Labeling and Automatically Identifying Basic-Level Categories

Chad Mills

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Gina-Anne Levow, Chair

Fei Xia

Luke Zettlemoyer

Program Authorized to Offer Degree:

Linguistics

University of Washington

Abstract

Labeling and Automatically Identifying Basic-Level Categories

Chad Mills

Chair of the Supervisory Committee:

Gina-Anne Levow

Department of Linguistics

Basic-level categories are the primary categories humans use to think and communicate; they are the first categories learned, with numerous psychological advantages including quick exemplar recognition time. They are valuable in a range of applications such as assessing text readability. Using WordNet, we create the first broad, representative dataset to build and evaluate systems to identify basic-level categories. We show there is significant label bias in the limited labels available in the psychology literature, and we add one novel label value since we find some chains in a hypernym/hyponym hierarchy do not include basic-level categories. We expand the number of labels available by a factor of 72, from 152 to 11,221. We build a heuristic baseline system to detect basic-level categories, showing systems evaluated on the previously-available data can look twice as effective as they perform on a more broadly-representative dataset. We take advantage of the increased quantity of labeled data to build a classifier-based system that improves performance to an f-measure of 0.607 from 0.381 for the heuristic-based system. We demonstrate basic-level categories may be useful in a range of applications. For measuring text readability, we show lower reading levels have proportionally more basic-level categories and our comparison of the reading levels of Wikipedia and Simple Wikipedia using basic-level categories alone aligns well with existing research in the area. We also show that image captions tend to be much more likely to include basic-level categories than normal text, further suggesting that basic-level categories may be a useful signal in language grounding applications.

Acknowledgements

I would like to thank my advisor, Gina-Anne Levow, for her support and encouragement throughout my graduate studies. Her directional advice and thoughtful comments have helped to strengthen my work. She also shared relevant experiences which have helped me to progress faster than I otherwise would have, something critical to the completion of this work given my pursuit of a full-time career in parallel with this work. I would also like to thank Fei Xia and Luke Zettlemoyer for their helpful conversations, useful feedback, and suggestions. Thanks to Meliha Yetisgen for giving her time to serve as my Graduate School Representative to preserve the integrity of the process.

Thanks to the Linguistics Department Administrator, Mike Furr, for going out of his way to make it easy to navigate processes and paperwork which have seemed much more cumbersome in my experiences outside of this institution.

Thanks to my wife Ginger for her constant encouragement and support, without which this dissertation would not exist.

Dedication

To Ginger

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Basic-level categories are the primary categories humans form to understand and communicate about the world around them. They group together entities along the most basic cuts in reality. Examples include ***table, car, tree, bird, guitar, shirt, fish,*** and ***apple*** (Rosch et al. 1976). This is at an intermediate level of specificity; in the example ***car***, the basic-level is not as broad as ***vehicle*** (its superordinate) or as narrow as ***sedan*** (a subordinate). These basic-level categories have been shown to have a number of interesting and useful properties, including that they are the first type of categories a child learns and they are used to identify or describe an object in a neutral context (Rosch et al. 1976).

Given that basic-level categories are important to human cognition and communication, researchers have found that knowing them is helpful for a variety of practical applications. Basic-level categories are helpful in word sense disambiguation (Legrand 2006), image searches (Rorissa et al. 2008), ad targeting (Wang et al. 2015), accurately measuring the readability of a text (Lin et al. 2009), making search result entity cards more easily consumable (Wang et al. 2015), linking together different domain-specific information classification systems (Green 2006), and user-centered design of image-browsing interfaces (Rorissa et al. 2008).

Despite this wide variety of practical applications, to the best of our knowledge all attempts to identify basic-level categories have been ad hoc as part of a broader practical system. Although a handful of examples of basic-level categories are available in the psychology literature, we are not aware of any attempts to identify a broad-coverage dataset including basic-level category labels for use in a variety of applications. Without a common data set, the several systems that have been built to identify basic-level categories have not been evaluated as such and rather their assistance in an application has been measured.

We attempt to fill this gap by using crowd-sourcing to build a labeled dataset for the creation and evaluation of systems to detect basic-level categories. We use properties of basic-level categories confirmed by experiment (Rosch et al. 1976) to generate instructions for crowd-source workers to follow in identifying basic-level categories. We then use the small number of examples from psychology experiments on basic-level categories (Rosch et al. 1976, Markman et al. 1997) to train the annotators as well as to validate the labeling process, producing a large set of labeled data.

This labeling process, as well as our subsequent work, depends heavily on the taxonomic structure of Princeton WordNet (Miller 1995), which is a widely-used, broad-coverage lexical database that organizes English word senses (synsets) into a hypernym/hyponym hierarchy. This aligns well to the framework used in research on basic-level categories, where these are contrasted against more general (superordinate) and more specific (subordinate) categories (Rosch et al. 1976).

Using our labels, we build a baseline heuristic-based system to identify synsets in WordNet which correspond to basic-level categories. We use this system to show that the previously-available examples of basic-level categories from psychology experiments are biased; when used to build and evaluate systems to detect basic-level categories, these examples can lead to overreporting effectiveness by a substantial margin.

We next take advantage of the large set of labeled data to build a classifier-based system to identify basic-level categories in WordNet, showing it performs substantially better than our heuristic baseline. Finally, we show that this basic-level category data appears to be a useful signal in practical applications including measuring text readability and automatic image captioning.

1.2 Contributions

The primary contributions of this work include:

- A demonstration that previously-available basic-level category labels contain substantial bias
- The first broad, representative set of labels for basic-level noun categories
- An extension to the theoretical framework used for basic-level categories which acknowledges the existence of hypernym/hyponym chains which do not contain any basic-level categories

- A system to identify basic-level categories in WordNet
- An efficient algorithm for selecting the maximum likelihood set of basic-level category candidates in a tree structure obeying the constraint that a maximum of one node may be chosen in each chain from leaf to root in the tree
- A demonstration that knowing basic-level categories appears promising for use in a range of practical applications

Chapter 2

Literature Survey

We review foundational psychology literature on basic-level categories as well as epistemology literature on first-level concepts. We then discuss Princeton WordNet, a lexical resource well-suited to work in this area, as well as work on detecting hypernym/hyponym relationships. We follow this with a discussion of identifying entry-level categories and the few existing approaches to identifying basic-level categories. We also describe relevant work on Amazon Mechanical Turk to crowd-source labels since this resource was instrumental in building our dataset. Finally, we discuss work to measure the readability of a text, which is an area of application we consider for this work.

2.1 Basic-level Categories

We first describe categories themselves in §2.1.1, and then discuss the basic-level in §2.1.2.

2.1.1 Categories and Concepts

A category is a means of classification where a single term is used to refer to any one of many distinct entities, where each is treated as essentially the same in an important respect (Rosch 1999). As an example, one individual entity may be referred to as a "cow," and even though there is no other entity with the exact same size, shape, coloring pattern, sound, etc., we naturally group many similar entities under this same category "cow." Humans do this naturally, but how this works has been a subject of much debate over millennia (Porphyry 270, Acquinas 1274, Ockham 1323, Kant 1781, Mill 1884, Rand 1966a, Rosch 1975, Aristotle c.350 BC, Plato c.380 BC); the way we are able to use a single category to denote an unlimited number of entities that have no single attribute exactly in common is referred to as the problem of universals.

Some researchers refer to categories and others to concepts; some have attempted to make clear distinctions between concepts and categories, such as denoting a category as the set of entities referred to by a term while a concept is a mental representation of this set (Rips et al. 2012). This distinction being inconsequential to our current work, and having no reason to suppose categories themselves exist

independent of a human mind doing the conceptualization, we treat these terms as synonymous and use them interchangeably. Nonetheless, due to the heavy reliance of this work on experimental results from the psychology literature where the term “category” is used, outside of this literature review we generally use “category” rather than “concept”.

Izquierdo et al. (2007) is careful also to point out the differences between basic-level categories and the similarly-named base concepts, which could be a source of confusion for those familiar with the latter. Base concepts are a set of concepts core to many relations and which tend to occur relatively high in the hierarchy (Izquierdo et al. 2007). These are often used to identify key concepts showing up across multiple languages and as a baseline set of concepts that should be included in a WordNet for a new language. On the other hand, while there is certainly overlap, basic-level categories tend to occur closer to the middle of the hierarchy and tend to have less relations (Izquierdo et al. 2007).

There are two different viewpoints addressing the problem of universals relevant to this work.

The first, used as the basis for the discussion of basic-level categories in §2.1.2, is prototype theory (Rosch 1973). In this view, categories are not well-defined groups with strict boundaries as in the classical view where every member of the category is similarly representative of the category as a whole; rather, people latch onto a prototypical member of the category and judge category membership by determining an exemplar is similar to the prototype. This allows for some members to be more strongly associated with the category label than others, some easier to recognize than others, and other properties which have been verified experimentally.

The second viewpoint addressing the problem of universals is within the classical tradition in epistemology and describes a related idea of first-level concepts; this is discussed in §2.2.

2.1.2 The Basic Level

Brown (1958) noticed that children learn a middle level of concepts before concepts that are broader or more specific. Building on this insight, Rosch et al. (1976) distinguish between three levels of categories: basic-level, superordinate (hypernyms of the basic-level), and subordinate (hyponyms of the basic level). They describe the basic-level as the “level of abstraction at which the most basic category cuts are made”

(Rosch et al. 1976). An important motivation for this idea is that attributes in the world aren't uniformly distributed; entities with feathers are also likely to have wings but not fur (Rosch et al. 1976). These "basic category cuts," then, are groupings of entities possessing a set of these related properties. Rosch et al. (1976) describe advantages in cognitive economy in using categories that cut across these natural boundaries, as well as describing how categories at this level of inclusiveness have a high cue validity (attribute presence is a strong predictor of the category). While perhaps not offering the most succinct definition, Rosch et al. (1976) do describe and experimentally validate many properties of basic-level categories, discussed infra.

Markman et al. (1997) offer what may be a more fundamental and clear definition of the basic-level as being the level with the most alignable differences. An alignable difference is a difference in degree rather than kind; for example, cars and motorcycles have a different number of wheels (alignable) but a car carries a jack and a motorcycle does not (non-alignable). ***Car*** and ***motorcycle*** here are both taken to be basic-level categories, while ***vehicle*** is a superordinate and ***coupe*** is a subordinate. The various subordinates of car (***coupe***, ***sedan***, etc.) vary in a handful of ways, but they have more similarities than differences. Cars and motorcycles, on the other hand, have many more differences and many of these are alignable (number of wheels, type of seat, steering controls, acceleration controls, etc.). According to (Markman et al. 1997), this abundance of alignable differences is a clear indicator that ***car*** and ***motorcycle*** are basic-level.

Though not offering a clear definition, Rosch et al. (1976) provide a broad array of experiments showing a number of interesting properties of basic-level categories. They find that basic-level categories are the most inclusive level at which exemplars include many attributes in common. Basic-level categories are also the most inclusive level at which people associate exemplars with similar motor movements they would expect to perform in relation to the exemplars. The basic-level is the most inclusive level at which exemplars share a similar shape, and thus also the most inclusive level at which people can readily identify objects based on their shape. Judging whether two objects are the same type of object or not is possible for both subordinates and basic-level categories, while it's faster for people to recognize category membership from an image for basic-level categories than either subordinates or superordinates. Young children can categorize two objects into their basic-level category but can't categorize two objects in different basic-level

categories into the same superordinate category; category membership is learned earlier for basic-level categories. Children also sort images, without prompting, into taxonomy-based groups at the basic-level but only do this at the superordinate level at later ages. The word used to represent a basic-level category is the most common word used to describe it in language, children learn these words first, and in languages without much taxonomic depth basic-level categories exist even while superordinate and subordinate levels are impoverished. These properties are all verified through experiments and analysis (Rosch et al. 1976).

There has been a wide variety of additional research in this area within psychology showing a range of properties, applications, and even several potential issues with basic-level categories.

Studies have shown children learn basic-level categories first, then subordinates, then superordinates (Jónsdóttir et al. 1996), with children not even considering a novel noun to potentially be a superordinate until around age 7 (Golinkoff et al. 1995).

At the same time, there are some limitations to these advantages. Adult experts in a domain may be so fluent with the subordinate level in that domain that three of the twelve advantages of the basic-level over the subordinate level demonstrated by Rosch et al. (1976) become greatly diminished (Tanaka et al. 1991). None of the properties identifying the basic-level as the most inclusive level at which some property is true are questioned in this follow-up work by Tanaka et al. (1991), suggesting these may be the more durable properties. Additional work has further shown the boundary between basic and superordinate concepts is an important one with qualitative differences in how they are represented, such as superordinate concepts (e.g. ***furniture***) often referring to groups of entities and basic-level (e.g. ***table***) referring to individuals (Murphy et al. 1989). Some interesting corner cases have also been found with abnormal subordinate exemplars, for example with ***penguin*** being shown to have some of the basic-level advantages while ***bird*** is the clear basic-level category when dealing with more typical exemplars of the category (Jolicoeur et al. 1984), again eroding the boundary between basic-level and subordinate concepts without undermining the significant differences found between the basic-level and superordinates..

2.2 First-level Concepts

Another line of work on this same phenomenon is in epistemology, where it is framed differently but with striking resemblances. This work is based on a different theory of concepts addressing the problem of universals. Rather than concepts being formed on the basis of similarity to a prototypical category member, a concept does have a clear definition in line with the classical view of concepts. Concepts are formed by identifying measurable characteristics that category members share and establishing a range of measurements that isolate included exemplars from other entities possessing the same characteristics For example, tables possess a range of shapes including a flat, level surface with supports as well as the purpose of supporting smaller objects; these can be differentiated from the shapes and purposes of other objects possessing these two characteristics, such as chairs.

This approach is in the classical tradition that Rosch was arguing against (Rosch 1999) insofar as concepts represent clearly-defined sets of referents, but the range of measurements included also allows for some referents to be more central and others borderline which prototype theory handles better than treating all referents as equally representative of the category as in most classical theories. Note that this approach to concept formation already bears striking resemblance to the explanation Markman et al. (1997) give for basic-level categories in terms of alignable differences (discussed in §2.1.22.1.1, supra). In this view coming from epistemology, all concepts inherently have alignable differences to their siblings in a hypernym/hyponym hierarchy and the explanation Markman et al. (1997) give that the basic-level has the most alignable differences translates straightforwardly. Additionally, advocates of both theories treat economy of thought and expression as a key motivation for the need for concepts or categories (Rand 1967, Rosch et al. 1976).

In epistemology, what Rosch et al. (1976) refer to as basic-level categories are usually called first-level concepts (Rand 1966a, Binswanger 2014) or occasionally basic-level concepts (Rand 1990) rather than basic-level categories, but they identify the same phenomenon from a different perspective. In this view, first-level concepts are called first-level in part because they’re the first ones learned (an observation also common to Rosch et al. (1976)). In addition, non-first-level concepts are formed by reference to these

first-level concepts and/or their derivatives (Rand 1966b), which makes them primary or “first” in a hierarchical sense in addition to a temporal one.

This is a very different theoretical framework for thinking about basic-level categories, but it is clear the same basic phenomenon is being described. Both theories even give a number of identical examples, including “bird”, “cat”, “chair”, “dog”, “fish”, and “table” (Rand 1966b, Rosch et al. 1976).

Of particular interest for future work, this view from epistemology treats only concepts of entities as truly first-level since concepts of entities are formed before concepts of motion, relationship, attribute, characteristic, etc. (Rand 1966a) However, within each of these other fundamental groups of concepts there is a first-level within that group (Binswanger 2014). For example, “blue” is a first-level attribute concept, formed directly from perceptual data like first-level concepts but only after forming concepts of entities first (Binswanger 2014). We focus on first-level concepts of entities since this is what both theories treat as the basic or first level, as well as because this is where there are many experimentally-validated examples to use as a set of gold-standard labels; however, this indicates much more work could be done to identify the first-level concepts in each of these other major groups of concepts as well.

While the work on first-level concepts in epistemology (Rand 1966a) started before the work in psychology (Rosch et al. 1976), the latter does not cite the former indicating no direct influence between the two. Nonetheless, Rosch et al. (1976) does cite Anglin (1975) on the topic of the first words children learn, which is one of the key properties of both basic-level categories and first-level concepts share. Furthermore, in an influential book on concepts begun five years earlier and published the following year, quotes a passage from Rand (1966a) that he agrees with, even mentioning the latter as a little-read book. Rosch also subsequently cites this book by Anglin in her future work in the area (Mervis et al. 1981). While there is no clear indication of a direct relationship between the two lines of inquiry, Anglin at least serves as an indirect link suggesting there may have been some indirect influence between the two before the present work.

2.3 Princeton WordNet

Princeton WordNet (Miller 1995), hereafter referred to as WordNet, organizes concepts into a hierarchy of hypernyms and hyponyms (Murphy 2003). While WordNet also identifies other information, such as meronymy, basic-level categories are not currently annotated in WordNet. As mentioned in Chapter 1 , basic-level categories are identified in contrast to their superordinates and subordinates (Rosch et al. 1976). This naturally aligns with hypernymy and hyponymy, making WordNet a basic organization of categories well-suited to our task of identifying basic-level categories. WordNet is an open-domain, dictionary-scale resource (Miller 1995).

It is difficult to overstate the extent to which WordNet is used. WordNet is used widely in ontologies, including building ontologies, matching ontologies together (Lin et al. 2008a), and evaluating ontologies (Brank et al. 2005). It is used for measuring text similarity (Gomaa et al. 2013), sentiment analysis (Liu et al. 2012), word sense disambiguation (Navigli 2009), named entity recognition (Nadeau et al. 2007), information retrieval (Haav et al. 2001), and text summarization (Gholamrezazadeh et al. 2009) among many others. This list only includes some of the broad surveys of fields of research where WordNet is widely used as it would be impractical to catalog the entire range of applications of a system cited over 14,000 times (Scholar 2018). The success of WordNet has led to the creation of numerous additional wordnets in other languages (Bond et al. 2012), corpora aligned to WordNet senses (Petrolito et al. 2014), and a variety of other, more specialized machine-readable lexical resources (Baker et al. 1998, Meyers et al. 2004, Schuler 2009).

Given the broad coverage and widespread adoption of WordNet, we frame our task as one which involves identifying basic-level categories within WordNet, rather than as an independent task using WordNet as a reference.

2.4 Hypernym/Hyponym Detection

While WordNet does not contain annotations of basic-level categories, there are a number of other lexical properties and relations which are included and for some of which there's been work to learn them automatically. Since this is indirectly related to the present work, we discuss some of this work here.

Hearst (1998) learns hypernym/hyponym relationships to extend the annotations present in WordNet. This included starting with some syntactic templates like “such NP1 as NP2” where NP2 is reliably a hyponym of NP1 when appearing in corpora inside sentences matching this template (Hearst 1998). They extend this to arbitrary relations by taking a set of manually-determined templates that work reliably, looking for example sentences in a corpus where the same words appear near one another in ways that aren’t consistent with the existing patterns, and identifying additional patterns to add to the set based on that. This results in new relations and concepts to be added to WordNet (Hearst 1998). Similar template-based approaches have been used to learn other relations including synonyms (Lin et al. 2003, Turney et al. 2003), and meronyms (Girju et al. 2003). In addition to work on nouns, these sorts of approaches have also been extended to verbs. Chklovski et al. (2004) use patterns to identify verb antonyms, temporal happens-before, and several other relations.

Later work includes going beyond hand-built templates in identifying hyponyms, including a classifier-based approach using logistic regression and a set of known hyponym and non-hyponym relationships to learn these relationships without relying on hand-crafted rules (Snow et al. 2005). Snow et al. (2006) then extend and generalize this to combine multiple classifiers on different relations to both learn taxonomies and avoid lexical ambiguity by operating at the sense level rather than the token level.

More recently, identifying hyponymy and entailment using distributional semantics has become an active area of research; we discuss several relevant examples here. Much of this recent work uses unsupervised approaches, particularly word embeddings, as an important component in the system (Henderson et al. 2016, Chang et al. 2017). Weeds et al. (2014) builds classifiers using operations on vectors representing the distributional semantics of two words to identify hyponymy and co-hyponym relations between them. Roller et al. (2016) present qualitative analysis on an entailment classifier and show that the most recognizable and common patterns learned are Hearst patterns (Hearst 1992) like “such as” (e.g. “animals such as cats”). These modern approaches are much more sophisticated and scalable solutions; the fact that this captures and heavily relies on Hearst patterns without needing them to be specified by the experimenter highlights how recent work in this area builds on the earlier, more heuristic approaches to identifying hyponymy in corpora.

2.5 Identifying Basic-level Categories

There has been very little work specifically on detecting basic-level categories at scale. The experiments in psychology have dozens of examples of basic-level categories (Rosch et al. 1976, Markman et al. 1997), with 29 validated individually in experiments (Rosch et al. 1976) and another 63 proposed by later researchers and shown to share tendencies toward basic-level categories in aggregate (Markman et al. 1997).

There have only been a few efforts to use this data to learn patterns and extrapolate to a broader set of basic-level categories, all working with WordNet, though some of the psychology literature also points out attributes of basic-level categories that may be helpful.

Stephen et al. (2009) starts with all nouns and does some filtering of superordinates and subordinates by depth in the hierarchy. This is followed by a voting scheme to pick the best candidate on each path from the top of the hierarchy to a leaf node, considering how short the word is, how frequently the word is used, and how many words are in the synset all as positive features while having few hyponyms and fewer relationships with other synsets more broadly as negative features (Green 2006). There is no effort to reconcile results from nearby paths down the hierarchy, though, and the list of basic-level categories generated is fed into a downstream system to map information systems together, with no evaluation of the categories themselves. This is purely heuristic, and all of the available data is used to define the rules, making evaluation difficult.

Another effort focuses on word sense disambiguation, with Izquierdo et al. (2007) using a simpler approach that filters out the lower levels of the hierarchy and searches up the hypernym hierarchy exclusively looking for a synset with a large number of WordNet relations. These features are already included by Green (2006) and here as well the evaluation is only performed on the applied system and an evaluation is not performed on this basic-level category identification system as such.

Lin et al. (2009) attempt to identify basic-level categories by looking for words that are shorter than their hyponyms and where the word is frequently contained within its hyponyms as a compound. Again, this is

only evaluated in the application of measuring text readability, and like the other experiments they use all the available data for forming the rules without holding aside any data for an independent evaluation.

These systems each share many attributes in common, including many of the attributes used in heuristics, the basic approach itself, and the lack of a direct evaluation.

2.6 Identifying Entry-level Categories in Images

Motivated by the work on basic-level categories, Ordonez et al. (2013) work to identify entry-level categories, adapted from the "entry point level" terminology Jolicoeur et al. (1984) use to denote the word used to refer to an object from perceptual data. This is one of the properties Rosch et al. (1976) note is true of basic-level categories, though in their more detailed analysis of this particular property Jolicoeur et al. (1984) show that basic-level categories and entry point level categories are not always the same. Of the twelve properties of basic-level categories identified by Rosch et al. (1976), this entry level property is one of the three properties found to vary between inexperienced individuals and experts by Tanaka et al. (1991) as discussed in §2.1.2. These both suggest that being entry-level may be highly correlated with basic-level categories but not an essential property of basic-level categories as originally indicated by Rosch et al. (1976).

Ordonez et al. (2013) show that the word used to refer to an object is strongly correlated with typicality of usage even when diverging from the basic-level category. Discrepancies between basic-level and entry-level categories particularly include atypical exemplars. For example, ***penguin*** is an entry-level category while a more inclusive category, ***bird***, is the corresponding basic-level category. The work by Tanaka et al. (1991) also suggests even typical exemplars may present similar discrepancies for individuals with an expert level of knowledge.

Ordonez et al. (2013) work to identify entry-level categories present an image. Though these are not identical to basic-level categories, these two are often correlated. They use the SBU Captioned Photo Dataset (Ordonez et al. 2011), a large corpus of image caption data, as well as ImageNet (Deng et al. 2009), a large-scale database of images depicting objects aligned to WordNet synsets widely used in computer vision for object recognition tasks. They use frequently-used words in image captions as well as

frequency information from the web generally as signals for identifying entry-level categories. The most relevant portion of this work is a subsystem which maps leaf nodes in ImageNet to entry-level categories by optimizing a function that prefers words with high frequency on the web offset by a cost proportional to the height in the hierarchy above the leaf node. They also consider another approach using information retrieval ranking on image captions. They find their best system achieves an f-score of 0.161 overall at identifying the entry-level category on a broadly representative dataset of consensus annotations obtained using Amazon Mechanical Turk. They also find that their mapping subsystem that most closely relates to our task achieves 37% agreement with human-supplied annotations, indicating that this is a difficult problem.

The way the problem is framed by Ordonez et al. (2013) in identifying entry-level categories allows a strong dependence on frequency information, though we also find this to be a strong signal in identifying basic-level categories as well.

2.7 Amazon Mechanical Turk for Crowd-Sourcing Labels

Amazon Mechanical Turk is a platform supporting two sets of customers: requestors post tasks to be completed and make payments, while workers complete these tasks and receive payments. This platform has become a widely-used tool for data collection which is both economical and attracts workers who pay at least as much attention to directions as workers obtained through more traditional means (Paolacci et al. 2010). The data produced has also been shown to have a quality level at least as high as traditional alternatives (Buhrmester et al. 2011).

Nonetheless, quality concerns lead to the practice of having multiple workers complete the same task with agreement rate and other approaches used to measure and maintain high-quality output (Ipeirotis et al. 2010). Using qualification tests to ensure workers understand the task's directions and are able to apply them correctly has proven effective at improving the quality of the labels obtained from Mechanical Turk (Rashtchian et al. 2010).

Mechanical Turk is now widely used for generating labels, and can produce labels of sufficient quality to replace expert annotators across a wide variety of tasks (Snow et al. 2008, Alonso et al. 2009).

We use Mechanical Turk to generate our labels, using both agreement by multiple workers per task and qualification tests to improve label quality.

2.8 Measuring Text Readability

In order to demonstrate the practical applications enabled by our work to identify basic-level categories, one area of application we consider is using basic-level categories as a signal to help measure the readability of a text. Automatically determining the reading level of a text is an active area of research. Recent examples include novel classification-based approaches (Dalvean et al. 2018) as well as those using unsupervised word embeddings (Cha et al. 2017).

Despite more advanced recent systems, simple heuristic formulas have nonetheless proven useful and are widely used today. The Flesh Reading Ease Test (Flesch 1979) and Flesch-Kinkaid Grade Level (Kincaid et al. 1975), for example, are formulas only requiring the average number of syllables per word and average number of words per sentence to assign a reading level corresponding to grade levels in school. This is the most widely cited and used readability formula (Colmer 2018), even being written into law in Florida to force insurance companies to write more readable policies (Legislature 2003).

In more complex systems, some researchers have attempted to incorporate high-frequency word lists or words likely to be known by students of a certain age in order to build vocabulary-based readability into the system as a feature. The Dale 3000 list of words familiar to most fourth-graders (Chall et al. 1995) is a widely-used list for this purpose (Collins-Thompson 2014). Since we suggest essentially using a list of basic-level categories could be helpful in assessing text readability, we directly compare our list to this one.

With the more sophisticated approaches, research into the readability of medical texts, using domain-specific datasets, has found that more concrete words tend to be more common in texts that are easier to read and can be used to improve text readability systems (Tanaka et al. 2013). Since basic-level categories specifically identify words used most commonly to describe concrete objects (Rosch et al. 1976), this provides additional motivation for using basic-level categories to assess text readability and extending this benefit to systems not focused specifically on the medical domain.

We mention in §2.5 that Lin et al. (2009) create a simple heuristic for identifying basic-level categories along the way to measuring text readability and found this helpful in their system. However, the basic-level category identification was not the focus of this work and so our purpose here is to show that our basic-level category data, with more focus on the identification of these basic-level categories, can also be of help with this problem.

Chapter 3

Annotation

We describe a process for annotating basic-level categories in WordNet. We start by describing the initial labels from the psychology literature we start with in §3.1, discuss aligning those labels with WordNet senses in §3.2, explain how we used crowd-sourced annotations to extend the labels in §3.3, discuss a novel label option in §3.4, describe the labeling process in §3.5, discuss a review of low-confidence annotations in §3.6, describe the resulting labels in §3.7, investigate the accuracy of our labels in §3.8, break the labels into sets for experimentation in §3.9, measure the difficulty of our labeling task in §3.10, and summarize what we learn from this process in §3.11.

3.1 Rosch-Markman Labels

We are aware of two major lists of basic-level categories as well as corresponding superordinates and some subordinates, which we describe as the Rosch-Markman labels.

The original experiments that inspire much of the work in this area (Rosch et al. 1976) include nine superordinate taxonomies for their first two experiments. For the three of these superordinates falling in the biological taxonomy, the experimental results show the presumed superordinate level (***tree***, ***fish***, ***bird***) is actually the basic-level. So, for these three groups the taxonomy is shifted down one level (e.g. superordinate to basic) and new superordinates (***plant***, ***animal***, ***animal***) are added to ensure the experimental results are accounted for. Additionally, eight additional basic-level categories are used in their later experiments 3-4 (Rosch et al. 1976), so these are also added. Markman et al. (1997) also provide a large list of superordinates, basic-level categories, and subordinates, though there is overlap with the aforementioned list. Note that Rosch et al. (1976) individually validate each of the categories included in their lists, while Markman et al. (1997) propose a larger set of possible basic-level categories and then show they tend to have basic-level properties in aggregate. There are some conflicts, in which case we take the labels from Rosch et al. (1976).

Some of the categories chosen (e.g. ***chicken pox***, ***TV show***) seem questionable in Markman et al. (1997), so we treat those by Rosch et al. (1976) to be of higher quality and do not substantially base our labeling process on Markman et al. (1997). However, since we also build a heuristic system using the Rosch-Markman labels, and the quantity of data in Rosch et al. (1976) alone is very small while the vast majority Markman et al. (1997) seem reasonable, we do use these labels for that purpose and hence they are also described here.

A summary of the lists is shown in Table 1. The full lists themselves are presented in Appendix A. The labels from experiments in Rosch et al. (1976) are included in Table 39 and Table 40, the labels from experiments in Markman et al. (1997) are included in Table 41, Table 42, and Table 43, and the unique combination is listed in Table 44.

Table 1: Categories with known classification by level of abstraction

Level	Rosch	Markman	Combined
Superordinate	8	24	**24**
Basic-level	29	86	**91**
Subordinate	45	25	**68**

This data in Table 1 is used for training and evaluating a heuristic system, which we also update with the labels output from our broader labeling effort.

3.2 Aligning Rosch-Markman Labels to WordNet Senses

Starting with the labels from the psychology literature described in §3.1, we map these to synsets in WordNet. Fortunately, given the fact that the labels are aligned to superordinate, basic-level, and subordinate categories, this context is sufficient to resolve all but a couple sense ambiguities.

Some categories with labels do not have an equivalent concept in WordNet. For example, while ***table*** maps to *table.n.01* and ***kitchen table*** maps to *kitchen_table.n.01*, we do not find an equivalent concept for ***dining room table***. In these cases we keep the labels that align to WordNet senses and discard the ones with no equivalent in WordNet.

Another challenge in mapping the Rosch-Markman labels to WordNet senses involves the hierarchy being structured differently in WordNet than the one proposed by Markman et al. (1997). Note this conflict

does not arise for any of the examples in Rosch et al. (1976). As an example, Markman et al. (1997) consider ***weights***, ***bicycle***, and ***pool*** to be basic-level categories under the superordinate ***exercise equipment***. In WordNet, however, *weight.n.02* falls under *sports_equipment.n.01*, *exercise_bike.n.01* falls under *exercise_bike.n.01*, and *swimming_pool.n.01* falls under *athletic_facility.n.01*. In cases like this, where each of the basic-level categories fall under a different superordinate and these are far enough apart in the hierarchy that there is not a nearby common ancestor of roughly similar generality to other superordinates in the list, these are also discarded.

However, in cases where the superordinates are not so spread out, with no more than two superordinates and multiple categories remaining under a superordinate, the labels are all retained. This includes *ball.n.01* and *football.n.02* under *game_equipment.n.01* as well as *racket.n.04* and *net.n.05* under *sports_implement.n.01*; all four of these basic-level categories are listed under ***sports equipment*** in Markman et al. (1997).

The combined labels, aligned to WordNet synsets, are listed in Table 45 of Appendix B. A summary of the number of labels is shown in Table 2.

Table 2: Label counts by WordNet alignment and level of abstraction

Level	Before Alignment	WordNet-Aligned
Superordinate	24	**26**
Basic-level	100	**79**
Subordinate	68	**50**

Aligning to WordNet senses increases the number of superordinates since the taxonomies do not match exactly and some basic-level categories are split across multiple superordinates in WordNet; while several are discarded, the total count does increase by 8%. Basic-level categories decrease by 21% and subordinates by 26%. Many of the basic-level categories not included are due to superordinate alignment issues, discussed supra. Many of the subordinates that fail to align, however, are due to the concept itself not being present in WordNet rather than these superordinate alignment issues; the labels most affected by these superordinate alignment issues are in an experiment that included superordinates and basic-level categories but not subordinates.

