Learning Physical Commonsense Knowledge

Carlos E. Jimenez
University of Utah

UUCS-20-005

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

27 April 2020

Abstract

As speakers and writers omit details about common human experiences, a great deal of relevant information eludes concrete attestation in the corpora that many language models and other NLP systems rely on. Physical information about everyday objects is a perfect representative of this type of elusive commonsense knowledge. Through extracting dependency-based contextual representations for training classifiers, we present a means of learning the physical attributes of scores of new words from corpora using only a small number of seed words.

Learning Physical Commonsense Knowledge

by

Carlos E. Jimenez

A Senior Thesis Submitted to the Faculty of
The University of Utah
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Computer Science

School of Computing
University of Utah
April 26, 2020

Approved:

Ellen Riloff ______________________ Date: 4/27/2020

Ellen Riloff
Supervisor

______________________ Date: ________

H. James de St. Germain
Director of Undergraduate Studies
School of Computing

______________________ Date: ________

Ross Whitaker
Director
School of Computing

Abstract

As speakers and writers omit details about common human experiences, a great deal of relevant information eludes concrete attestation in the corpora that many language models and other NLP systems rely on. Physical information about everyday objects is a perfect representative of this type of elusive commonsense knowledge. Through extracting dependency-based contextual representations for training classifiers, we present a means of learning the physical attributes of scores of new words from corpora using only a small number of seed words.

Contents

List of Figures

List of Tables

1 Introduction

In natural speech and text one encounters constant reference to material objects in the human environment and descriptions of their relations to one another. That is to say, unsurprisingly, language is often used to refer to artifacts and events in the natural world. It can also be noted that humans often omit descriptions of the natural world when they expect their audience to assume an approximation of these omitted facts sufficiently well, reflecting Grice's maxim of quantity [Grice, 1975]. This is partly reflected in the historical difficulty and significance of tasks such as the Winograd Schema Challenge (WSC) [Levesque et al., 2012], which can be seen as a problem of pragmatics [Saba, 2019]; a problem of inferring meaning using world knowledge, rather than syntactico-semantic information. Tasks like the WSC are difficult *because* they require external knowledge.

Physical properties of common objects represent a kind of commonsense knowledge that is often omitted in text, since humans rely on assumptions about common experiences to make communication more concise and efficient [Havasi and Alonso, 2007]. For example, instances of coffee cups in natural language are rarely coupled with the size, weight, and rigidity of such cups. One is unlikely to include an object's physical properties unless that object is novel, or otherwise diverges from common experience. Yet extra-linguistic physical world knowledge can be fundamental to understanding language [Katz and Fodor, 1963] such as in the WSC question "I put the [**heavy book / butterfly wing**] on the table and it broke. What broke?" This question captures a problem of coreference resolution, but one that very likely requires some sense of physical reasoning; an understanding of real and relative sizes, weights, etc.

Common systems for solving comprehension tasks in natural language processing include the large pretrained neural models trained on massive corpora that have become popular recently for solving a wide variety of tasks. Yet it has been shown that these systems' representations often fail to model a variety of meaningful aspects, especially relationships that are not directly attested to in its training corpus [Forbes et al., 2019, Rubinstein et al., 2015], such as the very type of physical knowledge we've discussed.

Humans acquire this knowledge through everyday experiences. We have years of experience in physical interaction with the many objects that we so discuss. Ultimately, it is suspected that the real solution for imbibing machines with the full gamut of human style reasoning and knowledge would require a form of embodied learning [Lucy and Gauthier, 2017], involving extensive real human-like sensory experience of machine agents in the real world. This is yet to be achieved.

How might we improve existing systems' performance in solving tasks requiring common knowledge of the physical world? We propose a method of predicting nouns' physical properties through training a classifier on distributional context-based representations and a small number of annotated seed words. We evaluate the predictions provided by our classifier on the gold annotation data for nouns provided in the [Wang et al., 2018] dataset, as well as their relative property values to the object-pair relative physical knowledge provided in [Forbes and Choi, 2017].

2 Background and Related Work

The concept of commonsense and commonsense knowledge remains difficult to define in natural language processing. Notable recent approaches attempting to acquire commonsense knowledge have focused on relative physical attributes [Forbes and Choi, 2017, Yang et al., 2018, Tandon et al., 2014] or acquiring real valued distributions of quantitative data for various physical attributes from text [Elazar et al., 2019]. Our approach attempts to learn knowledge in a medium-resolution landmark based format, introduced in [Wang et al., 2018].

2.1 Physical Plausibility Task

[Wang et al., 2018] introduces a crowd-sourced dataset to measure the semantic plausibility for 3,062 different events, with a vocabulary of 150 verbs, and about 450 nouns. They seek to classify some event represented by a (*subject, verb, object*) triplet as either physically plausible, or implausible. Consider for example three events: *(goose, eat, rice)*, *(goose, eat, quarter)*, and *(goose, eat, piano).* The first event is rather plausible and we might be unsurprised to see it attested in corpora. The second event is certainly less common, but it is still a physically plausible event; we can very well imagine a goose swallowing a quarter, even if such an event is unlikely, and thus unlikely to be attested in corpora. The last however is both unlikely to be attested in corpora and not physically plausible. Despite both the second and third events being unlikely to appear in corpora, the third event is simply unacceptable to entertain. In the dataset, events similar to the first and second are labeled positive since they are physically plausible, regardless of whether they are likely to appear in corpora or not, while those physically implausible events, like the third event, are labeled negative. Additionally, there are cases of semantically non-sensical events, such as the triplet (*cloth, erase, wind*), which do appear in the dataset and are labelled negative, of course.

