencies $f_{sin}(a)$ and $f_{plu}(a)$ of the occurrences of a in singular and plural form within a text corpus, using standard grammar rules. If $f_{sin}(a) \gg f_{plu}(a)$ or $f_{plu}(a) \gg f_{sin}(a)$, then we consider a to be uncountable. The threshold for these comparisons is determined from a set of known uncountable concepts.
3) We compare the grammatical forms of a and b. If the majority of a and b occurrences in the same sentence is in the form {singular, singular} (e.g., {handle, bike}), then we set $c = 1$. If the majority of occurrences has the form {plural, singular} (e.g., {wheels, bike}), and the supporting patterns include a numeric token (e.g., 2, 3, . . .), numbers in text forms (”two”, ”three”, . . .), or cues such as “both” or “couple of”, then we set $c = 2$ for patterns indicating 2, and $c = 3+$ for all others. For the remaining cases where a, b co-occur in the forms {singular, plural} or {plural,plural}, we use default settings: $c = 1$ for $\prec_P$ and $c = 3+$ for $\prec_M$.

As English articles and determiners (“the”, “some”, “any”, etc.) do not easily discriminate singular and plural, Step 3 is error-prone. We thus tapped German and Italian corpora (Google-books n-grams) where plural forms are more easily detectable by variants of articles and inflections of nouns. For the resulting assertions, we use Wiktionary to map the German or Italian words back to English.

5.3 Results

Input data. We construct PWKB from the following:

i) *Google Web 1T N-gram* Dataset Version 1 (Brants and Franz, 2006) which contains frequencies of n-grams (n=1,..,5) for English web pages;

ii) *Wikipedia* (2010 snapshot) (Shaoul, 2010) which contains all English Wikipedia articles as of April 2010;

iii) *Google books n-grams* in English, Italian and German (2010 snapshot) which contains POS-tagged 4-grams and 5-grams from millions of books;

iv) *Yahoo! Flickr* images (Shamma, 2014), which contains 100 million images from `www.flickr.com` with title, description, and tags.

Baselines. We consider two types of baselines: *KB baselines* and *methodology baselines*.

As KB baselines, we consider the manually constructed *WordNet (WN)*, the recall-oriented text-based *Specificity Ranker (SR)* of (Tandon et al., 2011), the image-based *NEIL* (Chen et al., 2013), and the crowdsourcing-based *ConceptNet (CN)* of (Havasi et al., 2007). The part-whole relations of SR and CN are not refined into the more specific relations that PWKB has. To make a fair comparison with SR and CN, we partitioned its assertions into the relations $\prec_P$, $\prec_M$, $\prec_S$ by domain-range type restriction (see Table 5.1), and set the visual attribute in case the arguments of $\prec_P$ map to Flickr tags (identically to our method). SR and CN contain many part-whole assertions that are encyclopedic rather than commonsense (e.g., Castro-district partOf California), in addition to noise (e.g., misspellings). Such concepts cannot be mapped to WordNet, so we drop them. Further, for SR and CN, we optimized the score thresholds for coverage. This explains the difference in numbers from original papers on CN and SR versus our setting.

As methodology baselines for scoring assertions, we include the widely used *Espresso* (Pantel and Pennacchiotti, 2006) and SR, both run on our input data. For word disambiguation, we compare against the widely used and strong *Most Frequent WordNet Sense (MFS)* heuristic. For the quality of noun phrases, *NPMI*

Table 5.4: Precision (first line) and coverage (second line)

	$\prec_P$	$\prec_M$	$\prec_S$	vis.	card.	overall
WN	1.00	1.00	1.00	1.00	-	1.00
	12892	3714	609	1304	-	17215
SR	0.19	0.20	0.23	0.16	-	0.20
	0.49M	0.49M	0.15M	0.15M	-	1.13M
CN	0.82	0.45	0.43	0.85	-	0.68
	921	516	56	665	-	1493
NEIL	0.15	-	-	0.15	-	0.15
	68	0	0	68	-	68
PWKB	0.89	0.96	0.71	0.98	0.80	0.89
	6.65M	0.04M	0.06M	0.74M	6.69M	6.75M

alone is used as a baseline.

PWKB statistics and evaluation. In total, PWKB contains 6.75 Million assertions for the three fine-grained part-whole relations, with disambiguated arguments, and, to some extent, with the two additional attributes. To evaluate the quality of PWKB, we compiled a random sample of 1000 assertions from $\prec_P$,$\prec_M$,$\prec_S$, with at least 200 assertions from each relation. We relied on human annotators to judge each assertion. An assertion was marked as correct if the judge stated that the disambiguation of the arguments was correct and the part-whole relation was correct.

For the baselines, we generously evaluated the assertions based on their surface forms as the baselines do not have disambiguated arguments. We compute the precision as $\frac{c}{c+i}$, where c and i are the counts of correct and incorrect assertions, respectively. For statistical significance, we computed Wilson score intervals for $\alpha = 95\%$ (Brown et al., 2001). The inter-annotator agreement for three judges in terms of Fleiss' κ was 0.78. We used majority voting to decide on the gold-standard labels.

The per-relation results are reported in Table 5.4. PWKB clearly outperforms all baselines in terms of coverage. In terms of quality, PWKB has an overall average precision of 89%, which seems good enough for many downstream applications (e.g., in computer vision) where commonsense can be used for distant

Table 5.5: PWKB anecdotal examples

mouth#1 $\prec_P^{\{1,V\}}$ man#1	electron#1 $\prec_P^{\{3+,NV\}}$ atom#1
sheep#1 $\prec_M^{3+}$ herd#1	musician#2 $\prec_M^{2}$ duet#2
fibre#1 $\prec_S$ cloth#1	steel#1 $\prec_S$ boiler#1

supervision or as a prior in probabilistic computations. Such applications need to cope with uncertain inputs anyway, so ∼90% precision is useful.

PWKB is much larger than all prior KBs while having higher precision than all except the manually curated WordNet. This holds also for the visibility assertions, where we outperform NEIL, constructed from 2M images, by an order of magnitude.

Table 5.5 presents some anecdotal results from PWKB.

Evaluating the PWKB construction pipeline. We evaluated the performance of the components of the three-phase PWKB pipeline. For each phase, we had three judges assess the output. For statistical significance, we again computed Wilson score intervals for $\alpha = 95\%$.

For the first phase – the initial construction of $\prec_P$,$\prec_M$,$\prec_S$, assertion ranking is the most important component which in turn relies on the ranking of patterns. Our assertion ranking model (0.85±0.05) outperforms the baseline Espresso ranking (0.34±0.07) and also the Specificity Ranker (0.55 ±0.06) by a large margin. For the disambiguation of arguments, our IMS-based method (0.80±0.07) achieves substantially better precision than the MFS baseline (0.70±0.07). Table 5.7 lists some prominent patterns for the three part-whole relations. Note that some of them are of mixed quality: good for recall, but poor in precision – for example, "*y's x*" for $x \prec_P y$, which would be matched by *Alice's husband*. Note, however, that the patterns are further restricted by the domain and range of the relations (refer Table 5.1). Candidates such as (x=Alice, y=husband) are rejected because Alice is an instance rather than a concept of WordNet type "*physical entity*".

For the second phase – enrichment, our noun phrase ranker (0.60±0.04) significantly outperforms the baseline NPMI (0.25 ± 0.05), yielding 36,498 high quality noun phrases that we attach to WordNet. We performed an ablation study on the influence of the logical rules; Table 5.6 shows the results. The C

rules for deduction increased the coverage from ca. 382K assertions to 6.75M. The Q rules for constraint checking, on the other hand, were able to remove nearly 150K false assertions that exhibited inconsistencies. For coverage, each of the two rules individually yields a major increase in the size of the PWKB; their combined effect boosts the size even more. Note, though, that even without any logical rules at all, PWKB with 382K assertions is already an order of magnitude larger than WordNet or any other prior KB of similarly high quality.

Table 5.6: Ablation study on the logical rules of Phase 2

	No Rule	+Rule 1	+Rule 2	+Rule 1,2
C rules	382K	+55K	+700K	+6.4M
Q rules	+6.4M	-476	-146K	-146K

In the third phase – cardinality inference, our method achieved a precision of 0.80±0.07, significantly improving upon relying solely on English (0.61±0.09). As for the cardinality values, we found that we achieve a high precision for cardinalities 1 and 2. However, we did not compute many assertions with 3+. This was because our heuristic method preferred a cardinality of ω (uncountable) in many cases.

5.4 Discussion

We presented the methodology for automatically constructing a large, high-quality KB of part-whole relations. We improve the state-of-the-art in several ways:

Table 5.7: Prominent patterns for PWKB relations

$\prec_P$	$\prec_M$	$\prec_S$
y have x	x (be) member of y	y (be) made of/from x
y 's x	x be in y	x found in y
x be part of y	x be of y	y (be) composed of x

i) We capture many instances for the refined relations `physicalPartOf`, `memberOf`, and `substanceOf`,

ii) We disambiguate the arguments of assertions onto WordNet senses,

iii) We additionally infer visibility and cardinality information for part-whole instances, and

iv) We do all this at a very large scale using a novel combination of statistical pattern-based extraction and logical reasoning.

The resulting knowledge base, *PWKB*, is freely available at (`http://people.mpi-inf.mpg.de/~ntandon/resources/readme-partwhole.html`). *PWKB* contains more than 6.75 million assertions; sample-based manual assessment shows that this output is of high quality.

Strengths:

- The building blocks in this work require only little variations in order to be used for other types of commonsense relations (e.g., topological or spatial relations). For example, spatial knowledge like the location of an object including near/above/under is also visually verifiable relations and have multiple sub-relations.

  Our extensions of a weighted version of pattern/assertion ranking can be easily applied to any pattern-based IE setting where a seed can potentially belong to multiple relations as observed in several commonsense relations (including `hasProperty`).

- Our dataset does not contain trivial assertions like atom $\prec$ man. To achieve this, we limit the transitive closure to paths up to length two. To keep the error rate under control, we apply the transitivity rule only to physical objects and we limit the transitive closure to paths up to length two.

Weaknesses:

- Inheritance based deduction rule $C2$ in Table 5.2 may lead to incorrect assertions. For instance, if we know that `lace` $\prec_P$ `shoe` and `laceless shoe hyponymOf shoe`, then `lace` $\prec_P$ `laceless shoe`. This is incorrect because a `laceless shoe` obviously has no lace. In the evaluation (see Table 5.6), we found that there are very few exceptions of this kind and this rule gives an impressive increase in coverage.

- Our cardinality estimation method does not take into account the confidence (or support frequency) of the cardinality estimates from multiple languages.

One solution would be to re-scale/normalize the frequencies as the dataset sizes across different languages have a large variance. Currently, frequency estimates from English are the largest.

Lessons learned:

- Multimodal data has an ensemble effect and provides visually salient part-whole relationships. This helps overcome any reporting bias in text.
- Image tags are a good substitute for prominent object detections in the image. With an overlap of more than 20% between the image tags and detected objects, we found that these 20% overlap prunes non-salient objects, e.g., button of a shirt in an image depicting a *golfer*. We call such a visual verification as *quasi-visual verification* because we use the tags instead of the low-level features of an image overcoming the low accuracy and scalability challenges associated with object detection.

Assumptions:

- We follow the approach of existing works (Smith, 1995) which considers commonsense assertions to be true even if there are a few counter-examples (e.g., birds can (usually) fly although penguins cannot).
- We assume that if a *part* and *whole* is visible or appear as tags in an image, then the part-whole relationship is visible. Similar to the previous assumption, we assume this to be true even if there are few counter-examples (e.g., some parts are visible to a mechanic or a surgeon but invisible to the majority).

Extensions:

- We estimate the cardinality using majority voting over multiple languages. We can scale this to more languages and estimate a weighted aggregation over these languages.
- We can perform reasoning over cardinality using rules to infer for instance, `hands` $\prec^{\{2\}}$ `woman` given `hands` $\prec^{\{2\}}$ `human`.
- We can mine spatial commonsense with a visibility attribute using the approach described in this chapter. Visible spatial commonsense could provide knowledge about what concepts are likely to co-occur, leading to priors for object detection.

6 Commonsense on Interactions

6.1 Introduction

In this chapter, we investigate the third category of commonsense relations: interactions. This chapter investigates methods for acquisition and organization of knowledge about `activities`.

Methods for acquisition of a holistic, multimodal activity commonsense knowledge base have never been explored before. In this chapter, we present methods and results on acquisition of activity commonsense.

Motivation. With groundbreaking new products like Amazon Echo as well as assistants like Google Now, Microsoft's Cortana, and Apple's Siri, there is a strong need for commonsense knowledge enabling smart interpretation of queries relating to *everyday human activities*.

Intelligent systems need background knowledge about human activities. For example, consider an activity: *climbing a mountain*. An intelligent system should know that this involves a participant — a human, especially a *climber*, and the typical location is a mountain, it is a daytime activity. Additionally, *climbing a mountain* and *hiking a hill* are semantically equivalent and one needs to go out of the house to go for hiking. A visual representation of the activity further helps the system to identify the activity and the scene. Beyond the kind of knowledge discussed in the previous chapters, activity commonsense will take the intelligence of machines to the next level.