An additional issue arose during alignment which is of theoretical interest but does not affect the number of labels successfully aligned. WordNet often includes many intermediate concepts between a basic-level category and what the labels would indicate are its superordinate or subordinates. As an example, consider the partial chain up the hypernym/hyponym hierarchy shown in Listing 1. This includes three intermediate categories between ***animal*** and ***fish*** as well as another four between ***fish*** and ***salmon***. In Rosch et al. (1976), however, ***animal*** is listed as the superordinate of ***fish*** which is the basic-level category under which ***salmon*** is a subordinate. Less salient categories, like ***aquatic vertebrate*** and ***teleost fish***, are not included in the psychology experiments. In most cases the subordinates are directly nested under the basic-level categories in the WordNet hierarchy, but there is more commonly an intermediate or several between the basic-level category and its superordinate. The example in Listing 1 is a relatively extreme example of this issue with several intermediates on both sides of the basic-level category; this is relatively common in the biology taxonomy portion of WordNet.

Listing 1: Abstruse categories between salient ones

animal

chordate

vertebrate

aquatic vertebrate

fish

bony fish

teleost fish

soft-finned fish

salmonid

salmon

This issue is interesting theoretically since the salient contrasts are not directly connected by edges in the WordNet hierarchy. This also poses an interesting set of challenges for labeling since the psychology experiments depend on having a comparison between easily-comprehensible categories with large increases in inclusiveness between levels. In this framework, however, the distinctions are more granular and the salient comparisons are spread out and non-obvious.

3.3 Crowdsourcing Labels

We use Amazon Mechanical Turk to obtain crowd-sourced labels for basic-level categories. We ask the crowd-source workers to select a single category as the basic-level category in a chain up the WordNet hypernym hierarchy.

We do not obtain labels for superordinates or subordinates explicitly, but rather treat this as a binary classification problem to identify the basic-level categories against all others. When one category in a chain is chosen as the basic-level category in that chain, the other categories can be inferred as not belonging to the basic level.

An example of the prompt we use for obtaining a label on a chain is shown in Figure 1.

The goal of this task is to pick the "basic" word from a list of related words. A "basic" word is one where:

- A young child can easily learn the word early on, before learning other related words (assuming the child is exposed to a number of examples)
- When shown a picture of a very specific thing, it's the word you'd commonly refer to it as (e.g. seeing a specific dining room chair, you'd typically refer to it as a chair)
- It is *often* a short, simple word. Unfortunately there are some exceptions to this (e.g. television).
- It's the broadest term to describe a thing that you can still picture as a single thing representing everything that word refers to (e.g. chair but not furniture, whistle but not device)

You'll be presented with a list of words, from broadest to most specific. Please choose the "basic" word, if there is one. In some cases, there may not be (e.g. the list "function", "polynomial", "quadratic"). In these cases, choose "None".

Ask yourself: ***"Which word do you think a young child would learn first? (assume exposure to all these things)"***. That is not perfect--note some lists don't have a "basic" word in them!--but it is often correct so it's a good place to start. If none seem like a young child would learn them, even given exposure to some examples, that's another clue it could be "None"

Here is a simple example with the correct answer in **bold**:

- organism *(a living thing that has (or can develop) the ability to act or function independently)*
- animal *(a living organism characterized by voluntary movement)*
- **bird** *(warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings)*
- thrush *(songbirds characteristically having brownish upper plumage with a spotted breast)*
- robin *(small Old World songbird with a reddish breast)*
- None *(none of these answers may be pictured or learned by a young child)*

For this list, **bird** is the right answer.

Basic Word

Choose the basic word in the list below, or "None" if there aren't any.

- organism *(a living thing that has (or can develop) the ability to act or function independently)*
- animal *(a living organism characterized by voluntary movement)*
- chordate *(any animal of the phylum Chordata having a notochord or spinal column)*
- vertebrate *(animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)*
- aquatic vertebrate *(animal living wholly or chiefly in or on water)*
- fish *(any of various mostly cold-blooded aquatic vertebrates usually having scales and breathing through gills)*
- bony fish *(any fish of the class Osteichthyes)*
- teleost fish *(a bony fish of the subclass Teleostei)*
- soft-finned fish *(any fish of the superorder Malacopterygii)*
- salmonid *(soft-finned fishes of cold and temperate waters)*
- salmon *(any of various large food and game fishes of northern waters; usually migrate from salt to fresh water to spawn)*
- chinook *(large Pacific salmon valued as food; adults die after spawning)*
- None *(none of these answers may be pictured or learned by a young child)*

Figure 1: Example Mechanical Turk Question

This is a wordy prompt. In addition to including the question itself, there are a number of additional aspects. The prompt includes a description of basic-level categories as well as an example before the question. The full hierarchy is not shown, starting at ***organism*** rather than ***entity***. A *'None'* option is included in the list as well, discussed in §3.4. We describe each component of the prompt as well as the motivations behind them in §3.3.1 through §3.3.3.

3.3.1 Building and Testing the Prompt

We build the prompt by initially testing, in person, a series of questions about five basic-level categories on several non-expert adults who were unfamiliar with basic-level categories, WordNet, and natural language processing more generally. We initially just use simple, diverse examples from our label set described in §3.2 and listed in Appendix B. Rosch et al. (1976) and Markman et al. (1997). This initial set includes ***piano***, ***milk***, ***gun***, ***grape***, and ***tree*** as the basic-level categories included. We randomly choose one chain including a subordinate also in the label set for each of these five categories and use the same chain for each person we interview.

We iterate based on which questions lead to answers that align with the gold-standard labels, as well as based on feedback from observing the participants and getting their verbal feedback on the process and their interpretation of both the questions and their responses. We then switch to Mechanical Turk and, due to the ease of scaling the labeling there, increase the number of basic-level categories we request labels on from five to sixteen, obtaining ten labels per chain including these basic-level categories. We iterate further on the prompt based on the responses to these prompts, finally achieving 100% accuracy on aggregated answers per chain labeled based on simple voting; this also has a very high accuracy of 98% on the individual responses.

After obtaining high-accuracy results on simple cases, we choose four examples that are much more complicated to further refine the prompt and help ensure our labeling process scales outside the simple, salient categories used in the psychology experiments used to motivate our existing label set. We include three chains where '*None'* is the correct answer and one where there is a basic-level category (***trousers***) but it is a narrower distinction between nearby categories than the more canonical cases. We iterate on the prompt until we obtain a 75% accuracy on aggregated answers using simple voting on these four

tricky cases while maintaining the same accuracy of 100% with aggregated labels and 98% on individual labels in the original, more canonical cases, where we do not observe any improvements (or regressions) from these prompt changes. This is the final prompt we use as shown in Figure 1.

3.3.2 Choosing the Prompt

The prompt we use for our labeling efforts is a question directly asking a labeler to pick the basic-level category. This is a straightforward question but requires the labeler to understand, at least at some level, what a basic-level category is.

Our direct prompt right above the set of choices is: “Choose the basic word in the list below, or ‘None’ if there aren’t any.” This is under a large heading: “Basic Word”. In both of these cases, we use “basic word” rather than “basic-level category”, “basic category”, “basic-level concept”, “first-level concept”, or other descriptions which would be more technically accurate. Labelers appear confused by these terms so simply calling them basic words is simpler to understand. This language is less precise and may account for some of the mistakes we later observe as discussed further in §3.11, but using more technical language was among the worst alternatives we tried.

We explore asking more basic questions about whether or not a given synset from WordNet had various of the properties of basic-level categories which may be accessible to non-experts. Questions include identifying the easiest word to learn, choosing the word a child would be expected to learn first, choosing the first in the list the labeler could picture (with synsets listed from more to less inclusive), gauging the complexity of the word on a scale, and identifying which word is the simplest to understand. The two most promising questions about properties include:

- Starting from the top of the list, which is the first where a clear picture comes to your mind?
- Which word do you think a child would learn first? (assume exposure to all these things)

The latter, which involves speculation about which word a child would learn first, is the most promising but both were imperfect. The latter tends to result in some over-broad answers (e.g. ***fruit*** instead of ***grape***), while the former tends to generate answers corresponding to less-inclusive categories than the basic-level category in the chain (e.g. ***pine*** instead of ***tree***). Placing multiple answers side-by-side leads many

labelers to selecting the same option for both questions though these tend to generate somewhat different answers separately. We believe we could have achieved around 80-90% accuracy on the simple concepts tested using a combination of these two questions asked separately, but this would have required additional reconciliation between the different answers and a more complicated process.

The challenge with our solution, which involves directly asking users the broader question we are trying to answer, becomes needing to educate the user about what the question means since it's less obvious. We use several means for this education. Most importantly, we provide this explanation at the top of every question rather than only in a separate instruction reference. In addition to this, we use a qualification test to limit participation to labelers who demonstrate the ability to perform well at this task (discussed in §3.3.3) and we provide a more detailed set of instructions for labelers to read when preparing for the test or for later reference (see Appendix C for these slightly more detailed instructions).

The explanation we provide includes several key pieces of information: the meaning of a basic-level category, awareness that "None" is a valid answer, a simpler question to help focus the labeler, and a full example with the answer highlighted.

We describe the meaning of a basic-level category by listing four of the properties associated with basic-level categories. There are many properties, but those chosen are ones we believe to be easier to interpret and apply than the others. As an example comparison between an included property and an excluded property, we believe it should be more straightforward for a labeler to think about whether he can picture a category rather than what his reaction time is in recognizing an object belonging to that category.

Another critical component in the explanation is that 'None' is a valid answer. This is of theoretical interest and discussed in more detail in §3.4. Of relevance here, however, is that we provide not only the mention that this is an option but also an example of an abstract chain of hyponyms (***function***, ***polynomial***, ***quadratic***) where none of the available options would be considered basic-level categories.

We next provide, in bold italics, a key question: "Which word do you think a young child would learn first? (assume exposure to all these things)". This corresponds to the first property of basic-level categories

listed earlier in the description, and to the narrow property-focused question that performs best. This enables us to ask the direct question about which is a basic-level category while also guiding the labeler to a useful question for focusing the labeler's attention on a more concrete issue that leads to good results in the event the direct question is difficult to interpret. We find this works better than either alternative alone.

Next we show a complete example of a question like the one the labeler will receive, with the correct answer in bold. In addition to making this more complete, we believe this provides a simple, concrete reminder not to choose an option that is too inclusive (e.g. ***animal***) or too narrow (e.g. ***robin***), but instead to choose the option that is at the basic-level (***bird***).

This explanation is above the direct question asked (discussed supra) each time a label is requested.

3.3.3 Presenting the Options

The options shown in a labeling question involve listing each of the optional categories for the labeler to choose between. They are listed from the most inclusive category at the top of the list to the most narrow at the bottom of the list.

In cases where there are large number of options, given that basic-level categories never appear to occur near the top of the hierarchy, we remove up to five of the most inclusive categories in the list, starting from the top. Since some leaves in the hierarchy are relatively close to the top (albeit generally not including any basic-level categories), we do not remove options that would lead the resulting list to contain less than five options, excluding 'None'.

There are two motivations for removing options near the top. First, while even leaf nodes in the hierarchy are occasionally basic-level, in the gold-standard labels basic-level categories generally do not appear in the top seven levels of the hierarchy, making the removal of no more than five levels relatively low-risk while also making the task easier for labelers.

Second, in our early in-person testing of the questions we would ask, we heard some feedback that some of the words near the top have some of the properties of basic-level categories; in particular, some are very short words. Additionally, being very abstract, some of these words have alternate meanings that

seem like they could plausibly be learned early by a child, even if they aren't basic-level. For example, the words "whole" has the abstract definition "an assemblage of parts that is regarded as a single entity". Another concept represented by that same word, though, means "including all components without exception; being one unit or constituting the full amount or extent or duration; complete". While this latter is also not a basic-level category, it is a short word and some labelers can imagine a child learning it early (e.g. "I want the *whole* cookie"). So, to avoid making some decisions more difficult we remove some of the top of the hierarchy.

In WordNet, many synsets have multiple words associated with them. We choose to only show the first word in the synset to avoid confusion and make it easier for users to have a simple list of words to choose an individual word from, rather than requiring the labeler to compare lists of words. To help with disambiguation, however, we do provide the gloss for the term. Usually seeing the words representing a chain up the hierarchy disambiguates which sense of each word is intended based on the fact that they're all related through hypernym/hyponym relations, but this is not always true and it may not be obvious to all labelers. Since this is not usually required but can be helpful in some circumstances, adding the gloss risks complicating the task or confusing the labeler. We therefore make the font size smaller and place it in parentheses and italics so the words being compared still jump out while making glosses available in the event they are needed.

3.3.4 Using a Qualification Test

The quality of labels can vary widely in Mechanical Turk, where labelers are paid by the label. Since the purpose of Mechanical Turk is to obtain results that are difficult to automate, it can be difficult to verify the accuracy of an individual label and many get accepted even if incorrect; as a result, malicious turkers may submit many low-quality labels and still get paid for them, creating quality issues for those using Mechanical Turk for labels (Ipeirotis et al. 2010).

One effective strategy for obtaining higher-quality labels is to use qualification tests, where a labeler is required to follow the instructions and label a set of sample questions with known labels correctly before being allowed to participate in the labeling tasks (Alonso et al. 2008). We use a qualification test to increase the quality of the labels obtained. Our test asks for ten labels. Eight of these come from our gold-

standard labels and two are ones we chose which clearly have ‘None’ as the correct answer. We require a score of 90% to be eligible for participating in the labeling project. This ensures that labelers are effective at the easier cases and at least somewhat capable at the more complicated ones.

3.4 The “None” Option

In the psychology literature containing experimentally-verified basic-level categories (Rosch et al. 1976, Markman et al. 1997), basic-level categories are contrasted against more inclusive superordinates and less inclusive subordinates. In §3.2 we discuss how this simple model does not map cleanly to WordNet. The main example discussed there is how the examples used are often salient words at each level while WordNet often includes less-salient categories in between.

An additional challenge we encounter is that some chains in the WordNet hierarchy do not appear to include basic-level categories at all. Three examples are shown in Figure 2.

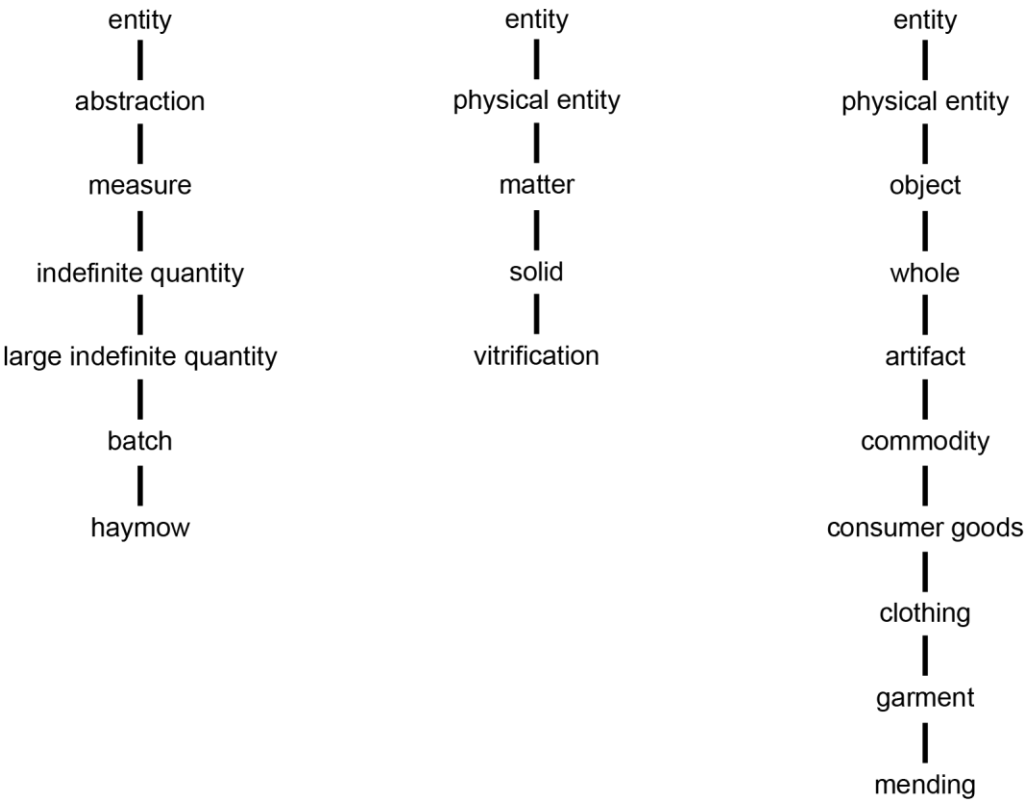

Figure 2: Example chains with no basic-level categories

These examples do not fit into the theoretical framework from the psychology literature. It's unclear whether this discrepancy is an oversight in the psychology work in this area or alternatively whether this framework is not intended to apply to all nouns and WordNet including out-of-scope nouns in its hypernym/hyponym hierarchy doesn't pose a problem for the framework.

In the epistemology literature, however, all concepts are either first-level or derived from a process starting with first-level concepts (Binswanger 2014), so this is a more interesting case that requires explanation. This theoretical framework, however, regards the relationships formed between concepts as more complicated than simple hypernym-hyponym relationships.

For example, ***mending*** ("garments that must be repaired") would fall under what this line of research calls a cross-classification (Binswanger 2014). A concept like this would get formed by starting with first-level concepts like ***shirt*** and ***pants***, generalizing from there up to more inclusive concepts like ***garment***, and then—going the other direction—subdividing this concept based on different properties than those used to form the generalization. When a concept is subdivided in this way multiple times along independent axes (i.e. cross-classified), a particular referent may fall under multiple paths in the resulting hypernym/hyponym chain.

This leads to many awkward sections in the WordNet hierarchy. One case that motivates the inclusion of this 'None' option is ***freestone***. ***Freestone*** ("fruit (especially peach) whose flesh does not adhere to the pit") is most commonly used to refer to a subset of the referents of ***peach***, but ***freestone*** is broader than that and reflects a distinction that also applies to ***plum*** and a handful of other fruit. This is illustrated in Figure 3. In WordNet, though, ***freestone*** couldn't be a hyponym of ***peach***, ***plum***, and several other fruits; the hyponym relation requires that the hyponym would need to be a subset of any hypernym category. It wouldn't be possible for the freestone plums to be a subset of ***peach***, or for the freestone peaches to be a subset of ***plum***. Instead, ***freestone*** is instead nested under a broader category ***edible fruit*** where its referents really are a subset of the parent's referents. This is illustrated in Figure 4. ***Edible fruit*** has a set of hyponyms which are not mutually exclusive like ***peach*** and ***plum*** are (or ***freestone*** and ***clingstone*** are for that matter). But it is not the goal of WordNet to capture these more subtle relationships.

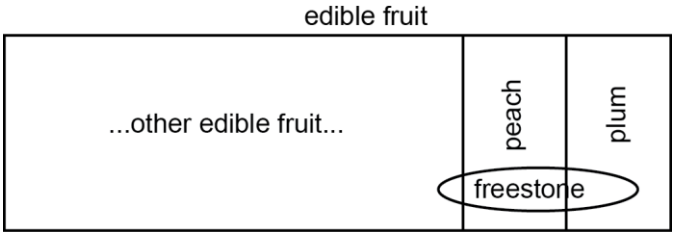

Figure 3: Edible fruit subdivisions

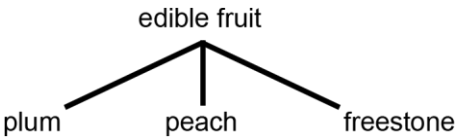

Figure 4: Edible fruit partial hierarchy in WordNet

Cross-classification is not captured explicitly by relationships in WordNet even in simpler cases like this where the categories are at or near the basic-level. This difference leads to concepts descendant from basic-level categories ending up in chains not including a basic-level category. This suggests the discrepancy between first-level concepts being regarded as the starting point for all concepts and there being some chains without any first-level concepts may be a result of a mismatch between the goals of WordNet and the goals of this work. While WordNet is the best available resource we are aware of to build on in identifying basic-level categories, we also imagine a resource that captures more relations in which these basic-level category labels would be even more well-suited and central.

We have been discussing this primarily in terms of the epistemology literature since this is the framework that appears to have the most insights to offer in this particular case. To the best of our knowledge, the psychology work in this area does not consider this issue. Since basic-level categories and first-level concepts describe the same phenomena, as discussed in §2.2, we suspect the same analysis would apply to basic-level categories as described in the psychology literature and the discrepancy is between the purposes of WordNet and our purposes in identifying basic-level categories.

3.5 The Labeling Process

We have described the prompt (§3.3.1, §3.3.2, §3.3.3,), qualification test (§3.3.4), and some of the theoretical issues (§3.4) involved in obtaining labels using Mechanical Turk. With this framework in place, we now discuss the actual process of using Mechanical Turk to obtain labels.

The chains we ask labelers to label are critical to how representative the resulting labels are, and this selection also affects the number of distinct labels we can obtain. We used three selection mechanisms for choosing new chains of hypernyms/hyponyms to label in search for new basic-level categories as shown in Figure 5.

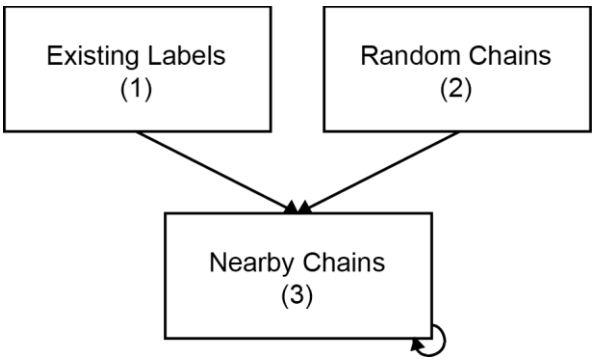

Figure 5: Process for Finding Chains to Label in Search of New Basic-Level Categories

First, we include the synsets from the aligned labels described in §3.2. Since what we label is a chain, not a synset, we randomly choose 3 distinct chains passing through each selected synset and label each. Second, we added to this a random collection of chains across WordNet to expand the scope of labels well beyond the salient examples selected by Rosch et al. (1976) and Markman et al. (1997).

Additionally, as a third (but iterative) source of chains to label, we expand outward from labeled chains with a basic-level category to find other nearby chains that may also contain a basic-level category. This is where most of our labels come from, though since this is an iterative process assuming an existing set of labels exists to expand outward from, the two aforementioned sources are extremely influential in determining which portions of the WordNet hierarchy are labeled.

An example of this third source of labels is shown in Figure 5, which shows a fragment of a tree structure like the WordNet hierarchy and how this works in a simple case. In Figure 5(a), a fragment of the hierarchy is shown with one node labeled as a basic-level category (noted in black). We start by expanding to the nearest neighbors on the same level by going up one level in the hierarchy and down one level to find all the siblings not yet labeled; these are highlighted in Figure 5(b). Chains going through these nodes are then labeled; since this is a fragment often somewhere in the middle of the hierarchy, each of the nodes appearing as leaf nodes actually have children in WordNet, so again we select three random chains passing through the gray node. In an ideal case, the basic-level categories are all on the same level, so we presume this case and show the new labels as black in Figure 5(c).

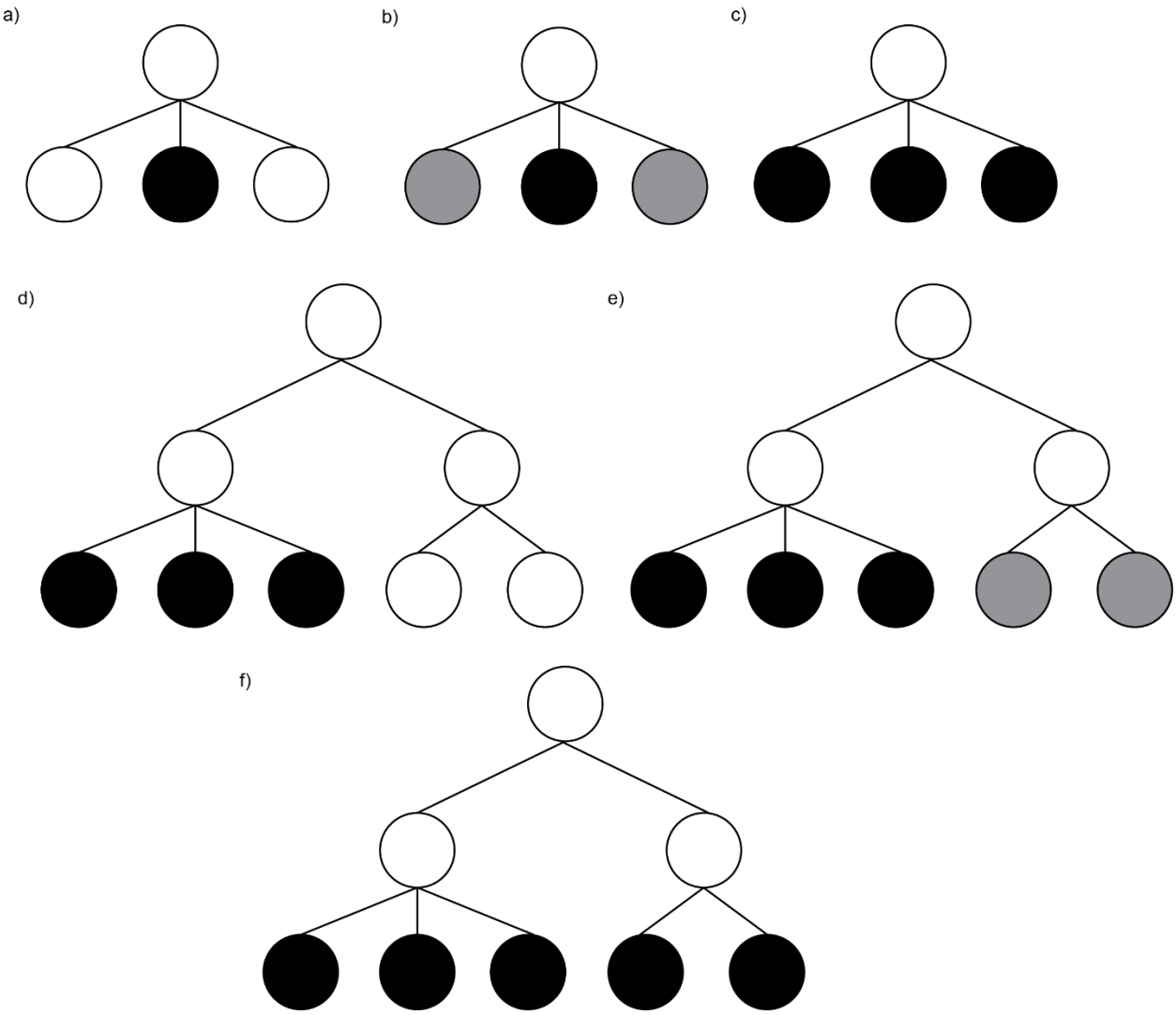

Figure 6: Expanding from Basic-Level Categories to Nearby Synsets to Label

Note that if siblings of basic-level categories are necessarily basic-level categories, this labeling process would not be needed for these nodes. So, while in this diagram we presume the proposed nodes are in fact confirmed to be basic-level categories, but in reality it could end up being a different node, such as one of the proposed node's children.

Figure 5(d) reminds us that this is a fragment of the hierarchy and shows the three labeled nodes in a broader context with more nodes remaining to label. Here, we cannot expand outward to siblings since all the siblings are labeled from the previous step. So instead, we expand to additional nearby nodes on the

same level by going up two levels in the hierarchy and then down two levels, this time finding the cousins as illustrated in gray in Figure 5(e). Again three chains through these nodes are randomly chosen to be labeled, and presuming the proposed nodes turn out to be basic-level categories the result is shown in Figure 5(f) where these nodes are now black as well.

We also expand to a third level out by going up three levels and then down to all descendants three levels below that looking for chains to label that aren't already labeled. This is not pictured for brevity. In each case, we expand to the nearest group with any remaining nodes to be labeled, only expanding another level out when all of these nodes are successfully labeled.

A key motivation for this expansion is that many siblings of basic-level categories are also basic-level categories, so it is a straightforward way to find new nodes that contain basic-level categories. Randomly selecting chains may result in many chains chosen from dense portions of the hierarchy, for example under ***person***, a basic-level category with more than 400 subordinates and many more chains passing through it. By starting at a basic-level category and carefully expanding outward, we avoid submitting many labels through the same basic-level category, reducing the labeling effort required. This does create clumps of labels rather than having them randomly distributed throughout the hierarchy, but to the extent the locations of the clumps are chosen randomly this can still be a representative sample, albeit with higher variance.

Another issue related to bias in the nodes labeled is that we do not expand from 'None' answers. When a chain does not have any basic-level categories, we have no basic-level category in the list to expand from. This could artificially limit the number of 'None' labels.

There are two distinct type of 'None' labels. In one case, like ***freestone*** as discussed in 3.4, the 'None' labels are siblings of basic-level categories and as long as we select one chain passing through a basic-level category nearby we can find these through expansion. Since we always select three distinct chains through any possible basic-level categories, this also helps reduce the likelihood of missing 'None' chains.

Another category of ‘None’ labels is those that are clumped together rather than adjacent to basic-level categories. It has been proposed, for example, that all nodes passing through ***abstraction*** (synset abstraction.n.06) may not contain any basic-level categories (discussed further in §4.2.2.1.14). These would inherently be underrepresented in our labeling process. We may select some randomly at the beginning, but since many of our labels come from expansion from these labels we do not find the nearby chains which would also be labeled ‘None’. The ratio of randomly-selected chains labeled as ‘None’ was slightly less than the ratio from expanding from basic-level categories, though, so this appears to be a relatively small problem since it didn’t drive a substantial difference in the other direction here.

We have described the process of selecting which new chains get labeled. We have also mentioned that we label three chains passing through each possible basic-level category. Whenever we label a chain, we have three independent labelers label the chain. We use voting to choose the most common label for the chain. This is sufficient for labeling the chain itself, but for chains with a basic-level category already labeled in them we can get one positive label and additional negative labels. All the nodes in the chain not labeled as basic-level are taken as negative examples.

Further, if we have three or more chains labeled through this same basic-level category, each with that basic-level category chosen as basic-level, we infer that this is a strong basic-level category label and infer that it is the basic-level category for all the chains running through that node. This enables us to infer additional negative labels; most of these are the subordinates of the basic-level category, but also since some nodes have multiple hypernyms we also have a small additional number of superordinate negative labels as well. As mentioned previously, WordNet senses are much more fine-grained and on average less salient than those in the previously-available labels, so we are using the terms subordinate and superordinate loosely in referring to all of the nodes above and below the basic-level category in the hypernym/hyponym hierarchy, respectively.

This also leads to one additional source of chains to label not mentioned previously. Rather than finding new basic-level categories, we also have a small number of chains we want labeled to get stronger confirmation that an existing basic-level category label is strong enough to extrapolate to the other chains running through that node. The process for finding new chains to label attempts to handle this

automatically by expanding outward from basic-level categories and selecting three random chains including each node at the same hierarchical level as a nearby basic-level category. If the basic-level category is at a higher level than expected (e.g. the aunt of the node being expanded from), there will be at least three labeled chains already through that node. If the nodes chosen as basic-level by the labelers are at a lower level in the hierarchy, however, more labels are likely required since there will usually only be one chain labeled through each of these new basic-level categories. So in this case, we choose a couple more chains passing through the already-labeled basic-level category to confirm the label. This is not looking for a new basic-level category, the process described previously in Figure 5, but adding this as an additional source of new chains to label helps to extrapolate better to additional negative labels.

Finally, occasionally there will be a disagreement between labelers in selecting the basic-level category on the chain. While voting works well, we request more labels on a chain in some circumstances. We request an additional seven labels for a chain when a minority of the labels is 'None' or when there is no clear winner (e.g. when there are three different answers). Since this is an indication of a potentially difficult case, getting a larger number of labels helps to make the voting more reliable by reducing the variance in the responses.

In summary, we start with both an existing set of gold-standard labels and random sampling to label, and expand that outward by choosing new chains to label that are likely to lead to new basic-level category labels. We have several labels per labeled chain and use voting to determine the winner, though in some circumstances we request additional labels on the chain to make a more informed decision. We are specifically focusing the prompt on finding the positive labels, but we extrapolate both up and down the hierarchy from basic-level categories to obtain the related negative labels.

3.6 Reviewing Targeted Labels

In some cases, since parallel chains are being labeled, conflicts appear where a category present in multiple graded chains is labeled as basic-level in one chain and not basic-level in another. We review these cases as well as cases where 'None' was a minority choice, correcting egregious errors while leaving borderline cases with their original labels. Since the number of positive labels is relatively small, we also review these. Through these reviews, we make corrections to thirty-five labels, representing 0.3%

of the total number of labels resulting from this labeling process. Note that these label corrections are after the voting process, which already filters out some individual labeling errors.

We notice three main sources of error.

The first is that a number of superordinates are chosen as basic-level. There are some common superordinates, particularly including ***fruit***, ***plant***, ***animal***, and ***food***. These cases are easy to identify since there are many other categories below these in the hierarchy that are labeled as basic-level. It would be possible to automatically identify and remediate these mistakes based on conflicts, but since it is rare it was sufficient for us to correct these cases manually. Interestingly, these examples are all relatively short, frequent words.

Another source of error is chains that should be labeled as 'None', particularly those with short, polysemous words elsewhere in the chain as well as meronyms and materials. Examples include ***plastic*** and ***handle***. There is some overlap with the first error case, such as with the 'None' chain including ***freestone*** (an example discussed in more detail in §3.4), which provides an opportunity for ***fruit*** being chosen over 'None' even if 'None' does appear as a minority response that does not win the vote.