State-of-the-art and its limitations. Interest in human activities goes back to Schank and Abelson's early work on scripts (Schank and Abelson, 1977), where procedural knowledge was gathered manually. More recently, such knowledge has been crowdsourced via Amazon Mechanical Turk (Regneri et al., 2010), but this data only covers 22 stereotypical scenarios. Other research has developed ways to mine activity knowledge from the Web using text analysis (Chambers and Jurafsky, 2009) and deep neural networks (Modi and Titov, 2014). These methods aim at solving small temporal ordering tasks. Shibata and Kurohashi

(2011) analyze a collection of event similarity data, but do not construct any new activities.

To gather activity knowledge from multimodal data, Chen et al. (2013) attempt to mine a large-scale collection of simple conceptual knowledge from images, e.g., that wheels are parts of cars. However, this work does not recognize activities. Regneri et al. (2013) relate crowdsourced activity descriptions to videos. However, this is a very small collection of only 26 activity types. Varadarajan and Vincze (2012) present a KB of object affordances for robotics, but cover only 250 objects. In the vision community, activities have been much less explored than analyzing objects in images. The root problem is that there is no comprehensive list of activity classes with corresponding images for training.

Taxonomy induction has a rich body of prior work in NLP and AI. However, this is primarily for type hierarchy (`isA`, `subclassOf`) between general concepts (classes, noun senses) (see, e.g., Ponzetto and Strube (2011); Pasca (2014a) and references given there). There is little research on taxonomy induction for verbal phrases (Lin and Pantel, 2001; Chklovski and Pantel, 2004; Nakashole et al., 2012). However, this line of work does not consider rich attributes for actions, and is about general verbs rather than focused on human activities.

Formal upper-level ontologies such as Cyc (Matuszek et al., 2005) and SUMO (Niles and Pease, 2001) contain some activity knowledge like agents involved in concepts expressed by verbs. For example, SUMO knows that *kissing* involves two humans as agents and their lips. However, this is manually modeled and the amount of activity knowledge in these ontologies is tiny. These ontologies focus on knowledge that is expressible in first-order logic, and lack commonsense knowledge such as participants, typical locations, times, temporal alignment between activities. Thus, manual and curated approaches are very limited in size.

ConceptNet comes closest to this task because it possesses some activity commonsense including temporal relations like `hasSubevent`, `hasFirstSubevent`, `hasLastSubevent`. It does not provide direct interpretation of activity attributes because it does not distinguish between an activity and a concept explicitly. Further, the activities in ConceptNet are not disambiguated. It does not contain information about some attributes like participating agents. There is no visual data attached to concepts or activities in ConceptNet. The knowledge about activities in general is not organized explicitly as expected in an *activity commonsense KB*.

Thus there is an important void of a *semantically organized activity commonsense KB* that needs to be addressed.

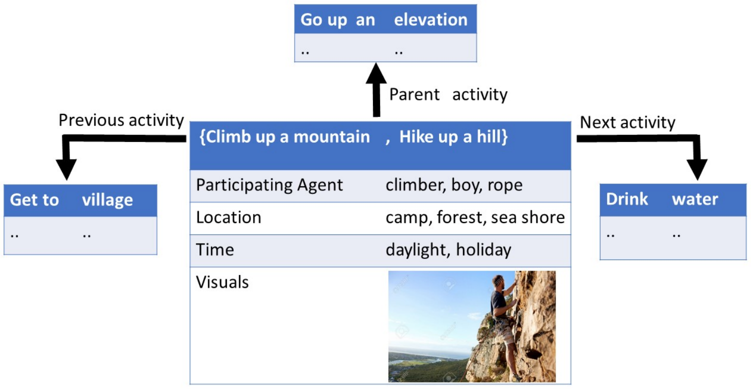


Figure 6.1: Activity Frame Example

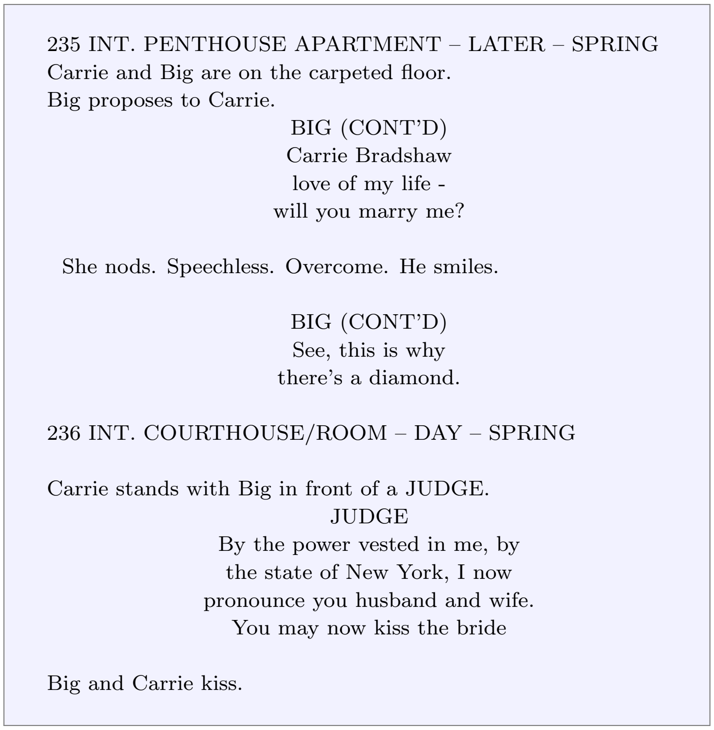
Problem statement. The goal of this work is to fill this void by automatically compiling large amounts of knowledge about human activities from *narrative text*. For example, *climbing a mountain* should be a known activity, along with attributes like participating agents — *a human*, especially *a climber*, typical location, and time of day. This knowledge should be organized in a frame-style representation, as illustrated in Figure 6.1. Further, the activities must be semantically grouped (here with *hiking up a hill*), and these semantic groups must be hierarchically arranged. These activity groups should also be temporally linked to typical previous and next activities. Having this sort of data can greatly improve computer behavior in tasks like natural language dialog, scene understanding, or video search.

Recent work in computer vision (Rohrbach et al., 2012) has manually compiled a small collection of activity scripts, based on short videos about cooking. This contains about 65 different activities such as *melting butter* or *cooking pasta*, with attributes *tool=pan* or *tool=sieve*.

Our goal is to broaden and automate the construction of these kinds of semantic frames, in order to populate a comprehensive *activity knowledge base*, in which, similarly to previous chapters, all concepts are sense-disambiguated and thus canonicalized with regard to high-quality linguistic resources like WordNet or VerbNet (Schuler et al., 2009; Kipper et al., 2006).

The input to our methods is primarily scripts about movies or episodes of TV series. Figure 6.2 shows an example from the movie *Sex and the City* (obtained from the website `imsdb.com`). Such script data is suitable to our problem as opposed to other resources. Resources such as news articles and Wikipedia articles mostly contain encyclopedic activities. Blogs and personal diaries might contain some activity knowledge but they may not contain temporal granularity of activities that we need (previous and next activities). These resources would usually not contain the aligned visual data that we need.

Although scripts are in a free format, we can exploit some structure. Specifically, there are cues for detecting scene boundaries, we can identify speakers, and we can extract short descriptions about the setting of a scene that typically precede the actual dialog. In addition, there are short narrative texts in between dialogs. Our methods are primarily geared for narrative snippets such as *Big proposes to Carrie* or *Big and Carrie kiss.* The next section discusses how to further process these snippets and extract semantically cleaner information.



235 INT. PENTHOUSE APARTMENT – LATER – SPRING
Carrie and Big are on the carpeted floor.
Big proposes to Carrie.

BIG (CONT'D)
Carrie Bradshaw
love of my life -
will you marry me?

She nods. Speechless. Overcome. He smiles.

BIG (CONT'D)
See, this is why
there's a diamond.

236 INT. COURTHOUSE/ROOM – DAY – SPRING

Carrie stands with Big in front of a JUDGE.

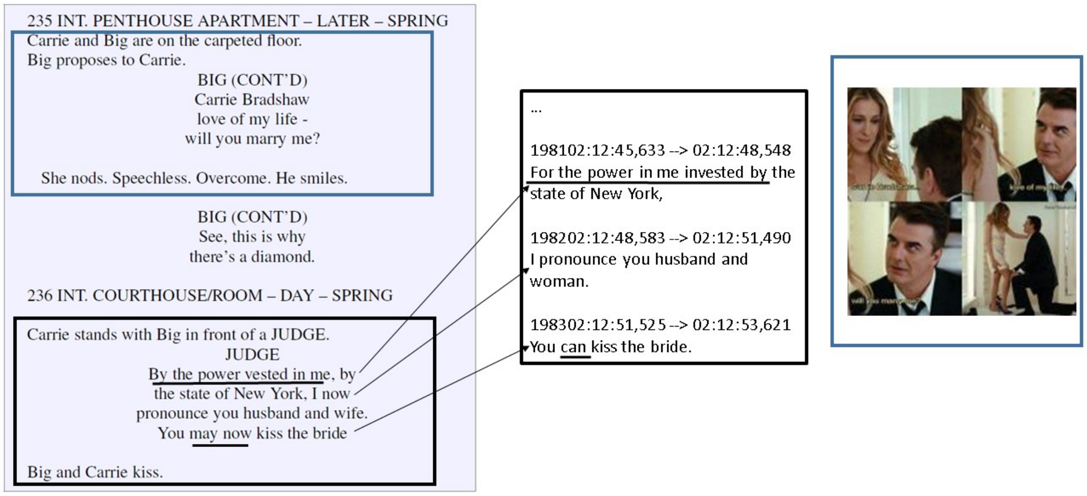
JUDGE
By the power vested in me, by
the state of New York, I now
pronounce you husband and wife.
You may now kiss the bride

Big and Carrie kiss.

Figure 6.2: Excerpt from a movie script

Obviously, individual scripts may be too noisy for automated methods to extract any meaningful information. Our method leverages that certain cues for activities appear in several scenes of different movies. Further, our scope is beyond movie scripts, including sitcoms, TV series, and novels, providing us with a broad spectrum of activities and higher redundancy.

We treat *verbal phrases* in narrative snippets as surface expressions for activity candidates. Using NLP techniques, this gives us cues such as *propose to a woman* and *kiss someone.* Generally, we extract *verb-object* pairs, where the verb can



235 INT. PENTHOUSE APARTMENT – LATER – SPRING
Carrie and Big are on the carpeted floor.
Big proposes to Carrie.
BIG (CONT'D)
Carrie Bradshaw
love of my life -
will you marry me?

She nods. Speechless. Overcome. He smiles.

BIG (CONT'D)
See, this is why
there's a diamond.

236 INT. COURTHOUSE/ROOM – DAY – SPRING

Carrie stands with Big in front of a JUDGE.
JUDGE
By the power vested in me, by
the state of New York, I now
pronounce you husband and wife.
You may now kiss the bride

Big and Carrie kiss.

```
...

198102:12:45,633 --> 02:12:48,548
For the power in me invested by the
state of New York,

198202:12:48,583 --> 02:12:51,490
I pronounce you husband and
woman.

198302:12:51,525 --> 02:12:53,621
You can kiss the bride.
```

Figure 6.3: Excerpt from a movie script and its corresponding mapping to visuals via the subtitles, using techniques from Laptev et al. (2008)

.

have a preposition (e.g., *propose to*) and the object is a noun phrase, potentially a multi-word phrase (Definition 6.1.1 defines an activity). Such a definition of an activity provides a balance between redundancy (e.g., having subject verb object as an activity leads to drop in redundancy) and specificity (e.g., having only a verb as an activity loses semantics quickly). Initially, these are still ambiguous words that may have many different meanings. Our methods map both verb and object to unambiguous senses in WordNet. This is crucial for semantic interpretation, and also key to being able to combine cues from different scenes and to organize activities in a clean taxonomy. In the example, we would obtain $\texttt{propose}^5$ $\texttt{woman}^1$ where 5 and 1 are the WordNet sense numbers of the ambiguous words *propose* and *woman*, respectively. The result of this sense disambiguation forms the core of an activity frame.

Definition 6.1.1 - Activity.

An *activity* consists of ($\texttt{v}$,*prep*,$\texttt{o}$) where $\texttt{v}$ is a WordNet sense for a verb or verb phrase and $\texttt{o}$ is a WordNet sense for a noun or noun phrase and *prep* is a surface form preposition linked with $\texttt{v}$.

Activities are then enriched by *attributes* (or frame slots[1]) about location, time, and involved participants. The latter includes both the humans in an activity (e.g., *man*, *woman*, *judge*) and objects or props that play role in the activity (e.g., *diamond* or *ring*). We obtain cues for the location and time attribute values using NLP techniques as well as from the scene description (before the dialog starts: see Figure 6.2), e.g., *apartment*, *courthouse*, *spring*, *day*. For the participants attributes, we extract cues from both the characters in a scene and noun phrases in narrative snippets or dialogs. All these will also be sense-disambiguated in the output for the KB.

Definition 6.1.2 - Activity frame.

An *activity frame* is an activity enhanced with attribute values for location, time, and participants.

- For location, the allowed values are WordNet senses that are hyponyms (specialization) of the WordNet sense `location`^{1}.
- For time, the allowed values are hyponyms of the sense `time period`^{1} or `event`^{1}.
- For participants, the allowed values are hyponyms of `living thing`^{1} or `physical object`^{1}.

Each attribute can have zero, one or multiple values.

Finally, as we may extract activity candidates from each scene, we can relate activities from successive scenes (if there is a typical pattern found in several movies). To this end, we introduce frame attributes *prev* for a previous activity and *next* for a following activity. This way we link different activity frames to form entire chains. In the example, `propose`^{5} `woman`^{1} would be *next*-linked to `kiss`^{1} `someone`^{1}.

Definition 6.1.3 - Activity chain.

An *activity chain* is a sequence of temporally related activities connected by *prev* and *next* links. $A.next = B$ and B.prev = A denote that activity A is often followed by activity B.

[1] not to be confused with frames in the field of Logic.

Approach. We have developed an advanced pipeline for semantic parsing and knowledge distillation. This allows us to systematically compile semantically organized activity frames from scripts and novels (see the example illustrated in Figure 6.1). Figure 6.4 illustrates the Knowlywood pipeline of methods and tools. For automatically building the Knowlywood KB, we take the following main steps:

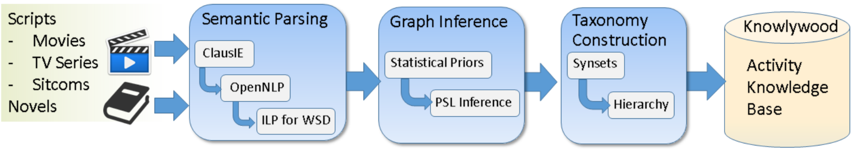


Figure 6.4: Knowlywood System Overview

- **Semantic parsing:** We first apply information extraction techniques on our input sources and then feed the output into a novel technique for semantic parsing, based on identifying clauses, mapping words and phrases to WordNet and VerbNet, and using integer linear programming (ILP) for the final disambiguation and construction of candidate activity frames.

  Semantic parsing has received much attention in computational linguistics recently; see Artzi et al. (2013) and references there. So far, it has been applied only to specific use-cases like natural-language question answering (Berant et al., 2013; Fader et al., 2014) or understanding temporal expressions (Lee et al., 2014). We believe that our work is the first to apply semantic parsing to large amounts of narrative text for KB construction.

  Semantic role labeling (SRL) (Gildea and Jurafsky, 2002; Palmer et al., 2010) is highly related to semantic parsing, the goal being to fill the slots of a predefined frame with arguments from a sentence. However, state-of-the-art methods are slow and do not work well for our task of activity knowledge acquisition. Moreover, SRL methods typically consider Propbank (Palmer et al., 2005) as a backbone, and Propbank lacks the semantic organization of verbs that VerbNet provides.

  Word sense disambiguation (WSD) (Navigli, 2009) is another component in semantic parsing and semantic role labeling. We use the state-of-the-art tool IMS (It-Makes-Sense) (Zhong and Ng, 2010) as a WSD prior for our joint disambiguation ILP. Note, though, that WSD alone focuses on the lexical semantics of individual words – this is still far from full-fledged semantic parsing for populating an activity KB.

- **Graph inference:** We use the output data of the first stage to construct a preliminary activity knowledge graph, with noise and false positives. We then use Probabilistic Soft Logic (PSL) (Kimmig et al., 2013) for efficient inference to construct a cleaner graph as consistent, high-quality output.
- **Taxonomy construction:** We merge activities into equivalence classes, so-called *synsets* in the terminology of lexical resources (e.g., WordNet). An example is merging activity *propose to girlfriend* with *propose to fiancee.*

  Finally, we construct a subsumption hierarchy of activity synsets, which connects activities by the `hasType` relation. An example is: *propose to girlfriend* hasType *propose to someone.*

We additionally attach video frames to activities by aligning scenes in movie scripts with their respective video frames. We leverage the timestamp information in subtitle data to perform this alignment (see Figure 6.3). We maintain a record of the position in the scene where an activity is spotted. If there is visual data corresponding to this position in text, then we assume that this visual data represents the activity. To align scenes in movie scripts with the video frames, we follow the procedure described in Laptev et al. (2008): for movies, subtitle data includes timestamps that can point us to the movie frames at that timestamp. Scripts contain nearly the same dialogues as the subtitles, thus making an alignment easy. The TV series and sitcoms data that we use sometimes contains video frames as part of an episode description. We use the image caption and its position in the surrounding HTML page to heuristically align the image with the text of its corresponding scene.

We have processed nearly 2 million scenes from 560 movies, 460 TV series, and 100 books, and constructed a high-quality activity knowledge base with almost one million frames. Specifically, we represent activities as JSON objects which gives us typed attributes (also known as slots in knowledge representation terminology) and set-valued entries for attributes (also known as values or fillers). JSON is a popular format for data export/import. Our frames can also be easily cast into RDF triples.

While parts of our approach could be applied to other genres, we focus on narrative text because it possesses some attractive yet under-utilized properties. Rather than being limited to newsworthy events, narrative text may include descriptions of common, rather mundane everyday activities. These are often described in a very detailed way and in chronological order with marked boundaries. For instance, we may find that one often *unlocks a door* before *entering a building.* Finally, movie narratives allow us to connect our knowledge to visual content in movies.

Contributions. Overall, our contributions are:

- The first system, called *Knowlywood* that automatically acquires detailed knowledge about activities, by tapping into movie/TV scripts and narrative texts and combining semantic parsing techniques for candidate generation with probabilistic inference for candidate cleaning.
- New techniques for sense disambiguation of multi-word phrases (mapping them to WordNet) and taxonomy induction for activities, as building blocks of our construction pipeline.
- A large knowledge collection with nearly one million activities in the form of semantic frames, and with linkage to visual contents where activities occur. This Knowlywood collection is publicly accessible. Its high quality has been confirmed by manual assessment with extensive sampling.

Our activity frames are valuable for use-cases such as video search (such as *movie scene search*, discussed in Section 7.4), provide background knowledge for human-computer dialog, and can aid tasks like video scene understanding and the generation of textual descriptions for visual contents. Note also that the developed methodology is general and can be applied to other input sources if available, for example, personal diaries or travel logs.

6.2 Semantic Parsing

We have devised a customized pipeline for semantic parsing that starts with the input scripts and extracts and disambiguates constituents, all the way to constructing a frame structure for candidate activities.

Consider the input sentence *He shot a video in the moving bus.* The output frame for this input is shown in the last column of Table 6.1. The activity name is given by the verb followed by an object (i.e., $\texttt{shoot}^{4}$ $\texttt{video}^{2}$)[2]. If a phrase is not present in WordNet, e.g., *moving bus*, then we merely map its head word (*bus*). The other columns in Table 6.1 show the input phrases (after chunking) and their mappings to WordNet senses and entries in VerbNet (Kipper et al.,

[2] Footnote 6.2: The WordNet senses for *shoot* and *video* are:

• $\texttt{shoot}^{1}_{v}$: hit with missile …	$\texttt{video}^{1}_{n}$: picture in TV transmission …
• $\texttt{shoot}^{2}_{v}$: kill by firing missile …	$\texttt{video}^{2}_{n}$: a recording of …
• …	…
• $\texttt{shoot}^{4}_{v}$: make a film …	$\texttt{video}^{4}_{n}$: broadcasting …

2006), as discussed below.

Table 6.1: Semantic parsing example

Phrase	WordNet Mapping	VerbNet Mapping	Output Frame
the man	`man`^{1}	Agent . animate	Agent: `man`^{1}
began to shoot	`shoot`^{4}	`shoot`$^{3}_{vn}$	Action: `shoot`^{4}
a video	`video`^{2}	Patient . solid	Patient:`video`^{2}
in	in	PP . in	
the moving bus	`bus`^{1}	NP . Location . solid	Location: `moving bus`^{1}

Sentence analysis. We first use ClausIE (Corro and Gemulla, 2013) to decompose sentences into shorter clauses, whenever possible. These are then further decomposed by applying the OpenNLP (`opennlp.sourceforge.net`) maximum entropy model for chunking the text of each individual clause. In the example in Table 6.1, this results in the sentence being split as shown in the first column.

Sense and argument analysis. Understanding the verb is the most critical task for semantic interpretation. We address this by mapping the verb or verb phrase to its proper sense in WordNet, which in turn is linked with VerbNet (Kipper et al., 2006), a manually curated linguistic resource for English verbs. For each verb class, VerbNet lists relevant thematic roles, semantic restrictions on the arguments, and syntactic frames. For example, for the main predicate verb *shoot* in our example sentence, VerbNet lists multiple candidate senses, and for the first of these, `shoot`$^{1}_{vn}$, it provides, among others, the following syntactic frame:

Agent.animate V Patient.animate PP Instrument.solid

This would match "*he shot the man with a gun*". Here, several roles (e.g., *Patient* in this example) are accompanied by a semantic constraint (e.g., *animate* in this example), known as a *selectional restriction.* A selectional restriction such as `animate` for the patient requires that this patient be a living being when used in the given syntactic frame. This can guide the choice of the proper WordNet mappings for the objects and for other words. For instance, *the man* in our example sentence could be disambiguated as `man`$^{1}_{n}$, which in turn is in a `hasInheritedHypernym` relationship with `living thing`$^{1}_{n}$, which leads us to the `animate` label from VerbNet, and helps us find the right VerbNet sense for the verb. As the alignments of VerbNet roles to WordNet are not available, we manually map these roles to the WordNet senses.

These dependencies are captured as constraints in our joint disambiguation method, based on Integer Linear Programming (ILP) — discussed below. The ILP method uses prior weights obtained from simpler heuristics for word sense disambiguation (WSD) — discussed next.

WSD priors. For an initial disambiguation of individual words and phrases, we use the state-of-the-art WSD system It-Makes-Sense (IMS) (Zhong and Ng, 2010), which relies on supervised learning. We obtain the following scores for mapping a word i to sense j:

$$\xi_{ij} = \begin{cases} \text{score from IMS} & \text{for } j \in S_{\mathrm{W}} \\ \sum\limits_{j'} \xi_{ij'} & \text{for } j \in S_{\mathrm{V}} \text{ linked to } j' \in S_{\mathrm{W}} \end{cases}$$

Here, S_{W} denotes the set of candidate WordNet senses for the verbs and S_{V} denotes the set of candidate VerbNet senses. Note that VerbNet is much smaller and thus coarser-grained than WordNet, hence the summation over all WordNet senses linked with the same VerbNet verb.

An additional feature used in the ILP later is the most-frequent-sense ranks that WordNet provides, based on manual annotation of a large news corpus:

$$\phi_{ij} = \begin{cases} \text{1/ 1+ rank(j for i)} & \text{for } j \in S_{\mathrm{W}} \\ \sum\limits_{j'} \phi_{ij'} & \text{for } j \in S_{\mathrm{V}} \text{ linked to } j' \in S_{\mathrm{W}} \end{cases}$$

Finally, we compute syntactic and semantic priors based on how well the input verb matches a VerbNet entry:

$$\begin{array}{rl} \mathrm{syn}_{ij} & \text{frame match score for word } i \text{ and VerbNet sense } j \\ \mathrm{sem}_{ij} & \text{selectional restriction score of the roles} \\ & \text{in a VerbNet frame } j \text{ for word } i \end{array}$$

ILP model. For the *joint* disambiguation of all words in the input sentence, we have devised an ILP with *binary decision variables* x_{ij} set to 1 if word i is mapped to sense j (in WordNet and/or VerbNet). V denotes the set of all input words or phrases i that are verb chunks. Our ILP is defined as follows:

$$
\begin{array}{lll}
\multicolumn{3}{l}{\textbf{maximize}} \\
\multicolumn{3}{l}{\quad \sum_{i,j} x_{ij}(\beta_1 \xi_{ij} + \beta_2 \phi_{ij} + \beta_3 \mathrm{syn}_{ij} + \beta_4 \mathrm{sem}_{ij})} \\
\multicolumn{3}{l}{\textbf{subject to}} \\
\quad \sum_{j \in S_{\mathrm{V}}} x_{ij} & \leq \; 1 & \forall i \in V \\
\quad x_{ij} & \leq \; x_{ij'} & \forall i \in V, j \in S_{\mathrm{W}}, \\
 & & \textit{j mapped to } j' \in S_{\mathrm{V}} \\
\quad x_{i_0 j_0} & \leq \; x_{ij} & \forall i_0 \in V, j \in S_{\mathrm{V}}, \\
 & & x_{ij} \in \textit{role-restr}(x_{i_0 j_0}) \\
\quad \sum_{j} x_{ij} & \leq \; 1 & \forall i \notin V \\
\quad x_{ij} & \in \; \{0,1\} &
\end{array}
$$

The objective function combines the various prior scores, with coefficients tuned on withheld training sentences that are manually labeled. The first constraint ensures that at most one VerbNet sense is chosen for each verb. The second one ensures consistency between choices of WordNet senses and corresponding VerbNet ones. The third constraint covers the selectional restrictions described earlier. The fourth constraint ensures that at most one sense is chosen for each non-verb word.

We instantiate a separate ILP for every sentence, and thus the ILP size and complexity remain tractable.