A third source of error is basic-level categories not common in countries where English is the primary language. For example, ***niqab*** represents a piece of clothing that's very common in Saudi Arabia where Islamic women often cover their faces in public, but rare in the United States where this is less common. It would be natural to think of this as a basic-level category just like ***shirt*** or ***pants***. However, since one of the properties of basic-level categories is that they're frequent, this is not true in this case. We mostly encounter this error with Islamic clothing and some locale-specific fruits.

3.7 Basic-Level Category Labels

Our labeling process results in a total of over eleven thousand labels; we show this by label category in Table 3. The terms "Superordinate" and "Subordinate" are used loosely hereafter when discussing these labels, referring to categories above or below a basic-level category in the WordNet hypernym/hyponym hierarchy, respectively.

Table 3: Label Frequency by Subcategory

Label Category	Labels
Superordinate	126
Basic-level	258
Subordinate	10,348
None	489
Total	**11,221**

The overwhelming majority, 92% of the labels, are ‘Subordinate’ labels. WordNet is heavily weighted toward the bottom of the hierarchy, and this reflects that. The next most common category is ‘None’ at 4.4%. Only 2.3% of the labels are positive examples of basic-level categories, a label bias that could make it difficult for automated systems to identify this relatively rare class.

The WordNet synsets corresponding to the positive (basic-level) labels are listed in Table 46 in Appendix D. With a total of 82,115 noun synsets in WordNet, we have labeled 14% of all synsets. From this label set we can estimate that there are around 1,888 basic-level categories in WordNet. This supersedes our previous estimate of 1,620 (Mills et al. 2018).

3.8 Label Accuracy

As previously mentioned, our labeling process is initially seeded, in part, with categories that are labeled and experimentally verified in the psychology literature (Rosch et al. 1976, Markman et al. 1997). While we seed the list of categories to label, we do not simply accept these labels as given and expand from these (and the random set we also include). We actually label these seed categories using the standard process, which provides an opportunity to measure alignment between the two sets of labels.

This is not a perfect measure of accuracy since we have argued the Rosch-Markman labels are a biased subset of labels, and because one of the existing label sources (Rosch et al. 1976) is inherently more reliable than the other (Markman et al. 1997) given the experimental process used, as discussed in §3.1. Nonetheless, we believe it is the best estimate of labeling accuracy available, at least within the context of the way the problem is thought about in the psychology literature.

We find that 95% of the labels, where the WordNet synset being labeled was in common across the two sets, are in agreement. Interestingly, each of these disagreements is labeled as ‘None’ by our labeling process, an option which has not been discussed in experiments in this area.

Each of the disagreements is interesting, albeit in different ways. We do not believe these are all errors in the labeling process but, not having a clear and objective way of resolving this small number of disagreements, we present potential interpretations for each.

Two of the disagreements, synsets *freestone.n.01* and *cling.n.01*, are closely related and are clearly examples that fall within the ‘None’ option as we have described it; in fact, ***freestone*** is used as an example in §3.4. However, this is not as clearly a mistake in the experimentally-validated labels as it may appear. We align “freestone peach” and “cling peach” from Rosch et al. (1976) to the WordNet synsets *freestone.n.01* and *cling.n.01*. While these are easily the categories most closely aligned to one another across sets, the conceptual vocabulary that WordNet provides does not align perfectly to the conceptual vocabulary used by Rosch et al. (1976). *Freestone.n.01*, for example, is more general than “freestone peach” since it also includes “freestone plum” and others even if ***freestone*** is almost always used to refer to peaches. The error here may be in our alignment between Rosch et al. (1976) and WordNet, in the decision to treat “freestone peach” as an independent category rather than a combination of categories in Rosch et al. (1976), or in WordNet’s failure to include the more granular category “freestone peach”. Regardless, we believe that ‘None’ is the appropriate label given the synset *freestone.n.01* that appears in WordNet, that ‘Subordinate’ is the appropriate label within the options provided by Rosch et al. (1976), and that this discrepancy does not substantially affect our systems derived from these labels since both of these are negative (non-basic) category labels.

Shelter.n.01 is the disagreement most indicative of a potential mistake in our labeling process, although not with the individual response of a particular labeler on a particular hyponym chain of categories. In WordNet there are many hyponyms of *shelter.n.01* and some of these chains include basic-level categories while others do not due to the inclusion of both basic-level subordinates and cross-classifications like the previously-discussed ***freestone***. In this case, *shelter.n.01* was in a number of chains labeled that all happened to have their correct answer as ‘None’.

Underwear.n.01 is another interesting case but in our opinion it is less clear whether this is a labeling mistake on either side or just a complicated case. *Underwear.n.01*, at least as defined by WordNet, is a very broad term encompassing essentially anything worn next to the skin and under outer garments. Given the wide range of clothing included in this range, it is not entirely surprising that annotators do not choose this as a basic-level category. On the other hand, we do have *underpants.n.01* as a basic-level category. Strangely, in WordNet this is a sibling of *underwear.n.01*, sharing the common hypernym *undergarment.n.01*. There may be a reasonable explanation of this, such as it being possible that what people typically think of as standard examples of ***underwear*** could be worn over other clothing and arranging the synsets this way to avoid incorrect implications. While we generally do not quibble over individual difficult decisions made in developing such a comprehensive and challenging-to-develop resource as WordNet, this is an instance where our intuitions differ from the hierarchy in WordNet. This resulting issue could be due to a mistake in WordNet, a mistake in our mapping the word 'underwear' to *underwear.n.01* instead of *underpants.n.01*, or a mistake by our annotators (and one we did not make easier for them with examples or training).

Overall we find the agreement rate is very high and each of the disagreements is instructive in understanding the sorts of issues we face in bridging data across psychology experiments and a lexical resource like WordNet, with both process challenges and difficult decisions for annotators along the way. Fortunately, these cases are a rarity and we consider there to be strong alignment between our labels and the Rosch-Markman labels.

3.9 Canonical Sets for Experimentation

We divide the labels up into sets for experimentation, including a train, development, and test set. Since experiments may take advantage of local tree-based features, we divide the sets at a superordinate level, placing all synsets under each superordinate in the same set. An additional motivation for this division is that since many of the labels are obtained through outward expansion from known basic-level categories the actual task of extrapolating to unlabeled categories more closely resembles needing to apply learned patterns to new portions of the hierarchy.

Various trade-offs are needed due to trying to place hierarchically nearby categories together, the sets do not divide perfectly. We particularly prioritize dividing up the positive labels (basic-level categories) across the three sets, and within the basic-level categories we prioritize getting a relatively even split between train and test.

While we believe this is the best way to divide the set for experiments, this does lead to some imbalances. One imbalance in the quantity of basic-level categories per set, where the development set is somewhat impoverished relative to the more closely-balanced train and test sets. The most substantial imbalance, however, is in the count of subordinates across sets since some regions in the graph are much more dense with subordinates than others. This is particularly pronounced for the biological categories; we placed these in the train set, which makes 91% of all subordinate labels appear in the train set. However, subordinates are nonetheless the most common in each set. The breakdown is shown in Table 4.

Table 4: Label Experiment Set Sizes

Set	Superordinate	Basic-level	Subordinate	None	Total
Train	59	102	9,387	303	9,851
Development	26	61	319	88	494
Test	41	95	642	98	876
Total	126	258	10,348	489	11,221

The basic-level categories in each set are listed in Appendix E, with the train set in Table 47, the development set in Table 48, and the test set in Table 49. While not fully explaining the divisions, roughly speaking the biological categories are in the train set; clothing, eating-related, and cleaning-related objects are in the development set; and everyday objects like tools, devices, furniture, and means of transportation are in the test set.

3.10 Task Difficulty

We use crowdsourcing, ask annotators to select from a list rather than making binary judgments, and then we use multiple levels of agreement to accept a label; while this process appears to be efficient at

generating high-quality labels as discussed in §3.8, this complex process does not lend itself well to standard measures of task difficulty.

To this end, we train two individuals with linguistics backgrounds as expert annotators to determine how difficult it is to label an individual concept as basic-level or not. We use the train set, as described in §3.9, to train the annotators by showing the correct labels. We provide the annotators with the same training material used on the Mechanical Turk labeling task showing how to choose the basic-level category in a chain. We then reframe this as a binary basic-level categorization task without providing the chain in the hierarchy, instead providing a list of individual concepts without their hierarchical context, indicating whether each concept presented is basic-level or not for the annotator to learn from.

We then split the development set into two, having one half setup as a test where the annotators have the answer key to reference, and the other half being a preliminary test where annotators do not have the answer key to ensure they perform at a reasonable level before exposing them to the test set. Finally, we have the annotators grade the test set.

Since these sets are large, we use stratified sampling per label subcategory to ensure reasonable representation with around 100 categories annotated per label subcategory, excepting superordinates where only forty-one labels exist in the test set. In total we obtain 334 labels with only the subordinate label subcategory being subsampled since this is the only one with over one hundred labels in the test set. After annotation, however, we re-weight the sample to reflect the overall test set when calculating inter-annotator agreement and the corresponding term in the kappa coefficient calculation.

We report our results in Table 5, which shows that the task is relatively hard. The inter-annotator agreement itself is substantial though imperfect at 92%; when taking into account the difference across label subclasses with the kappa coefficient this drops to 0.61.

Table 5: ITA and Kappa for Basic-level Category Annotation

Metric	Value
Inter-annotator Agreement	92%
Kappa coefficient	0.61

This value for the kappa coefficient barely falls within the “substantial agreement” bucket in a scale published by Landis et al. (1977) and widely-used(Viera et al. 2005). It is barely across the boundary from moderate agreement, which extends up to 0.60. This shows that our task is relatively difficult, yet there is still a substantial degree of agreement among annotators which indicates the data should provide useful signal to learn from.

The largest source of disagreement between annotators is on cases where *None* is the gold-standard label. In each of the other label subcategories, agreement rates both between annotators and comparing annotators to gold-standard labels fall between 93% and 98%. On *None*, however, the agreement rate is only 86% between annotators and 67% to 69% when comparing annotators to the gold-standard labels. This shows a non-trivial amount of overlap in these mistakes between the annotators. We find in our Mechanical Turk-based labeling work that these are the most difficult cases and the presence of a minority of *None* labels is still an indicator of a *None* label since this is challenging and *None* labels are rare. Recall from §3.6 that we review all cases where a minority of labels are *None* for this very reason. Given the consistency of this being a problem for both labeling approaches, we believe these disagreements are an accurate reflection of the difficulty of the labeling task; this further justifies our work to reduce error in our labeling process when encountering a minority of *None* labels.

While we report this inter-annotator agreement and kappa score on a binary labeling task to better compare the difficulty of this task with other labeling tasks reported in this way, our labeling task is not a binary labeling task. Instead, we have asked annotators to choose the basic-level category among a list of choices. We would expect this to be more challenging than binary labeling since there are more options and thus more opportunities to disagree. We therefore compare the inter-annotator agreements between the expert annotators on their binary labeling task and the crowd-source annotators on the more complicated multi-option labeling task in Table 6. The inter-annotator agreement for crowdsource annotators on the multi-option labeling task is calculated as the portion of annotations which correspond to the most popular response for the respective chain being annotated, with ties for the most popular response per chain broken arbitrarily to avoid overcounting.

Table 6: ITA Comparison Between Expert and Crowdsource Annotators

Annotation Method	Inter-annotator Agreement
Expert annotators, binary labeling task	92%
Crowdsource annotators, multi-option labeling task	71%

We do find that the multi-option labeling task is more challenging, which further motivates our use of multiple layers of redundant labeling to maintain high accuracy, including multiple annotations on each chain as well as multiple chains being annotated with the same basic-level category.

3.11 Lessons Learned from the Labeling Process

The most substantial lesson learned from the labeling process is that chains up the WordNet hypernym/hyponym hierarchy do not fit perfectly into the theoretical framework described in the psychology literature. Notable differences include WordNet often having uncommon, specialized words in between basic-level categories and their superordinates or subordinates. Perhaps even more challenging, however, is that some chains do not include basic-level categories at all; this requires the addition of a 'None' option when labeling chains.

These differences highlight that the Rosch-Markman labels from psychology experiments, while certainly interesting for motivating and experimenting on the psychological consequences of basic-level categories, are not broadly representative of how basic-level categories fit into the broader English lexicon.

We also learn that the basic level subsumes vastly different numbers of subordinates in different parts of the hierarchy. In particular, basic-level categories within the domain of organisms we label tend to have substantially more subordinates than others. The many subordinates under ***person*** indicate this is not just due to an abundance of scientific terms for creatures, but at least in this case it illustrates how categories that help people make distinctions between the many sorts of phenomena they deal with on a daily basis form a sizeable portion of this large number of subordinates.

While the previously-available labels are biased, providing a broader spectrum is also harder with both theoretical and practical issues making the problem challenging.

In addition to the problem itself, we also learned that the wording of instructions and questions in Mechanical Turk is critical to obtain high-quality labels. This requires being careful to word text for a non-technical audience not familiar with the theoretical framework underlying the task. This does lead to shortcuts which may cause additional problems.

For example, we have strong evidence that using “basic word” instead of “basic-level category” or “first-level concept” leads to better answers. However, we also believe this conceivably could be causing some of the common error cases, such as people choosing simple words that represent a broad, general set of referents (e.g. ***fruit***) rather than more complicated words (e.g. ***pomegranate***) that represent a simple basic-level category. In cases where the basic-level category is also a simple word (e.g. ***apple***) the labelers are able to make the correct choice, but it seems harder for them to choose complicated words that represent simple concepts since mistakes did occur in this area. This could very well be due to our usage of “word” rather than a more accurate term like “category” or “concept” in the prompt.

We attempt to strike a balance of using simple language but also providing sufficient detail to explain the ideas, along with a qualification test to ensure the labelers have read and applied the guidelines successfully. This enables us to use simple language to remind labelers of the question they are answering. Errors are very low overall, so oversimplification in the prompt language may be an issue but this was not as substantial as the improvement using more common words that do not perfectly align to technical distinctions we could make.

We have struck a balance between clear and technically-precise language, which overall is net positive for labeling. The need for this balance, as well as the expansion of the theoretical framework underlying the labeling process, comprise the main insights we learn from the labeling process.

Chapter 4

Heuristic-based System

We build a heuristic-based system to identify basic-level categories in WordNet as a baseline to compare later machine learning-based approaches against. This system takes all WordNet noun synsets as input and outputs which of these are basic-level categories. As a heuristic-based system, it relies on rules; most of these rules are based on information contained within WordNet, and we supplement this with some external resources to provide additional context including a list of stopwords (Bird et al. 2009), the Brown, SEMCOR, and Gutenberg corpora for word frequency (Francis et al. 1964, Landes et al. 1998, MacWhinney 2000, Gutenberg 2018), the CHILDES corpus (MacWhinney 2000) for a list of words learned in early childhood, Dolch's Word List of words spoken by kindergarteners(Dolch 1948), and the CMU Pronunciation Dictionary (Weide 1998).

4.1 Background

Francis Bond taught a class and in two separate terms assigned a class project to build a system to identify basic-level categories in WordNet (Mills et al. 2018). This resulted in 29 student projects each independently trying to solve this problem. We catalog the types of approaches and rules considered, taking these as inspiration for our system and combining a slightly-constrained set of these, as well as novel rules, into a combined system.

4.2 System Description

While the goal is to produce one system by evaluating the collective set of rules, some boundaries are needed to constrain this. For example, one student only considered words also appearing in the 'adventure' category of the Brown corpus (Francis et al. 1964), a small, categorized corpus of English, which restricts the project beyond the goals of this work. We therefore start with a general approach common to most solutions (§4.2.1) and describe the relevant rules (§4.2.2)

4.2.1 General Approach

We start with all noun synsets in WordNet as input to our system. We restrict ourselves to nouns because the available labels discussed in §3.1 and aligned to WordNet in §3.2 are all nouns, as well as to make the problem more tractable, though it is worth noting some research has indicated it is likely possible to extend the basic-level to other parts of speech (Lemaitre et al. 2013).

Taking the synsets in WordNet with labels, the goal becomes to extrapolate from these labels to other WordNet synsets that are also at the basic-level and not at the superordinate or subordinate levels. In the psychology experiments (Rosch et al. 1976, Markman et al. 1997) this is done with words whose senses are disambiguated by context, so we operate at the sense level. For our purposes, category and synset will be used interchangeably.

The student projects mentioned in §**Error! Reference source not found.** identify words, not synsets, t hough each student tries to map words to synsets to use WordNet features before producing a final list of words from there, losing the synset distinctions. For this work, we treat the basic-level as operating at the sense level and ensure our labels for training and evaluation are on WordNet synsets to remove this unnecessary complexity.

Essentially everything the students do to identify basic-level categories can be generalized as one of two approaches:

1. filtering out nouns that are not basic-level or
2. on a particular path from the root to a leaf node in the hypernym/hyponym hierarchy, score each node and choose the optimal one as the basic-level on that path

We adopt both of these approaches, first applying a set of Filtering Rules to remove synsets unlikely to be basic-level and then choosing at most one per path based on a set of Voting Rules. The system diagram is shown in Figure 7.

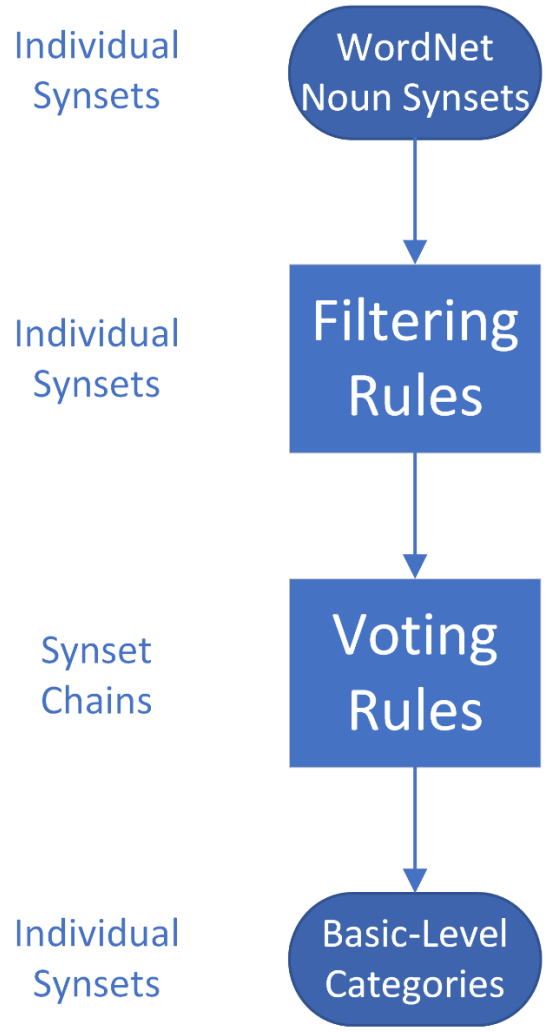

Figure 7: Heuristic System Architecture Diagram

Students consider a few other extensions, such as taking the top 2000 results with a provided sorting function, but since we do not want to assume a particular number of basic-level categories or otherwise restrict our system, we do not incorporate these approaches. Many students also dedupe their final list, deal with lemmatization, and other issues that are not necessary when operating at the synset level and thus are omitted here.

4.2.2 Rules

We catalog the rules students use and add four of our own novel rules. In addition to adding novel rules, we generalize and parameterize student rules where possible to enable experimenting with different

thresholds. Some rules are dependent on having a word, while we have synsets as our labels and these may have multiple words (lemmas) associated with them. We typically consider two possibilities for rules when a word is required to apply the rule: filtering the synset if any of the lemmas associated with the synset triggers the rule, or taking the first lemma associated with the synset (typically the most frequently used) and only applying the rule to that lemma. Students choose the latter, which is often best, but we consider both possibilities.

4.2.2.1 Filtering Rules

We use a variety of filtering rules, which filter out synsets from consideration as basic-level categories. This is a first pass that is intended to remove many synsets, with the Voting Rules described in §4.2.2.2 used for resolving cases where multiple synsets have not been filtered out in the same chain up the hierarchy.

We show a summary of the filtering rules in Table 7. Parameter ranges used by students, or examples in cases where there are long lists of parameters, are shown after the rule. Ranges are given in interval notation to avoid boundary condition ambiguity. We then describe these rules in more detail. Our novel rules are Rules 36-39, all four Filtering Rules.

Table 7: Filtering Rules Used to Filter Non-basic Categories

Filtering Rules
1. Filter words with a set of suffixes (-ing, -ment, … [59 total])
2. Filter words with a set of prefixes (un-, th-)
3. Filter words of length n or greater [7, 16]
4. Filter words of length n or fewer [1, 4]
5. Filter space-separated compound words
6. Filter hyphenated words ('-')
7. Filter joined compounds (e.g. ‘racetrack’)
8. Filter words with numbers
9. Filter words with symbols
10. Filter words with more adjective than noun senses
11. Filter words with more adverb than noun senses
12. Filter words with over 1 more verb than noun sense
13. Filter words that are not substrings in immediate subordinate nodes
14. Filter words containing any word at a higher level
15. Filter stopwords
16. Filter plural words
17. Filter words with no vowels

18. Filter words with over n vowels [1]
19. Filter capitalized words
20. Filter synsets with average depth ((min+max)/2, recursive) outside the range a to b [4.2, 9)
21. Filter synsets with average height ((min+max)/2, recursive) outside the range a to b [1.1, 2.2)
22. Filter synsets with avg_depth/(avg_depth+avg_height) outside the range a to b [.74, .91]
23. Filter the top n levels of the hierarchy [2-7]
24. Filter nodes with n levels below them (5)
25. Filter synsets with an average depth ((max+min)/2) outside the range a to b [0, 5.4)
26. Filter the bottom n levels of the hierarchy [1, 3]
27. Filter synsets n or more levels deep [9, 15]
28. Filter siblings of synsets with 0 hyponyms
29. Filter nouns with a to b hyponyms [0,2], [5,inf)
30. Filter synsets in the Brown corpus with frequency < n (1-10)
31. Filter synsets in the Brown corpus with frequency > n (40)
32. Filter all synsets under abstraction.n.06
33. Filter all synsets except those under set S (combinations of physical_entity.n.01, thing.n.08, substance.n.01, process.n.01)
34. Filter all words in the CHILDES corpus
35. Filter words in the CMU Pronouncing Dictionary with > *n* phonemes
36. Filter all synsets with *n* or more siblings having no hyponyms
37. Filter all synsets with at least *p* percent of siblings having no hyponyms
38. Filter synsets with less than *n* siblings
39. Filter words not in the Childes corpus

4.2.2.1.1 Filtering Prefixes and Suffixes

Rules 1-2 involve filtering words with a particular beginning (in the case of prefixes) or a particular ending (in the case of suffixes), motivated by the idea that basic-level categories should be simple and affixes often indicate a more complicated, derived category. The parameters to be set for each of these rules include:

- the set of affixes to filter
- the minimum length of a lemma to consider filtering it based on finding an affix
- whether the rule is applied if it matches any lemma or only the first lemma in the synset

For the set of affixes to filter, we only consider those suggested by students. The 59 total suffixes suggested are shown in Table 9. The only prefixes suggested by students are ‘un-‘ and ‘th-‘, so those are the only two we consider.

Table 8: Prefixes Considered in Filtering Rule 1 to Filter Categories with a Prefix

-ability	-cess	-est	-ize	-sis	-tics
-acy	-d	-ful	-le	-sity	-tion
-age	-dity	-hood	-less	-sm	-tion
-al	-dle	-ian	-logy	-some	-tive
-ance	-dosis	-ic	-ment	-ssary	-tor
-ate	-e	-ing	-ness	-ssory	-tory
-bent	-ed	-ism	-re	-tal	-ture
-bess	-ence	-ist	-s	-tant	-ty
-c	-er	-ity	-ship	-th	-y
-ce	-ess	-ive	-sion	-tic	

4.2.2.1.2 Filter Words by Length

Rules 3-4 involve filtering out long and short words, respectively. Basic-level categories tend to be relatively short words. The parameters to be set for these rules include:

- the boundary length (minimum length when filtering long words, maximum length when filtering short words)
- which lemma(s) the rule is applied to:
    - all lemmas associated with the synset (if any matches, the synset is filtered)
    - the first lemma in the synset
    - the shortest lemma in the synset

4.2.2.1.3 Filter Compound Words

Rules 5-7 filter out compound words. For Rule 5, multi-word expressions (words with at least one space in them) are filtered out. Note that WordNet uses an underscore ('_') to represent a space, so while the semantics involves filtering out words with spaces the technical solution actually involves filtering underscores. Rule 6 filters out hyphenated words.

Rules 5 and 6 have only one parameter:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

Rule 7 filters out words that are joined compounds, like ‘racetrack’, which is a compound of ‘race’ and ‘track’. Since we are using WordNet for this task, we assume the components are also lemmas in WordNet. We look for any combination of other noun lemmas in WordNet The purpose of a parameter of only considering the first lemma in a synset is to apply the rule to the commonly-used form of the synset rather than a rarely-used alternative; in this case, for finding compounds we allow any lemma (not just the first) of any noun synset in WordNet to be a component in a compound. The parameters for Rule 7 include:

- whether the rule is applied if any lemma in the synset is a compound or only if the first lemma in the synset is a compound (in both cases, as noted supra, the components may include non-first lemmas of other synsets)
- the minimum length of a lemma to be considered as a possible component of the compound

This last parameter for Rule 7 is intended to avoid spurious compounds with several small words combining together to form larger words that are unrelated (e.g. ‘are’ and ‘a’ forming ‘area’).

4.2.2.1.4 Filter Numbers and Symbols

Rules 8 and 9 filter out words with numbers and symbols, respectively. Words with numbers are those that contain any 0-9 digit.

Words with symbols in Rule 9 are defined using a negative definition: words with characters that aren’t alpha-numeric, a space, or an underscore (which is how WordNet encodes spaces). Rather than using a known list of symbols, we instead use this negative definition to avoid missing any. For all practical purposes, however, given that WordNet is a hand-curated lexicon developed in the English language, we do not expect any substantial difference between this and using a long list of symbols with a positive definition. Note that hyphen (‘-‘) here would be considered a symbol, so Rule 9 also filters out everything filtered by Rule 6, all else being equal (i.e. considering the same lemma(s) with respect to the synset being filtered).

The only parameter for Rules 8-9 is:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

4.2.2.1.5 Cross-Part-of-Speech Sense Counts

Rules 10-12 involve counting the number of senses of a word as a noun as compared to one other part of speech. Rules 10 and 11 filter out a word if the number of adjective or adverb senses of the word, respectively, is greater than the number of noun senses of the word. Rule 12 is slightly more complicated, requiring the number of verb senses of a word to be over one more than the number of noun senses of the word in order to filter it out.

In each case, we find the number of senses of the word by counting the number of synsets including the word as a lemma in the synset.

The only parameter for Rules 10-12 is:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

This parameter only applies to the word used for filtering. For example, if only the first lemma in the synset is used, we check how many senses of that lemma exist both as a noun and the other respective part of speech relevant to the specific rule being considered. If the word appears as a non-first lemma in another synset, it will still be included in the sense count. On the other hand, if we consider all lemmas we repeat this exact same procedure for each lemma in the synset being considered for filtration, filtering it out if the condition is met for any of these lemmas, and again including in the sense count even synsets where the word appears as a non-first lemma.

4.2.2.1.6 Substrings in Hypernym/Hyponym Chain

Rules 13 and 14 involve looking at other words associated with synsets in the same hypernym/hyponym chain and using substring matches as a criterion for filtering. Both rules are based on the intuition that subordinates often include the basic-level category above them in the hierarchy as part of their name; for example, ***ball-peen hammer*** includes its corresponding basic-level category, ***hammer***.

These two rules are two of only three Filtering Rules which do not have any parameters to be set. We do not look at whether non-first lemmas in a synset appear elsewhere in the hypernym/hyponym hierarchy.

However, while we do focus on the first lemma as the word to find elsewhere in the hierarchy, we do check whether it appears in non-first lemmas of the other synsets in the hierarchy.

Rule 13 works by assuming basic-level categories will tend to have at least one subordinate that follows this pattern. It filters out all words where there are no immediate subordinates (hyponyms) using that word as a substring. If none of the lemmas for any of the direct hyponyms of ***hammer*** included the word 'hammer', the synset would get filtered out by this rule.

Rule 14, on the other hand, filters out any words that contain any word higher in the hierarchy as a substring. Since ***ball-peen hammer***'s first lemma, 'ball-peen hammer', contains its ancestor 'hammer' in its name, ***ball-peen hammer*** is thus filtered out by this rule.

Rule 13 aggressively assumes this pattern will apply across all basic-level categories, while Rule 14 takes advantage of the pattern whenever it is observed at a cost of reducing the number of categories filtered out.

Comparisons for both of these rules are case-sensitive, preventing many trivial substrings (e.g. atomic element symbols like 'Ar') from appearing to be a substring of many words that are derivationally unrelated.

4.2.2.1.7 Stopwords

Rule 15 filters out all words that are stopwords. The list of stopwords used is the standard set in the NLTK package in Python (Bird et al. 2009). Comparisons are case-insensitive.

The only parameter for Rule 15 is:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

4.2.2.1.8 Plurals

Plural words are filtered in Rule 16 by filtering all words ending in the suffix '-s'.

The only parameter for Rule 16 is:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

4.2.2.1.9 Number of Vowels

Rules 17-18 filter out words based on the number of vowels they contain. By vowels, we specifically refer to the five vowels of the English language (‘a’, ‘e’, ‘i', ‘o’, and ‘u’). We only count vowels that are clearly vowels by including one of these characters in their English orthography. We do not use any morphological or phonetic rules, such as words ending in ‘-y’ or including syllabic /ɹ/ in their pronunciation.

Rule 17 filters out words that contain a specified number of vowels or fewer. Rule 18 filters out words that contain more a specified number of vowels or more.

Both rules have the following parameters:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset
- the boundary number of vowels to trigger the filter (the maximum filtered for Rule 17, or the minimum required for filtering for Rule 18)

4.2.2.1.10 Capitalized Words

Rule 19 filters out words that are capitalized. This is determined by whether or not the first letter in the word is an upper-case alphabetic character. While this is a simple rule, it may not perfectly align to the common conception of capitalized words; for example, it includes acronyms and capitalized abbreviations in addition to the standard mixed-case examples that are more canonically considered capitalized words.

The only parameter for Rule 19 is:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

4.2.2.1.11 Depth and Height

Basic-level categories tend to occur at a middle level of the hierarchy, and this was perhaps the first observation (Brown 1958) that eventually led to the later research into basic-level categories. Since this is an important factor in identifying basic-level categories, and there are many different ways to filter based on the position in a complex hierarchy, Rules 20-27 all attempt to use this insight as a filter. Since here we are filtering by location in a hierarchy, we use synsets directly and do not need to address the issue of determining the right word to use to represent a synset.

For each of these rules we have a candidate synset being considered as a possible synset to filter. We use height to refer to the distance from leaf hyponym nodes up to a candidate node in the hierarchy, and depth to refer to the distance from the root hypernym down to a candidate node in the hierarchy. How we measure distance varies by rule.

Rule 20 filters synsets with an average depth falling outside a specified target range, while Rule 21 filters synsets if their average height falls outside a specified target range. The distance measurement follows the same formula for these two rules, just in opposite directions.

We describe this height measurement in detail for average height through the use of an example. A fragment of the WordNet noun hierarchy is shown in Figure 8. We consider calculating the height for ***orange*** as we would calculate it for Rule 21. In this example, ***Jaffa orange***, ***navel orange***, and ***Valencia orange*** are all leaf hyponyms which we take to have a height of one by definition. ***Sweet orange*** is the parent of these three synsets and no others. Our distance measurement for a parent is one plus the average of the minimum and maximum height of each child; since all the children of ***sweet orange*** are of height one, the minimum and maximum are both one and this makes ***sweet orange*** have a height of two.

Sweet orange's parent is ***orange***, which has two other children, both height one synsets without hyponyms: ***temple orange*** and ***bitter orange***. The height of ***orange***, then, is one plus the average of the minimum and maximum height of its children. The minimum height is one (from ***temple orange*** and ***bitter orange***) and the maximum is two (from ***sweet orange***). So, the average is 1.5 and the height for ***orange*** is then 2.5. These heights are all annotated in Figure 8.

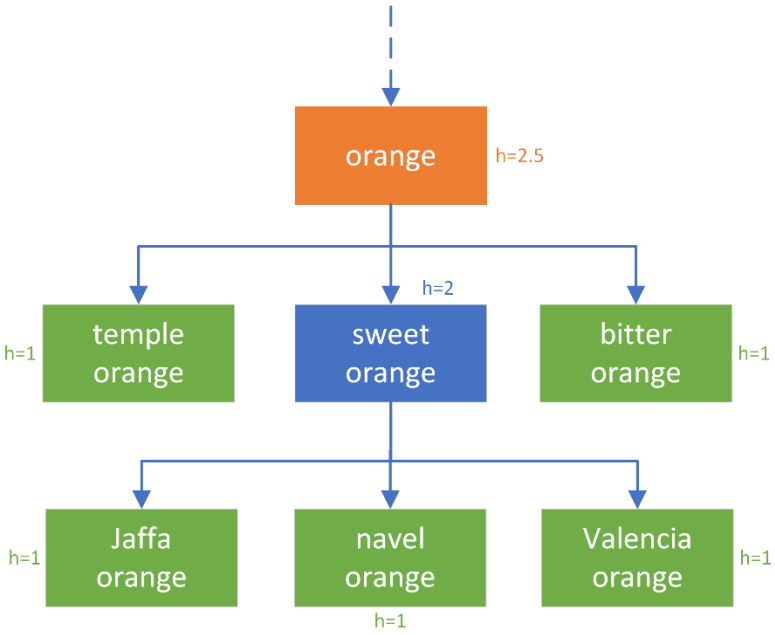

Figure 8: Example Calculating Average Height in WordNet – Orange

This example shows how we measure average height for Rule 21 as one plus the average of the minimum and maximum heights of a synset's children in the hierarchy. In Rule 20, we measure depth using an analogous formula in the opposite direction, which we do not repeat for brevity.

Rule 22 takes a ratio of the average height and average depth (as measured in Rules 20-21) using the formula shown in Equation 1.

Equation 1 $$AvgDepthHeightRatio = \frac{average\ depth}{average\ depth + average\ height}$$

Rules 20-22 each have two parameters:

- the minimum value allowed
- the maximum value allowed

The specific value being compared is the height, depth, or ratio being computed; any synset with a measurement outside the range from the minimum to the maximum value, inclusive, is filtered out.

Rules 23-24 and 26-27 use a much simpler distance measure for both depth and height. In each case, the maximum depth and maximum height is used without any averaging at each intermediate node in the chain. All paths from the candidate node to a hypernym root (for measuring depth) or hyponym leaf (for measuring height) are considered and the measurement is taken over the longest path.

Rules 23 and 26 filter out the top and bottom levels of the hierarchy, respectively. Each has a single parameter:

- the number of levels to filter

For Rule 23, if the depth is measured at this parameter or smaller, the synset is filtered. For Rule 26, if the height is measured at this number or smaller, the synset is filtered.

It may also be possible to filter synsets that are not strictly near the very bottom or top, but which are relatively deep or shallow in the hierarchy. Since the hierarchy varies considerably in the distance between root and leaf nodes, we also use extreme depth or height as an indication of being on the periphery in a longer chain.

Rule 24 filters out synsets with many others below them, while Rule 27 filters out synsets with many others above them. These share the same single parameter as used in Rules 23 and 26, listed supra. Rule 24 filters out synsets with a height at least as high as the parameter, while Rule 27 filters out synsets with a depth at least as high as the parameter.

The last rule pertaining to depth and height is Rule 25, which filters synsets by average depth. The average again is not the average across all chains but the midpoint between the minimum and maximum depth. Rather than doing this in a recursive approach like in Rules 20-22, though, these values for the minimum and maximum are simply the length of the shortest and longest chains from a root hypernym down to the candidate node.

Rule 25 has the same two numeric parameters as Rules 20-22, listed supra. Synsets are filtered if their average depth falls outside the range specified by the minimum and maximum parameters, inclusive.

4.2.2.1.12 Hyponyms

Rules 28-29 and 36-37 involve filtering synsets based on the number of hyponyms in some relationship to the candidate synset.

Rule 28 filters all siblings of synsets having zero hyponyms. Since basic-level categories should appear somewhere near the middle of the hierarchy, a sibling not having any hyponyms could indicate that the candidate is a sibling to a non-basic category and thus itself be unlikely to be a basic-level category. Rule 28 is one of only three Filtering Rules without any parameters to be set.

Rule 29 filters all synsets with a number of hyponyms in a specified range, requiring two parameters:

- the minimum number of hyponyms required for filtering
- the maximum number of hyponyms required for filtering

Hyponyms falling within the range specified by these two parameters, inclusive, are filtered.

Rules 36-37 are both novel rules.

Rule 36 is a more restrictive version of Rule 28; rather than filtering all siblings of synsets having zero hyponyms, it filters synsets with at least a specified number of siblings having zero hyponyms. This requires one parameter:

- the number of siblings without any hyponyms required for filtering

Rule 37 looks at the percentage of siblings having no hyponyms and filters based on this ratio rather than a count. It also requires one parameter:

- the minimum percentage of siblings having no hyponyms required for filtering

4.2.2.1.13 Brown Corpus Frequency

Since basic-level category names will tend to be used to refer to an object unless a more specific context requires a different level of granularity (Rosch et al. 1976), basic-level categories may be frequently used in a corpus. Although this seems intuitive, we have not seen this validated in any prior research. There are many factors affecting when a word is used, including how often the referents are of relevance to a discussion; relatedly, more inclusive categories may apply in a wider range of contexts.

We consider two rules related to corpus frequency. We use the Brown corpus (Francis et al. 1964) a one million word corpus of American English, to determine the frequency of a word.

Rule 30 filters out synsets lower than a specified frequency, while Rule 31 filters out synsets over a specified frequency. In each case, two parameters are required:

- whether the frequency is for the first lemma in the synset or the sum of the frequencies for all lemmas in the synset
- a numerical frequency that marks the boundary between the filtered and unfiltered (the specified frequency itself is not filtered)

4.2.2.1.14 Local Hypernyms

In the examples of basic-level categories provided in Rosch et al. (1976) and Markman et al. (1997), many of the examples are living things, tools, several other broad groupings. There are other parts of the WordNet hierarchy that appear to be unlikely to contain basic-level categories. We have two rules aimed at using patterns like this to filter out non-basic-level categories. Rule 32 filters all synsets under a specified set of synsets, while Rule 33 takes the opposite approach and filters all synsets except those under a specified set of synsets.

In each case, one parameter is required:

- the set of synsets to use

Students only proposed filtering synsets under *abstraction.n.06* and filtering all synsets except those under *physical_entity.n.01*, *thing.n.08*, *substance.n.01*, and *process.n.01*. So, in setting parameters here, these are the only ones we considered in building the sets used for the parameter.

4.2.2.1.15 CHILDES

Since basic-level categories are the first categories children learn, we use early childhood language data to inform two filters. The CHILDES corpus (MacWhinney 2000) is a collection of transcripts of early language acquisition. One student, the only who used this corpus, proposed a rule where we filter out all words in the CHILDES corpus. We include this as Rule 34. Since childhood language would be expected to indicate basic-level categories, this is the opposite of what we would expect the ideal rule to be using this corpus. It is unclear whether this was an accident on the student's part or if this was intentional; either way, we propose novel Rule 39 as the opposite: to filter out all categories not in the CHILDES corpus.

Each of these rules requires a single parameter:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset

4.2.2.1.16 Long Pronunciation

Basic-level categories are generally short words; aside from word length itself, another measure of the complexity of a word is the number of phonemes included in its pronunciation. The CMU Pronouncing Dictionary (Weide 1998) is a machine-readable English pronunciation dictionary which maps words to phonetic translations. We use this resource to determine the number of phonemes in a word, and this allows Rule 35 to filter out words with more than a specified number of phonemes.

Rule 35 requires two parameters:

- whether the rule is applied if it matches any lemma or only the first lemma in the synset
- the maximum number of phonemes a word can have without being filtered

4.2.2.1.17 Siblings

Many of the example basic-level categories from Rosch et al. (1976) and Markman et al. (1997) map to synsets with large numbers of siblings. We propose a novel rule, Rule 38, which filters synsets with less than a specified number of siblings. This requires a single parameter:

- the minimum number of siblings a synset must have to remain unfiltered

4.2.2.2 Voting Rules

Voting rules are used after all of the Filtering Rules have been applied. Whereas the Filtering Rules apply at an individual synset level, with logic being applied to a synset to determine whether or not it should be filtered, Voting Rules are applied to a synset chain. A Voting Rule can be thought of as giving a single point to one or more synsets in the chain. Only synsets not already filtered out remain as potential basic-level categories, but the entire chain is used for applying the Voting Rules.

So for example, if a chain of length eight only has two synsets that have not been filtered out by the Filtering Rules, those are the only two which may be proposed as basic-level categories. However, if a Voting Rule awards a point to the word with the greatest frequency in a particular corpus, even a filtered synset may be awarded that point. After applying the Voting Rules, either zero or one categories in the chain may be proposed as basic-level categories. In order to be proposed, the category must have the majority of the points awarded by Voting Rules, be over a certain minimum threshold, and not have been filtered out by a Filtering Rule.

The Voting Rules considered are listed in Table 9. We then describe each of the rules in more detail. The rule numbering continues from where we left off numbering the Filtering Rules so that each rule has a unique number. Unlike Filtering Rules, which almost all include parameters, only a few of the Voting Rules have numerical thresholds; we used the student values provided rather than setting parameters as we did with the Filtering Rules. We describe these only when present.

Table 9: Voting Rules Used to Select the Best Basic-Level Candidate in a Hypernym Chain

Voting Rules
40. Top frequency in the chain (sum of lemma frequencies in synset)
41. Top frequency in the chain in SEMCOR and frequency <= n (60)

42. Word length between a and b [3, 7]
43. Synset is of depth a to b in the hierarchy [6, 10]
44. The word appears in Dolch's Word List
45. The word appears in compound nouns
46. Maximum % of descendants including the term as a compound in the chain
47. The synset has hyponyms
48. The highest value in the chain for (frequency in Brown + 1)/15 + (compounds in hyponym subtree containing word + 1)/5
49. Highest frequency in Brown + Gutenberg corpora combined in the chain
50. Maximum word length in chain
51. Maximum number of meronyms in the chain
52. Minimum word length in chain

4.2.2.2.1 Frequency

Rules 40-41 and Rules 48-49 both involve using corpus frequency to identify the synset in a chain which has the highest frequency. Rule 48 includes a combination of both frequency information and information about compounds, which are discussed in §4.2.2.2.5, so we defer discussion on that rule to §4.2.2.2.5. The other mentioned rules exclusively focus on frequency, so we discuss them here. In all cases, where word frequency is involved we arrive at a synset frequency by summing the frequencies of each of the lemmas associated with the synset.

The main difference between the three frequency rules is the corpora used to compute the frequency, though Rule 41 and Rule 49 also each have an additional complexity.

Rule 40 simply finds the synset with the highest frequency using the frequencies built into WordNet.

Rule 41 does the same thing using the SEMCOR corpus, but also restricts consideration to words under a particular frequency. SEMCOR (Landes et al. 1998) is a WordNet sense-tagged corpus. Rule 41 requires a single parameter: the maximum SEMCOR frequency allowed for a synset to be considered a basic-level category. The value used for this parameter is 60 as proposed by the student who used it.

Rule 49 takes the highest frequency in a much larger combination of two corpora, both Brown (Francis et al. 1964) and Gutenberg corpora. The Gutenberg Corpus is a subset of the public domain books available on Project Gutenberg (Gutenberg 2018) and made available by the Natural Language Toolkit (Loper et al. 2002).

4.2.2.2.2 Word Length

Rule 42, Rule 50, and Rule 52 all involve the length of a word. In each case, this is only applied to the first lemma in the synset.

Rule 42 applies to any word with a length in a specified range, which requires a minimum and maximum length to define the range of length allowed without filtering. We use a minimum of three and a maximum of seven as proposed by the student who used this rule.

Under this rule, multiple synsets in the chain may equally fall within the range; in this case, this rule votes for each of them (i.e. gives each one a point).

Rule 50 votes for the longest word in the chain, while Rule 52 votes for the shortest word in the chain. In the case of a tie, the rule votes for each of the corresponding synsets (i.e. gives each one a point).

4.2.2.2.3 Depth

Rule 43 is the only rule which relates to depth. Since we are looking at a specific chain in the hypernym/hyponym hierarchy we have no need to deal with the complexity of calculating depth as in §4.2.2.1.11. Instead, we simply use the depth of a synset within the current chain being considered. For the purposes of this rule, the top synset has a depth of zero and each synset below that has a depth one greater than the synset above it.

Rule 43 votes for all the synsets within a specified depth range, inclusive, which requires two parameters: the minimum depth and the maximum depth allowed without filtering. We used values of six and ten for these two parameters, respectively, in accordance with the student who used this rule.

4.2.2.2.4 Dolch's Word List

Dolch's Word List (Dolch 1948) is a list of 510 words commonly spoken by kindergarteners. Rule 44 votes for all synsets where the first lemma is in Dolch's Word List.

4.2.2.2.5 Compounds

Rules 45-46 and Rule 48 each involve compound words. A compound, here, specifically refers to a lemma name in WordNet that has a space in it (represented as an underscore ['_'] in WordNet).

Rule 45 votes for any synsets where the first lemma in its entirety is used in any other compounds in the WordNet noun hierarchy, not just in the specific chain in question. Compounds may be found in non-first lemmas in other synsets, so although the candidate synset being voted on only uses its first lemma, it will still count if that lemma only appears as a component of a compound in a second or third lemma of another synset.

Rule 46 votes for the synset with the highest portion of its descendants (including those outside the particular chain in question) having its first lemma as a part of a compound word in its descendants' lemmas.

Rule 48 combines compounds with frequency information. As with other frequency measures, this rule votes for the highest value in the chain. However, instead of voting for the highest frequency value, it votes on the highest value for the quantity represented by Equation 2, which incorporates both frequency and compound information. *Brown Frequency* represents the sum of the frequencies of the synsets' lemmas in the Brown corpus (Francis et al. 1964), while *Compounds in hyponym subtree* represents the number of descendants of the candidate node with its first lemma as a component of a compound in the descendant node.

Equation 2 $$\frac{Brown\ Frequency+1}{15}+\frac{Compounds\ in\ hyponym\ subtree+1}{5}$$

4.2.2.2.6 Hyponyms

Rule 47 votes for any synset with hyponyms. This means it votes for all but one (the bottom synset) in each chain.

4.2.2.2.7 Meronyms

Rule 51 votes for the synset in the chain with the most meronyms. Since WordNet includes meronym relationships, this used the WordNet meronym relationships and not another source.

4.3 Experiments

We use the train, development, and test sets described in §3.10 to build a system combining the Filtering Rules from §4.2.2.1, the Voting Rules from §4.2.2.2, and to evaluate their combination. The process we use for this experimentation is illustrated in Figure 9.

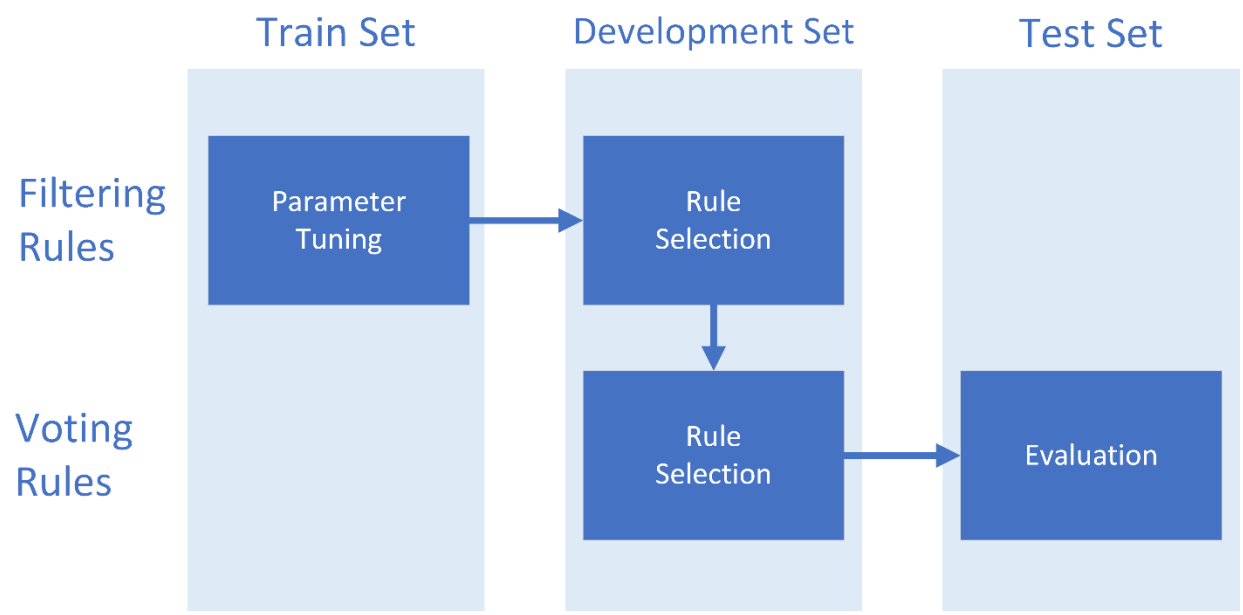

Figure 9: Experiment Design Flow Chart

We start by using the train set to tune parameters on the Filtering Rules, evaluating each rule in isolation across its parameter space and choosing the best parameters. This is discussed in §4.3.1. We then apply these rules with the parameters from the first step to the development set, and only keep the rules that generalize well to this new set. We describe this in §4.3.2. With the set of Filtering Rules finalized, we look at the Voting Rules and on the development set, choosing the combination of rules and a threshold of votes needed to select a category as basic-level, which we discuss in §4.3.3. Finally, we take this complete system and evaluate its performance on the test set in §4.3.5.

4.3.1 Parameter Tuning Filtering Rules

We start by tuning the parameters of the Filtering Rules. To do this, we start with any range of parameters specified by students and exhaustively explore the relevant parameters for whole-number and categorical parameters. For parameters with decimal parameters, within the range of relevant parameter values we explore the values to two significant figures.

Rule 1, which filters prefixes, deserves special mention as an exception to the exhaustive exploration of the parameter space. The number of possibilities is a prohibitive 2^{59}, so we instead settle for a greedy

search where we start at the beginning of the list and add prefixes one at a time, keeping them if they help the rule perform better and rejecting them if they harm the rule's performance or produce no effect.

We mention optimizing the system performance based on the values of the parameters, but we have not discussed an objective function being optimized. We optimize for the number of synsets filtered out by the rule, but only insofar as we maintain a precision of 100%. We do not allow any basic-level categories to be filtered out on the train set in order to keep the rule.

In Table 10 we show the best parameter settings for each rule and whether or not the rule produces an improvement and thus was kept for Rule Selection or not.

Table 10: Chosen Parameters for Filtering Rules

Rule	Abbreviated Description	Parameters	Result
1	Suffixes	Affixes: 'ment', 'ism', 'ness', 'tion', 'ing', 'ty', 'tor', 'c', 're', 'th', 's', 'ive', 'ance', 'ence', 'ssory', 'ssary', 'tant', 'bent', 'tory', 'ist', 'dle', 'tal', 'est', 'ful', 'hood' Minimum lemma length: 0 Lemma(s): first only	keep
2	Prefixes	Affixes: 'un', 'th' Minimum lemma length: 0 Lemma(s): first only	keep
3	Long words	Minimum length: 14 Lemma(s): first only	keep
4	Short words	Maximum length: 3 Lemma(s): first only	discard
5	Space-separated compounds	Lemma(s): first only	discard
6	Hyphenated words	Lemma(s): first only	keep
7	Joined compounds	Minimum length: 6 Lemma(s): first only	keep
8	Words with numbers	Lemma(s): first only	discard
9	Words with symbols	Lemma(s): first only	keep
10	Adjective senses	Lemma(s): first only	keep
11	Adverb senses	Lemma(s): first only	discard
12	Verb senses	Lemma(s): any	discard
13	No subordinate substrings	*No parameters*	discard
14	Substring of superordinate	*No parameters*	discard
15	Stopwords	Lemma(s): any	keep
16	Plurals	Lemma(s): any	discard
17	No vowels	Maximum vowels filtered: 0 Lemma(s): any	discard

18	Many vowels	Minimum vowels filtered: 6 Lemma(s): first only	keep
19	Capitalized words	Lemma(s): first only	keep
20	Average depth	Minimum: 3 Maximum: 14	keep
21	Hyponym depth	Minimum: 1.0 Maximum: 3.5	keep
22	Depth ratio	Minimum: 0.57 Maximum: 0.94	keep
23	Top of hierarchy	Levels to filter: 4	discard
24	Many levels below	Levels to filter: 10	keep
25	Low min-max depth	Minimum: 4 Maximum: 14	keep
26	Bottom of hierarchy	Levels to filter: 1	discard
27	Many levels above	Levels to filter: 14	keep
28	0-hyponym siblings	*No parameters*	discard
29	Hyponym range	Minimum: 1.1 Maximum: 2.0	discard
30	Low Brown frequency	Lemma(s): all Minimum unfiltered: 1	discard
31	High Brown frequency	Lemma(s): all Maximum unfiltered: 200	discard
32	Under particular synset	Synsets: *abstraction.n.06*	discard
33	Not under particular synset	Synsets: *physical_entity.n.01*, *thing.n.08*, *substance.n.01*	keep
34	In CHILDES corpus	Lemma(s): any	discard
35	Long pronunciation	Maximum unfiltered: 10 Lemma(s): any	keep
36	Many 0-hyponym siblings	0-hyponym siblings: 65	keep
37	Fraction 0-hyponym siblings	Percent of siblings: 100%	discard
38	Few siblings	Siblings: 1	discard
39	Not in CHILDES corpus	Lemma(s): any	discard

The portion of each label type filtered out by each parameterized rule is shown in Table 11. The rules filtered out are displayed in gray.

Table 11: Performance on Train Set by Label Subcategory

		Percent Filtered by Label			
Rule	**Abbreviated Description**	**Superordinates**	**Basic-level**	**Subordinates**	**None**
1	Suffixes	27%	0%	15%	19%
2	Prefixes	0%	0%	1%	0%
3	Long words	14%	0%	11%	15%

4	Short words	2%	6%	1%	1%
5	Space-separated compounds	32%	9%	3%	28%
6	Hyphenated words	0%	0%	3%	2%
7	Joined compounds	5%	0%	6%	3%
8	Words with numbers	0%	0%	0%	0%
9	Words with symbols	0%	0%	3%	3%
10	Adjective senses	3%	0%	2%	1%
11	Adverb senses	0%	0%	0%	0%
12	Verb senses	2%	1%	1%	3%
13	No subordinate substrings	0%	26%	72%	47%
14	Substring of superordinate	22%	6%	22%	23%
15	Stopwords	2%	0%	0%	0%
16	Plurals	5%	4%	7%	15%
17	No vowels	0%	0%	0%	0%
18	Many vowels	14%	0%	10%	16%
19	Capitalized words	0%	0%	17%	4%
20	Average depth	5%	0%	2%	1%
21	Hyponym depth	10%	0%	0%	1%
22	Depth ratio	19%	0%	0%	0%
23	Top of hierarchy	22%	4%	0%	9%
24	Many levels below	27%	0%	0%	0%
25	Low min-max depth	12%	0%	2%	1%
26	Bottom of hierarchy	0%	65%	79%	76%
27	Many levels above	0%	0%	3%	1%
28	0-hyponym siblings	0%	65%	79%	76%
29	Hyponym range	0%	3%	4%	6%
30	Low Brown frequency	34%	49%	67%	7%
31	High Brown frequency	8%	4%	1%	2%
32	Under particular synset	10%	1%	1%	6%
33	Not under particular synset	7%	0%	0%	4%
34	In CHILDES corpus	37%	53%	12%	14%
35	Long pronunciation	10%	0%	5%	5%
36	Many 0-hyponym siblings	0%	0%	10%	0%
37	Fraction 0-hyponym siblings	0%	6%	33%	31%
38	Few siblings	0%	4%	8%	9%
39	Not in CHILDES corpus	63%	47%	88%	86%

4.3.2 Rule Selection on Filtering Rules

We next take each of the rules for which there is an acceptable set of parameters on the train set and apply it to the development set to evaluate whether it generalizes well to another part of the hierarchy. The results of this evaluation are shown in Table 12.

Table 12: Effectiveness of Filtering Rules on Development Set

		Percent Filtered by Label				
Rule	**Abbreviated Description**	**Superordinates**	**Basic-level**	**Subordinates**	**None**	**Result**
1	Suffixes	31%	0%	0%	1%	keep
2	Prefixes	4%	5%	1%	3%	discard
3	Long words	19%	0%	5%	5%	keep
6	Hyphenated words	0%	0%	4%	2%	keep
7	Joined compounds	4%	2%	3%	1%	discard
9	Words with symbols	0%	0%	5%	3%	keep
10	Adjective senses	4%	0%	0%	2%	keep
15	Stopwords	0%	0%	0%	0%	discard
18	Many vowels	12%	0%	0%	3%	keep
19	Capitalized words	0%	2%	3%	1%	discard
20	Average depth	0%	0%	0%	0%	discard
21	Hyponym depth	19%	0%	0%	0%	keep
22	Depth ratio	4%	0%	0%	*0%*	keep
24	Many levels below	12%	0%	0%	0%	keep
25	Low min-max depth	0%	0%	0%	0%	discard
27	Many levels above	0%	0%	0%	0%	discard
33	Not under particular synset	0%	0%	0%	0%	discard
35	Long pronunciation	15%	0%	1%	2%	keep
36	Many 0-hyponym siblings	0%	0%	0%	0%	discard

Almost half of the rules that work well on the train set perform poorly or have no impact on the development set, while a slight majority is retained as the final set of ten Filtering Rules.

4.3.3 Rule Selection on Voting Rules

The Voting Rules (Table 9) are applied to categories not already filtered by Filtering Rules. These are applied along each chain from the bottom to the top of the hypernym hierarchy. Like Filtering Rules, these rules are also applied to a category although evaluated in the context of a chain.

Using a greedy search starting with the most accurate Voting Rules, we identify a set of the rules which together enabled high accuracy on the development set. This combination is listed in **Error! Reference s ource not found.**.

We determine that by using these rules together, and only selecting categories with three of these Voting Rules being fulfilled, high accuracy could be obtained on the development set. This does limit the number

of basic-level categories that can be selected to one in each chain from the bottom to the top of the hypernym hierarchy. However, with three of the four rules only being fulfilled for one node in the chain, it is possible not to select a basic-level category in some chains.

Table 13: Selected Voting Rules

40. Top frequency in the chain (sum of lemma frequencies in synset)
47. The synset has hyponyms
49. Highest frequency in Brown + Gutenberg corpora combined in the chain
51. Maximum number of meronyms in the chain

4.3.4 Full System

With thresholds set on the Filtering Rules, the Filtering Rules that generalize well selected, a set of selected Voting Rules, and a combination strategy for those Voting Rules, we describe our full system in Table 14.

Table 14: Full Heuristic System Description

First, filter out all synsets meeting any of the selected Filtering Rules:
1. Filter synsets where the first lemma has any of the following suffixes: 'ment', 'ism', 'ness', 'tion', 'ing', 'ty', 'tor', 'c', 're', 'th', 's', 'ive', 'ance', 'ence', 'ssory', 'ssary', 'tant', 'bent', 'tory', 'ist', 'dle', 'tal', 'est', 'ful', 'hood'
3. Filter synsets where the first lemma is 14 characters or longer
6. Filter synsets where the first lemma is hyphenated ('-')
9. Filter synsets where the first lemma contains a symbol
10. Filter synsets where the first lemma has more adjective than noun senses
18. Filter synsets where the first lemma has over 6 vowels
21. Filter synsets with average height ((min+max)/2, recursive) outside the range 1.0 to 3.5, inclusive
22. Filter synsets with avg_depth/(avg_depth+avg_height) outside the range 0.57 to 0.94, inclusive
24. Filter synsets with 10 or more levels of hyponyms below them
35. Filter synsets where any of its lemmas are in the CMU Pronouncing Dictionary with > 10 phonemes
Next, on each chain up the hierarchy, apply the following Voting Rules to each synset in the chain and select any unfiltered synsets triggering at least three of these Voting Rules:
41. Top frequency in the chain (sum of lemma frequencies in synset)
48. The synset has hyponyms
50. Highest frequency in Brown + Gutenberg corpora combined in the chain
52. Maximum number of meronyms in the chain

4.3.5 Evaluation and Discussion

Our system's overall performance on the test data is listed in Table 15.

Table 15: Overall Effectiveness of the Heuristic System

Metric	Value
Accuracy	88%
Precision	42%
Recall	35%
F-score	**0.381**

While our system only attempts to distinguish between basic-level categories and non-basic-level categories, we break down our system's performance across each of the label subcategories as we describe them in §3.7 and §3.9. This breakdown is shown in Table 16.

Table 16: Accuracy of the Heuristic System by Subcategory

Subcategory	Accuracy
Superordinate	83%
Basic-level	35%
Subordinate	98%
None	73%

Accuracy is measured as the percentage of categories filtered (or not filtered) correctly based on the test data. So, for example, an accuracy of 83% on subordinates, which should be considered non-basic-level, means that 83% of subordinates are correctly labeled as non-basic-level. And for the basic-level, 35% accuracy means 35% of these categories are correctly labeled as basic-level. Our system did well at filtering out subordinates which predominate the overall label set. It performed much more poorly on basic-level categories, our focus, than the other classes.

In Table 17 we show that out of the basic-level categories mistakenly chosen as non-basic, most mistakes are in chains were a superordinate is chosen as basic-level. There are several superordinates that are relatively short words with high frequency, including ***tool***, ***seat***, and ***device*** which are superordinates chosen as basic-level by the heuristics. Since the same superordinate may be a strong

choice to the system across multiple chains this results in a small number of superordinates accounting for most false negatives.

Table 17: Subcategories Chosen by Mistake as Basic-Level Categories

Subcategory	Portion of False Negatives
Superordinate	90%
Subordinate	1%
None	9%

Our Filtering Rules are parameterized on the train set and selected on the development set, only including rules which make no mistakes on basic-level categories in each set. Only 53% of the Filtering Rules with acceptable performance on the train set are selected based on their extrapolation to the development set. Then, these only result in an accuracy of 35% on the test set. This indicates that extrapolating from one set to another is difficult.

When we repeat this entire procedure of parameterization and rule selection just on the Rosch-Markman labels as described in §3.1, the rules extrapolate much better. Our system performance when trained, tuned, and evaluated on this data is shown in Table 18 and Table 19. Note that there is no 'None' subcategory in Table 19 because this is a novel label that does not exist in the Rosch-Markman Labels.

Table 18: Heuristic System Effectiveness on Rosch-Markman Labels

Metric	Value
Accuracy	77%
Precision	68%
Recall	84%
F-score	**0.749**

Table 19: Heuristic System Accuracy by Subcategory on Rosch-Markman Labels

Subcategory	Accuracy
Superordinate	100%
Basic-level	84%
Subordinate	44%

This performance is almost twice as effective on the Rosch-Markman Labels as on our new, expanded labels. This indicates how the Rosch-Markman Labels, using salient rather than representative examples as discussed in §3.3.1 and §3.5, is an easier set to experiment on and systems trained and evaluated on these labels will drastically over-estimate their performance. While this highlights the value of our labeling process, it also shows our heuristic system has plenty of room for improvement.

While our system's accuracy on the broader labels is 88%, which seems high, both the precision and recall are much lower. An important consideration in comparing the Rosch-Markman Labels with our new labels is the balance of labels across subcategories. The Rosch-Markman Labels are built around experiments on basic-level categories and thus have over half of the labels as positive examples (basic-level categories). In our labels, however, there is a much greater imbalance with only 2.3% being basic-level categories while 92% are subordinates. This leads to our system needing a very high accuracy on subordinates, where we achieve 98% accuracy, in order just to achieve a precision of just 42%. Filtering out so many subordinates in turn causes us to be aggressive at filtering in general, which leads to the issues with recall. The class imbalance makes achieving great performance challenging.

We next discuss subsystem performance with the effectiveness of our two main subsystems shown in Table 20.

Table 20: Subsystem Effectiveness for the Heuristic System

Subsystem	Precision	Recall	F-score
Filtering Rules	13%	94%	0.221
Filtering Rules + Voting Rules	42%	35%	0.381

The Filtering Rules have a very high recall, indicating that not many basic-level categories are being filtered out. However, the precision is also low. The Voting Rules improve precision by a factor of 3.2 while decreasing recall by a relative 63%. This results in an f-score 72% higher than the Filtering Rules alone, but the overall number is still relatively small.

4.4 Conclusion

We build a heuristic system to automatically identify basic-level categories using WordNet. We are effective at including most basic-level categories and excluding superordinates, but not as effective at excluding subordinates.

Chapter 5

Classifier-based System

We build a classifier-based system to identify basic-level categories in WordNet.

5.1 General Approach

Our classifier-based system is similar in design to the Heuristic System discussed in Chapter 4 , except with the Filtering Rules replaced by a Filtering Classifier, and in addition to Voting Rules we also consider two alternative strategies using classifier prediction scores for combining synset-level decisions to make the optimal decision within a particular synset chain. Our system diagram is shown in Figure 10.

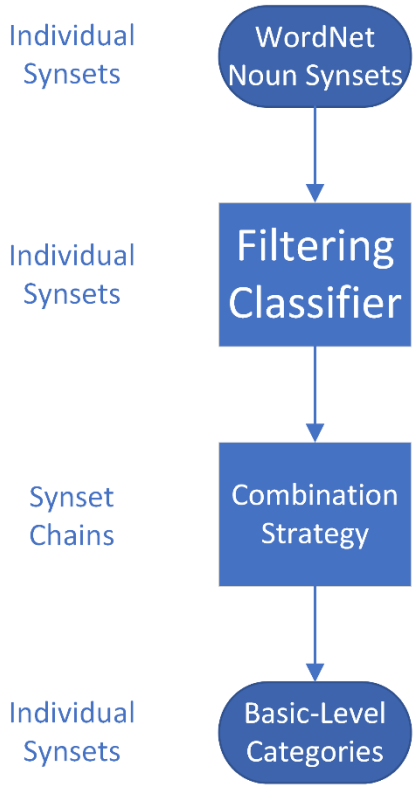

Figure 10: Classifier System Architecture Diagram

We start with all noun synsets in WordNet, and filter out many of these that are unlikely to be basic-level categories according to a Filtering Classifier, discussed in §5.2. We then try several Combination

Strategies (§5.3) to take the synsets left after filtering, and in some cases the classifier score for each, to determine which category to pick through each chain. The output of this is the proposed set of basic-level categories, which we then evaluate (§5.4).