6.3 Graph Inference

Based on the output frames of the semantic parsing phase, we derive connections between different activity frames: parent types (hypernyms), semantic similarity edges, and temporal order (previous/next). We cast this as a graph inference problem, denoting the three types of connections as T, S, and P (previous) edges. We tackle this task using the Probabilistic Soft Logic (PSL) framework (Kimmig et al., 2013) for relational learning and inference.

The verb or noun in an activity is either a single word that is directly mapped to a WordNet/VerbNet sense, or it is a multi-word phrase. In the latter case, we only map the head word of the phrase to WordNet or VerbNet.

Edge priors. As we define an activity as a `verb` prep, `noun` pair, we can leverage WordNet's taxonomic hierarchy to estimate parent types and similarities between activities.

Our model starts off with prior probabilities for each of the three kinds of edges. The prior for T (parent type) edges between two pairs (v_1,n_1), (v_2,n_2) is calculated as a multiplicative score $\mathrm{t}(v_1, v_2) \cdot \mathrm{t}(n_1, n_2)$. For the noun senses, we use the WordNet hypernymy: The score is 1 if parent and child are connected by hypernymy, and 0 otherwise. For the verb senses, we check both WordNet hypernymy and VerbNet verb hierarchy.

Finally, we derive edges from the subsumption of activity participants, retrieved from WordNet, e.g., between *drinking tea* and *drinking beverage.*

We create S (similarTo) edges based on the similarity between (v_1, n_1), (v_2, n_2) using the multiplicative score: $\tau_{\mathrm{VV}}(v_1, v_2) \cdot \tau_{\mathrm{NN}}(n_1, n_2)$. The taxonomic relatedness score between two noun senses $\tau_{\mathrm{NN}}(n_1, n_2)$ is computed using WordNet path similarity measure (Pedersen et al., 2004). Scores between two verb senses $\tau_{\mathrm{VV}}(v_1, v_2)$ are computed using WordNet verb groups and VerbNet class membership (Schuler et al., 2009).

P (previous) edges: Scripts come with scene boundaries. We assume that the activity sequences that occur in a scene are temporally alignable. While an exact sequence of activity does not bring much redundancy, a gap-enabled sequence of activities can have rich statistics. Secondly, generalizing activities to potential parent nodes brings more redundancy, and hence richer statistics. We use the Generalized Sequence Pattern mining (GSP) algorithm (Srikant and Agrawal, 1996) that efficiently estimates sequences taking into account gaps. GSP uses a-priori based observation that every sub-sequence of an infrequent sequence can be pruned. While an exact sequence of activity does not bring much redundancy, a gap-enabled sequence of activities can have rich statistics and GSP can also take this into account. We assign the following values to the two parameters: minimum support = 3 and maximum gap = 4. By using GSP, we efficiently estimate the P edges and provide the following scores to the P edges: an activity a_1 precedes a_2 with probability proportional to the support $\dfrac{\mathrm{freq}(a_1 \text{ prev } a_2)}{\mathrm{freq}(a_1)\,\mathrm{freq}(a_2)}$.

Inference. Based on the initial prior scores for T, S, P, our task is to compute a cleaner graph of T, S, and P edges with scores reflecting their joint dependencies. These dependencies are captured in our PSL model with the following soft first-order logic rules. Since these are soft rules, they do not need to hold universally. The model automatically determines to what extent they should contribute to the final solution.

1. Parents often inherit prev. (P) edges from their children:
   $P(a, b) \wedge T(a, a') \wedge T(b, b') \Rightarrow P(a', b')$.

2. Similar activities are likely to share parent types
   $S(a,b) \wedge T(b,b_0) \Rightarrow T(a,b_0)$.

3. Likely mutual exclusion between edge types:
   $T(a,b) \wedge S(a,b) \Rightarrow \neg P(a,b)$.

4. Siblings are likely to be similar:
   $T(a,c) \wedge T(b,c) \Rightarrow S(a,b)$.

5. Similarity is often transitive:
   $S(a,b) \wedge S(b,c) \Rightarrow S(a,c)$.

6. Similarity is normally symmetric:
   $S(a,b) \Rightarrow S(b,a)$.

The inference weights w_i are tuned based on withheld data, using the PSL system's weight learning module.

6.4 Taxonomy Construction

Activity merging. The previous steps of our pipeline yield fairly clean activity frames, but may produce overly specific activities such as *embrace spouse*, *hug wife*, *hug partner*, *caress someone*, etc. These are sufficiently similar to be grouped together into a single frame (with slightly generalized semantics). Thus, the relation S from the previous step provides a pruned starting point for activity merging.

Definition 6.4.1 - Activity synset.

An *activity synset* is a group of activities with highly related semantics. For a synset $\{(v_1,o_1),(v_2,o_2),\ldots\}$ of verb-sense/object-sense pairs, we require that $a_i = (v_i,o_i)$ and $a_j = (v_j,o_j)$ have a semantic distance in WordNet below a certain threshold.

Specifically, we consider *WordNet path similarity* (Pedersen et al., 2004) as a measure of semantic distance. To this end, we construct a graph between activity frames based on the synset (i.e., equivalence) and hypernym/hyponym relations in WordNet. The edges in this graph could be weighted by relatedness strength, such as gloss overlap (Banerjee and Pedersen, 2003), but we simply used uniform weights, i.e. simple path lengths $dist(v_i, v_j)$. For two activities a_i, a_j, we compute

$$\frac{1}{2}\left(\frac{1}{1+\mathrm{dist}(v_i,v_j)}+\frac{1}{1+\mathrm{dist}(o_i,o_j)}\right).$$

what are the Pi participants In addition, we consider the participants sets P_i, P_j in the frames of a_i,a_j, respectively. Recall that each P_i is a set of WordNet noun-phrase senses. We compute the WordNet path similarity for each element in $P_i \times P_j$, and aggregate them into an overall measure by taking the maximum (or alternatively the average). The final distance between a_i and a_j is the average of the verb-sense/object-sense distance and the participants distance.

The threshold for merging two activities into a synset is determined by manually grouping a small sample of activities and computing the threshold that achieves the synsets in the sample. We transitively merge activities whenever their distance is below that tuned threshold. We perform a transitive closure on this pruned neighborhood to allow grouping of activities.

Hierarchy induction. The above techniques provide us with a suitably grained but still flat collection of activity synsets. However, some of these may semantically subsume others. For example, *divorce husband* is subsumed by *break up with a partner*. Again, the relation T from the previous step provides a pruned starting point for hierarchy induction.

Definition 6.4.2 - Activity taxonomy.

An *activity taxonomy* is a DAG (directed acyclic graph) of activity synsets such that $a_i \sqsubset a_j$ is an edge in the DAG if a_i is semantically subsumed by a_j. That is, the verb or object of a_j is more general than that of a_i.

To construct the hierarchy, we again use WordNet path similarity but consider only *hypernym* relations now (i.e., disregard hyponyms). For this asymmetric measure, we again tune a threshold by manually assessing a small sample. The resulting taxonomy graph initially contains all subsumption pairs with semantic distance below the threshold. As this may create cycles, we finally break cycles by greedily removing low-weight edges. In building the Knowlywood KB, we had to eliminate only few cycles.

6.5 Results

To evaluate our approach, we conducted a series of experiments to thoroughly examine our pipeline for semantic parsing and knowledge distillation, as well as the resultant Knowlywood activity frames.

Data processing. Knowlywood is constructed by processing 1.89 million scenes from several sources:

- 560 movie scripts, scripts of 290 TV series, and scripts of 179 sitcoms. We crawled this data from Web sites like `wikia.com` and `dailyscript.com`.
- The Novels dataset comprises 103 novels from Project Gutenberg Faruqui and Pado (2012).
- Crowdsourcing: We use the data from Rohrbach et al. (2012), which consists of textual descriptions of videos portraying humans engaging in cooking related activities.

6.5.1 System components

Semantic parsing. In order to gain deep insights about our system, we had human judges annotate at least 250 random samples of the outputs of the different stages in our semantic parsing method, i.e., sentence extraction by pre-processing datasets, clause level splitting, the basic NLP pipeline (tagging, chunking, etc.), and finally disambiguation and VerbNet-based role assignment. Table 6.2 presents the resulting precision scores with statistical significance given as Wilson score intervals for $\alpha = 95\%$ Brown et al. (2001).

We observe that most of the errors stem from the NLP pipeline, especially chunking. This could be addressed by using more advanced NLP tools, which, however, tend to be slower. Processing the sitcom and TV series data is the most challenging and error-prone due to the nature of these texts: the sentences are long and often filled with slang (e.g., *hold'em*). Some errors are also introduced at the early stage of pre-processing movie scripts, where we rely on regular expressions to parse the semi-structured text files (e.g., the introductory text for each scene that introduces the location, time, etc.).

Graph inference. Next, we evaluate the PSL-based graph inference. Our findings indicate that it was instrumental in cleaning the candidate relations between activities and also in acquiring new edges between them. Table 6.3 shows the precision and size of the graph before and after the inference step. For example,

Table 6.2: Evaluation of the semantic parsing

	sentence	clause	senses	participant roles	overall
Movie scripts	0.79 ±0.11	0.84 ±0.07	0.96 ±0.04	0.96 ±0.03	0.91 ±0.03
TV series	0.90 ±0.06	0.96 ±0.04	0.65 ±0.10	0.79 ±0.08	0.74 ±0.04
Sitcoms	0.91 ±0.07	0.93 ±0.06	0.67 ±0.12	0.72 ±0.11	0.73 ±0.05
Novels	0.94 ±0.05	0.91 ±0.07	0.85 ±0.09	0.93 ±0.06	0.90 ±0.04
Crowdsourcing	0.96 ±0.04	0.96 ±0.04	0.75 ±0.11	0.91 ±0.07	0.91 ±0.03

from the P edge between $\texttt{acquire}_v^1$ $\texttt{cutting knife}_n^1$ and $\texttt{use}_v^1$ $\texttt{cutting knife}_n^1$, a new P edge is derived from $\texttt{acquire}_v^1$ $\texttt{knife}_n^1$ to $\texttt{use}_v^1$ $\texttt{knife}_n^1$. The transitive closure on the S relationships adds new edges. Thus, our graph inference increases Knowlywood's coverage and accuracy by inferring missing edges and removing inconsistent ones.

Table 6.3: Effect of PSL inference

	Before inference		After inference	
	Precision	#Edges	Precision	#Edges
T	0.77±0.04	1,906,520	0.87±0.03	4,511,203
S	0.84±0.02	1,022,700	0.85±0.04	3,421,210
P	0.78±0.04	116,186	0.84±0.09	205,678

Synset and hierarchy construction. We performed a static analysis of the hierarchy as well as an empirical evaluation. There were 543 cycles in the graph. These were of a very small length (average length 3). After breaking the cycles, the DAG consists of 505,788 synset nodes without any cycles. The maximum depth of the graph is 5.

Over a random sample of 119 activity synsets, human judges were asked if the edge between random synset members was indeed a synonymy relation, i.e. semantically equivalent activities. To evaluate the hierarchy, in a similar way, human judges were asked if the edge between two activity synsets was one of hypernymy, i.e. subsuming activity synsets.

The synset grouping achieved a very high accuracy of 0.976±0.02 (Wilson score intervals for $\alpha = 95\%$ Brown et al. (2001)). One of the reasons for this high accuracy was the tight threshold for taxonomic similarities. We had empirically chosen a high threshold of 0.40 for the synset similarity.

The hierarchy grouping achieved a high accuracy of 0.911±0.04 (Wilson score intervals for $\alpha = 95\%$). An example error case was *walk with a fly* having a hypernymy link to *travel with a beast*. This is because *animal* and *beast* are synonymous. Such mildly incorrect cases led to a slightly lower precision.

6.5.2 Knowlywood KB evaluation

In total, the Knowlywood pipeline produced 964,758 unique activity instances, grouped into 505,788 activity synsets. In addition to the edges mentioned above, we also obtain 581,438 `location`, 71,346 `time`, and 5,196,156 `participant` attribute entries over all activities.

Quality. To evaluate the quality of these activity frames, we compiled a random sample of 119 activities from the KB, each as a full frame with values for all attributes (participants, location, time, previous and next activity, etc.). We relied on expert human annotators to judge each attribute for each of these activities. An entry was marked as correct if it made sense to the annotator as typical knowledge for the activity. The judgement were aggregated separately for each attribute, and we computed the precision as $\frac{c}{c+i}$, where c and i are the counts of correct and incorrect attribute values, respectively. For statistical significance, we again computed Wilson score intervals for $\alpha = 95\%$. The per-attribute results are reported in Table 6.4. The inter-annotator agreement for three judges in terms of Fleiss' κ is 0.77.