5.2 Filtering Classifier

The Filtering Classifier is the first of two components in our system. This stage applies to WordNet noun synsets, filtering out ones that are unlikely to be basic-level categories. We discuss the features (§5.2.1), the learning algorithm chosen (§5.2.2), and our optimization grid search for that classifier (§5.2.3). We look at the learning curve (§5.2.4) to project what the effectiveness of the classifier might be with additional data available.

5.2.1 Features

Since we already have a heuristic system which incorporates domain knowledge in the form of rules, we draw on the Filtering Rules discussed in §4.2.2.1 as inspiration for our features.

In many cases there are multiple rules using the same underlying number. For example, Rule 3 filters words with length of *n* or greater while Rule 4 filters words with length *n* or fewer. In our classifier, we can replace these two rules with a number representing the word's length. This results in fewer features being needed than the number of rules.

In some cases, the Filtering Rules are fundamentally binary and there's no number involved. In these cases, we adopt the Filtering Rule itself as a binary feature, mapping true to 1 and false to 0 since our classifiers take numeric inputs. As an example, Rule 5 filters out compound words that contain a space. While we could conceivably include a feature for the number of space-separated tokens in the word, this naturally lends itself to binary representation and thus we treat this as a binary feature equivalent to the rule.

Some binary features do pose one additional challenge in this framework: if the rule is parameterized with non-numeric values, we could have a number of possible choices: the rule as parameterized in the Filtering Rules of our Heuristic System, the most inclusive version of the rule, a separate rule for each possible parameter, etc. In many cases the parameter is a simple binary one such as whether the rule

only applies to the first lemma in the synset or any lemma in the synset; in these cases we simply use the parameter values from the heuristic rules. In cases where there is a wider variety of options, such as in choosing a set of suffixes, given our relatively sparse positive labels we prefer not to create many separate rules and multiply our features unnecessarily, so we take the rule as parameterized in the Filtering Rules in addition to the version of the rule with all of its available options included for maximum recall. This prevents an explosion of features for every individual parameter value while still allowing some additional signal beyond the Heuristic System at a cost of a handful of extra features.

Following these principles, we build our feature set. The full list is shown in Table 21. While many of these are straightforward when reviewing their comparison to the related Filtering Rule(s) in the Heuristic System, since we have specified values for some parameters we briefly describe each feature for clarity. Nonetheless, the more verbose descriptions of relevant Filtering Rules in §4.2.2.1 may be of assistance in understanding some features—particularly including the motivation for including them—since the features are based on Filtering Rules which were described in more detail supra.

Table 21: Features Used in the Filtering Classifier

Feature	Description	Feature Type
1	The word has one of the suffixes selected in Rule 1	Binary
2	The word has any of the suffixes considered in Rule 1	Binary
3	The word has one of the prefixes selected in Rule 2	Binary
4	The length of the shortest lemma in the synset	Numeric
5	The length of the first lemma in the synset	Numeric
6	The first lemma contains a space	Binary
7	The word is hyphenated (‘-‘)	Binary
8	The word is a joined compound (e.g. ‘racetrack’)	Binary
9	The word contains a number	Binary
10	The word contains a symbol	Binary
11	Adjective-Noun ratio	Numeric
12	Adverb-Noun ratio	Numeric
13	Verb-Noun ratio	Numeric
14	No immediate subordinates include the word as a substring	Binary
15	No hypernyms have the word as a substring	Binary
16	The word is a stopword	Binary
17	The word is a plural	Binary
18	The number of vowels in the word	Numeric
19	The word is capitalized	Binary
20	The average depth of the synset	Numeric
21	The average height of the synset	Numeric

22	The average depth-height ratio of the synset	Numeric
23	The maximum depth of the synset	Numeric
24	The maximum height of the synset	Numeric
25	The number of siblings without hyponyms	Numeric
26	The number of hyponyms	Numeric
27	The log of the Brown Frequency of the word	Numeric
28	The synset is under *abstraction.n.06*	Binary
29	The synset is under *physical_entity.n.01*, *thing.n.08*, or *substance.n.01*	Binary
30	The word is in the CHILDES corpus	Binary
31	The number of syllables in the CMU Pronouncing Dictionary	Numeric
32	The percent of siblings without hyponyms	Numeric
33	The number of siblings	Numeric

5.2.1.1 Prefixes and Suffixes

Feature 1 and Feature 2 are both binary features indicating whether the first lemma in the synset ends with one of the suffixes on the list. The difference between the two rules is the set suffixes used. Feature 1 uses the set of suffixes chosen for Rule 1, which we list in Table 22.

Table 22: Prefixes Considered for Feature 1 to Identify Words with Relevant Prefixes

-ance	-est	-ism	-ness	-ssory	-tion
-bent	-ful	-ist	-re	-tal	-tor
-c	-hood	-ive	-s	-tant	-tory
-dle	-ing	-ment	-ssary	-th	-ty
-ence					

Feature 2, on the other hand, includes the entire list of suffixes we considered, listed in Table 23.

Table 23: Prefixes Considered for Feature 2 to Identify Words with any Prefixes

-ability	-cess	-est	-ize	-sis	-tics
-acy	-d	-ful	-le	-sity	-tion
-age	-dity	-hood	-less	-sm	-tion
-al	-dle	-ian	-logy	-some	-tive
-ance	-dosis	-ic	-ment	-ssary	-tor
-ate	-e	-ing	-ness	-ssory	-tory
-bent	-ed	-ism	-re	-tal	-ture
-bess	-ence	-ist	-s	-tant	-ty
-c	-er	-ity	-ship	-th	-y
-ce	-ess	-ive	-sion	-tic	

Feature 3 is similar to Feature 1 and Feature 2, except it indicates whether the first lemma in the synset starts with a prefix on a list of prefixes rather than suffix on a list of suffixes. Since Rule 2 only considers two prefixes, ‘un-‘ and ‘th-‘, and both are chosen during the parameterization of Filtering Rules, we only need one feature for prefixes.

5.2.1.2 Word Length

Feature 4 and Feature 5 are numeric features indicating the length of a word as measured in characters. For Feature 4, the word used is the shortest lemma in the synset. For Feature 5, the word used is the first lemma in the synset.

5.2.1.3 Compounds

Features 6-8 are binary features relating to whether or not the word is a compound based on two different ways of identifying compound words.

Feature 6 identifies whether the first lemma in the synset contains a space, which is encoded in WordNet as an underscore (‘_’). Feature 7 identifies whether the first lemma in the synset contains a hyphen (‘-‘).

Feature 8 identifies whether the first lemma in the synset is a joined combination of other lemmas in WordNet. We departed from the strict minimum length parameter in Rule 7, which required the other lemmas to be six characters or longer to be considered as part of a compound, and instead used a smaller threshold of three characters as the minimum for a component.

5.2.1.4 Numbers and Symbols

Feature 9 and 10 are binary features, with Feature 9 identifying words containing a number and Feature 10 identifying words containing a symbol. In both cases, only the first word in the synset is considered when evaluating the feature. As with Rule 9 in the Heuristic System, symbols are defined as any characters which are not alphabetical, numeric, space, or underscore.

5.2.1.5 Cross-Part-of-Speech Senses

Features 11-13 are numeric features representing ratios.

Feature 11 is the ratio of adjective senses of a word to the number of noun senses of that same word. In this case, as well as with Features 12-13, the word used is the first lemma of the synset the feature is being calculated for. Feature 12 is the ratio of adverb senses to noun senses of the word, and Feature 13 is the ratio of verb senses to noun senses of the word.

5.2.1.6 Substrings in Hypernym/Hyponym Chain

Features 14 and 15 involve the relationship between the first lemma of the synset and the lemmas of other synsets with hypernym or hyponym relationships with that synset. Both are binary features.

Feature 14 identifies synsets where none of the immediate hyponyms have any lemmas that contain the synset's first lemma as a substring. Feature 15 identifies synsets where none of the hypernyms up the WordNet hierarchy contain the synset's first lemma as a substring in any of their lemmas.

5.2.1.7 Stopwords

Feature 16 is a binary feature identifying whether or not the first lemma of the synset is a stopword on the standard English stopword list in the NLTK package in Python (Bird et al. 2009). The comparisons are case-insensitive.

5.2.1.8 Plurals

Feature 17 is a binary feature identifying whether the first lemma of the synset ends in the suffix '-s'.

5.2.1.9 Number of Vowels

Feature 18 is a numeric feature that counts the number of vowels ('a', 'e', 'i', 'o', and 'u') in the first lemma of the synset.

5.2.1.10 Capitalized Words

Feature 19 is a binary feature identifying whether the first letter of the first lemma of the synset is capitalized.

5.2.1.11 Depth and Height

Features 20-24 are all numeric features capturing aspects of where the synset falls vertically in the WordNet hierarchy between root hypernyms and leaf hyponyms. Feature 20 and 21 calculate the average

depth and height of the synset, respectively. We describe this calculation with a detailed example and diagram in the discussion of Depth and Height Rules in §4.2.2.1.11. Feature 22 is the ratio of the average depth to the average height, both values from Features 20 and 21. Features 23 and 24 represent the maximum depth and maximum height of the synset, respectively, calculated as the highest value of the depth or height across all chains passing through the synset.

5.2.1.12 Hyponyms

Features 25-26 and Feature 32 are all numeric features related to the number of hyponyms in some relationship to the candidate synset.

Feature 25 counts the number of siblings (other hyponyms of the candidate synset's hypernym) which themselves have no hyponyms. Feature 26 counts the number of hyponyms the candidate synset itself has.

Feature 32 is similar to Feature 25 except it is calculated as a percentage of siblings without hyponyms rather than the count of siblings without hyponyms.

5.2.1.13 Brown Corpus Frequency

Feature 27 is numeric and represents the base-10 logarithm of the frequency of the synset's lemmas in the Brown corpus (Francis et al. 1964). The frequency is obtained by summing the frequencies of each of the lemmas, and the base-10 logarithm is taken on that sum.

5.2.1.14 Local Hypernyms

Features 28 and 29 are binary features indicating whether the synset is in a certain region in the WordNet hierarchy by indicating whether the synset is a hyponym of specified hypernyms.

These are two separate features because the hypernym specified in Feature 28 is one (*abstraction.n.06*) which indicates the synset is likely not to be a basic-level category while those specified in Feature 29 (*physical_entity.n.01*, *thing.n.08*, *substance.n.01*) are hypernyms of many basic-level categories. As a result, we depart from our general attempt to reduce the number of rules into a smaller set of features and instead convert them into separate binary features with the intention of them providing signal in opposite directions.

5.2.1.15 CHILDES

Feature 30 is a binary feature indicating whether any of the lemmas associated with the synset appear in the CHILDES corpus (MacWhinney 2000).

5.2.1.16 Pronunciation

Feature 31 is a numeric feature indicating the number of syllables, according to The CMU Pronouncing Dictionary (Weide 1998), which the first lemma associated with the synset includes in its pronunciation.

5.2.1.17 Siblings

Feature 33 is a numeric feature which is a count of the total number of siblings of the candidate synset.

5.2.2 Choosing a Classifier

We explore using a variety of classifiers with the features listed in §5.2.1. We train the classifiers on the train set and evaluate them on the development set. The results are shown in Table 24.

Table 24: Different Classifiers - Effectiveness on Development Set

Classifier	Precision	Recall	F-Score	Accuracy
Logistic Regression	38%	64%	**0.477**	84%
Naïve Bayes	**46%**	43%	0.444	89%
Decision Tree (depth=5)	34%	57%	0.424	83%
AdaBoost (Decision Stumps)	39%	43%	0.409	86%
Random Forest	45%	30%	0.361	88%
Linear SVM	22%	54%	0.313	71%
Quadratic Discriminant Analysis	14%	**90%**	0.242	28%
kNN (n=1)	30%	2%	0.038	89%
RBF SVM	0%	0%	0.000	89%
Neural Network	0%	0%	0.000	89%

We used the default parameters in scikit-learn (Pedregosa et al. 2011) for these experiments, except that we adjust the sample weights for models that allow this. There is an extreme class imbalance, with subordinates comprising over 95% of all labels and basic-level categories only comprising 1%. This motivates us to use sample weighting. Without using weights, many classifiers tend to classify everything or nearly everything as non-basic-level and result in high accuracy but low or no recall. With these weights, we obtain reasonable basic-level recall. We select the weights by looking at the per-label-class

accuracy, with a special focus on subordinates due to their prevalence and basic-level due to them being the positive cases we are trying to learn, though we also attempt to ensure the mistakes aren't all pushed into the superordinate or 'None' classes.

Our weighting scheme, which reduces subordinates to 8% of their original weight, still results in subordinates comprising the majority of the aggregate weight. We also double the weight of basic-level categories, which takes them from 1% to 18% of the total weight. We finally decrease the weight of 'None' labels by 50% since this is the second largest label class and with subordinates so heavily down-weighted this ensures most weight for negative samples is still placed on subordinates. The aggregate impact of this weighting scheme is shown in Table 25 on the train set, with weight being taken from subordinates and redistributed to the other label subclasses, especially for the basic-level.

Table 25: The Aggregate Impact of Sample Weighting on the Train Set

Label Subclass	Original Weight	Adjusted Weight
Superordinate	0.60%	5.1%
Basic-level	1.0%	18%
Subordinate	95%	64%
None	3.1%	13%

We use f-score as our primary measure of performance due to the class imbalance, though we also report accuracy.

Logistic regression is the classifier that works best in these experiments. Naïve Bayes is the next best classifier, but it has substantially lower recall and higher precision; as a first pass at filtering out categories unlikely to be basic-level, we prefer higher recall and thus the choice between these two is straightforward even despite the higher accuracy for Naïve Bayes.

Decision trees and boosted decision stumps also perform reasonably well by comparison, but there is a sharp drop-off after that. Most of the worst classifiers do not support sample weighting, at least as implemented in scikit-learn, and have very low f-scores as a result, even as some have relatively high accuracy. For example, k-nearest neighbors is only able to identify 2% of basic-level categories, which leads to a very low f-score even with a precision value of 30% which has much less of a gap between it

and the better classifiers. Surprisingly, quadratic discriminant analysis produces a recall of 90%, by far the highest, despite not supporting sample weighting; the precision is very low, however, with an f-score barely half as much as the best classifier.

Based on these experiments, we select logistic regression as the classifier to use for our system and hence our subsequent experiments all use this classifier.

5.2.3 Optimizing the Classifier

Having selected logistic regression as our classifier, we now attempt to improve the performance of the classifier by tuning its parameters with a grid search.

We have two parameters to tune:

- The regularization penalty to use (L1 or L2)
- The coefficient *C* that has an inverse relationship with the strength of regularization (a high *C* will result in less impact from regularization)

L1 regularization penalizes strong individual feature weights by adding the magnitude of the feature weights in as a penalty term in the cost function being minimized. L2 is a stronger penalty, penalizing proportional to the square of the weights.

In the cost function being minimized, *C* is a coefficient on the loss term before regularization is incorporated, and regularization is added in without such a coefficient. So, *C* determines the weight of the standard cost function in the function being minimized, relative to a fixed regularization cost for a given set of weights; if *C* is higher this standard cost function will be more important, while if *C* is lower it will be less important. Meanwhile, the regularization term is unaffected by *C*. This results in the inverse relationship mentioned between *C* and the strength of regularization. When *C* is high the ordinary logistic regression cost function is more important and regularization has less impact; when *C* is low the impact of the standard cost function is diminished and regularization plays a larger role.

We experiment with these parameters using a grid search, where we vary the penalty between L1 and L2 while varying *C* exponentially from 0.01 to 1,000 by factors of 10. Our results are shown in Table 26. We

find that there are several different settings, using both L1 and L2 regularization, provide strong performance.

Table 26: Parameter Optimization for Logistic Regression on Development Set

Penalty	C	Precision	Recall	F-Score	Accuracy
L2	0.01	17%	41%	0.236	66%
L2	0.1	26%	57%	0.358	75%
L2	1	38%	64%	**0.477**	84%
L2	10	39%	62%	**0.477**	84%
L2	100	37%	59%	0.458	84%
L2	1000	38%	59%	0.460	84%
L1	0.01	18%	21%	0.191	78%
L1	0.1	25%	62%	0.352	72%
L1	1	37%	64%	**0.472**	83%
L1	10	37%	62%	**0.463**	83%
L1	100	42%	59%	**0.488**	85%
L1	1000	37%	57%	0.448	83%

As mentioned in §5.2.2, we prefer our classifier to have high recall to facilitate global optimization, though the best-performing parameters have low recall compared to the other top parameter settings. As a result, we also consider global optimization of the system in addition to local optimization of the classifier in choosing our settings. Using our best global optimization strategy, described in §5.3.3, we compare the overall system performance using the five highest-performing parameter settings in Table 27.

Table 27: Global Optimization of Classifier Parameters on Development Set

Penalty	C	Precision	Recall	F-Score	Accuracy
L2	**1**	64%	**75%**	**0.693**	**93%**
L2	10	64%	67%	0.669	92%
L1	1	62%	70%	0.659	92%
L1	10	62%	70%	0.659	92%
L1	100	**65%**	64%	0.646	92%

Based on the holistic impact to the system's overall performance, we use L2 regularization with a *C* value of 1 to achieve the best results on the development set.

5.2.4 Learning Curve

We construct a learning curve, shown in Figure 11, where we train our classifier on subsampled portions of our train data to show how the amount of data available affects the effectiveness of the classifier. We then extrapolate up to larger data sets for an indication of how effectiveness may increase with additional labeled data.

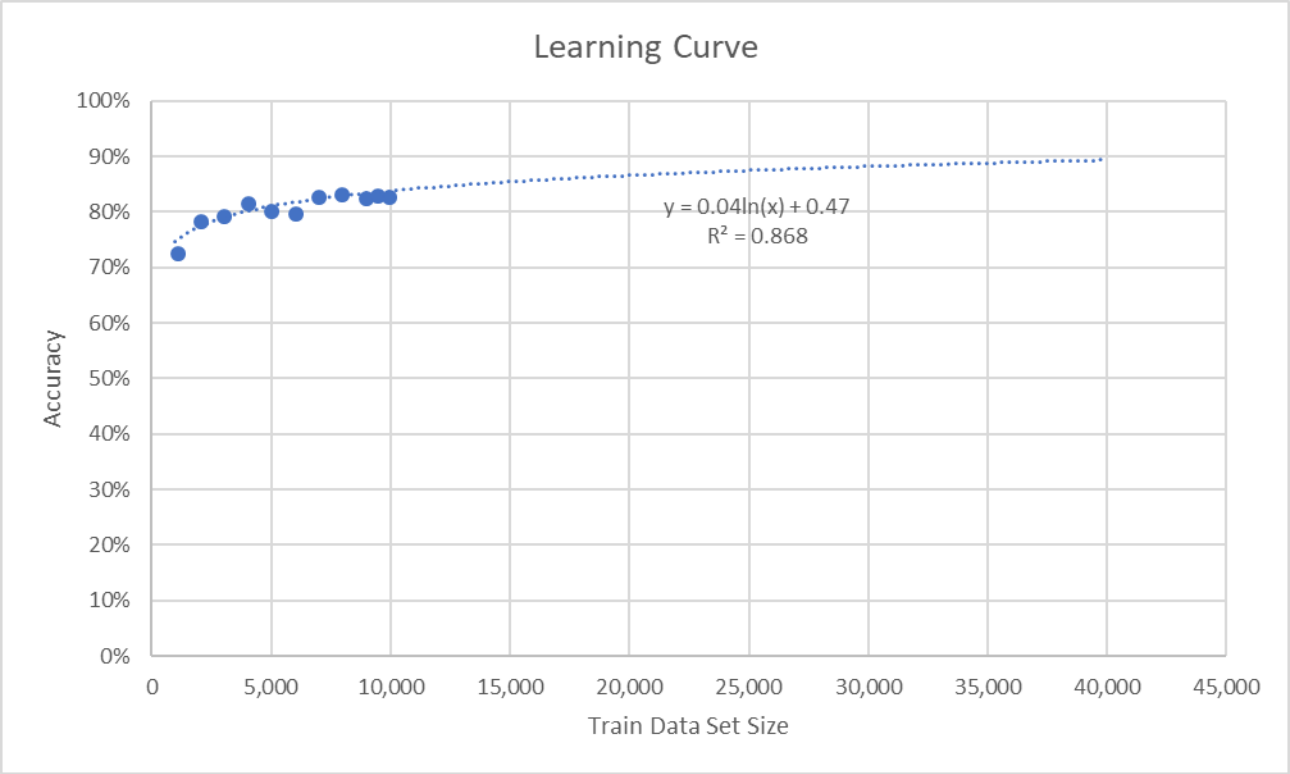

Figure 11: Learning Curve for Logistic Regression Classifier with Projection to Larger Data Sets

We find that effectiveness generally increases, as expected with additional training data, in a shape best fit by a logarithmic function. We find that accuracy would increase from the low-80s to around 90% if we were to obtain a data set roughly four times bigger, which would amount to nearly half of WordNet noun synsets being labeled.

Interestingly, with 100% of WordNet labeled this predicts we would expect to achieve an accuracy of 92%, which is the same as the 92% value for inter-annotator agreement we report in §3.10.

5.3 Combination Strategies

We consider three strategies for combining the Filtering Classifier output on individual synsets into a coherent set of basic-level category identifications up a chain or throughout the WordNet hierarchy as a whole. We first consider using the same Voting Rules we used in our Heuristic System, which we have described previously in §4.2.2.2 and revisit here in §5.3.1. We then discuss in §5.3.2 an approach using the Filtering Classifier's probability outputs at the synset level to choose the most likely combination of synsets we should predict as basic-level in order to maximize the likelihood of our overall predictions. Finally, in §5.3.3 we discuss revising this maximum likelihood approach to increase recall at the expense of truly maximizing likelihood by using a minimum probability threshold for basic-level prediction rather than requiring it to be more likely than a non-basic-level prediction. After describing these three approaches, we compare the systems in Table 28 of §5.4.

5.3.1 Voting Rules

Our first approach to combining the Filtering Classifier outputs into a consistent set of basic-level predictions is to essentially replace the Filtering Rules in our Heuristic System with our Filtering Classifier's binary judgments. In this system, illustrated in Figure 12, the Filtering Classifier's binary output decisions are used, and its more fine-grained probabilities are not considered when comparing synsets in a chain. Instead of using more granular classifier output, we instead turn to the Voting Rules described in §4.2.2.2 as part of our Heuristic System. These operate identically to how they operate in the Heuristic System. The only difference is the input to the Voting Rules; instead of synsets being filtered out from consideration by Heuristic Rules, they are instead filtered out by the Filtering Classifier.

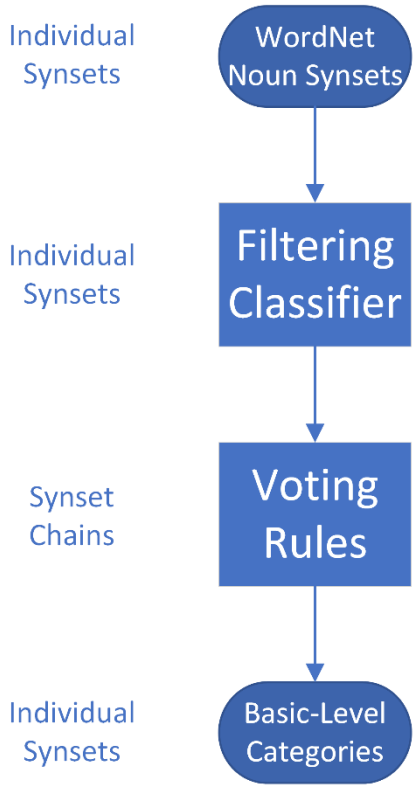

Figure 12: System Architecture Diagram with Voting Rules Combination Strategy

5.3.2 Maximum Likelihood

Our second approach to combining the output of the classifier on individual synsets into a coherent set of basic-level category identifications is to use a maximum likelihood approach. Rather than functioning on the synset chain, as we do with the Voting Rules, we now adopt a broader view of the entire hypernym/hyponym hierarchy. We illustrate this system in Figure 13.

Each synset in WordNet has been assigned a probability of being a basic-level category by our Filtering Classifier. We impose the constraint that on any chain up the hierarchy, a maximum of one synset may be labeled as a basic-level category. Of course, many chains may run through the same synset, so a decision made on one chain may affect others as well. Within the bounds of this constraint, our approach is to choose the combination of basic-level and non-basic-level labels that maximize the probability of our predictions, according to our classifier, for the entire hierarchy.

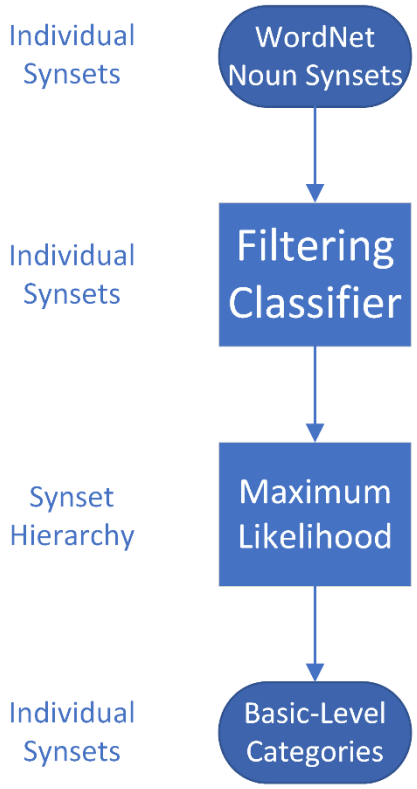

Figure 13: System Architecture Diagram with Maximum Likelihood Combination Strategy

While this is simple to state, implementing such a system is non-trivial since it is computationally infeasible to consider every possible combination of basic-level categories chosen, calculate the aggregate probability for each, and choose the one with the highest aggregate probability.

We present pseudocode for the algorithm we use in Figure 14. In summary, starting with leaf nodes in the hierarchy we track whether it's more likely the node is basic or non-basic. This proceeds up the hierarchy using a queue to keep track of when all the children of a node have been processed and it's safe to consider the parent. When a node with children is considered, we determine whether the node should be marked as basic by comparing the best probability below combined with the current node not being basic against the probability that none of the nodes below are basic with the current node being basic. At each node we track two probabilities for the subtree rooted at this node: the probability that the entire subtree is non-basic and the best overall probability for the subtree regardless of which labels have been chosen.

We mark any node that locally appears to be a good choice as a potential basic-level category at the time the node is evaluated (i.e. considering it and its descendants).

Once we've completed this upward pass, we then make a downward pass down the hierarchy. We branch out to each child, taking the first marked as basic-level we encounter and stopping there for each of the branching paths we traverse. Since the higher nodes always incorporate more information than those below, stopping at the marked nodes prevents us from selecting a node that locally appears to be basic-level but which would not be when considered in a broader context. There may be additional nodes lower in the tree that are marked but not chosen as basic-level because, when deciding over a larger portion of the hierarchy, it makes more sense to choose a higher node in the hierarchy as basic-level.

This maximizes the aggregate probability of the resulting hierarchy, though this makes one critical assumption: each probability computation is independent, and thus multiplying all of the probabilities together results in the overall probability of that hierarchy having the corresponding labels.

We present a simple example to show how this works. We show a sample subtree in Figure 15, where each node is numbered for identification and associated with a probability of being basic (as would be produced by the Filtering Classifier). We use a tree of height three because this is the smallest tree which enables us to show how a node can consider its children without worrying about its other descendants, and because we can illustrate some nodes appearing locally good that are overridden above alongside others not overridden.

```
MAXIMUM LIKELIHOOD TREE(BASIC_LOG_PROBS):
    CHILD_STATS ← Dictionary()
    Q ← Queue()
    for each noun SYNSET without hyponyms in WordNet: Q.put(SYNSET)
    while not Q.empty(): #Upward Pass
        SYNSET ← Q.get()
        PARENT ← PARENT of SYNSET #dummy "empty->SYNSET" if root node
        STATS ← CHILD_STATS[SYNSET] #empty if not present
        LOG_P_BEST ← sum(LOG_P_BEST in STATS.CHILDREN)
        LOG_P_BEST_NON_BASIC ← sum(LOG_P_BEST_NON_BASIC in STATS.CHILDREN)
        if BASIC_LOG_PROBS[SYNSET].LOG_P_BASIC + LOG_P_BEST_NON_BASIC >
        BASIC_LOG_PROBS[SYNSET].LOG_P_NON_BASIC + LOG_P_BEST:
            Mark SYNSET
            LOG_P_BEST ← BASIC_LOG_PROBS[SYNSET].LOG_P_BASIC + LOG_P_BEST_NON_BASIC
        else:
            LOG_P_BEST ← BASIC_LOG_PROBS[SYNSET].LOG_P_NON_BASIC + LOG_P_BEST,
        LOG_P_BEST_NON_BASIC ← BASIC_LOG_PROBS[SYNSET].LOG_P_NON_BASIC +
        LOG_P_BEST_NON_BASIC
        CHILD_STATS[PARENT].CHILDREN.Add(LOG_P_BEST, LOG_P_BEST_NON_BASIC)
        if CHILD_STATS[PARENT].CHILDREN.COUNT == PARENT.CHILDREN.COUNT:
            if exists(PARENT) Q.put(PARENT)
            else DOWN_Q.put(SYNSET)
    BASIC_SYNSETS ← Dictionary()
    while not DOWN_Q.empty(): #Downward Pass
        SYNSET ← DOWN_Q.get()
        if SYNSET is Marked: BASIC_SYNSETS.Add(SYNSET)
        else for each CHILD of SYNSET: DOWN_Q.put(CHILD)
    return BASIC_SYNSETS
```

Figure 14: Pseudocode for Maximum Likelihood Tree Algorithm

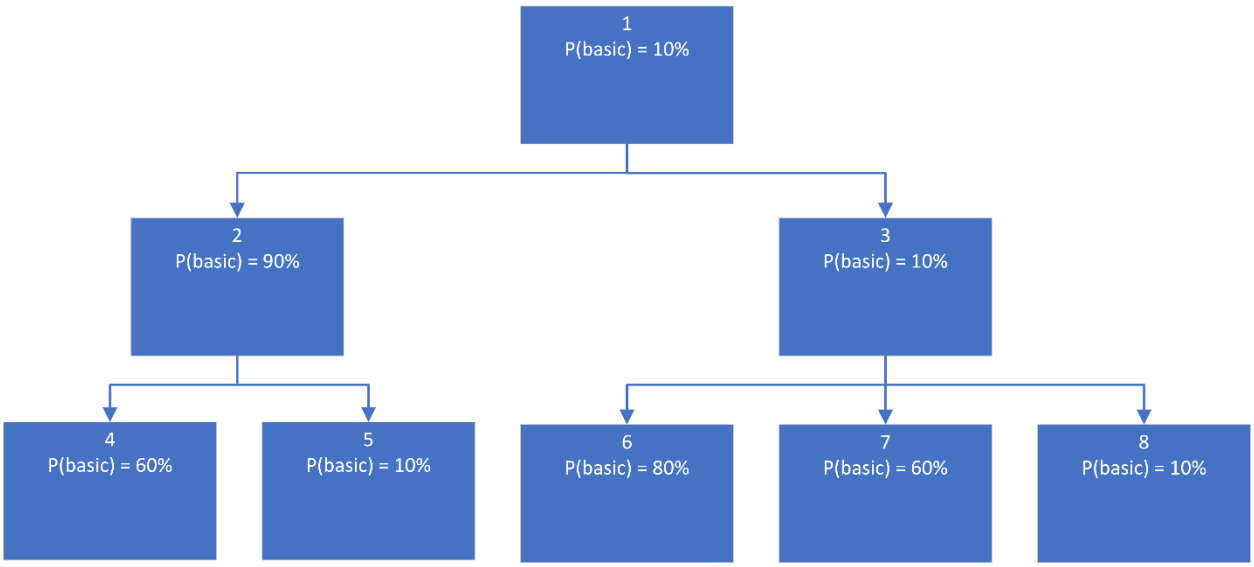

Figure 15: Maximum Likelihood Example Part 1, Classifier Probabilities

We start at the bottom of the hierarchy in Figure 16. Each leaf node is evaluated independently, though we represent the result of each in this single figure for brevity. A leaf node is taken as a subtree (of height one at this stage), and we calculate two probabilities and determine whether the node is *Marked* or not. The first probability is *P(best)*, which is the higher of *P(basic)* and *P(non_basic),* the latter which is calculated as *1 - P(basic)*. So in node 4, where *P(basic)* is 60%, *P(non_basic)* is 40%, and thus *P(best)* is the higher of those, 60%. The node is *Marked* since the highest-probability result is that the node under consideration is a basic-level category in this subtree. For *Marked* nodes at this point, *P(best)* + *P(non_basic)* will add up to 100% since *P(best)* is just *P(basic)*. In nodes where *P(non_basic)* is higher than *P(basic)*, like nodes 5 and 8, *P(best)* = *P(non_basic)* since the highest probability is that the node is not a basic-level category; thus these nodes are unmarked. *Marked* nodes are also highlighted in a lighter blue color.