Table 6.4: Knowlywood coverage and precision

Source	#Scenes	#Unique	Parent	Parti.	Prev	Next	Loc.	Time	Avg.
		Activities							
Movie scripts	148,296	244,789	0.87	0.86	0.78	0.85	0.79	0.79	0.84
TV series	886,724	565,394	0.89	0.85	0.81	0.84	0.82	0.84	0.86
Sitcoms	286,266	200,550	0.88	0.85	0.81	0.87	0.81	0.83	0.87
Novels	383,795	137,365	0.84	0.84	0.78	0.88	0.85	0.72	0.84
Crowdsrc.	3,701	9,575	0.82	0.91	0.91	0.87	0.74	0.40	0.86
Knowlywood	1,708,782	964,758	0.87	0.86	0.84	0.85	0.78	0.84	**0.85±0.01**
ConceptNet 5	-	4,757	0.15	0.81	0.92	0.91	0.33	N/A	**0.46±0.02**

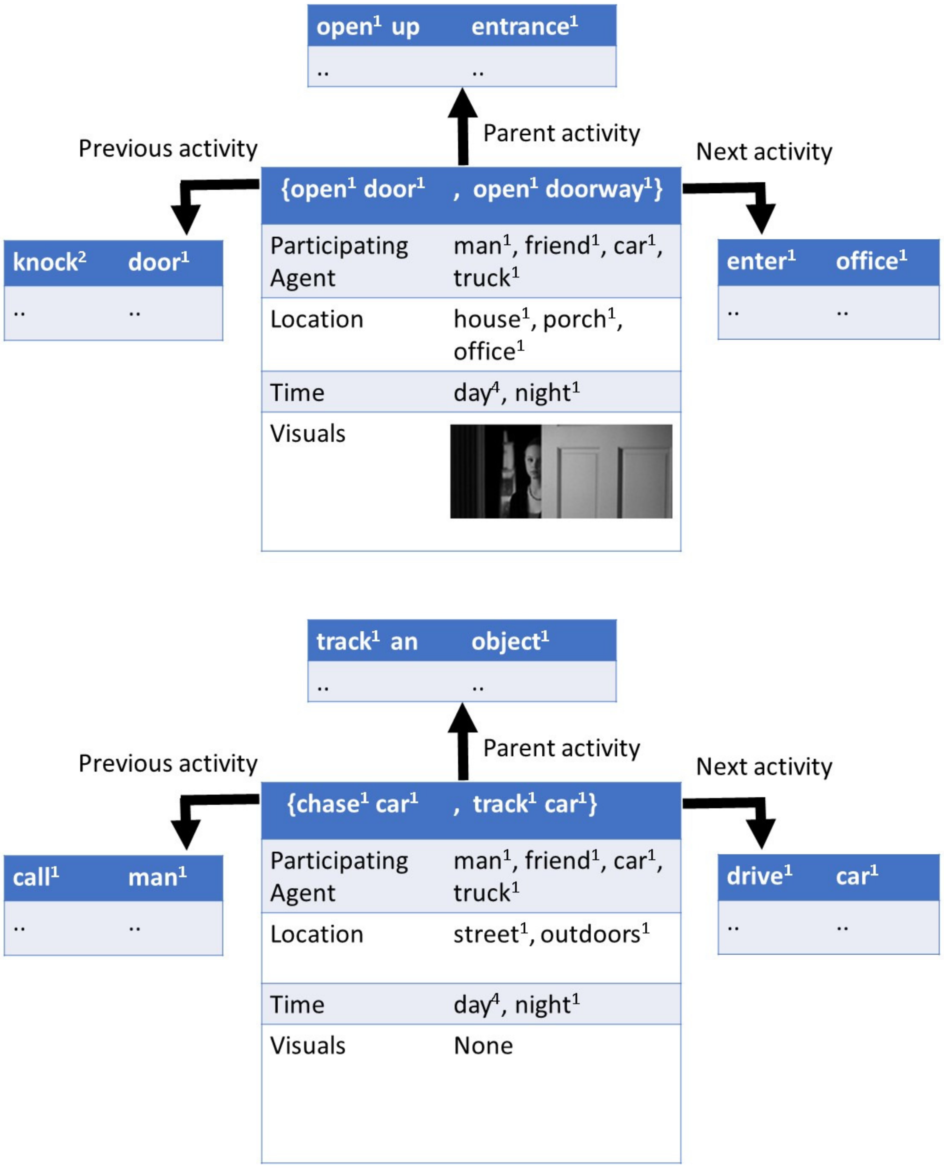


Figure 6.5: Anecdotal Examples

We can observe from these assessments that Knowlywood achieves good precision on most of the attributes. In some datasets like the Crowdsourcing collection, no information on time or location is available. This accounts for the low scores.

Examples. Figure 6.5 presents anecdotal examples of Knowlywood's activity frames, with specific sense numbers from WordNet.

Comparison with ConceptNet. There is no direct competitor that provides frames of semantically organized activities. We thus compared Knowlywood with ConceptNet 5 (CN), the closest available resource, assuming that any concept name matching the pattern *verb [article] object* is an activity. We mapped CN's relations to our notion of activity attributes as listed in Table 6.5.

Table 6.5: Organizing ConceptNet relations by aligning them with Knowlywood attributes

ConceptNet relation	Knowlywood attribute
`IsA`, `InheritsFrom`	`type`
`Causes`, `ReceivesAction`, `RelatedTo`, `CapableOf`, `UsedFor`	`agent`
`HasPrerequisite`, `HasFirstSubevent`, `HasSubevent`, `HasLastSubevent`, `MotivatedByGoal`	`prev/next`
`SimilarTo`, `Synonym`	`similarTo`
`AtLocation`, `LocationOfAction`, `LocatedNear`	`location`

The activities derived this way from CN were manually assessed by the same pool of annotators that assessed the Knowlywood frames. We randomly sample 100 activities from CN and take all their relations but adding further relationships if we encountered too few of any one relationship type. The last row of Table 6.4 shows the results — both coverage and precision. We see that CN works well for eliciting previous/next activities. Here its quality exceeds that of Knowlywood. CN's crowdsourcing-based knowledge acquisition leads to fine-grained temporal knowledge that is rather difficult to mine from narrative texts (e.g., that *riding a horse* is preceded by *keeping your heel down*, and followed by *your bottom getting sore*).

However, these high precision values also result from the specific nature of CN's knowledge representation. Since CN's concepts are essentially strings (not word senses), we instructed our annotators to evaluate an attribute value as correct even if it holds true for just one possible interpretation of the concept names, ignoring ambiguity. The data also contains duplicates (e.g., *you open your wallet*, *open your wallet*, *open wallet*, . . .) that were all judged as correct as predecessors of *taking out money*. CN's less formalized nature is particularly apparent from the fact that the parent type attribute obtains a precision of only

15%. Generally, except for the temporal ordering of activities, the precision of CN is substantially below that of Knowlywood.

Most importantly, Knowlywood's coverage of activities dwarfs that of CN. CN merely provides 4,757 activities, most of which are also included in Knowlywood, while the latter additionally contains nearly a million activity frames.

Comparison with ReVerb. We also compare Knowlywood with ReVerb (Fader et al., 2011b), the most widely used system for broad-coverage open information extraction. Open information extraction aims at mining all possible subject-predicate-object triples from text. We mine activity knowledge from these triples such that the subject is an agent, and the predicate and object together form an activity, e.g., *drink* + *coffee*.

For role assignment, we mine `MovieClips.com` to obtain mappings from words to labels. MovieClips contains high-quality human-annotated and categorized tags for nearly 30,000 movie scenes (e.g., *action:singing*, *prop:violin*, *setting:theater*). These tags have a direct correspondence to our attributes (see Table 6.7). The tag co-occurrence statistics can be used to create a Bayesian classifier as $P(\mathrm{c}|\mathrm{w}) = \frac{P(\mathrm{c},\mathrm{w})}{\sum_{c_i} P(c_i, w)}$, relying on the joint probabilities for classes (c) and words (w) from `movieClips.com`. One may also consider using semantic role labeling systems as an alternative. However, they cannot solve our semantic parsing task because they require large amounts of domain-specific labeled training data. Moreover, they suffer from poor scalability.

We consider two different datasets as input to ReVerb. First, all the input Script data that we used for our system (setup called ReVerbMCS). Second, all of ClueWeb09 dataset (setup called ReVerbClue). ReVerb extractions over ClueWeb09 are already available in the form of a publicly available dataset Fader et al. (2011b), consisting of 15 million unique SVO (Subject Verb Object) triples. The ReVerbClue data does not contain enough context to use the MovieClips-based role classifier because it consists of only SVO triples.

Since both ReVerbMCS and ReVerbClue extractions are strings (not word senses), we leniently evaluated an attribute value as correct if it holds true for any possible sense of the concept. This is thus a much easier task than Knowlywood's, for which we required the correct sense disambiguation.

In Table 6.6, we list the number of activities as well as numbers and precision of several roles. The precision values are obtained by evaluating the frames corresponding to the activities overlapping with the Knowlywood test set of 119 activities resulting in more than 400 attribute triples. Knowlywood outperforms both the ReVerb based baselines (compare to Table 6.4), in terms of both pre-

Table 6.6: ReVerb baselines (counts and precision scores)

	Activities	Participant	Location	Time
ReVerbMCS	0.37M	0.37M, 0.77	0.17M, 0.83	0.05M, 0.66
ReVerbClue	0.86M	1.47M, 0.41	0.055M , -	0.008M, -

Table 6.7: Mappings between MovieClips.com and Knowlywood

MovieClips tag	**Knowlywood attributes**	**Example**
action	activity.v	cut
prop	activity.o	knife
setting	location	bar
occasion	time	thanksgiving
charactertype	participant	policeman

cision and counts. The role labels score in ReVerbMCS reflect the rich statistics (though limited in size) obtained from the manually curated MovieClips. We also see that extractions from ClueWeb09 data, which is an order of magnitude larger than our scripts data, did not entail better quality.

Multimodal content. By automatically aligning the movie scripts with subtitled videos, we were also able to associate 27,473 video frames with Knowlywood's activities. We believe that this will be an important asset for computer vision, because existing systems for activity detection in videos suffer from a lack of training data and background knowledge, and hence have been quite limited in their coverage.

6.5.3 Use-case: movie scene tagging

In order to evaluate the usefulness of the Knowlywood KB extrinsically, we introduce the task of predicting the activity portrayed in a movie clip, without task-specific training data, given only the location and participants in the corresponding scene.

As ground truth, we consider `Movieclips.com`, which contains high quality, manually curated categorized tags for nearly 30,000 movie clips/ scenes. Examples of these include: *location/setting: cemetery*, *participating object/prop: rose*, *action: obituary speech*. By analyzing the co-occurrence statistics over the tags of these clips, we obtain a scored list of activities for a given [participant(s), location(s), time(s)]. We randomly select 1,000 clips from this gold data.

The evaluation task is to assess Knowlywood's (or any baseline activity KB's) top-k activity recommendations given only [participant(s), location(s), time(s)]. This task is more complex than a simple tag recommendation that would ignore any tag categories. As KBs, we use the Knowlywood KB, and the various baselines: ConceptNet, ReverbMCS, and ReverbClue.

Table 6.8: Movie Scene Tagging evaluation

	MRR	Hit rate
ReVerbClue	0.070	0.180
ConceptNet	0.143	0.345
ReVerbMCS	0.254	0.415
Knowlywood	0.327	0.610

The evaluation is based on a comparison of the predicted top-k activity list with the ranked gold list of activities. We report the standard IR-metric Manning et al. (2008) Mean Reciprocal Rank (MRR) that rewards early hits in the predictions. We also report Hit-Rate metric which is one whenever the top-10 results contain at least one good tag.

Mean reciprocal rank (MRR). Given a query q, we define r_q as the topmost rank of the relevant outcomes in the ranked list. If no relevant result is present in the list, assume $r_q \to \infty$. Averaging over queries, we obtain $MRR = \frac{1}{|Q|} \sum_q \frac{1}{r_q}$.

We then evaluated the various KBs on the movie scene tagging task. This is an automated evaluation, as the ground truth gold data is already available. For both the KBs and the gold-set, we uniformly set k=10, i.e. we compare the top-10 predictions against the top-10 ground truth rankings. The results in Table 6.8 demonstrates that although Knowlywood has not been trained or mined from `Movieclips.com` tags at all, the system is able to outperform the baselines by a large margin both on MRR and Hit rate. ReverbMCS outperforms other baselines because the role label classifier in ReverbMCS uses `Movieclips.com` statistics already. Knowlywood also yields a much better coverage in terms of the hit rate.

6.6 Discussion

We have presented Knowlywood, the first comprehensive KB of human activities. It provides a semantically organized hierarchy of activity types, participating agents, spatio-temporal information, information about activity sequences, as well as links to visual contents. Our algorithms ensure that the entries are fully disambiguated and that inconsistent attributes are removed. Our experiments show that Knowlywood compares favorably to several baselines, including in use-cases such as tag recommendations.

We believe that the resulting collection of approximately one million activity frames is an important asset for a variety of applications such as image and video understanding. The resulting algorithms in the Knowlywood pipeline could also serve as a building block for other applications, e.g., Rohrbach et al. (2015) employ a trimmed version of our semantic parsing and show improvements in image captioning task. Knowlywood KB is freely available at `http://people.mpi-inf.mpg.de/~ntandon/resources/readme-activity.html`

Strengths:

- In addition to scripts, our methods also work on less descriptive, more literary texts such as novels.
- Our method does not require training data for SRL because we jointly leverage the semantics from WordNet and VerbNet for semantic and syntactic constraints.
- Our framework combines and builds upon existing techniques to build a pipeline that can be used to build other domain-specific knowledge bases such as a sports activity KB through commentary scripts.

Weaknesses:

- We do not compile longer chains of activities, but only provide the previous and next activities.