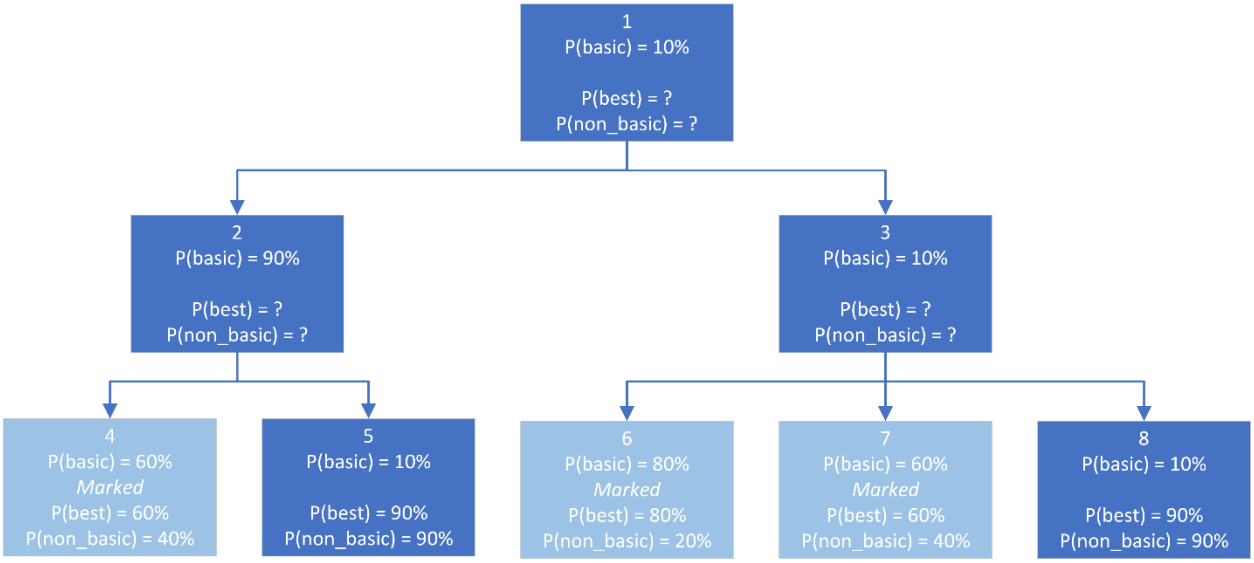

Figure 16: Maximum Likelihood Example Part 2, First Layer of the Upward Pass

Next, we consider the next layer of nodes up the tree in Figure 17. We include evaluations for nodes 2 and 3 in one diagram for brevity though they are evaluated independently. For node 2, we are considering the subtree rooted at node 2, including nodes 2, 4, and 5. We calculate *P(best)* and *P(non_basic)* for the node 2's subtree now, starting with its children in aggregate. *P(best)* for the children is 54% (60%*90%), while *P(non-basic)* for the children is 36% (40% * 90%). The two candidates for node 2's best option, then, are the best value for the children and node 2 being non-basic (54% * 10% = 5.4%) or the children all being non-basic and node 2 being basic (36% * 90% = 32.4%). The latter is the best option, so *P(best) = 32.4%* for the tree and the node is *Marked* since the best option so far is for it to be basic. The overall *P(non_basic)* for the tree is 3.6% (36% * 10%).

Node 3 is calculated in a similar manner. *P(best)* for the children of 3 is the product of each of their values of *P(best)*, or 43.2% (80% * 60% * 90%). *P(non_basic)* for the children is the product of each of the children's *P(non_basic)*, or 7.2% (20% * 40% * 90%). To find the best probability for node 3, we need to check the probability that node 3 is basic and its children are all non-basic against the probability that node 3 is non-basic and the best probabilities of all its children; these numbers are 0.72% (10% * 7.2%)

and 38.9% (43.2% * 90%), respectively. *P(best)* for the subtree, then, is 38.9% with node 3 being non-basic but having two basic children. *P(non_basic)* for the entire tree is then 6.5% (90% * 7.2%).

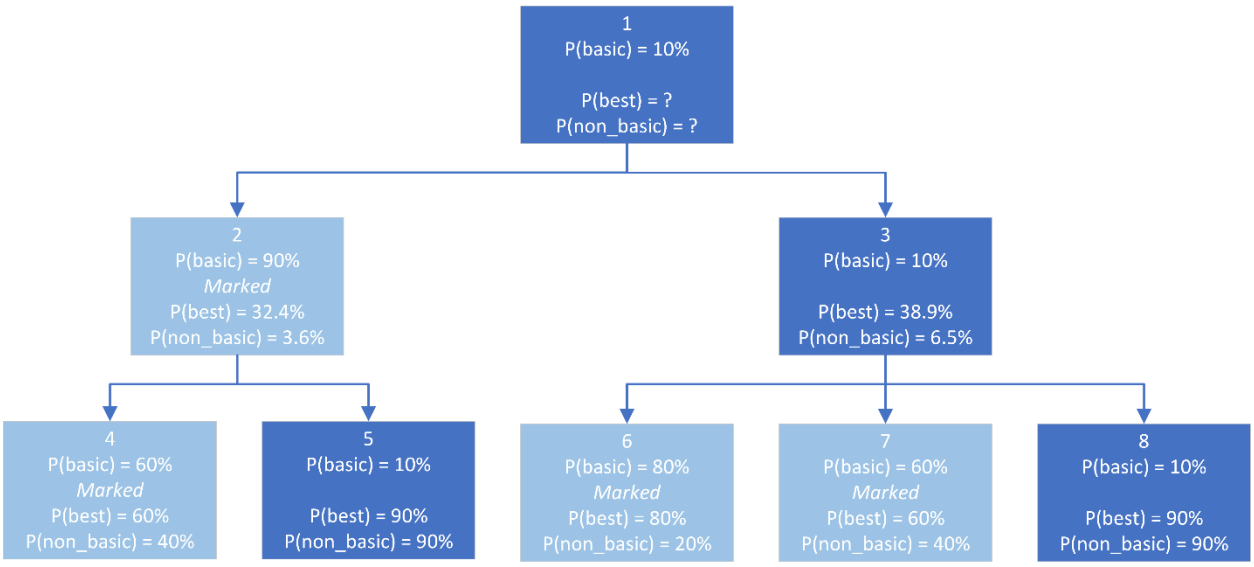

Figure 17: Maximum Likelihood Example Part 3, Second Layer of the Upward Pass

We repeat this procedure for the top node in the subtree we've been working with, node 1, in Figure 18. In this case the subtree has more than one level below it, but since we've retained *P(best)* and *P(non_basic)* for the entire subtrees of each of node 1's children, we only need to consider node 1 and its children in calculations for the subtree rooted at node 1. *P(non_basic)* for the children is 0.23% (3.6% * 6.5%), while *P(best)* for the children is 12.6% (32.4% * 38.9%). The two options for *P(best)* for the entire subtree are that node 1 is basic with probability 0.023% (0.23% * 10%), or that node 1 is non-basic with subtree probability 11.3% (12.6% * 90%). The latter is higher, so *P(best) = 11.3%* and the node remains unmarked since the better option was with node 1 not being labeled as a basic-level category. The overall *P(non_basic)* for the subtree is 0.2% (0.23% * 90%). This completes the upward pass computing all the probabilities needed and marking candidate nodes which could eventually be selected as basic-level categories.

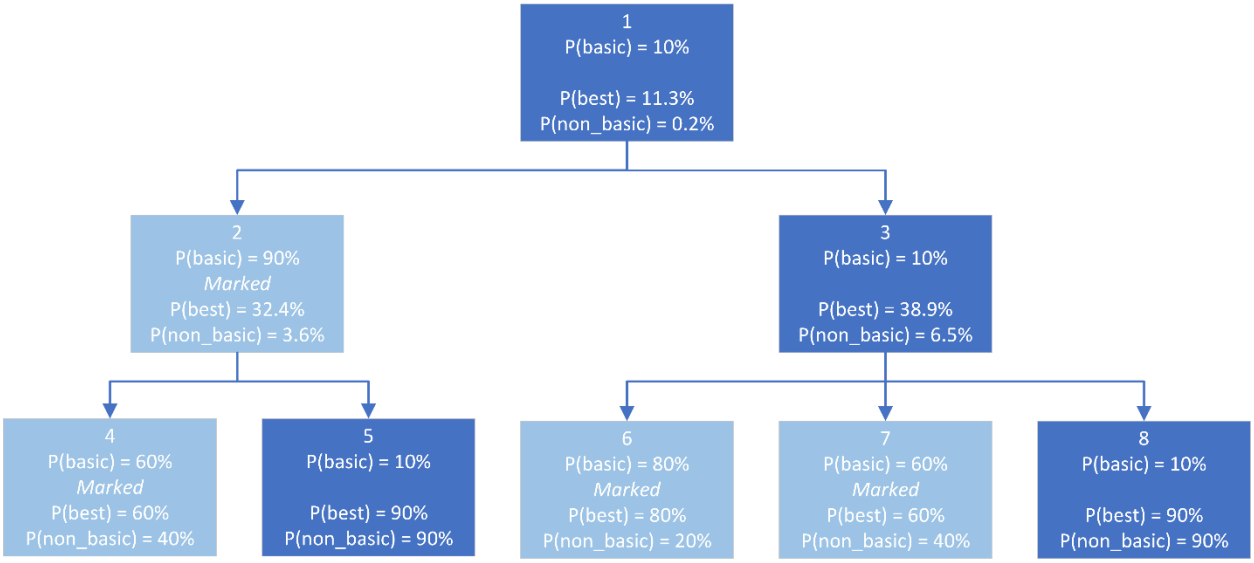

Figure 18: Maximum Likelihood Example Part 4, Top Layer of the Upward Pass

In Figure 19 we show the result of the entire downward pass, which is much simpler than the upper pass and involves no new computations. We traverse down the tree, branching as necessary, selecting the first *Marked* node in each branch as a basic-level category and stopping the traversal of that branch once one has been selected. We have highlighted the nodes selected as basic-level categories in green. Note there is one *Marked* node, node 4, which is not selected as a basic-level category. This is because node 4's parent, node 2, is also marked when including the context from node 4 and additional nodes in the tree, so when node 2 was selected as a basic-level category this is a more informed decision that essentially overrules the more local decisions below. In the algorithm we simply stop traversal down that branch at node 2, but node 2 incorporating more context is the reason why this works to produce the best possible combination.

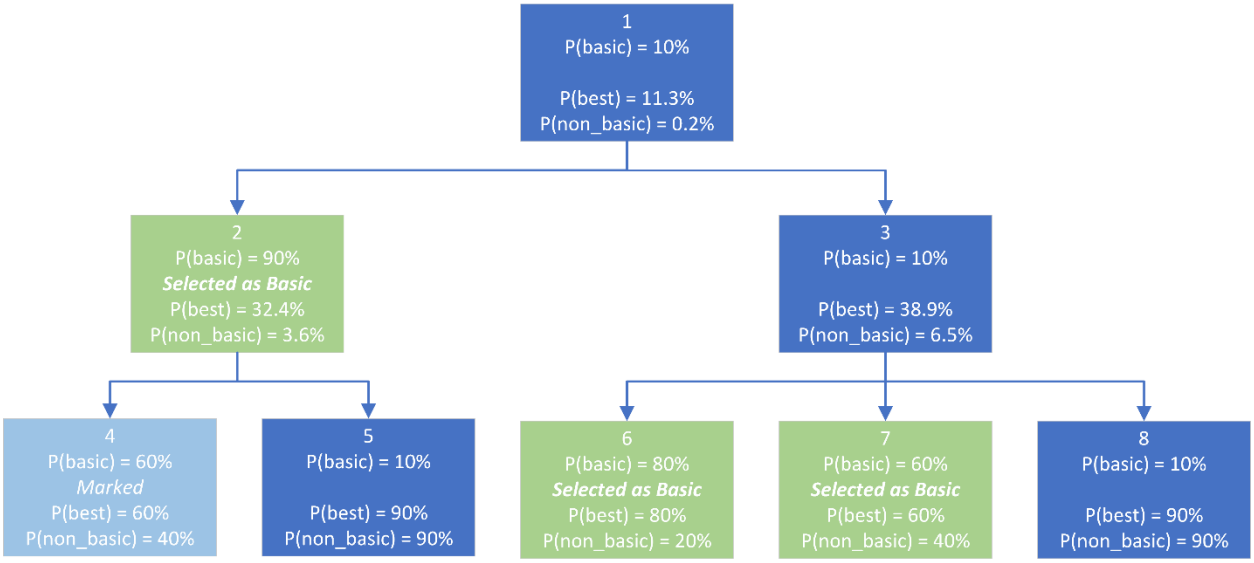

Figure 19: Maximum Likelihood Example Part 5, Downward Pass and Final Selection

This algorithm runs on the order of $O(n)$, where n is the number of nodes in the tree. Superficially it appears to run in $O(n \times c)$ where c is the number of children (hyponyms) per node, or slightly more precisely on the order of $O(n + \sum_{i=1}^{n} c_i)$, where c_i represents the number of children (hyponyms) for node i. However, at least under the simplified condition that each node has a single hypernym (which is true in trees but has a small handful of exceptions amounting to less than 2% of nouns in WordNet), each node is only a child of one other node and the sum of child counts will add up to $n - 1$, resulting in a linear $O(n)$ runtime.

5.3.3 Maximum Likelihood Relaxed with Minimum Threshold

In §5.3.2 we propose a relatively complicated tree traversal algorithm for finding the most likely combination of basic-level category predictions obeying the constraint that only a single basic-level category should be selected in each chain. Here we propose a modification to this algorithm based on observations on some of the mistakes it made on the development set during evaluation. Our classifier tends to assign higher probabilities to basic-level category candidates lower in the hierarchy and this leads to excess subordinates being treated as basic-level.

As a result, we propose an additional approach. We use the algorithm described in §5.3.2, but we also introduce an additional minimum probability threshold required for a node to be selected as basic-level and we *Mark* the node as basic-level if it is above that threshold even if it does not improve the overall *P(best)* for that node. Note, however, that we still follow the algorithm in §5.3.2 except that additional nodes may get marked along the way.

5.4 Evaluation and Discussion

Our combined system effectiveness, using each of the three combination strategies described in §5.3, is shown in Table 28.

Table 28: Classifier-based System Effectiveness with Different Second Stages

Combination Strategy	Precision	Recall	F-Score
Voting Rules (from Heuristic System)	50%	52%	0.511
Maximum Likelihood Estimate	54%	49%	0.512
Maximum Likelihood with Min Threshold	**56%**	**66%**	**0.607**

Our heuristic system, evaluated in §4.3.4 and listed in Table 15, achieves an f-score of 0.381 with 42% precision and 35% recall. By comparison, our Classifier-based system compares favorably under each of the combination strategies we evaluate, with better precision, recall, and f-score across the board.

Just having a Filtering Classifier, even using the second stage Voting Rules from the Heuristic System, helps tremendously with an f-score of 0.511 relative to the entire Heuristic System's 0.381. Using the maximum likelihood estimate approach described in §5.3.2, system performance only improves modestly with an f-score of 0.512, with slightly more gains in precision than loss in recall relative to the Heuristic System. The maximum likelihood approach modified with a minimum probability threshold, described in §5.3.3, performs the best with the highest precision, recall, and f-score. Recall in particular is substantially higher at 66% relative to the next highest at 52%.

Our original hope in building this minimum threshold into the maximum likelihood estimate was to choose a high-confidence threshold (e.g. 80%) at which we prefer higher nodes in the hierarchy to lower ones while still allowing the algorithm to work as intended for the lower probability predictions. However, in

tuning the parameter on the development set, we find the system performs better with a low setting of 20% as the minimum probability threshold. This gives us the ability to mark more nodes as basic-level higher in the tree despite a lower classifier prediction than we otherwise would; in order to be selected in a maximum likelihood approach, the probability would need to be at least 50%. So, the benefit of this modification does not come from taking a narrow selection of high-probability predictions and preferring higher nodes in the hierarchy on those. Rather, this achieves improved performance by increasing the recall of the basic-level category predictions and then using the second stage of the system preferring higher nodes in the hierarchy to correct mistakes lower in the hierarchy. This suggests further improving our ability to filter out subordinates or identify basic-level categories higher in the hierarchy with high confidence would be helpful at enabling a more principled maximum likelihood approach to be successful.

For this best system configuration, we present the system performance including accuracy in Table 1. While the f-score is 59% higher than our Heuristic System, the accuracy only improves from 88% to 91%.

Table 29: Full System Performance

Metric	Heuristic System	Classifier-based System
Accuracy	88%	91%
Precision	42%	56%
Recall	35%	66%
F-score	**0.381**	**0.607**

Our Classifier-based system is slightly more accurate than our Heuristic System; the f-score is much better for our Classifier-based System because it performs better on the less prevalent label subcategories. This is shown in **Error! Reference source not found.**, where the Heuristic System is a ctually slightly more accurate on the most prevalent subcategory, subordinates, while the classifier-based system is better on each other subcategory. The most substantial difference is in the critical basic-level subcategory, which corresponds to our positive labels and thus represents our system's recall.

One substantial difference between the Heuristic System and Classifier-based System is that the latter is trained using sample weighting while the former does not take sample weighting into account. When tuning the Heuristic System, however, we tune each rule to filter out as much as we can without filtering

any basic-level categories. This sets the decision boundary at the most aggressive parameter value feasible before the first basic-level category is filtered out. This will not change based on the way we weight samples together since our parameters only depend on the positive labels and any weighting within the negative labels will be irrelevant since more will be filtered out with more aggressive thresholds regardless of how we weight them together. Thus, while this is an apparent difference between the two systems, incorporating sample weighting while tuning the Heuristic System would not result in different performance; additionally, since sample weighting for the Classifier-based System only applies to the train data and not to the test data used to report results, this also does not lead to a discrepancy in the results reported.

Table 30: Full Classifier-based System Effectiveness by Subcategory

Subcategory	Heuristic System Accuracy	Classifier-based System Accuracy
Superordinate	83%	90%
Basic-level	35%	66%
Subordinate	98%	97%
None	73%	74%

Having looked at our system's overall performance and compared how different second stages affect that performance, we also evaluate the performance of our Filtering Classifier described in §5.2. For the purposes of this analysis, we use the binary output as to whether each synset is a basic-level category or not before any filtering by synset chain or by optimizing choices throughout the hierarchy. Incidentally, this is the same output as is fed into the Voting Rules in the variant of our Classifier-based System described in §5.3.1. We show the results of this subsystem in Table 31.

Table 31: Filtering Classifier Subsystem Effectiveness

Metric	Value
Accuracy	83%
Precision	35%
Recall	45%
F-score	**0.393**

While this classifier alone achieves an f-score better than our entire Heuristic System, this is primarily driven by recall with a lower precision and accuracy than the Heuristic System. The Filtering Rules of the Heuristic System, however, only achieve an f-score of 0.221, so comparing the respective subsystems does indicate an even stronger improvement from using a classifier instead of heuristic rules.

We break down the performance of this Filtering Classifier further in Table 32 by showing the accuracy on each of the label subcategories. Basic-level accuracy is the most important and the lowest, which provides additional motivation for using the minimum threshold modification to our maximum likelihood estimate combination approach to increase recall.

Table 32: Filtering Classifier Subsystem Effectiveness by Subcategory

Subcategory	Accuracy
Superordinate	93%
Basic-level	45%
Subordinate	87%
None	91%

Chapter 6

Error Analysis

Having described our system, we now provide some additional context to understand how it works, and particularly to describe the sorts of mistakes it makes in our final application of our system to the test set.

6.1 Filtering Classifier Feature Strength

Feature weights are not perfect indicators of how useful each feature is since it's their combination that is optimized. Additionally, some features may be more rare than others, or may have a large numeric range with smaller coefficients yet still having similar impact to a binary feature with a larger weight. Despite these sorts of concerns, looking at these weights still provides some indication about which features tend to have a stronger or weaker impact on the classifier output.

We find the following to be some of the strongest indicators of a synset being a basic-level category:

- The Brown Frequency is high
- The word is in the CHILDES corpus
- The synset is deep in the hierarchy
- It has a low frequency in the Brown Corpus

On the other hand, the following are strong indicators that the synset may not be a basic-level category:

- The word is capitalized
- The word is a plural
- The words frequently appear in other parts of speech
- The synset has many siblings without hyponyms
- The synset is not under *physical_entity.n.01*, *thing.n.08*, or *substance.n.01*
- All of the synset's hyponyms siblings have hyponyms
- The word has many vowels
- The word is long

- The word has a suffix

Additionally, while not a strong predictor, we include a bias feature in our classifier and this also received a score indicating that by default a synset should be predicted as not a basic-level category. The coefficient for this feature is about half the strength needed to qualify for the list of strong predictors.

In general, we find more features are strong indicators of a synset not being a basic-level category than we find as strong indicators of being a basic-level category. While this is true, note also that some negative features (e.g. long word length) could be reinterpreted as positive features for identifying basic-level categories (e.g. short word length).

Many of these strong indicators are straightforward to interpret and are consistent with the intent behind the features. One in particular stands out as unexpected: having a low frequency in the Brown Corpus is an indication that the category is a basic-level category. While this is not nearly as strong as the feature indicating a high frequency in Brown, we believe this relates to the fact that presence in the Brown Corpus at all is still an indicator of being basic-level while a word not appearing in the Brown Corpus is less likely to be a basic-level category.

6.2 False Negatives

We now discuss the synsets our system predicts are not basic-level categories but which actually are basic-level.

Of these, 78% received a predicted probability of being a basic-level category, according to the Filtering Classifier, of less than 20%. Using the Maximum Likelihood approach the probability would need to be at least 50% to be selected as a basic-level category, though our revised approach relaxes this down to allow categories with predictions as low as 20% to be selected as basic-level categories. However, most of our issues with recall occur with categories even below this very aggressive threshold.

On the other hand, only 13% of the false negatives have predicted probabilities above 50%, with each of these probabilities all happening to fall in the relatively high range of 79% to 87%. Over half of these high-probability cases are issues where a more inclusive synset was chosen with a lower probability due to our rule of aggressively choosing more abstract terms with moderate probability to avoid selecting too many

subordinates as basic-level; examples of these include choosing a moderate probability *seat.n.03* (51% probability of being basic-level) over *chair.n.01*, *sofa.n.01*, and *stool.n.01* with probabilities 79%, 85%, and 87%, respectively. Despite these errors and the clear room for improvement, this choice to choose superordinates in these cases is a net positive for the system and it only contributing to 7% of our false negatives reinforces this decision.

Since our errors come mainly from synsets with low prediction probabilities, we focus our error analysis on this set.

48% of these low-probability misses come from words that are compounds, including “stopwatch”, “pencil sharpener”, “bottle opener”, “pipe cutter”, and others. Many of these are small tools or appliances which are named in accordance with their function and appear to actually be basic-level categories with less straightforward names than most of those found in the psychology literature.

As an example, “pencil sharpener” is the synset *pencil_sharpener.n.01*, which has parent *sharpener.n.01* (“any implement that is used to make something (an edge or a point) sharper”), with other children including *grindstone.n.01*, *steel.n.03*, *strickle.n.01*, and *strop.n.01*. It seems unreasonable to expect a child to learn the more generic *sharpener.n.01* immediately from perceptual data, including grindstones and pencil sharpeners in the same category. This does appear to be an example where a compound word is a relatively clunky way to refer to a relatively simple everyday object like ***hammer***, ***saw***, and other first-level categories.

Our classifier struggles with these not just because they are compound words; there are a number of features that contribute to these looking unlike a basic-level category. We do include space-separated compounds, hyphenated compounds, and compounds consisting of a conjoined combination of other words in WordNet both to be features representing compounds. But beyond this, these terms also tend to have other properties that make it challenging to identify them as basic-level. Many of these are relatively long words, whereas basic-level categories tend to be short. Many of them end in suffixes like *-er*. And they tend to have a large number of vowels, going along with their overall length.

The train data, which is segmented as a different part of the WordNet hierarchy to avoid cheating, does include a small handful of examples of compound words that are basic-level, so this is not purely an issue of a difference between the train and test sets. For example, *spider_web.n.02* appears in the train set. Nonetheless, only 0.3% of the synsets in the train data that are compounds are basic-level categories; in the test set this is higher but still low at 1.2%.

Every single basic-level category we mistakenly failed to classify as basic-level either has a low Brown frequency or is not present in the Brown corpus at all. A high Brown frequency one of the strongest features predicting a synset should be considered a basic-level category; that said, many of our true positives do not have high Brown frequency, so while it's a missed opportunity this feature is not dispositive.

On the other hand, the false negatives are much more likely to be in the Brown corpus than true positives, with a low frequency ($p<0.01$), and this is a strong positive feature. The false negatives also are helped by having a lower maximum height ($p<0.01$). Despite these two features helping the basic-level categories we miss more than in the set we classify correctly, there are more and stronger features that hurt the false negatives than those that help them.

The false negatives have longer words ($p<0.01$), aren't substrings in subordinates as frequently ($p<0.01$), are much less frequently in the CHILDES corpus ($p<0.01$), and they contain more vowels ($p<0.05$).

These are all patterns expected of compound words, though the lower frequency in the CHILDES corpus and not having subordinates with substrings also apply to many of the other false negatives as well.

Of the non-compound false negatives with low predicted basic-level probabilities, tools are by far the most common comprising 85% of this set. Many of these are low-frequency or contain the suffix *-er*, including *gutter.n.01* and weeder.*n.02*. For most of these and the scattered non-tool errors, our system does not propose any of the synsets in paths through the false negative as basic-level; the low classifier score is a missed opportunity without competing with other errors from the classifier output.

Our test set contains a number of categories related to furniture, means of transportation, and tools; overall, both with compound words and with these low-frequency and suffix-containing words, tools are by

far the biggest source of our false negatives. The average value for the average height of our true positives is 1.9 while for our false negatives this is 1.5 ($p<0.01$). Having a lower average height tends to indicate a synset is not basic-level. While this does not stand out as a substantial contributor to misclassifications overall, the difference is even more pronounced for tools where the average height is only 1.2. Basic-level categories tend to have subordinates, but basic-level tools are an area where this is less commonly true. This is manifest in a number of features with smaller impact that add up, such as having fewer hyponyms, a higher average depth to height ratio, and being near the bottom of the hierarchy, all of which add up even though none of these features individually stands out as a major driver of the misclassifications on its own.

6.3 False Positives

We now discuss the synsets our system predicts are basic-level categories but where the actual label is not basic-level.

Since false positives include any non-basic-level label type, we show the breakdown of false positives by label type in Table 33. Interestingly, even though 'None' labels only comprise 11% of the test set, they comprise half of the false positives within this set. This comes at the expense of subordinates, which are under-represented in the false positive set.

Table 33: Label Subcategory Proportions in False Positives and Test Set Overall

Label Subcategory	Test Set	Test FPs
Superordinate	5%	8%
Basic-level	11%	n/a
Subordinate	73%	42%
None	11%	50%

Part of this discrepancy may be due to our weighting scheme described in §5.2.2; the proportions across sets are different since the sets are split based on common hypernym ancestors to avoid cheating as described in §3.9. Even so, the weights only apply during training, so the proportions in the train set are relevant to how the errors are traded off across label subcategories. In the train set, where subordinates are higher in frequency, 'None' labels still only correspond to 15% of the total after weighting. So, this

could affect the numbers somewhat but still doesn't explain 'None' labels comprising half of the false positives.

60% of our false positives are synsets with a low probability of being basic-level according to our Filtering Classifier, with low probability defined as less than 50%. Since our best combination strategy involves allowing lower-probability categories to be chosen to increase our recall, this sort of error is expected. While there is risk of a superordinate getting chosen over a basic-level category given the specific way we implement this strategy, we show in §6.2 that this is a rarity; on the other hand, this reduction in precision by aggressively proposing additional candidates as basic-level despite weak classifier probabilities is a problem but one that we expect as part of this trade-off.

With so many false positives coming from an aggressive threshold, we find that unlike the false negatives which are most heavily concentrated around tools, the errors are spread out across a much wider range as shown in Table 34.

Table 34: False Positive Breakdown by Superordinate Category

Category	Percent of FPs
implement	32%
tool	16%
musical instrument	14%
furniture	14%
vehicle	8%
part	8%
weapon	4%
abstract	4%

Implement is a relatively abstract term, so since it is the leading source of false positives we provide several examples: *lever.n.01*, *paintbrush.n.01*, *baton.n.01*, and *cane.n.01*. Tools also make this list of top errors in the second position. Overall, false positives are much less driven by a single subtree in the hierarchy than false negatives.

There are many statistically significant differences between the false positives and true negatives in terms of their feature values, but unlike the false negative case the broader patterns here are less clear as many of these cancel one another out.

The false positives actually receive more of a benefit than true negatives from average height, average depth, and average depth-height ratio features, but max depth and max height hurt them and this combined effect is 47% stronger by feature weight. All of these differences are statistically significant at $p < 0.01$ except for max height, which is significant at $p < 0.05$. For both max and average features, the false positives have lower height and higher depth, so these features trend in the same direction in these cases. Having so many related features facilitates our classifier assigning weights that counterbalance one another and the end result hurts the synsets that end up as false positives without a clearly interpretable explanation.

That said, the false positives being further down in the hierarchy is interesting; subordinates are the label subcategory with 42% of the false positives, but many of the 'None' labels also exhibit this property (e.g. *megaton_bomb.n.01*, which has no hyponyms).

Half of the false positives are on chains where no basic-level category is included. Of the other half with a basic-level category in the hierarchy, only 2% of the time is the basic-level category correctly given a higher score by the Filtering Classifier, indicating the problem is with the Filtering Classifier itself or our aggressive threshold used, rather than a result of our Combination Strategy. Additionally, the average predicted probability of being a basic-level category across all false positives is less than 50%, indicating our aggressive threshold is playing a substantial role in these mistakes.

We attribute the false positives to a general level of aggressiveness to improve recall, which spreads errors across many diverse examples. While there are some trends which can be observed across the two sets, they are difficult to interpret and no clear pattern of errors emerges for the false positives.

6.4 Error Analysis Summary

The Filtering Classifier is the source of most errors. For false negatives, these are concentrated around tools, where compounds, suffixes, and long words make many basic-level tool categories appear non-

basic-level. In the case of false positives, the pattern is less clear with a very aggressive threshold to improve recall driving a wide assortment of classification errors that do not follow patterns as clearly as the false negative case.

Chapter 7

Suitability in Applications

We have indicated several places in the literature where researchers attempting to solve a practical problem use a less well-developed attempt at identifying basic-level categories in the process of solving their more specific problem (Green 2006, Izquierdo et al. 2007, Lin et al. 2008b, Lin et al. 2009, Stephen et al. 2009). Nonetheless, we are approaching a novel task in identifying basic-level categories as a problem worthy of special attention itself. Since these other approaches are in the context of the researchers' own systems and approaches to their problems, we here provide some empirical evidence that having a set of basic-level categories would be useful for real-world applications.

In exploring the applicability of basic-level categories in real-world applications, we use our gold-standard labels rather than our system output. This is intended to demonstrate that knowing basic-level categories is helpful in real-world applications. Since our system has a moderate precision, we expect the gold-standard labels to provide a stronger signal than system output. Furthermore, since generating a large, broad-coverage set of labels is a core contribution of our work, covering 13.7% of all WordNet synsets compared with a biased 0.2% coverage with previously-available labels, this analysis is made possible by our work and these approaches could be used just with our gold-standard labels in many cases given their relatively broad coverage. We briefly confirm with one example in §7.1.2 that the gold-standard labels provide a stronger signal than the system output, but also show that the system output is a useful signal in an application; elsewhere, all numbers reported are using our gold-standard labels.

7.1 Automatically Measuring Text Readability

7.1.1 Introduction

As children develop their language skills they can read more complicated texts. Texts tend to target specific audiences. Determining whether a reader is capable of reading a particular text is useful for a wide range of applications, from displaying search results to recommending a book to buy. Readability also extends beyond children to adults as well. A straightforward example of this is foreign language

learners, but people with specialized knowledge are also able to read texts that those without that domain knowledge would not be able to comprehend. Nonetheless, most work on measuring text readability focuses on levels of language development that are more closely related to childhood language development and second language learning.

7.1.2 Basic-Level Categories as a Useful Signal for Text Readability

We now show that our basic-level category data is a useful signal in measuring text readability. We do so by using a standard readability dataset to show that the presence of basic-level categories in the text correlates with reading level as well as some of the most widely-used features in the field. We also show it correlates substantially better than a widely-used alternative word list containing commonly-understood words.

For our dataset, we use the Common Core Appendix B corpus (National Governors Association Center for Best Practices 2010). This is a collection of texts by grade level, grouped into two to three year buckets (K-1, 2-3, 4-5, 6-8, 9-10, and 11-12). The collections include different types of texts relevant to different ages, including stories, poetry, informational texts, and drama. Some texts are referenced but the majority of the document consists of texts reproduced as examples of appropriate texts for the relevant grade levels. We take all of the texts reproduced in this corpus except the poetry, which we exclude. We exclude poetry not primarily because it is not representative of normal text, but rather because the number of words per sentence is one of the best and most widely-used metrics used for computing the readability of a text and including a text type without clear and regular sentence boundaries would artificially make our approach seem better by comparison.

We tokenize these texts and part-of-speech tag them using NLTK (Loper et al. 2002). We determine the number of syllables in each word using the CMU Pronouncing Dictionary (Weide 1998). This enables us to calculate the two most widely-used features for assessing reading level, as discussed in §2.8: the number of words per sentence and the number of syllables per word.