  One solution would be to probabilistically construct longer chains of activities using a decoding algorithm like Viterbi.
- Our method relies on VerbNet, a handcrafted resource which is not updated frequently.

  Whenever there is no matching entry of a verb in VerbNet, we fallback to use a simple *S V preposition O* style extraction that gives us the activity *V preposition O* and the participating agent as *S*. It is non-trivial to tell apart most prepositions of time/location (e.g., at), however, we can infer the time by maintaining a dictionary of time (from hyponyms of `time`). If it is not time, then we label it as a location.

- The current set of images for activities is small due to a small set of movies.

  We propose a solution to this problem while discussing the extensions that uses Flickr images.

- We cannot incrementally update the knowledge base when new data comes. Though it is possible to perform semantic parsing on the incremental input, graph inference is not incremental.

  A workaround to this problem could be to perform graph inference periodically and not for every update.

- We do not generalize the values in a frame slot which could improve the frequency estimates due to redundancy. For example, consider sample values for `participant`: `man`, `boy`, `male`. We could generalize all of these to `male`, thereby increasing the frequency estimate for `male` and helping in ranking the values within a frame slot.

  One solution would be to leverage the WordNet hierarchy as our frame slot values are already mapped to WordNet.

Lessons learned:

- Unlike the kinds of commonsense relations addressed in the previous chapters, activity commonsense is much more implicit or not mentioned in text. In this situation, specialized sources like scripts or very structured Web contents like `WikiHow.com` are the suitable sources of knowledge, apart from getting this knowledge from humans as in ConceptNet.

Assumptions:

- We assume that the activity sequences that occur in a scene are temporally alignable.

Extensions:

- Flickr image recommendation for activities in Knowlywood. Given a Flickr post, we can infer the activity in the post using the scene tagging system from Section 6.5.3. This would lead to a very high coverage of visuals for activities.
- We can estimate a visibility attribute for the activities, similar to the previous

chapter. For this, we could use Flickr as a signal (see previous bullet point) and other linguistic cues like the abstractness of the verb in the activity.

- Modeling human activities from the Web is another interesting direction. As our approach is applicable to diverse data including scripts and novels, we believe that for textual sources on the Web with clear boundaries, e.g., WikiHow, our approach can be applied with minor changes.

7 Resulting KB: WebChild KB & Applications

In this dissertation, we presented methods for acquisition of commonsense knowledge. The resulting Commonsense KB, *WebChild KB*, is a very large KB containing these relations. WebChild is available for browsing at `https://gate.d5.mpi-inf.mpg.de/webchild/`, and the project page with datasets is hosted at `http://www.mpi-inf.mpg.de/yago-naga/webchild/` .

7.1 WebChild KB statistics

WebChild KB contains 2.3 million disambiguated concepts and activities, and more than 18 million assertions about these concepts connected by more than 6000 relations.

Table 7.1: WebChild statistics

Relation	#Sub-relations	#Assertions
Properties	19	4.3M
Comparatives	6331	1.1M
Part whole	3	6.7M
Activities	7	6.1M

In this chapter, we will present new applications where we applied the knowledge present in WebChild KB. This can be seen as a holistic evaluation of WebChild KB because eventually KBs act as structured background knowledge for applications. This chapter presents one application for each of the three paradigms of commonsense relations that we have discussed.

7.2 Commonsense on Object Properties: Set Expansion

As a use-case that demonstrates the application benefits of WebChild, we studied the problem of populating semantic classes, such as `river`, `car`, or `singer`. This problem is often addressed as a set-expansion task (Wang and Cohen, 2007): Given a small number of seeds, which are instances (or hyponyms) of the same class, find as many additional instances as possible and rank them into a high-precision list. For example, for seeds like *Mississippi*, *Nile*, and *Ganges*, we would like to collect other rivers such as *Danube*, *Rhine*, *Seine*, *Mekong*, etc. A good baseline to compare with is the *Google Sets* tool, which is part of the `docs.google.com` service. Other methods like (Dalvi et al., 2012) may be better, but they are also much more complex and need extensive Web data not available to us.

Our method for class population is fairly simple; its main point is the way it harnesses the WebChild knowledge. For a given noun sense `n` corresponding to an instantiable semantic class, we perform the following steps:

1. We select the m highest ranked adjective senses $\mathtt{a}_1, \ldots, \mathtt{a}_m$ that are connected with `n` by one or more of WebChild's fine-grained assertions. As these are senses, we can further expand them by their WordNet synonyms, thus constructing a ranked list of adjectives $\mathtt{a}_1, \ldots, \mathtt{a}_l$ (where usually $l \geq m$), now cast into simple words.
2. To avoid extensively used adjectives of unspecific or ambiguous nature (e.g., `great`), we compute PMI scores between the starting noun `n` and each of the adjectives $\mathtt{a}_i$ $(i = 1 \ldots l)$:

$$\mathrm{PMI}(n, a_i) = \log_2 \frac{P[n \wedge a_i]}{P[n]\ P[a_i]}$$

   We prune all adjectives from the ranked list whose PMI score is below a specified threshold. From the remaining adjectives, we take the top k words $a_1, \ldots, a_k$ (e.g., with $k = 10$).
3. Now we apply the linking-verb patterns that we introduced in Section 2 to the Google N-gram corpus and extract noun phrases from the matching N-grams. This yields frequencies for (n,a_i) co-occurrences. The noun phrases are the candidates for populating the class denoted by n.
4. We rank the collected noun phrases p by aggregating over the co-occurrence

frequencies:

$$\text{score}(p) = \sum_{i=1}^{k} \text{freq}(p, a_i) \times \text{weight}(a_i)$$

where weight(a_i) is the score of the original $\mathtt{a}_i$ for noun sense `n` in WebChild (based on pattern-matching statistics, see Section 3.3).

As a demonstration of the high quality that our method achieves, we evaluate its *precision@5*, in comparison to the top-5 results from Google Sets. We did this for the following 10 test cases (5 common nouns and 5 proper nouns): `river`, `ice cream`, `mountain`, `chocolate`, `keyboard`, `nile river`, `bangalore`, `tiger lily`, `parsley`, `florine`. We evaluate Google Sets with 1 seed (G-1) and 2 seeds (G-2) against WebChild, which only takes the class noun as input (W-1). G-1 runs into limitations, but G-2 performs reasonably well even in this extreme situation. For example, with seed *river* as input, G-1 gives as output *boca*, *estudiantes*, *independiente*, *racing*, *san lorenzo*; with the seed *tiger lily* as input, G-1 produces no output. G-2, with the seeds *river*, *river valley*, gives as output *canyon*, *arizona*, *valley*, *colorado*; with the seeds *tiger lily*, *panther lily* as input, G-2 gives as output *peacock iris*, *meadow saffron*, *pancratium*, *peruvian lily*, *flag.*

Table 7.2 shows the results. WebChild outperforms G-1 and G-2 on common nouns. On proper nouns, G-2 outperforms WebChild, but WebChild performs as well as G-1. Tables 7.3 and 7.4 show the top-10 WebChild adjectives, and the top-5 set expansions for the input `chocolate` and `keyboard` respectively.

Table 7.2: Results for set expansion

Approach	Genre	P@5
G-1	common noun	0.52
G-2	common noun	0.72
W-1	common noun	**0.92**
G-1	proper noun	0.52
G-2	proper noun	**0.68**
W-1	proper noun	0.52

Table 7.3: `chocolate`: top-10 adj, top-5 expansions

top-10 adjectives	`smooth`, `assorted`, `dark`, `fine`, `delectable`, `black`, `decadent`, `white`, `yummy`, `creamy`
top-5 expansions	*chocolate bar*, *chocolate cake*, *milk chocolate*, *chocolate chip*, *chocolate fudge*

Table 7.4: `keyboard`: top-10 adj, top-5 expansions

top-10 adjectives	`ergonomic`, `foldable`, `sensitive`, `black`, `comfortable`, `compact`, `lightweight`, `comfy`, `pro`, `waterproof`
top-5 expansions	*keyboard*, *usb keyboard*, *computer keyboard*, *qwerty keyboard*, *optical mouse*, *touch screen*

7.3 Commonsense on Relationships: Image Classification.

As a use-case that demonstrates the application benefits of PWKB, we use PWKB for image classification. The task is to recognize unseen image categories by transferring knowledge from known categories. For example, being able to recognize *wheels* of cars and *seats* of chairs might allow us to recognize a *wheelchair* even if we have no training image for *wheelchair*. This "zero-shot recognition" is crucial as many categories have no (or very sparse) training data.

For this task, we repeated the experiment of (Rohrbach et al., 2011), who trained classifiers for 811 part categories to recognize unseen categories. To associate the unseen categories with the parts, part-whole patterns (Berland and Charniak, 1999) were retrieved with Yahoo search. For comparability, we used the same visual features and the same image classification architecture as in the original study. We solely replaced the original part-whole relation with the relations from PWKB.

On the zero-shot task of recognizing 200 unseen categories, the top-5 accuracy increases from 23.8% (best single part-whole variant *Yahoo Snippets*) to 25.5% by using PWKB. We note that Rohrbach et al. (2011) achieved better performance, up to 35% accuracy, with a hierarchy-based transfer or combining multiple measures, which is orthogonal to the use of our part-whole knowledge. We could combine the PWKB asset with this technique. Note that this task is inherently difficult; we are not aware of any methods that achieve more than 40% accuracy.

7.4 Commonsense on Interactions: Scene Search

As a use-case that demonstrates the application benefits of Knowlywood, we use Knowlywood to build a search platform. The corpus comprises of movie scripts, the Crowdsourcing dataset, TV series, sitcoms, and novels (introduced in Section 6.5). This search system takes a text query q as input, which is expected to correspond to some activity. Examples of such queries are *animal attacks man*, *kissing during a romantic dinner*. As output, we expect a ranked list of scenes over the indexed corpus.

Approach. We use the textual (not visual) content of the scenes to obtain the score of a scene s for a given query.

Given an activity $a \in K$, where K denotes the Knowlywood knowledge base, let a_p be the set of

participants according to K and $A_\mathrm{p} = \bigcup_{a \in K} a_\mathrm{p}$ be the set of all participants associated

with activities in K. We derived a query-likelihood statistical language model as follows.

The probability that the scene s generates a query q is given by

$$P(q|s_t) = \sum_{a \in K} \sum_{p \in A_\mathrm{p}} P(q|a) \cdot P(a|p) \cdot P(p|s_t)$$

- s_t is the textual representation of the scene,
- $P(p|s_t)$ is the probability that the scene generates participant p of an activity (e.g., girl, ring, etc.), estimated from noun-phrase occurrences in t with corpus smoothing,
- $P(a|p)$ is the probability that participant p generates activity a, again with smoothing, and

Table 7.5: Query Frames.

Frame	Semantic restriction
S	WN physical entity
V	WN verb (compulsory)
O_1	WN physical entity
O_2	WN physical entity
L	WN location or WN physical entity
T	WN time-period

Table 7.6: Performance of the two search methods.

Algorithm	NDCG	MAP	Precision@5	MRR
Knowlywood	0.8972	0.9512	0.8809	0.9840
Text based	0.0772	0.0696	0.0404	0.0730

- $P(q|a)$ is the query likelihood of activity a, estimated by the occurrences of the verb-object words of a in the query, once more with smoothing.

Experimental setup. As there is no similar activity search system or evaluation dataset, we construct a benchmark dataset by gathering 100 queries of a predefined frame (`S V` O_1 O_2 `Location Time`), such as, *man kissed the girl on the cheek at the movie theater in the evening.* For this, we relied on a user interface as in Table 7.5, asking two people (one outsider and one of the authors) to enter arbitrary queries of their choice, as long as it fit the template. Further examples of these gathered queries include *frying onion* and *killing a bird.*

For this set of 100 queries, we generate search results using our generative model over the Movie script, Crowdsourcing, Sitcom, TV series, and Novels datasets.

For comparison, we also obtain the search results using a text-retrieval baseline, in particular, a statistical language model with Dirichlet smoothing, as implemented in the well-known INDRI system (Strohman et al., 2005).

Two annotators evaluated the top-10 results for each of these queries for both the baseline and the Knowlywood search system. Each result was scored between 1 (irrelevant) to 5 (perfectly relevant). The final rating for each result is given

Table 7.7: Anecdotal examples for Scene Search results

Query	**KB based**	**Text based**
man climbs mountain	.. following Jack , and helps him **climb a mountain** and find a crystal that will transport Jack home both... TV series: Samurai Jack	.. from deep in the **mountains** ... had entered the **Mountain** ...carried deeper into the **mountain** . .... **climbs** out of the river... looking for the **Mountain** of Skulls as well...turns his attention to Lysinka and the others down the **mountain** .. Novel: Conan
man shoots video	.. While **shooting Dixons music video** , Silver gets a call from the fertility clinic informing her that the IVF procedure has been moved up to the next day .. TV series: Beverly Hills	.. the woman **shoots Alex with a video game gun** as the woman traps her in the game that the.. TV series: Totally Spies
kill a bird	..mark go hunting with Sophie 's dad. Jeremy go hunting with Sophie 's dad. mark tries to **kill a bird** . the man injures it simply. the man tries to break its neck.. Sitcom: Peep Show	.. Carlos and Susan are still painting over the graffiti on the wall as those people discuss **To Kill a Mocking Bird** , however , while talking , .. TV series: Desperate Housewives

by the average of the ratings by the two annotators.