Both of these show a strong positive correlation by grade level. A least squares best fit line on words per sentence by reading level as shown in Figure 20. Syllables per word, shown in Figure 21, still shows strong positive correlation but is not quite as strong as words per sentence.

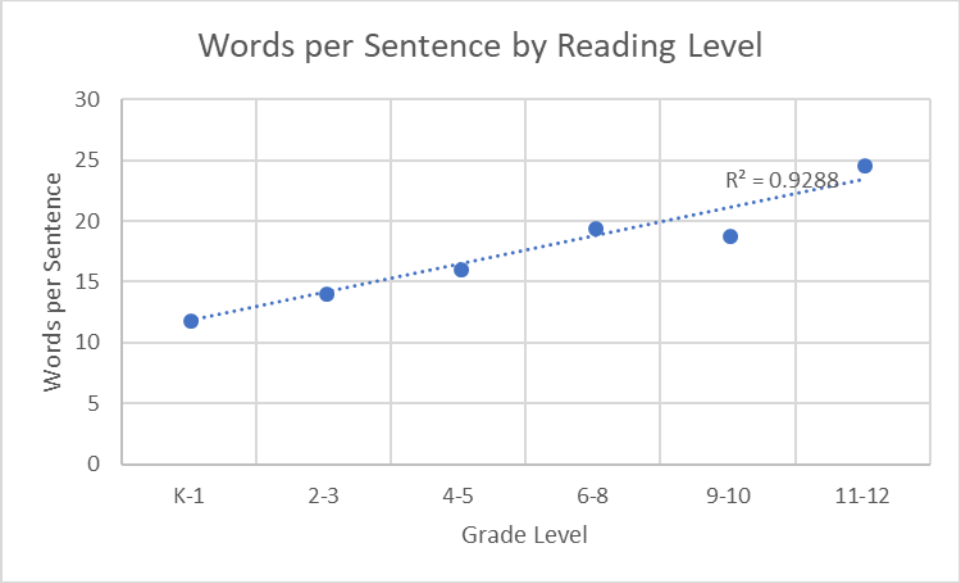

Figure 20: Average Number of Words per Sentence by Reading Level

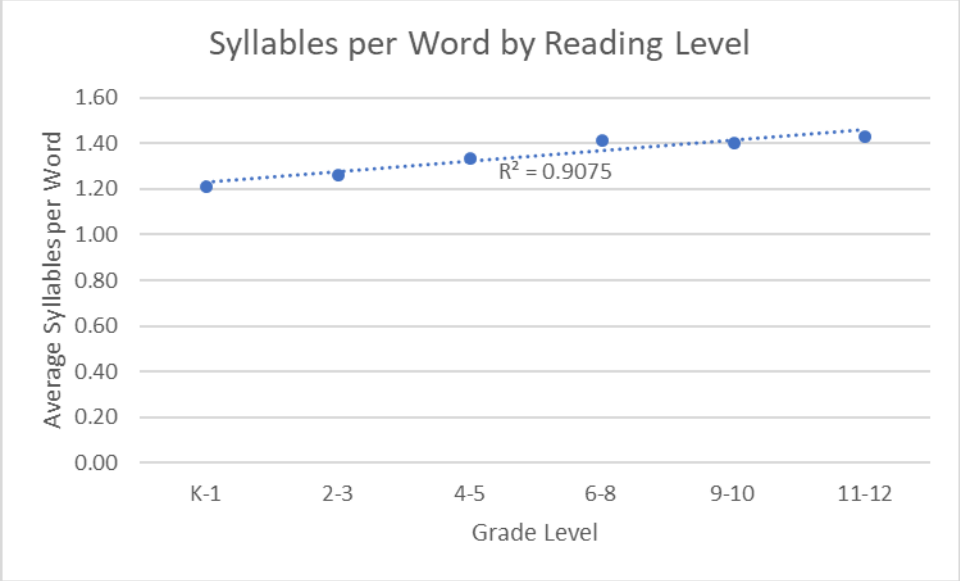

Figure 21: Average Syllables per Word by Reading Level

Basic-level categories are expected to be more common in text at lower reading levels. In Figure 22 we confirm this, showing that texts at lower reading levels tend to have a higher percentage of basic-level nouns. The number we show here is the percent, out of common noun tokens in the text we are able to associate with a label, that are basic-level. We exclude proper nouns based on the part-of-speech tags, as well as any common nouns we are not able to associate with a positive or negative label. This shows another strong correlation, with the lowest grade levels having the highest percentage of basic-level categories while the highest grade levels have the least.

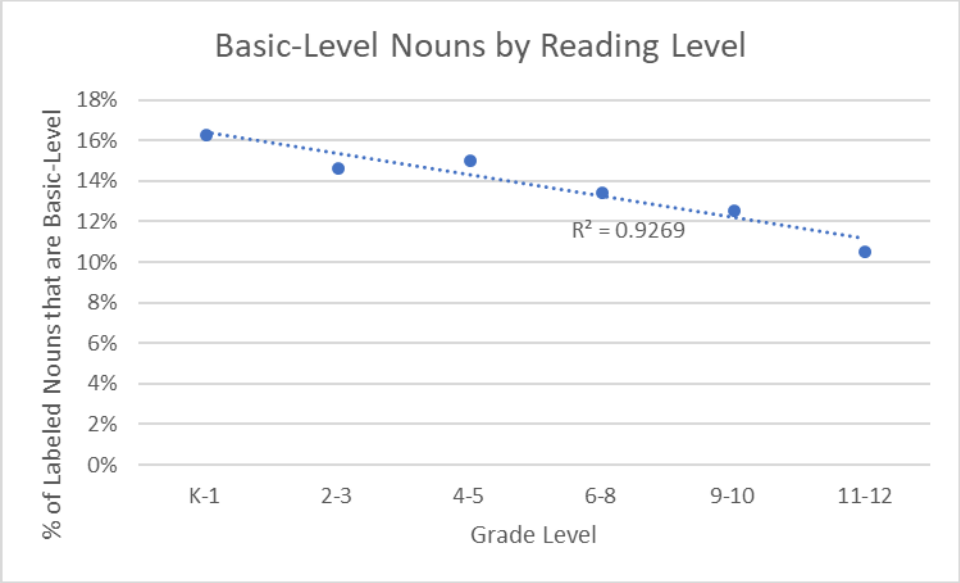

Figure 22: Basic-level Common Nouns as a Portion of Labeled Common Nouns by Reading Level

Since readability systems sometimes use other word lists as a feature, and the Dale3000 list is a widely used list, we also show the correlation of words on this list by grade level in Figure 23.

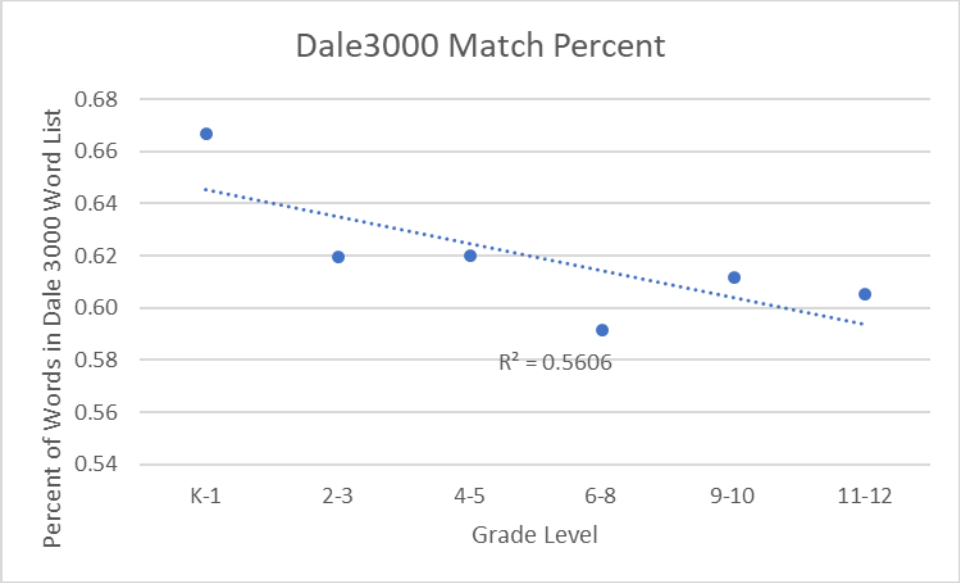

Figure 23: Portion of Words on the Dale3000 List by Reading Level

This list does exhibit the pattern, like basic-level categories, that more proportionally words on the list tend to appear in texts at lower grade levels. However, this correlation is weak.

We show the correlation coefficients of each of these features by grade level in Table 35. Words per sentence narrowly beats basic-level categories as having the highest Pearson correlation coefficient and R^2 value. Basic-level categories beats the other widely-used feature, syllables per word in correlating with grade level. The Dale3000 list matches, on the other hand, are only loosely correlated; basic-level categories perform much better than this alternative word list.

Table 35: Correlation with Grade Level by Readability Feature

Feature	Pearson Correlation Coefficient	R^2
Words per Sentence	**0.964**	**0.929**
Basic-Level Categories (Gold-Standard Labels)	0.963	0.927
Syllables per Word	0.953	0.908
Dale3000 Matches	0.749	0.561
Basic-Level Categories *(System Output)*	0.919	0.845

We also include a line in Table 35 showing the value of basic-level categories predicted by our system output rather than our gold-standard labels. This system output performance is not as strongly correlated with grade level as our gold-standard labels are, performing worse than both words per sentence and syllables per word features. Nonetheless, this still outperforms the alternative Dale3000 list by a wide margin. The R^2 of basic-level categories using our system output is 7% worse than that of the syllables per word feature, but the Dale3000 list is 38% worse than syllables per word by that same measure. This shows that our system output is also a useful signal for measuring text readability relative to an alternative broadly-used word list. This may be particularly useful in cases where our gold-standard labels aren't a sufficiently-large set or where broad coverage of English is required for integration in a system, and it also indicates an opportunity for higher performance if we improve our system in the future.

While this does not go so far as to implement a state of the art system and show basic-level categories provide incremental value, this does indicate that basic-level categories are a relevant and interesting signal that should be considered for applications in this area. Using basic-level categories provides signal on par with some of the most widely-used features in this space and substantially outperforms a widely-used competing list.

7.1.3 Readability: Wikipedia vs. Simple Wikipedia

Having shown our own correlation between the portion of documents that are basic-level categories and the reading level, with a best-fit line to enable us to predict reading level from the portion of basic-level categories, we now compare our predictions to a published study on the reading level of Wikipedia compared to the reading level of Simple Wikipedia. Simple Wikipedia is a simplified version of Wikipedia with generally shorter sentences and a restrictive vocabulary with the intention of making the text easier to read than the full version of Wikipedia.

Lucassen (2012) applied the Flesch Reading Ease Test (Flesch 1979) to both Wikipedia and Simple Wikipedia, showing that using this standard measure of text readability in fact Wikipedia is harder to read than Simple Wikipedia. Finding that many Wikipedia documents are very short and this skews the numbers substantially, Lucassen (2012) first filters out documents having no more than five sentences. They then compute the Flesch Reading Ease scores, arriving at a score of 61.69 for Simple Wikipedia

and 51.18 for full Wikipedia. These scores, subdivided into 10-point ranges, map to grade levels or buckets with multiple grade levels. Since the grade groupings in the scale do not match the groupings in the Common Core Appendix B corpus we used for our readability work (National Governors Association Center for Best Practices 2010), we use linear interpolation within buckets with multiple grade levels across both datasets to pick a single grade most closely associated with the score.

We repeat a similar procedure for determining the proportion of basic-level categories in documents across the two sets. We filter documents by the number of tokens rather than the number of sentences, though, removing corresponding documents where either Simple Wikipedia or full Wikipedia has less than 75 tokens. We then map the portion of basic-level categories in each set to a grade level using the best-fit line in Figure 22.

We report the grade levels across both sets, according to both the reported Flesch Reading Ease scores (Lucassen 2012) as well as our own approach using basic-level categories, in Table 36.

Table 36: Reading Level Predictions for Wikipedia and Simple Wikipedia

Reading Level Prediction Method	Simple Wikipedia	Full Wikipedia
Flesch Reading Ease	grade 8	grade 10
Basic-level categories	grade 8	grade 11

Using only the portion of labeled nouns that are basic-level, we predict the exact same grade level for Simple Wikipedia and within one grade of the more widely-used test for Full Wikipedia. While our method predicts the two corpora are slightly further apart than the widely-used Flesch Reading Ease Test, this alignment between the two is striking.

Additionally, while we map the portion of basic-level categories to a grade level, the portion itself (12.5% in Simple Wikipedia and 11.2% for full Wikipedia) is statistically significant with $p < 0.01$.

7.1.4 Readability Conclusions

We find that using a simple measure, the portion of basic-level categories among common nouns in a document with category predictions, correlates as well with reading level as other leading features used in more widely-used methods. We additionally predict the reading levels, using this measure alone, of

Simple Wikipedia and the full Wikipedia, showing that our predictions align well with existing research on this subject. This suggests that basic-level category data could be a useful signal for systems built to predict the readability of a text.

7.2 Image Captioning

7.2.1 Introduction

Basic-level categories may be particularly relevant for natural language grounding. Since basic-level categories describe those concepts which are learned directly from perceptual data and are frequently the terms used to refer to objects absent a context requiring a more specific or generic term, we believe systems combining language and perceptual data could benefit from a dataset or system which indicates which words are basic-level categories.

We consider an image captioning dataset, which is used by systems that learn how to write captions for novel images, and show that image captions contain substantially more basic-level categories than normal text represented by the readability analysis in §7.1. This suggests that basic-level category information could be a useful input into these systems, as well as others which combine perceptual and language data.

7.2.2 Analyzing Image Caption Data

We use COCO, a dataset consisting of images and metadata, including five captions for each image (Lin et al. 2014). This data includes 82,783 images with 99.8% of these having exactly five captions, with the remaining 0.2% having six or seven captions associated with them.

a bicycle lays against a rock by the ocean
a yellow bike that is behind a rock on the beach
a bike parked up against a rock on a beach
a bicycle is parked against a rock at the beach
a bike sits parked next to a giant rock

Figure 24: Example Image with Captions from COCO Dataset

An example is shown in Figure 24, where an image from the COCO dataset is shown with its five captions.

We tokenize these texts and part-of-speech tag them using NLTK (Loper et al. 2002), and calculate, as in §7.1.2, the portion of common nouns we can associate with a label where the label indicates it is a basic-level category. We compare our results to the previous text readability corpus and our Wikipedia experiments in Table 37 to show how it compares.

Table 37: Basic-Level Categories by Corpus

Corpus	Subset	Basic-Level Proportion
Common Core Appendix B	K-1	16.4%
Common Core Appendix B	2-3	14.7%
Common Core Appendix B	4-5	15.1%
Common Core Appendix B	6-8	13.5%
Common Core Appendix B	9-10	12.6%
Common Core Appendix B	11-12	10.6%
Wikipedia	Simple Wikipedia	12.5%
Wikipedia	Full Wikipedia	11.2%
COCO	Image Captions	**30.2%**

We find that the portion of common nouns that are basic-level is much higher in the image captions than any of the values from either of the readability corpora we consider. From a readability perspective, this number is too high to associate it with a grade level in school, being nearly double the value of even the earliest grades where students are just learning to read.

Table 38: Basic-Level Proportion for Overlapping Nouns

Noun Grouping	Basic-Level Proportion
All	30.2%
Nouns appearing in 2+ captions	31.5%

We further show in Table 38 that nouns which appear in multiple captions for the same image tend to be basic-level more often than in the overall set of captions (significant at $p < 0.01$). All of the standard metrics for automatically evaluating novel captions against the COCO reference set incorporate the overlap in words (and sometimes n-grams) across the reference captions and the candidate caption (Chen et al. 2015). While the difference in percentage is not extreme like comparing image captions to normal text, the large dataset enables us to show that this difference is nonetheless statistically significant. Given that word overlap is a critical component of evaluating systems, and basic-level categories are even more common in nouns that overlap across multiple captions, this further reinforces that basic-level categories could potentially help systems in choosing which noun to use in a caption.

This data indicates that basic-level categories may be particularly relevant for this area of application, where objects are being identified and named. Since Rosch et al. (1976) show that basic-level categories tend to be the terms used to describe objects absent a more specific context requiring otherwise, and since we find these terms to be more frequent in image captions than other texts, we believe the use of basic-level categories as an input to image captioning systems could potentially help generate more relevant captions.

Chapter 8

Conclusion

We approach the problem of identifying basic-level categories for the first time as an independent task worthy of a standard dataset and rigorous evaluation. Previous work has been done in psychology experiments on small numbers of words and subjects to show basic-level categories have theoretically interesting properties. Several applied systems have attempted to use data from these experiments to identify basic-level categories and use them in real-world applications, getting value from basic-level categories but without a way to effectively evaluate the identification itself except insofar as it helps in an applied task.

We use crowdsourcing to build a broadly-representative dataset for basic-level categories. We rely on psychology experiments to provide properties we can use to elicit accurate responses from annotators and use WordNet as a resource providing a strong hierarchy of hypernyms and hyponyms to align the task to the framework of superordinates, basic-level categories, and subordinates used in the field of psychology. We find that despite this theoretically well-grounded framework, the salient examples used in the psychology experiments omit the possibility of some chains of hypernyms/hyponyms not containing any basic-level categories, and also that there are often intermediate categories between the more salient ones which make the annotation process more difficult than anticipated. While we use a very targeted labeling process asking annotators to choose the basic-level category (or none) in a chain up the WordNet hypernym/hyponym hierarchy, with multiple levels of agreement required on individual chains and across multiple chains through the same synset, we also train two expert annotators on the binary task of identifying whether or not a concept is a basic-level category. We show that this is a challenging task but that annotators are still able to achieve substantial agreement.

Despite these challenges, our dataset increases the number of available labels by a factor of 72, from 155 to 11,221. In the process, though unsurprising given the shape of the WordNet hierarchy, we also identify that there is a much stronger label bias toward subordinates rather than the more balanced combination of superordinates, basic-level categories, and subordinates found in the previously-available

labels. We provide a standard division of these categories into a train, development, and test set for any future researchers who are interested in working on this problem, and we use this division for our subsequent experiments.

Using our dataset and the fraction of the overall WordNet hierarchy we label with any category labels, we can extrapolate to estimate that there are a total of 1,888 basic-level categories total in WordNet.

Based on and extending the ideas generated by students in a class project, we develop a Heuristic System to identify basic-level categories in WordNet. Using dozens of Filtering Rules to filter out WordNet synsets unlikely to be basic-level based on simple properties like the length of the word and the height of the synset in the WordNet hypernym/hyponym hierarchy, we arrive at a set of candidate basic-level synsets. We then apply another set of over a dozen Voting Rules to identify, in each chain up the hypernym/hyponym hierarchy, which synset is the best one to choose as basic-level, if any. This system achieves an f-score of 0.381, with recall being the most challenging factor. We show that applying our Heuristic System to previously-available data achieves an f-score of 0.749, which shows that our more broadly-representative label set is much more challenging than the salient examples used in psychology experiments. We believe this more representative dataset will be helpful in enabling more effective systems in practice.

Given the increased availability of data, we next build a Classifier-based System, training a logistic regression classifier using dozens of numeric and binary features based on the rules we use in our Heuristic System. We then take the output probabilities of this classifier and consider several ways to combine them into an optimal set of proposed basic-level categories. This includes a novel dynamic programming tree algorithm to make the calculation of the constrained overall maximum likelihood for the tree computationally tractable in linear time using two passes through the tree, where the constraint is that no more than one synset in any path up the hypernym/hyponym hierarchy may be proposed as a basic-level category. We settle on a relaxed version of our constrained maximum likelihood estimate; we relax this by encouraging lower probabilities to be selected and giving hypernyms additional preference over hyponyms to improve recall based on error analysis. This combined Classifier-based System achieves an f-score of 0.607, a relative 60% improvement over the Heuristic System.

We describe our errors in detail, showing opportunities for improvement given the f-score we report is still not extremely high. Our false negatives are concentrated around tools, which often use compound words to denote simple tools. Additionally, these tools tend to end in the suffix *-er*, while suffixes are often a sign that a category is not at the basic-level. The tools are also hurt by having sparse hyponyms, with many basic-level categories in this portion of the hierarchy not having hyponyms despite the presence of hyponyms being common across much of the rest of the hierarchy. As to false positives, we do not observe the same strong local properties but rather determine the errors are scattered across the spectrum as a result of our aggressive thresholding to increase recall and maximize f-score.

We next show that this work we have done to identify basic-level categories appears interesting for use in a range of applications. We show that the readability of a text correlates with the proportion of basic-level categories in the text on a standard corpus used in text readability research. We apply our best-fit line approximating reading level based on the proportion of basic-level categories in a text to the problem of measuring the readability of full Wikipedia and Simple Wikipedia. Using just this one feature, we show that Simple Wikipedia appears to be at an 8^{th} grade reading level while full Wikipedia is at an 11^{th} grade reading level; this finding closely matches existing research in text readability, where Simple Wikipedia is also projected to be at an 8^{th} grade reading level while full Wikipedia is measured at a 10^{th} grade reading level.

Additionally, we consider an application to image captioning. Using a standard dataset for image captioning, we show that the proportion of basic-level categories in text describing images is substantially higher than in normal text, suggesting that systems to generate automatic image captions could benefit from knowing which words represent basic-level categories.

Chapter 9

Future Work

Despite this progress, there is still plenty of room for improvement. In future work, we would like to explore other ways to improve recall, particularly with improvements in the Filtering Classifier by exploring additional features. Since frequency and early childhood language data are both very strong signals, we could explore larger corpora for frequency information and find additional early childhood development language data. We would also like to explore using multi-class classifiers for each label subcategory instead of only using binary classification as another way to potentially improve classifier effectiveness. Improving recall in this subsystem could enable us to use less aggressive thresholds in our Combination Strategy and thus mitigate our largest source of imprecision.

We would like to extend this work to other parts of speech beyond nouns, particularly to verbs, adjectives, and adverbs. While we are not aware of substantial work on basic-level categories in other parts of speech, one property Rosch et al. (1976) identified about basic-level categories is that they are the most inclusive categories sharing many attributes in common. To the extent awareness of attributes is present as children form their first categories, we may be able to find a basic-level for adjectives as well. Klibanoff et al. (2000) find that basic-level categories are useful to children forming concepts of adjectives, and we suspect using the work on basic-level noun categories could help us extend the work to other parts of speech including adjectives.

Lemaitre et al. (2013) shows that the sounds resulting from actions appear to be arranged around a basic-level while Van Dam et al. (2010) show that the concreteness of action verbs (with reference to a potential basic-level) is reflected in the neural response to action verbs; these both suggest the work could also potentially be extended to verbs. Additionally, nouns are learned before verbs (Gentner 1982) just as before adjectives, which suggests knowing the basic-level for nouns could also potentially help with verbs as it does for adjectives. While we are not yet aware of strong evidence for a basic-level in adverbs, by extension we expect it may exist in this remaining open-class part of speech as well.

In pursuing extending this work to other parts of speech, we also hope to help resolve the cross-classification complexities in WordNet. By providing a richer set of relations on how categories are related to one another across parts of speech based on the way concepts are formed, it may be possible using selectional restrictions and multiple parts of speech to resolve the sorts of issues we discuss in §3.4 with ***freestone*** and ***clingstone*** where semantically closely-related categories have artificially large distances between them in the hypernym/hyponym hierarchy due to overlapping referents.

We also hope to extend this work beyond the basic-level. We have noted in §3.2 that abstruse categories often tend to appear between a basic-level category and its superordinate and/or subordinate in the hypernym/hyponym hierarchy. The order we form concepts is not entirely the same as the order later organize them, a distinction Salmieri (2017) describes as leading to distinct epistemic and taxonomical hierarchies of concepts, where the epistemic hierarchy describes which concepts are learned before others while the taxonomic hierarchy shows the logical organization of all fully-formed concepts as in WordNet. Identifying basic-level categories is a first step in building an epistemic hierarchy, which ideally could be built by adding additional relations between synsets in WordNet based on the required or typical order of related concepts being learned. We believe that building these relations, even if just for the basic level and several layers out, could make distance measures within WordNet far more meaningful than distance measures through the existing taxonomic hierarchy which includes many abstruse concepts along many paths between common words.

An additional opportunity for future work is in extending this work across languages and cultures. Languages and cultures divide the world up differently, and it would be interesting to see the extent to which the basic-level exists and is shared across languages. Rosch et al. (1976) provided some evidence for cross-linguistic applicability of basic-level categories, and other studies have shown a similarity in groupings of even continuous variables like color into similar categories across cultures (Roberson et al. 2005). The work on first-level concepts in epistemology has also argued that although the specific order individuals learn particular concepts may theoretically be able to vary based on being exposed to extremely different circumstances (e.g. a child being raised in a furniture store and making distinctions between different types of tables first), these sorts of distinctions require much greater mental effort

without having simpler concepts (e.g. table) to build from and thus even if it is possible to learn them in a different order the eventual logical taxonomy including the simpler concept would result in the simpler concept being re-evaluated as first-level (Rand 1990) and thus likely having the basic-level advantages described in Rosch et al. (1976). However, these claims have not been validated with psychological experiments and this field generally is under-studied. On the other hand, other studies have more generally shown differences in childhood language development across languages (Choi et al. 1995). Since this is not a widely-studied problem, determining the extent to which basic-level categories are shared using broadly-representative datasets would be an interesting line of future work. We could specifically investigate the extent to which culture and language affect whether a particular category is basic-level, focusing in part on whether cultures with very different objects in their everyday life end up having basic-level advantages at different levels in the hierarchy for the objects more and less common in their respective cultures. Understanding these issues better could also help aid the development of multi-lingual WordNets.

Applying our system to different languages could pose some interesting challenges. While the classifier-based approach would ideally facilitate scaling across languages relatively easily, this would be expected to work better in some languages than others. While the features used in our system may translate well across languages similar to English, valuable features like word length may not be as valuable in logographic languages. Here also words that include their hyponyms as substrings may be difficult. Looking at the complexity of the character and shared components could regain some of these patterns, but the system may not be able to perform as well and additional features could be needed to achieve similar performance. In languages with less taxonomic depth, some of the features regarding height and depth may not be as effective, and this could require a language-specific model being trained rather than using a general-purpose model across languages. Additionally, tokenized corpora would be needed in languages where we apply these techniques since word frequency is a very strong feature for identifying basic-level categories.

While our system may generally perform best on languages more similar to English without additional feature engineering work, one alternative strategy could be to attempt automatic translation. Basic-level

categories are typically simple words denoting perceptual objects; these often have one-for-one mappings to words in other languages, while more complex categories can be very difficult to translate. This suggests translation of an English basic-level category list could potentially provide a strong starting point for identifying basic-level categories in other languages.

We would also like to extend this work to applications, particularly in the area of natural language grounding. Basic-level categories are learned directly from perceptual data and appear to be the simplest concepts, while also being the terms people use to refer to objects absent context conditioning the need for more inclusive or specific terms. This seems well-aligned to helping with language grounding and robotics applications. We show that basic-level categories appear to be interesting signals for image captioning, and we hope that it could also be helpful for systems of human-computer interaction relating to objects in the real world.

Appendix A Rosch-Markman Labels

Table 39: Labels from Rosch et al. (1976) Experiments 1-2

Superordinate	Basic	Subordinate (2 columns)	
musical instrument	guitar	folk guitar	classical guitar
	piano	grand piano	upright piano
	drum	kettle drum	base drum
fruit	apple	delicious apple	mackintosh apple
	peach	freestone peach	cling peach
	grapes	concord grapes	green seedless grapes
tool	hammer	ball-peen hammer	claw hammer
	saw	hack hand saw	cross-cutting hand saw
	screwdriver	phillips screwdriver	regular screwdriver
clothing	pants	levis	double knit pants
	socks	knee socks	ankle socks
	shirt	dress shirt	knit shirt
furniture	table	kitchen table	dining room table
	lamp	floor lamp	desk lamp
	chair	kitchen chair	living room chair
vehicle	car	sports car	four door sedan car
	bus	city bus	cross country bus
	truck	pick up truck	tractor-trailer truck
tree	maple	silver maple	sugar maple
	birch	river birch	white birch
	oak	white oak	red oak
fish	bass	sea bass	striped bass
	trout	rainbow trout	steelhead trout
	salmon	blueback salmon	chinook salmon
bird	cardinal	easter cardinal	grey tailed cardinal
	eagle	bald eagle	golden eagle
	sparrow	song sparrow	field sparrow

Note that although proposed and presented like this (Rosch et al. 1976), the examples relating to ***tree***, ***fish***, and ***bird*** in Table 39 are found to be basic-level in Rosch et al. (1976). Thus, reflecting the experiment output, we shift these labels down one level (superordinate to basic-level, basic-level to subordinate) in coding these as labels in our system as described in §3.1, also adding corresponding new superordinates ***plant***, ***animal***, and ***animal*** to correspond to ***tree***, ***fish***, and ***bird***.

Table 40: Labels from Rosch et al. (1976) Experiments 3-4

Superordinate	**Basic**
clothing	shoes
vehicle	airplane
	motorcycle
animal	cat
	dog
	butterfly
furniture	sofa
	bed

Table 41: Labels from Markman et al. (1997) Experiment 1

Superordinate	**Basic**	**Subordinate**
vegetable	beans	green beans
vehicle	bus	school bus
musical instrument	guitar	folk guitar
weapon	gun	shotgun
jewelry	necklace	pearl necklace
footgear	shoes	sandals
exercise equipment	weights	Nautilus weights
sports equipment	ball	football
clothing	pants	jeans
office equipment	paper	typing paper
kitchen utensil	plate	dinner plate
tool	saw	chainsaw
camping equipment	tent	pup tent
animal	dog	poodle
reading material	novel	mystery novel
beverage	milk	skim milk

entertainment	movie	horror movie
food	potatoes	mashed potatoes
furniture	table	kitchen table
plant	tree	pine tree

Table 42: Labels from Markman et al. (1997) Experiment 2

Superordinate	**Basic**
clothing	tie
	scarf
musical instrument	trumpet
	saxophone
weapon	sword
	spear
vehicle	bus
	truck
furniture	bed
	couch
reading material	magazine
	newspaper
kitchen utensil	spoon
	fork
human dwelling	apartment
	hotel
tool	screwdriver
	drill
beverage	coffee
	tea
fruit	apple
	pear
vegetable	onion
	radish
animal	horse
	cow
insect	ant
	termite
bird	robin
	canary
disease	measles
	chicken pox

Table 43: Labels from Markman et al. (1997) Experiment 3

Superordinate	Basic	Subordinate
vehicle	bus	school bus
		city bus
	truck	fire truck
		semitrailer truck
	car	limousine
reading material	novel	mystery novel
jewelry	necklace	
	ring	
	watch	
sports equipment	ball	
	racquet	
	net	
beverage	milk	
	soda	
	alcohol	
musical instrument	guitar	
	piano	
	drum	
clothing	pants	
	shirt	
	underwear	
animal	dog	
	fish	
	mouse	
exercise equipment	weights	
	bicycle	
	pool	
furniture	chair	
	table	
	lamp	
vehicle	bus	
	truck	
	car	
office equipment	paper	
	pencil	
	pen	
plant	tree	
	fern	
	flower	
vegetable	beans	
	pepper	
	onion	

tool	saw
	screwdriver
	hammer
food	potato
	cereal
	bread
weapon	gun
	knife
	bomb
kitchen utensil	plate
	spoon
	cup
entertainment	movie
	TV show
	museum
footgear	shoes
	boots
	skates
reading material	novel
	magazine
	reference

Table 44: Combined Labels from Rosch et al. (1976) and Markman et al. (1997)

Superordinate	**Basic**	**Subordinate (2 columns)**	
musical instrument	guitar	folk guitar	classical guitar
	piano	grand piano	upright piano
	drum	kettle drum	base drum
	trumpet		
	saxophone		
fruit	apple	delicious apple	mackintosh apple
	peach	freestone peach	cling peach
	grapes	concord grapes	green seedless grapes
	pear		
tool	hammer	ball-peen hammer	claw hammer
	saw	hack hand saw	cross-cutting hand saw
		chainsaw	
	screwdriver	phillips screwdriver	regular screwdriver
	drill		
clothing	pants	levis	double knit pants
		jeans	
	socks	knee socks	ankle socks
	shirt	dress shirt	knit shirt
	tie		
	scarf		

	shoes		
	underwear		
furniture	table	kitchen table	dining room table
	lamp	floor lamp	desk lamp
	chair	kitchen chair	living room chair
	sofa		
	bed		
	couch		
vehicle	car	sports car	four door sedan car
		limousine	
	bus	city bus	cross country bus
		school bus	
	truck	pick up truck	tractor-trailer truck
		fire truck	semitrailer truck
	airplane		
	motorcycle		
plant	tree	maple	birch
		oak	pine tree
	fern		
	flower		
animal	fish	bass	trout
		salmon	
	bird	cardinal	eagle
		sparrow	robin
		canary	
	cat		
	dog	poodle	
	butterfly		
	horse		
	cow		
	mouse		
insect	ant		
	termite		
vegetable	beans	green beans	
	onion		
	pepper		
	radish		
weapon	gun	shotgun	
	sword		
	spear		
	knife		
	bomb		
jewelry	necklace	pearl necklace	
	ring		
	watch		

footgear	shoes	sandals
	boots	
	skates	
exercise equipment	weights	Nautilus weights
	bicycle	
	pool	
sports equipment	ball	football
	racquet	
	net	
office equipment	paper	typing paper
	pencil	
	pen	
kitchen utensil	plate	dinner plate
	fork	
	spoon	
	cup	
camping equipment	tent	pup tent
reading material	novel	mystery novel
	magazine	
	newspaper	
	reference	
beverage	milk	skim milk
	coffee	
	tea	
	soda	
	alcohol	
entertainment	movie	horror movie
	TV show	
	museum	
food	potato	mashed potatoes
	cereal	
	bread	
human dwelling	apartment	
	hotel	
disease	measles	
	chicken pox	

Please note that in Table 44, ***shoes*** appears twice, once under ***footgear*** and another time under ***clothing*** from the two distinct sources. This is the only duplicated basic-level category, making for ninety-one unique categories given of the ninety-two listed with one duplicate. Note, however, that we have corrected ***bird***, ***fish***, and ***tree*** to basic-level per the experimental findings in Rosch et al. (1976). Despite this change, note that Markman et al. (1997) include ***bird*** as a superordinate and thus to avoid one category being counted at two different levels we defer to the more reliable source that validated the findings on an individual category basis rather than as a group (Rosch et al. 1976), leaving ***bird*** as basic-level rather than a superordinate in our labels. ***Robin*** and ***canary***, listed as basic-level in Markman et al. (1997), are here listed as subordinates given the aforementioned realignment of ***bird***.

Appendix B WordNet-Aligned Labels

Table 45: Labels Aligned to WordNet Senses

Superordinate	Basic	Subordinate
musical_instrument.n.01	guitar.n.01	
	piano.n.01	grand_piano.n.01
		upright.n.02
	drum.n.01	bass_drum.n.01
	cornet.n.01	
	sax.n.02	
fruit.n.01	apple.n.01	
	peach.n.03	freestone.n.01
		cling.n.01
	grape.n.01	concord_grape.n.01
	pear.n.01	
tool.n.01	hammer.n.02	ball-peen_hammer.n.01
		carpenter's_hammer.n.01
	saw.n.02	hacksaw.n.01
		crosscut_saw.n.01
		chain_saw.n.01
	screwdriver.n.01	phillips_screwdriver.n.01
		flat_tip_screwdriver.n.01
	drill.n.01	
clothing.n.01	trouser.n.01	levi's.n.01
		jean.n.01
	sock.n.01	knee-high.n.01
		anklet.n.02
	shirt.n.01	dress_shirt.n.01
	necktie.n.01	
	scarf.n.01	
	shoe.n.01	
	underwear.n.01	
furniture.n.01	table.n.02	kitchen_table.n.01
	lamp.n.02	floor_lamp.n.01
		table_lamp.n.01
	chair.n.01	
	bed.n.01	
	sofa.n.01	
vehicle.n.01	car.n.01	sports_car.n.01
		sedan.n.01
		limousine.n.01
	bus.n.01	
		school_bus.n.01
	truck.n.01	pickup.n.01
		tractor.n.02

		fire_engine.n.01
		trailer_truck.n.01
	airplane.n.01	
	motorcycle.n.01	
plant.n.02	tree.n.01	maple.n.02
		birch.n.02
		oak.n.02
		pine.n.01
	fern.n.01	
	flower.n.01	
animal.n.01	fish.n.01	bass.n.08
		trout.n.02
		salmon.n.01
	bird.n.01	cardinal.n.04
		eagle.n.01
		sparrow.n.01
		robin.n.01
		canary.n.04
	cat.n.01	
	dog.n.01	poodle.n.01
	butterfly.n.01	
	horse.n.01	
	cow.n.01	
	mouse.n.01	
insect.n.01	ant.n.01	
	termite.n.01	
vegetable.n.01	bean.n.01	green_bean.n.01
	onion.n.03	
	pepper.n.04	
	radish.n.01	
weapon.n.01	gun.n.01	shotgun.n.01
	sword.n.01	
	spear.n.01	
	knife.n.02	
weaponry.n.01	bomb.n.01	
jewelry.n.01	necklace.n.01	
	ring.n.08	
timepiece.n.01	watch.n.01	
footwear.n.02	shoe.n.01	sandal.n.01
	boot.n.01	
sports_equipment.n.01	skate.n.01	
game_equipment.n.01	ball.n.01	football.n.02
sports_implement.n.01	racket.n.04	
	net.n.05	
flatware.n.01	plate.n.04	dinner_plate.n.01

cutlery.n.02	fork.n.01	
	spoon.n.01	
crockery.n.01	cup.n.01	
shelter.n.01	tent.n.01	pup_tent.n.01
beverage.n.01	milk.n.01	skim_milk.n.01
	coffee.n.01	
	tea.n.01	
	pop.n.02	
	alcohol.n.01	
entertainment.n.01	movie.n.01	
	television_program.n.01	
	museum.n.01	
food.n.02	potato.n.01	mashed_potato.n.01
	cereal.n.03	
	bread.n.01	
disease.n.01	measles.n.01	
	chickenpox.n.01	

Appendix C Mechanical Turk Labeling Instructions

Instructions

The goal of this task is to pick the "basic" word from a list of related words. A "basic" word is one where:

- A young child can easily learn the word early on, before learning other related words (assuming the child is exposed to a number of examples)
- When shown a picture of a very specific thing, it's the word you'd commonly refer to it as (e.g. seeing a specific dining room chair, you'd typically refer to it as a chair)
- It is *often* a short, simple word. Some, like "television", are longer.
- It's the highest word on the list where the things named by the word tend to look somewhat similar (e.g. most chairs have a similar shape so that could fit this requirement, but pieces of furniture vary widely in appearance and would not fit the requirement)

You'll be presented with a list of words, from broadest to most specific. Please choose the "basic" word, if there is one. In some cases, there may not be a basic word in the list. For example, the list (function, polynomial, quadratic) does not have a basic word. In cases like this, choose "None".

A good guiding question is: ***"Which word do you think a young child would learn first? (assume exposure to all these things)"***. The answer to that question is often the right answer, so it's a good starting point. Still, note some lists don't have a "basic" word in them! If none seem like a young child would learn them, even given exposure to some examples, that's another clue it could be "None"

Example Question

Here is a simple example with the correct answer in **bold**:

- organism *(a living thing that has (or can develop) the ability to act or function independently)*
- animal *(a living organism characterized by voluntary movement)*
- **bird** *(warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings)*
- thrush *(songbirds characteristically having brownish upper plumage with a spotted breast)*
- robin *(small Old World songbird with a reddish breast)*
- None *(none of these answers may be pictured or learned by a young child)*

For this list, **bird** is the right answer. In relation to the four properties common to "basic" words: A child will learn this very young, if exposed to examples of birds. When seeing a particular bird it's usually the word "bird" that comes to mind first, not "thrush" or "animal". It does happen to be a short and simple word, which isn't a hard requirement but is additional evidence for "bird" being a "basic" word. And it's the broadest term that can be pictured as a single thing; most birds look roughly similar while different types of animals ("animal" is the next broader term on the list) are very differently shaped. So, "bird" fits all of these properties.

"Basic" and "Not Basic" Examples

Other example "basic" words: apple, bread, lamp, fish, cup, ring

Some words that are *not* "basic" because they are too broad (can't be pictured as one thing representing all): entertainment, furniture, jewelry

Some words that are *not* "basic" because they are too specific (a broader word can be pictured as one thing representing all): red delicious apple, salmon, flat tip screwdriver

Some words that are *not* "basic" because they are too abstract (not picturable as a thing): entitlement, magnetization, population

Appendix D Basic-Level Category Labels

Table 46: List of Synsets Corresponding to Basic-Level Categories from Labeling Process

aba.n.01	guava.n.03	radio.n.03
ackee.n.01	gutter.n.04	radish.n.01
anchovy_pear.n.02	haik.n.01	raisin.n.01
apple.n.01	hammer.n.02	rambutan.n.02
apricot.n.02	hyena.n.01	ribbon.n.01
armor.n.01	iceberg.n.01	robe.n.01
avocado.n.01	icepick.n.01	romper.n.02
awl.n.01	igloo.n.01	rose_apple.n.02
banana.n.02	iron.n.03	rug.n.01
band_aid.n.01	iron.n.04	sackcloth.n.01
bassinet.n.01	izar.n.01	sandglass.n.01
bed.n.01	jackal.n.01	sapodilla.n.02
bedspread.n.01	jacket.n.02	sapote.n.02
beet.n.02	jump_suit.n.01	saw.n.02
bell_apple.n.01	kanzu.n.01	scarf.n.01
berry.n.01	kettle.n.04	scraper.n.01
bevel.n.02	kiwi.n.03	screwdriver.n.01
bodkin.n.03	kumquat.n.02	seed.n.01
body_stocking.n.01	lamp.n.02	seed.n.02
bomb.n.01	land.n.02	shirt.n.01
bookcase.n.01	legging.n.01	shoe.n.01
boot.n.01	lemon.n.01	shovel.n.01
bottle_opener.n.01	leotard.n.01	shovel.n.03
brassiere.n.01	lime.n.06	shrub.n.01
breadfruit.n.02	lip-gloss.n.01	skirt.n.02
breechcloth.n.01	litchi.n.02	slipper.n.01
brick.n.01	longanberry.n.02	snowsuit.n.01
brush.n.02	loquat.n.02	snuffer.n.01
bubble.n.04	mamey.n.02	sock.n.01
burqa.n.01	mandarin.n.05	sofa.n.01
bus.n.01	mango.n.02	sorb.n.01
cabinet.n.01	mangosteen.n.02	sour_gourd.n.03
can_opener.n.01	marang.n.02	spade.n.02
canistel.n.02	mascara.n.01	spatula.n.02
canopy.n.01	mask.n.01	spear.n.02
car.n.01	mask.n.04	spider_web.n.02
carambola.n.02	matchbook.n.01	spreader.n.01
carriage.n.02	medlar.n.03	square.n.08
carrot.n.03	medlar.n.04	stick.n.01
carrycot.n.01	melon.n.01	stick.n.07

ceriman.n.02	motorcycle.n.01	stocking.n.01
chador.n.01	necktie.n.01	stool.n.01
chair.n.01	needle.n.03	stopwatch.n.01
cherry.n.03	neighbor.n.02	straightedge.n.01
citron.n.01	niqab.n.01	straitjacket.n.02
cloak.n.01	oar.n.01	suit.n.01
cloak.n.02	orange.n.01	swab.n.01
clock.n.01	paint.n.01	swatter.n.01
coat.n.01	pallet.n.03	sweater.n.01
cone.n.03	papaw.n.02	swimsuit.n.01
cracker.n.05	papaya.n.02	table.n.02
cradle.n.01	parsnip.n.03	table.n.03
crank.n.04	passion_fruit.n.01	tamarind.n.02
cravat.n.01	peach.n.03	tangelo.n.02
crib.n.01	pear.n.01	taro.n.03
cue.n.04	pen.n.01	telephone.n.02
custard_apple.n.02	pencil.n.01	television.n.01
cymbal.n.01	pencil_sharpener.n.01	tent.n.01
date.n.08	peplos.n.01	thimble.n.02
diaper.n.01	person.n.01	thumb.n.02
dibble.n.01	pestle.n.03	tights.n.01
dog.n.01	piano.n.01	timer.n.01
drum.n.01	pincer.n.01	toe.n.02
dry_ice.n.01	pineapple.n.02	toothbrush.n.01
durian.n.02	pinecone.n.01	toothpick.n.01
earmuff.n.01	pipe_cutter.n.01	train.n.01
eraser.n.01	pitahaya.n.02	tree.n.01
feijoa.n.02	pitchfork.n.01	triangle.n.05
field.n.14	plane.n.05	trouser.n.01
fig.n.04	plate.n.14	trowel.n.01
file.n.04	pliers.n.01	truck.n.01
file_folder.n.01	plum.n.02	turnip.n.02
flail.n.01	plumcot.n.02	underpants.n.01
flea.n.01	plunger.n.03	vest.n.01
float.n.05	poker.n.01	wall.n.02
flower.n.02	pole.n.01	washer.n.03
fox.n.01	pomegranate.n.02	watch.n.01
gem.n.02	pomelo.n.02	watermelon.n.02
genip.n.02	pot.n.01	weeder.n.02
genipap.n.01	potato.n.01	wet_suit.n.01
go-kart.n.01	prickly_pear.n.02	wire_stripper.n.01
golfcart.n.01	prune.n.01	wolf.n.01
gong.n.01	pulasan.n.02	wrench.n.03

grape.n.01	quandong.n.04	yam.n.03
grapefruit.n.02	quince.n.02	yam.n.04
graver.n.01	racket.n.04	yashmak.n.01

Appendix E Basic-Level Category Labels by Experiment Set

Table 47: Basic-Level Categories in the Train Set

ackee.n.01	guava.n.03	plum.n.02
anchovy_pear.n.02	hyena.n.01	plumcot.n.02
apple.n.01	iceberg.n.01	pomegranate.n.02
apricot.n.02	igloo.n.01	pomelo.n.02
avocado.n.01	jackal.n.01	potato.n.01
banana.n.02	kiwi.n.03	prickly_pear.n.02
beet.n.02	kumquat.n.02	prune.n.01
bell_apple.n.01	land.n.02	pulasan.n.02
berry.n.01	lemon.n.01	quandong.n.04
breadfruit.n.02	lime.n.06	quince.n.02
canistel.n.02	litchi.n.02	radish.n.01
carambola.n.02	longanberry.n.02	raisin.n.01
carrot.n.03	loquat.n.02	rambutan.n.02
ceriman.n.02	mamey.n.02	ribbon.n.01
cherry.n.03	mandarin.n.05	rose_apple.n.02
citron.n.01	mango.n.02	sapodilla.n.02
cone.n.03	mangosteen.n.02	sapote.n.02
cracker.n.05	marang.n.02	seed.n.01
custard_apple.n.02	medlar.n.03	seed.n.02
date.n.08	medlar.n.04	shrub.n.01
dog.n.01	melon.n.01	sorb.n.01
dry_ice.n.01	neighbor.n.02	sour_gourd.n.03
durian.n.02	orange.n.01	spider_web.n.02
feijoa.n.02	paint.n.01	tamarind.n.02
field.n.14	papaw.n.02	tangelo.n.02
fig.n.04	papaya.n.02	taro.n.03
flea.n.01	parsnip.n.03	tent.n.01
flower.n.02	passion_fruit.n.01	tree.n.01
fox.n.01	peach.n.03	turnip.n.02
gem.n.02	pear.n.01	wall.n.02
genip.n.02	person.n.01	watermelon.n.02
genipap.n.01	pineapple.n.02	wolf.n.01
grape.n.01	pinecone.n.01	yam.n.03
grapefruit.n.02	pitahaya.n.02	yam.n.04

Table 48: Basic-Level Categories in the Development Set

aba.n.01	iron.n.04	skirt.n.02
armor.n.01	izar.n.01	slipper.n.01
band_aid.n.01	jacket.n.02	snowsuit.n.01

bedspread.n.01	jump_suit.n.01	sock.n.01
body_stocking.n.01	kanzu.n.01	stocking.n.01
boot.n.01	legging.n.01	straitjacket.n.02
brassiere.n.01	leotard.n.01	suit.n.01
breechcloth.n.01	mask.n.01	sweater.n.01
brick.n.01	mask.n.04	swimsuit.n.01
bubble.n.04	matchbook.n.01	thimble.n.02
burqa.n.01	necktie.n.01	thumb.n.02
canopy.n.01	niqab.n.01	tights.n.01
chador.n.01	peplos.n.01	toe.n.02
cloak.n.01	plate.n.14	toothpick.n.01
cloak.n.02	robe.n.01	trouser.n.01
coat.n.01	romper.n.02	underpants.n.01
cravat.n.01	sackcloth.n.01	vest.n.01
diaper.n.01	scarf.n.01	washer.n.03
earmuff.n.01	shirt.n.01	wet_suit.n.01
file_folder.n.01	shoe.n.01	yashmak.n.01
haik.n.01		

Table 49: Basic-Level Categories in the Test Set

awl.n.01	gutter.n.04	screwdriver.n.01
bassinet.n.01	hammer.n.02	shovel.n.01
bed.n.01	icepick.n.01	shovel.n.03
bevel.n.02	iron.n.03	snuffer.n.01
bodkin.n.03	kettle.n.04	sofa.n.01
bomb.n.01	lamp.n.02	spade.n.02
bookcase.n.01	lip-gloss.n.01	spatula.n.02
bottle_opener.n.01	mascara.n.01	spear.n.02
brush.n.02	motorcycle.n.01	spreader.n.01
bus.n.01	needle.n.03	square.n.08
cabinet.n.01	oar.n.01	stick.n.01
can_opener.n.01	pallet.n.03	stick.n.07
car.n.01	pen.n.01	stool.n.01
carriage.n.02	pencil.n.01	stopwatch.n.01
carrycot.n.01	pencil_sharpener.n.01	straightedge.n.01
chair.n.01	pestle.n.03	swab.n.01
clock.n.01	piano.n.01	swatter.n.01
cradle.n.01	pincer.n.01	table.n.02
crank.n.04	pipe_cutter.n.01	table.n.03
crib.n.01	pitchfork.n.01	telephone.n.02
cue.n.04	plane.n.05	television.n.01
cymbal.n.01	pliers.n.01	timer.n.01

dibble.n.01	plunger.n.03	toothbrush.n.01
drum.n.01	poker.n.01	train.n.01
eraser.n.01	pole.n.01	triangle.n.05
file.n.04	pot.n.01	trowel.n.01
flail.n.01	racket.n.04	truck.n.01
float.n.05	radio.n.03	watch.n.01
go-kart.n.01	rug.n.01	weeder.n.02
golfcart.n.01	sandglass.n.01	wire_stripper.n.01
gong.n.01	saw.n.02	wrench.n.03
graver.n.01	scraper.n.01	

References

Acquinas, T. (1274). Summa Theologica.

Alonso, O. and Mizzaro, S. (2009). Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation.

Alonso, O., Rose, D. E. and Stewart, B. (2008). Crowdsourcing for relevance evaluation. ACM SigIR Forum, ACM.

Anglin, J. M. (1975). "The child's first terms of reference." Bulletin de Psychologie. La Memoire Semantique.

Aristotle (c.350 BC). Metaphysics.

Baker, C. F., Fillmore, C. J. and Lowe, J. B. (1998). The Berkeley FrameNet Project. Proceedings of the 17th International Conference on Computational Linguistics, Montreal, Quebec, Canada, Association for Computational Linguistics.

Binswanger, H. (2014). How We Know: Epistemology on an Objectivist Foundation, TOF Publications.

Bird, S., Klein, E. and Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit, O'Reilly Media, Inc.

Bond, F. and Paik, K. (2012). A survey of wordnets and their licenses. Proceedings of the 6th Global WordNet Conference (GWC 2012).

Brank, J., Grobelnik, M. and Mladenić, D. (2005). A survey of ontology evaluation techniques. Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005), Ljubljana, Slovenia, J. Stefan Institute.

Brown, R. (1958). "How shall a thing be called?" Psychological review Vol. 65, No. 1: 14-21.

Buhrmester, M., Kwang, T. and Gosling, S. D. (2011). "Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?" Perspectives on psychological science Vol. 6, No. 1: 3-5.

Cha, M., Gwon, Y. and Kung, H. (2017). "Language Modeling by Clustering with Word Embeddings for Text Readability Assessment." arXiv preprint arXiv:1709.01888.

Chall, J. S. and Dale, E. (1995). Readability revisited: The new Dale-Chall readability formula. Cambridge, MA, Brookline Books.

Chang, H.-S., Wang, Z., Vilnis, L. and McCallum, A. (2017). "Unsupervised Hypernym Detection by Distributional Inclusion Vector Embedding." arXiv preprint arXiv:1710.00880.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P. and Zitnick, C. L. (2015). "Microsoft COCO captions: Data collection and evaluation server." arXiv preprint arXiv:1504.00325.

Chklovski, T. and Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, Association for Computational Linguistics.

Choi, S. and Gopnik, A. (1995). "Early acquisition of verbs in Korean: A cross-linguistic study." Journal of Child Language Vol. 22, No. 3: 497-529.

Collins-Thompson, K. (2014). "Computational assessment of text readability: A survey of current and future research." ITL-International Journal of Applied Linguistics Vol. 165, No. 2: 97-135.

Colmer, R. (2018). "The Flesch Reading Ease and Flesch-Kincaid Grade Level." Retrieved 6/3/2018, from https://readable.io/blog/the-flesch-reading-ease-and-flesch-kincaid-grade-level/.

Dalvean, M. C. and Enkhbayar, G. (2018). "A New Text Readability Measure for Fiction Texts."

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, Miami Beach, FL, IEEE.

Dolch, E. W. (1948). Problems in reading, Garrard Press.

Flesch, R. (1979). "How to write plain English: Let's start with the formula." University of Canterbury.

Francis, N. and Kucera, H. (1964). "Manual of Information to Accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Department of Linguistics, Brown University, Providence, USA)." icame. uib. no/brown/bcm. html (accessed 12 October 2010).

Gentner, D. (1982). "Why nouns are learned before verbs: Linguistic relativity versus natural partitioning." Center for the Study of Reading Technical Report; no. 257.

Gholamrezazadeh, S., Salehi, M. A. and Gholamzadeh, B. (2009). A comprehensive survey on text summarization systems. Computer Science and its Applications, 2009. CSA'09. 2nd International Conference on, IEEE.

Girju, R., Badulescu, A. and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics.

Golinkoff, R. M., Shuff-Bailey, M., Olguin, R. and Ruan, W. (1995). "Young children extend novel words at the basic level: Evidence for the principle of categorical scope." Developmental Psychology Vol. 31, No. 3: 494-507.

Gomaa, W. H. and Fahmy, A. A. (2013). "A survey of text similarity approaches." International Journal of Computer Applications Vol. 68, No. 13: 13-18.

Green, R. (2006). "Vocabulary alignment via basic level concepts. Final Report 2003 OCLC/ALISE Library and Information Science Research Grant Project." Dublin, OH: OCLC Online Computer Library, Inc. Retrieved May Vol. 14: 2008.

Gutenberg, P. (2018). Retrieved 11/18/2017, from http://www.gutenberg.org.

Haav, H.-M. and Lubi, T.-L. (2001). A survey of concept-based information retrieval tools on the web. Proceedings of the 5th East-European Conference ADBIS.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. Proceedings of the 14th conference on Computational linguistics-Volume 2, Association for Computational Linguistics.

Hearst, M. A. (1998). "Automated discovery of WordNet relations." WordNet: an electronic lexical database: 131-153.

Henderson, J. and Popa, D. N. (2016). "A vector space for distributional semantics for entailment." arXiv preprint arXiv:1607.03780.

Ipeirotis, P. G., Provost, F. and Wang, J. (2010). Quality management on amazon mechanical turk. Proceedings of the ACM SIGKDD workshop on human computation, ACM.

Izquierdo, R., Suárez, A. and Rigau, G. (2007). Exploring the automatic selection of basic level concepts. Proceedings of RANLP, Citeseer.

Jolicoeur, P., Gluck, M. A. and Kosslyn, S. M. (1984). "Pictures and names: Making the connection." Cognitive psychology Vol. 16, No. 2: 243-275.

Jónsdóttir, M. K. and Martin, R. C. (1996). "Superordinate vs basic level knowledge in aphasia: A case study." Journal of Neurolinguistics Vol. 9, No. 4: 261-287.

Kant, I. (1781). "The Critique of Pure Reason."

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L. and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel, Naval Technical Training Command Millington TN Research Branch.

Klibanoff, R. S. and Waxman, S. R. (2000). "Basic level object categories support the acquisition of novel adjectives: Evidence from preschool-aged children." Child development Vol. 71, No. 3: 649-659.

Landes, S., Leacock, C. and Tengi, R. I. (1998). "Building semantic concordances." WordNet: an electronic lexical database: 199-216.

Landis, J. R. and Koch, G. G. (1977). "The measurement of observer agreement for categorical data." biometrics Vol. 33: 159-174.

Legislature, F. (2003). "Readable language in insurance policies." Chapter 627: INSURANCE RATES AND CONTRACTS. Retrieved 6/3/2018, from http://www.leg.state.fl.us/Statutes/index.cfm?App_mode=Display_Statute&Search_String=&URL=0600-0699/0627/Sections/0627.4145.html.

Legrand, S. (2006). "Word Sense Disambiguation with Basic-Level Categories." Advances in Natural Language Processing. Ed. Alexander Gelbukh, Research in Computing Science Vol. 18: 71-82.

Lemaitre, G. and Heller, L. M. (2013). "Evidence for a basic level in a taxonomy of everyday action sounds." Experimental brain research Vol. 226, No. 2: 253-264.

Lin, D., Zhao, S., Qin, L. and Zhou, M. (2003). Identifying synonyms among distributionally similar words. IJCAI, Acapulco, Mexico.

Lin, F. and Sandkuhl, K. (2008a). A survey of exploiting wordnet in ontology matching. IFIP International Conference on Artificial Intelligence in Theory and Practice, Milan, Italy, Springer.

Lin, S.-y., Su, C.-c., Lai, Y.-D., Yang, L.-C. and Hsieh, S.-K. (2008b). Measuring Text Readability by Lexical Relations Retrieved from Wordnet. ROCLING, Taipei, Taiwan.

Lin, S.-Y., Su, C.-C., Lai, Y.-D., Yang, L.-C. and Hsieh, S.-K. (2009). "Assessing text readability using hierarchical lexical relations retrieved from WordNet." Computational Linguistics and Chinese Language Processing Vol. 14, No. 1: 45-84.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision, Zurich, Switzerland, Springer.

Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Aggarwal, C. and Zhai, C. (eds) Mining text data. Springer, Boston, MA.

Loper, E. and Bird, S. (2002). NLTK: The natural language toolkit. Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Philadelphia, PA, Association for Computational Linguistics.

Lucassen, T. D., Roald; Schraagen, Jan Maarten (2012). "Readability of Wikipedia." First Monday Vol. 17, No. 9.

MacWhinney, B. (2000). The CHILDES project: The database, Psychology Press.

Markman, A. B. and Wisniewski, E. J. (1997). "Similar and different: The differentiation of basic-level categories." Journal of Experimental Psychology: Learning, Memory, and Cognition Vol. 23, No. 1: 54-70.

Mervis, C. B. and Rosch, E. (1981). "Categorization of natural objects." Annual review of psychology Vol. 32, No. 1: 89-115.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B. and Grishman, R. (2004). The NomBank Project: An Interim Report. HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation, Boston, MA.

Mill, J. S. (1884). A system of logic ratiocinative and inductive: Being a connected view of the principles of evidence and the methods of scientific investigation, Harper.

Miller, G. A. (1995). "WordNet: A Lexlcal Database for English." Communications of the ACM. Vol. 38, No. 11: 39-41.

Mills, C., Bond, F. and Levow, G.-A. (2018). Automatic Identification of Basic-Level Categories. Proceedings of the 9th Global WordNet Conference (GWC 2018), Singapore, Global Wordnet Association.

Murphy, G. L. and Wisniewski, E. J. (1989). "Categorizing objects in isolation and in scenes: What a superordinate is good for." Journal of Experimental Psychology: Learning, Memory, and Cognition Vol. 15, No. 4: 572-586.

Murphy, M. L. (2003). Semantic relations and the lexicon: Antonymy, synonymy and other paradigms, Cambridge University Press.

Nadeau, D. and Sekine, S. (2007). "A survey of named entity recognition and classification." Lingvisticae Investigationes Vol. 30, No. 1: 3-26.

National Governors Association Center for Best Practices, C. o. C. S. S. O. (2010). "Common Core State Standards for Language Arts & Literacy in History/Social Studies, Science, and Technical Subjects: Appendix B." Retrieved 6/4/2018, 2018, from http://www.corestandards.org/assets/Appendix_B.pdf.

Navigli, R. (2009). "Word Sense Disambiguation: A Survey." ACM computing surveys (CSUR) Vol. 41, No. 2: 1-69.

Ockham, W. o. (1323). "Summa Logicae."

Ordonez, V., Deng, J., Choi, Y., Berg, A. C. and Berg, T. L. (2013). From large scale image categorization to entry-level categories. Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, Institute of Electrical and Electronics Engineers.

Ordonez, V., Kulkarni, G. and Berg, T. L. (2011). Im2text: Describing images using 1 million captioned photographs. Advances in Neural Information Processing Systems, Granada, Spain.

Paolacci, G., Chandler, J. and Ipeirotis, P. G. (2010). "Running experiments on amazon mechanical turk." Judgment and Decision Making Vol. 5, No. 5: 411-419.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. (2011). "Scikit-learn: Machine Learning in Python." Journal of machine learning research Vol. 12: 2825-2830.

Petrolito, T. and Bond, F. (2014). A survey of wordnet annotated corpora. Proceedings of the Seventh Global WordNet Conference.

Plato (c.380 BC). Republic.

Porphyry (270). Isagoge.

Rand, A. (1966a). "Introduction to Objectivist Epistemology (I): Concept-Formation." The Objectivist Vol. 66, No. 7.

Rand, A. (1966b). "Introduction to Objectivist Epistemology (II): Abstraction from Abstractions." The Objectivist Vol. 66, No. 8.

Rand, A. (1967). "Introduction to Objectivist Epistemology (VII): The Cognitive Role of Concepts." The Objectivist Vol. 67, No. 1.

Rand, A. (1990). Introduction to Objectivist Epistemology: Expanded Second Edition, Penguin.

Rashtchian, C., Young, P., Hodosh, M. and Hockenmaier, J. (2010). Collecting image annotations using Amazon's Mechanical Turk. Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Association for Computational Linguistics.

Rips, L. J., Smith, E. E. and Medin, D. L. (2012). "Concepts and categories: Memory, meaning, and metaphysics." The Oxford handbook of thinking and reasoning: 177-209.

Roberson, D., Davies, I. R., Corbett, G. G. and Vandervyver, M. (2005). "Free-sorting of colors across cultures: Are there universal grounds for grouping?" Journal of Cognition and Culture Vol. 5, No. 3: 349-386.

Roller, S. and Erk, K. (2016). "Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment." arXiv preprint arXiv:1605.05433.

Rorissa, A. and Iyer, H. (2008). "Theories of cognition and image categorization: What category labels reveal about basic level theory." Journal of the American Society for Information Science and Technology Vol. 59, No. 9: 1383-1392.

Rosch, E. (1975). "Cognitive representations of semantic categories." Journal of Experimental Psychology: General Vol. 104, No. 3: 192.

Rosch, E. (1999). "Reclaiming concepts." Journal of consciousness studies Vol. 6, No. 11-12: 61-77.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M. and Boyes-Braem, P. (1976). "Basic objects in natural categories." Cognitive psychology Vol. 8, No. 3: 382-439.

Rosch, E. H. (1973). "Natural categories." Cognitive psychology Vol. 4, No. 3: 328-350.

Salmieri, G. (2017). Epistemic vs. Logical Hierarchy.

Scholar, G. (2018). "WordNet Citations." Retrieved 7/9/2018, from https://scholar.google.com/scholar?cites=18066988468300420639&as_sdt=5,48&sciodt=1,48&hl=en.

Schuler, K. K. (2009). "VerbNet Overview." NAACL HLT, Tutorials: 13-14.

Snow, R., Jurafsky, D. and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. Advances in neural information processing systems.

Snow, R., Jurafsky, D. and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics.

Snow, R., O'Connor, B., Jurafsky, D. and Ng, A. Y. (2008). Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics.

Stephen, H., Farwell, D., Reeder, F., Miller, K., Dorr, B., Habash, N., Hovy, E., Levin, L., Mitamura, T., Rambow, O. and Siddharthan, A. (2009). "Interlingual annotation of multilingual text corpora and FrameNet." TRENDS IN LINGUISTICS STUDIES AND MONOGRAPHS Vol. 200: 287-318.

Tanaka, J. W. and Taylor, M. (1991). "Object categories and expertise: Is the basic level in the eye of the beholder?" Cognitive psychology Vol. 23, No. 3: 457-482.

Tanaka, S., Jatowt, A., Kato, M. P. and Tanaka, K. (2013). Estimating content concreteness for finding comprehensible documents. Proceedings of the sixth ACM international conference on Web search and data mining, Association for Computing Machinery.

Turney, P. D., Littman, M. L., Bigham, J. and Shnayder, V. (2003). "Combining independent modules to solve multiple-choice synonym and analogy problems." arXiv preprint cs/0309035.

Van Dam, W. O., Rueschemeyer, S.-A. and Bekkering, H. (2010). "How specifically are action verbs represented in the neural motor system: an fMRI study." Neuroimage Vol. 53, No. 4: 1318-1325.

Viera, A. J. and Garrett, J. M. (2005). "Understanding interobserver agreement: the kappa statistic." Fam Med Vol. 37, No. 5: 360-363.

Wang, Z., Wang, H., Wen, J.-R. and Xiao, Y. (2015). An Inference Approach to Basic Level of Categorization. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Association for Computing Machinery.

Weeds, J., Clarke, D., Reffin, J., Weir, D. and Keller, B. (2014). Learning to distinguish hypernyms and co-hyponyms. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics.

Weide, R. L. (1998). "The CMU pronouncing dictionary." URL: http://www. speech. cs. cmu. edu/cgibin/cmudict.