The annotation ratings were then used to compute four widely-used IR evaluation metrics, namely NDCG, Precision@k, MAP, and MRR (Manning et al., 2008).

Scene search results. Table 7.6 gives a comparative analysis of the NDCG, MAP, Precision@5, and MRR scores for both search methods. Since MAP, Precision@5, and MRR involve binary notions of relevance, we assume that those scenes that are rated with a score of at least 3 are the only relevant scenes.

We observe that for all four metrics, the Knowlywood search method performs best. We observed that the text retrieval engine often returns scenes with script text that closely matches the words in the query, while Knowlywood achieves a higher level of abstraction. For example, given the query *man climbs mountain*, the text engine favors scenes with many occurrences of the keywords *mountain* and *climb*, but not used in the specific sense of climbing mountains. The Knowlywood search method, on the other hand, uncovers those scenes that portray the activity, even if they do not contain the word *mountain* explicitly, but just semantically related expressions such as *hiking up a hill* etc. The Knowlywood search also correctly identifies the true meaning of an activity even if it contains verbs with ambiguous meaning. For example, the query *shoot a video* is often interpreted wrongly by the text retrieval engine and therefore it returns irrelevant snippets referring to shooting with a gun, etc. Table 7.7 provides some anecdotal examples of queries and scene search results by the two competitors.

7.5 Discussion

We have presented the statistics and some applications of the WebChild KB, the first comprehensive KB of fine-grained and disambiguated commonsense knowledge. WebChild KB's statistics show that it is by far by the largest automatically constructed commonsense KB. We presented one application for each of the three genres of commonsense knowledge, including set expansion using property knowledge, image recognition with part-whole knowledge, and, movie scene search with activity knowledge.

Commonsense knowledge would serve as background knowledge for more intelligent activity recognition in images and videos. The activity taxonomy could serve as a backbone of classes used in the recognition task. Deep learning methods for text typically rely on word embeddings using co-occurrences. Chen et al. (2015) construct more meaningful word embeddings driven by commonsense knowledge bootstrapped from WebChild KB.

8 Conclusions and Outlook

8.1 Summary

This dissertation has revived the theme of commonsense knowledge bases, which had previously been handcrafted or crowdsourced, unimodal and ambiguous. We introduce new methods for automatic acquisition of commonsense knowledge, with semantic rigor. These methods are generalizable and go beyond the relations covered in this dissertation.

The first contribution of this dissertation is a new method to extract and organize large-scale property commonsense. We automatically construct the range and domain of the property relations, starting out with a small set of seed examples. These seeds are typically manually gathered but we observe that an ensemble of two very different, automated, and noisy sources can also produce good seeds. We construct a graph where the nodes are words and word senses and the edge weights are computed based on taxonomic and distributional similarities. Our graph-based semi-supervised method is generic to extract any type of fine-grained sub-property or attribute where we need only a few seeds to begin. Our methods are flexible enough to consider any lexical database that has a distinction across different senses of a word and provides short glosses of these senses (e.g., Wiktionary).

The second contribution of this dissertation is a new method to extract and organize large-scale comparative commonsense. Before our work, semantically organized comparative commonsense had never been studied or compiled before. The constituents of a comparative assertion are strongly related; our method builds upon this observation to jointly disambiguate and classify the assertions. We consider adjectival phrases as relations, however, the machinery allows for any type of phrase, e.g., verbal phrases common in openIE approaches. Thus, our method can generalize to semantic organization of openIE triples.

The third contribution of this dissertation is a new method to extract and organize large-scale part-whole commonsense. We acquire the assertions from text, distilling them with our statistical and logical components. Our pattern and assertion ranking methods generalize to any relation with finer-grained sub-relations. Further, we mine novel attributes like cardinality (using multiple languages) and visibility (using images). To estimate visibility, we verify the assertions in images (we call this quasi-visual verification). Quasi-visual verification leverages the best of both text only verification (which is inaccurate due to reporting bias), and visual only verification (which is inaccurate due to the object detectors inaccuracies). Our method generalizes to any commonsense relation that has multiple sub-relations and is observable in images, e.g., `hasLocation` relation (with sub-relations such as `hasLocationAbove`, `hasLocationBelow`, etc.).

The fourth contribution of this dissertation is a new method to extract and organize semantic frames of human activities, together with their visual content. We acquire knowledge about human activities from a novel, multimodal source of rich activities: movie scripts. Our method considers joint semantic role labeling and word sense disambiguation for parsing these scripts to generate candidate activity frames. We then perform inference using probabilistic graphical models that can encode joint dependencies among the candidate activity frames. Unlike the previous contribution, this method goes beyond disambiguation of the arguments of an assertion; and, additionally assign roles to these arguments.

Together, these methods have been used to create the WebChild KB, which is one of the largest commonsense knowledge bases available, describing over 2 million disambiguated concepts and activities, connected by over 18 million assertions. The WebChild KB is bigger, richer and cleaner than any other automatically constructed commonsense KB. WebChild KB can also be viewed as an extended WordNet (comprising not just words, but also activities and extended concepts), with orders of magnitude denser relation graph (connecting the concepts with novel relations such as comparatives), and additionally with some visuals.

From a resource perspective, people looking for commonsense knowledge bases had few options available before our construction of the WebChild knowledge base. The available alternatives do not offer the same level of size, richness and semantic rigor over multiple modalities. The WebChild KB has already been effective in providing background knowledge to various applications ranging from text to vision. WebChild KB is freely available for download from `http://www.mpi-inf.mpg.de/yago-naga/webchild/`.

8.2 Outlook

The WebChild knowledge base can serve as a catalyst for new research in text mining, computer vision, as well as multimedia search, as shown in Chapter 7.

As the WebChild KB is semantically organized and mapped to WordNet, it also allows for semantic reasoning.

With the computer vision systems, such as object detectors and activity detectors, getting more sophisticated and robust, multimodality will play a bigger role. Our approach to exploit multimodal data is simple, scalable, and allows for better computer vision (e.g., improving object detection by exploiting visible part-whole knowledge) as well as richer knowledge bases (e.g., inferring visibility of part-whole knowledge).

WebChild KB opens up new research avenues. To list a few: first, Knowlywood's activity taxonomy can be used to generate activity classes for a computer vision based activity recognizer. Some activities are not visual in nature, e.g., *chair a seat* while others are visual, e.g., *sit on a chair*. Thus, a research avenue is to automatically estimate the visual nature of activities in order to train the activity recognizers exclusively over visual activities. Second, even though some activities are visual in nature, their temporal duration is very long. An activity recognizer cannot be trained to robustly identify such activity classes. The research avenue is to automatically estimate the temporal scoping of activities, e.g., $(|dinner|_{duration} < |holiday|_{duration})$.

We believe that our research has set the stage for the next level of reasoning in applications that can use commonsense knowledge for natural human-computer interactions.

List of Figures

List of Tables

Bibliography

Akbik, A. and Löser, A. (2012). Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 52–56. Association for Computational Linguistics. (cited on page 24)

Akbik, A. and Michael, T. (2014). The weltmodell: A data-driven commonsense knowledge base. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA). (cited on page 22, 31)

Almuhareb, A. and Poesio, M. (2004). Attribute-based and value-based clustering: An evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 4, pages 158–165. (cited on page 22, 32, 53)

Anacleto, J., Lieberman, H., Tsutsumi, M., Neris, V., Carvalho, A., Espinosa, J., Godoi, M., and Zem-Mascarenhas, S. (2006). Can common sense uncover cultural differences in computer applications? In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 1–10. Springer. (cited on page 11)

Artzi, Y., FitzGerald, N., and Zettlemoyer, L. S. (2013). Semantic parsing with combinatory categorial grammars. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).* (cited on page 103)

Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R., and Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference (ISWC)*, LNCS 4825. Springer. (cited on page 2)

Baader, F., Horrocks, I., and Sattler, U. (2008). *Description logics*, volume 3. Elsevier New York, NY, USA. (cited on page 14)

Bagherinezhad, H., Hajishirzi, H., Choi, Y., and Farhadi, A. (2016). Are elephants bigger than butterflies? reasoning about sizes of objects. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).* (cited on page 62)

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).* (cited on page 110)

Bar-Hillel, Y. (1960). The present status of automatic translation of languages. *Advances in computers*, 1(1):91–163. (cited on page 29)

Baroni, M. and Zamparelli, R. (2010). Nouns are vectors and adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* (cited on page 22, 32)

Benamara, F., Cesarano, C., Picariello, A., Recupero, D. R., and Subrahmanian, V. S. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Web and Social Media (ICWSM).* (cited on page 59)

Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* (cited on page 103)

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).* (cited on page 82, 126)

Blanco, E., Cankaya, H. C., , and Moldovan, D. (2011). Commonsense knowledge extraction using concepts properties. In *Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS).* (cited on page 16)

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.* (cited on page 2)

Brants, T. and Franz, A. (2006). Web 1t 5-gram version 1. *Linguistic Data Consortium.* (cited on page 37, 91)

Brin, S. (1998). Extracting patterns and relations from the world wide web. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 172–183. Springer. (cited on page 25)

Brown, L. D., Cai, T. T., and DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science*. (cited on page 22, 53, 92, 112, 114)

Cao, Y., Cao, C., Zang, L., Wang, S., and Wang, D. (2010). Extracting comparative commonsense from the web. In *Proceedings of the International Conference on Intelligent Information Processing*, pages 154–162. Springer. (cited on page 63)

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. H., and Mitchell, T. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. (cited on page 25, 65, 83)

Chambers, N. and Jurafsky, D. (2009). Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*. (cited on page 97)

Chen, J., Tandon, N., and Gerard de Melo (2015). Neural word representations from large-scale commonsense knowledge. In *Proceedings of the International Conference on Web Intelligence (WI)*. (cited on page 8, 130)

Chen, X., Shrivastava, A., and Gupta, A. (2013). Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1409–1416. (cited on page 6, 28, 83, 91, 98)

Chklovski, T. and Pantel, P. (2004). VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 98)

Clark, P. and Harrison, P. (2009). Large-scale extraction and use of knowledge from text. In *Proceedings of the Fifth International Conference on Knowledge Capture (KCAP)*, pages 153–160. ACM. (cited on page 21, 33)

Conesa, J., Storey, V. C., and Sugumaran, V. (2008). Improving web-query processing through semantic knowledge. *Data & Knowledge Engineering*, 66(1):18–34. (cited on page 17)

Corro, L. D. and Gemulla, R. (2013). Clausie: clause-based open information extraction. In *Proceedings of the International World Wide Web Conference (WWW)*. (cited on page 22, 106)

Dahlgren, K., McDowell, J., and Stabler, E. P. (1989). Knowledge representation for commonsense reasoning with text. *Computational Linguistics*, 15(3):149–170. (cited on page 14)

Dalvi, B. B., Cohen, W. W., and Callan, J. (2012). Websets: Extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. (cited on page 40, 124)

Davidov, D. and Rappoport, A. (2010). Extraction and approximation of numerical attributes from the web. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1308–1317. Association for Computational Linguistics. (cited on page 62)

Davis, E. and Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103. (cited on page 15)

de Melo, G. and Weikum, G. (2009). Towards a universal wordnet by learning from combined evidence. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pages 513–522. ACM. (cited on page 58)

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 18, 29)

Divvala, S., Farhadi, A., and Guestrin, C. (2014). Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3270–3277. (cited on page 6, 27)

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam (2011). Open information extraction: The second generation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. (cited on page 22, 65)

Fader, A., Soderland, S., and Etzioni, O. (2011a). Identifying relations for open information extraction. In *Proceedings of the Conference on Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics. (cited on page 22)

Fader, A., Soderland, S., and Etzioni, O. (2011b). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* (cited on page 117)

Fader, A., Zettlemoyer, L., and Etzioni, O. (2014). Open question answering over curated and extracted knowledge bases. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD).* (cited on page 103)

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785. IEEE. (cited on page 33)

Faruqui, M. and Pado, S. (2012). Towards a model of formal and informal address in english. In *Proceedings of the Annual conference of the European Association for Computational Linguists (EACL).* (cited on page 112)

Fei, G., Liu, B., Hsu, M., Castellanos, M., and Ghosh, R. (2012). A dictionary-based approach to identifying aspects implied by adjectives for opinion mining. In *24th International Conference on Computational Linguistics*, page 309. (cited on page 32)

Fellbaum, C. and Miller, G. (1998). Wordnet: An electronic lexical database. (cited on page 2, 5)

Galárraga, L., Heitz, G., Murphy, K., and Suchanek, F. (2014). Canonicalizing Open Knowledge Bases. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM).* (cited on page 15)

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288. (cited on page 103)

Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).* (cited on page 82)

Girju, R., Badulescu, A., and Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics*. (cited on page 82)

Gupta, A. and Davis, L. S. (2008). Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proceedings of the European conference on computer vision (ECCV)*, pages 16–29. Springer. (cited on page 62)

Hartung, M. and Frank, A. (2010). A structured vector space model for hidden attribute meaning in adjective-noun phrases. In *Proceedings of the International Conference on Computational Linguistics (COLING)*. (cited on page 32, 53, 55)

Hartung, M. and Frank, A. (2011). Exploring supervised lda models for assigning attributes to adjective-noun phrases. In *Proceedings of the Conference on Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 540–551. Association for Computational Linguistics. (cited on page 32, 52, 53, 54, 55, 137)

Havasi, C., Speer, R., and Alonso, J. (2007). Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Proceedings of the International Conference on Recent advances in natural language processing (RANLP)*. (cited on page 20, 83, 91)

Hayes, P. J. (1979). *The logic of frames*, volume 46. Citeseer. (cited on page 14)

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An Electronic Lexical Database*, pages 305–332. (cited on page 43)

Hoffman, J., Guadarrama, S., Tzeng, E., Donahue, J., Girshick, R., Darrell, T., and Saenko, K. (2014). LSDA: Large scale detection through adaptation. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*. (cited on page 89)

Hsu, M.-H., Tsai, M.-F., and Chen, H.-H. (2006). Query expansion with conceptnet and wordnet: An intrinsic comparison. In *Proceedings of the Asia Information Retrieval Symposium*, pages 1–13. Springer. (cited on page 29)

Ittoo, A. and Bouma, G. (2010). On learning subtypes of the part-whole relation: do not mix your seeds. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*. (cited on page 82)

Ittoo, A. and Bouma, G. (2013). Minimally-supervised extraction of domain-specific part–whole relations using wikipedia as knowledge-base. *Data & Knowledge Engineering*, 85:57–79. (cited on page 82)

Jain, A. and Pantel, P. (2011). How do they compare? automatic identification of comparable entities on the Web. In *Proceedings of the IEEE International Conference on Information Reuse & Integration.* (cited on page 61)

Jang, M., Park, J.-w., and Hwang, S.-w. (2012). Predictive mining of comparable entities from the web. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).* (cited on page 61)

Jindal, N. and Liu, B. (2006). Mining comparative sentences and relations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1331–1336. AAAI Press. (cited on page 61, 78)

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D. A., Bernstein, M. S., and Fei-Fei, L. (2015). Image retrieval using scene graphs. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678. IEEE. (cited on page 28)

Keet, C. M. and Artale, A. (2008). Representing and reasoning over a taxonomy of part–whole relations. *Applied Ontology*, 3(1-2):91–110. (cited on page 81, 82, 88)

Kim, E., Helal, S., and Cook, D. (2010). Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48–53. (cited on page 30)

Kimmig, A., Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. (2013). A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications (NIPS Workshop).* (cited on page 104, 108)

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2006). Extending VerbNet with novel verb classes. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC).* (cited on page 18, 99, 105, 106)

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332.* (cited on page 20)

Kuo, Y.-l., Lee, J.-C., Chiang, K.-y., Wang, R., Shen, E., Chan, C.-w., and Hsu, J. Y.-j. (2009). Community-based game design: experiments on social games for commonsense data collection. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 15–22. ACM. (cited on page 19)

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* (cited on page 101, 104)

Lazaridou, A., Dinu, G., Liska, A., and Baroni, M. (2015). From visual attributes to adjectives through decompositional distributional semantics. *arXiv preprint arXiv:1501.02714.* (cited on page 33)

Lebani, G. and Pianta, E. (2012). Encoding commonsense lexical knowledge into wordnet. In *Proceedings of the Global WordNet Conference.* (cited on page 6, 21, 33)

Lee, K., Artzi, Y., Dodge, J., and Zettlemoyer, L. (2014). Context-dependent semantic parsing for time expressions. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).* (cited on page 103)

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38. (cited on page 5, 83)

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM. (cited on page 43)

Levesque, H. J., Davis, E., and Morgenstern, L. (2011). The winograd schema challenge. In *Proceedings of the AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.* (cited on page 29)

Lieberman, H., Liu, H., Singh, P., and Barry, B. (2004). Beating common sense into interactive applications. *AI Magazine*, 25(4):63. (cited on page 20, 29)

Lin, D. and Pantel, P. (2001). Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD).* (cited on page 98)

Lin, T., Etzioni, O., et al. (2012). Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 84–88. Association for Computational Linguistics. (cited on page 62)

Lin, X. and Parikh, D. (2015). Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2984–2993. (cited on page 28)

Ling, X., Clark, P., and Weld, D. S. (2013). Extracting meronyms for a biology knowledge base using distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction (AKBC)*, pages 7–12. ACM. (cited on page 82)

Liu, H. and Singh, P. (2004). Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226. (cited on page 20)

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge University Press. (cited on page 119, 130)

Mascardi, V., Cordì, V., and Rosso, P. (2007). A comparison of upper ontologies. In *Proceedings of the Workshop on Objects and Agents*, volume 2007, pages 55–64. (cited on page 15)

Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., and Lenat, D. (2005). Searching for common sense: Populating Cyc from the Web. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).* (cited on page 83, 98)

McCarthy, D. (2001). *Lexical acquisition at the syntax-semantics interface: diathesis alternations, subcategorization frames and selectional preferences.* Citeseer. (cited on page 31, 84)

McCarthy, D. and Carroll, J. (2003). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654. (cited on page 32)

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. A. (2007). Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33:553–590. (cited on page 2, 43)

Miller, G. A. and Hristea, F. (2006). Wordnet nouns: Classes and instances. *Computational Linguistics*, 32(1):1–3. (cited on page 15)

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2015). Never-ending learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. (cited on page 83)

Modi, A. and Titov, I. (2014). Inducing neural models of script knowledge. In *Proceedings of the International Conference on Computational Natural Language Learning (CoNLL)*. (cited on page 97)

Nakashole, N., Weikum, G., and Suchanek, F. (2012). PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 98)

Navigli, R. (2009). Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69. (cited on page 67, 103)

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems (FOIS)*. (cited on page 98)

Orbach, M. and Crammer, K. (2012). Graph-based transduction with condence. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML/PKDD)*. (cited on page 60)

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106. (cited on page 103)

Palmer, M., Gildea, D., and Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103. (cited on page 103)

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*. (cited on page 82, 91)

Pasca, M. (2014a). Acquisition of open-domain classes via intersective semantics. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 551–562. (cited on page 98)

Pasca, M. (2014b). Queries as a source of lexicalized commonsense knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1091. (cited on page 21)

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). WordNet::Similarity: Measuring the relatedness of concepts. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. (cited on page 67, 109, 110)

Ponzetto, S. P. and Strube, M. (2011). Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10). (cited on page 98)

Rajagopal, D. and Tandon, N. (2015). A proposal of the marriage of encyclopedic and commonsense knowledge. In *CMU LTI-SRS symposium*. (cited on page 8)

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, pages 41–47. (cited on page 82)

Regneri, M., Koller, A., and Pinkal, M. (2010). Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 979–988. Association for Computational Linguistics. (cited on page 29, 97)

Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., and Pinkal, M. (2013). Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36. (cited on page 98)

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1-2):107–136. (cited on page 26)

Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. (2015). A dataset for movie description. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3212. (cited on page 9, 120)

Rohrbach, M., Amin, S., Andriluka, M., and Schiele, B. (2012). A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 99, 112)

Rohrbach, M., Stark, M., and Schiele, B. (2011). Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 30, 81, 126, 127)

Ruiz-Casado, M., Alfonseca, E., and Castells, P. (2007). Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Data & Knowledge Engineering*, 61(3):484–499. (cited on page 82)

Sadeghi, F., Divvala, S. K., and Farhadi, A. (2015). Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1456–1464. IEEE. (cited on page 27)

Saxena, A., Jain, A., Sener, O., Jami, A., Misra, D. K., and Koppula, H. S. (2014). Robobrain: Large-scale knowledge engine for robots. *arXiv preprint arXiv:1412.0691*. (cited on page 18, 30)

Schank, R. C. and Abelson, R. P. (1977). Scripts, plans, goals and understanding: An inquiry into human knowledge structures. *Mhwah, NJ (US): Lawrence Erlbaum Associates*. (cited on page 97)

Schubert, L. (2002). Can we derive general world knowledge from texts? In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 94–97. (cited on page 22)

Schuler, K. K., Korhonen, A., and Brown, S. W. (2009). VerbNet overview, extensions, mappings and applications. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Tutorial Abstracts*. (cited on page 99, 109)

Shamma, D. (2014). One hundred million Creative Commons Flickr images for research. `http://labs.yahoo.com/news/yfcc100m/`. (cited on page 89, 91)

Shaoul, C. (2010). The westbury lab wikipedia corpus. *Edmonton, AB: University of Alberta*. (cited on page 91)

Shibata, T. and Kurohashi, S. (2011). Acquiring strongly-related events using predicate-argument co-occurring statistics and case frames. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP).* (cited on page 97)

Shutova, E., Tandon, N., and de Melo, G. (2015). Perceptually grounded selectional preferences. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL).* (cited on page 9, 33, 58)

Smith, B. (1995). Formal ontology, common sense and cognitive science. *International journal of human-computer studies*, 43(5):641–667. (cited on page 96)

Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A. L., and Rosse, C. (2005). Relations in biomedical ontologies. *Genome Biology*, 6(R46). (cited on page 81)

Speer, R. and Havasi, C. (2012). Representing general relational knowledge in conceptnet 5. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC).* (cited on page 6, 20, 83)

Speer, R., Havasi, C., and Lieberman, H. (2008). Analogyspace: Reducing the dimensionality of common sense knowledge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 8, pages 548–553. (cited on page 26)

Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *International Conference on Extending Database Technology*, pages 1–17. Springer. (cited on page 109)

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis.* (cited on page 128)

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In *Proceedings of the International World Wide Web Conference (WWW).* (cited on page 2)

Takamura, H. and Tsujii, J. (2015). Estimating numerical attributes by bringing together fragmentary clues. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).* (cited on page 62)

Talukdar, P. and Crammer, K. (2009). New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML/PKDD)*. (cited on page 35, 38, 39)

Tandon, N. (2011). Deriving a web-scale common sense fact knowledge base. Master's thesis, Universität des Saarlandes Saarbrücken. (cited on page 25)

Tandon, N. and De Melo, G. (2010). Information extraction from web-scale n-gram data. In *Proceedings of the Web N-gram Workshop at ACM SIGIR 2010*, volume 7. (cited on page 25)

Tandon, N., de Melo, G., De, A., and Weikum, G. (2015a). Knowlywood: Mining activity knowledge from hollywood narratives. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*. (cited on page 8)

Tandon, N., de Melo, G., Suchanek, F., and Weikum, G. (2014a). Webchild: Harvesting and organizing commonsense knowledge from the web. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. (cited on page 9, 83)

Tandon, N., de Melo, G., and Weikum, G. (2011). Deriving a web-scale common sense fact database. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. (cited on page 6, 25, 33, 34, 82, 83, 86, 87, 91)

Tandon, N., De Melo, G., and Weikum, G. (2014b). Acquiring comparative commonsense knowledge from the web. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 166–172. (cited on page 9)

Tandon, N., Hariman, C., Urbani, J., Rohrbach, A., Rohrbach, M., and Weikum, G. (2016). Commonsense in parts: Mining part-whole relations from the web and image tags. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*. (cited on page 8)

Tandon, N., Rajagopal, D., and de Melo, G. (2012). Markov chains for robust graph-based commonsense information extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING): Demos*, pages 439–446. (cited on page 9, 25)

Tandon, N., Weikum, G., Melo, G. d., and De, A. (2015b). Lights, camera, action: Knowledge extraction from movie scripts. In *Proceedings of the 24th*

International World Wide Web Conference: Companion, pages 127–128. International World Wide Web Conferences Steering Committee. (cited on page 9)

Trummer, I., Halevy, A., Lee, H., Sarawagi, S., and Gupta, R. (2015). Mining subjective properties on the web. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1745–1760. ACM. (cited on page 11)

Varadarajan, K. M. and Vincze, M. (2012). Afnet: The affordance network. In *Proceedings of the Asian Conference on Computer Vision*, pages 512–523. Springer. (cited on page 98)

Varzi, A. C. (2010). The stanford encyclopedia of philosophy. (cited on page 81)

Vedantam, R., Lin, X., Batra, T., Lawrence Zitnick, C., and Parikh, D. (2015). Learning common sense through visual abstraction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2542–2550. (cited on page 27)

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*. (cited on page 29)

Von Ahn, L. (2006). Games with a purpose. *Computer*, 39(6):92–94. (cited on page 19)

Wang, R. and Cohen, W. (2007). Language-independent set expansion of named entities using the web. In *Proceedings of the IEEE International Conference on Data Mining (ICDM*. (cited on page 124)

Winograd, T. (1972). Understanding natural language. *Cognitive psychology*, 3(1):1–191. (cited on page 29)

Winston, M. E., Chaffin, R., and Herrmann, D. (1987). A taxonomy of part-whole relations. *Cognitive science*, 11(4):417–444. (cited on page 81, 82)

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492. IEEE. (cited on page 30)

Xiao, P. and Blat, J. (2013). Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity*. (cited on page 76)

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: open information extraction on the web. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Demonstrations*, pages 25–26. Association for Computational Linguistics. (cited on page 22)

Zhong, Z. and Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*. (cited on page 87, 103, 107)

Zhu, Y., Fathi, A., and Fei-Fei, L. (2014). Reasoning about Object Affordances in a Knowledge Base Representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 27)