Since most distributional models are fully dependent upon the linguistic information attested in training corpora [Forbes et al., 2019, Lucy and Gauthier, 2017], the lack of physical world knowledge available to these models may hinder performance on comprehension and reasoning [Wang et al., 2018]. Our approach then seeks to circumvent the normal problems affecting distributional models, like reporting bias [Durme, 2010], by explicitly modeling physical properties to extract attribute labels for use directly. This is to say that though there may not often be explicit descriptions of the approximate size of different objects, we may still be able to infer information from distributional approaches if we represent words' physical attributes explicitly.

[Wang et al., 2018] shows that the semantic plausibility task benefits from explicitly modelling physical world knowledge. They compare results from training a selectional preference-based neural network model [de Cruys, 2014] trained on events represented as stacked Glove embeddings to a neural network model that is similar, but enriched with explicit physical world knowledge. The enriched model significantly outperforms the plain neural network model, suggesting, at least, that explicitly modeling physical world knowledge is useful in the semantic plausibility task. The performance of the selectional preference based neural network NN and the model enriched with physical world knowledge NN+WK are shown in Table 5.1.

2.2 Selectional Preferences

Selectional preferences refer to the semantic “preferences” of a predicate for its arguments. For example, in the sentence “The skiers wore helmets” the mere occurrence of the predicate “wore” tells us the likely semantic class of its subject. In determining selectional preferences, [Resnik, 1997] sought to measure the *selectional preference strength* of a predicate for its argument to classify word sense. In other words, selectional preference strength measures how informative a predicate is with respect to selecting its argument’s semantic class. The selectional preference strength that a predicate has for its subject’s ‘conceptual class’ (e.g., WordNet senses), was modeled as the Kullback Leibler divergence, $S_R(p) = D_{\mathrm{KL}}(P(c|p)||P(c))$, for predicate p, and conceptual class c. The predicate “eat” is likely to have a much higher selectional preference strength for its object argument, since its objects’ conceptual class is likely to be loosely restricted to foodstuffs, compared to the predicate “see”, which would be likely to occur with a huge variety of conceptual classes.

Selectional preferences are considered in [Wang et al., 2018] as potentially a means of learning the physical plausibility task. However, it is shown that such an approach may be insufficient, because by measuring the semantic preferences of predicates for their arguments, a model will necessarily learn only the attested events in a corpus, and not plausibile events that are simply uncommon. This is because by learning the conceptual class of a predicate’s arguments, the possibility of an event becomes tied to a taxonomic commonality; “eat” selects for objects that are edible, yet edibility of an object is not exactly what makes an eat event plausible. Thus, a system based on selectional preferences properly is likely to be unable to optimally learn the physical plausibility task, since it cannot consider the independent physical attributes of event arguments, which may sometimes be the reason for an event’s plausibility.

Interestingly, selectional preferences have been expanded upon to look at additional, more sophisticated forms of measuring the ‘preference’ that a predicate, or other relation, has for its arguments. An additional relationship may be to consider the preference that certain adjectives have for their argument’s conceptual class as investigated by [Ó Séaghdha, 2010, Hermann et al., 2012]; for example, we can imagine that the adjective ‘hairy’ might select more strongly for mammal arguments.

2.3 Learning Physical Attributes

There are some notable prior attempts to learning physical attributes from unstructured text. [Forbes and Choi, 2017] use a factor graph and belief propagation to model the relative physical relations between object pairs. For some attribute $a \in$ {SIZE, WEIGHT, STRENGTH, RIGIDNESS, SPEED}, they look to compare two objects (x, y) as $x >^{\text{attr}} y$, $x <^{\text{attr}} y$ or $x \approx^{\text{attr}} y$. For attr = size, for example $x <^{\text{size}} y$ represents x is smaller than y. They perform inference on a graph using nodes that model the relative sizes of various objects, as well as nodes modeling the physical implications of verbs (verb frame nodes). The verb frame node represents a prediction like $P(F_{\text{verb}}^{\text{attr}}(x, y) = >)$ for some verb, physical attribute dimension *attr*, subject x, and object y. Through this method they are able to acquire 'low resolution' relative physical attribute knowledge for 3,656 object pairs. This resource is useful of course for asking explicitly relative questions, such as "is a cat heavier than a mouse?" But, in the semantic plausibility task form [Wang et al., 2018] for example, there may be considerations that are more nuanced than simply relative knowledge. If our event is (*man, swallow, grape*), knowing that a grape is smaller than a man is useful in classifying whether such an event is plausible, but books are smaller than men, and yet it is not plausible that a man swallows a book.

Another approach to learning physical attribute knowledge by [Elazar et al., 2019], is to aggregate all real-valued co-occurrences of objects and quantitative values within some window-context, and count these as positive examples. They do this for attributes TIME, CURRENCY, LENGTH, AREA, VOLUME, MASS, TEMPERATURE, DURATION, SPEED, and VOLTAGE. They normalize all measurements by converting all quantitative values to some standard measure. This approach results in a frequency distribution over a range of values, providing richer statistical information than other results. It is effectively, a full resolution representation, which of course has some benefits, but, it may be argued, isn't always appropriate for common-sense tasks, where physical attributes are meant to be approximate, and many people don't actually have such rich information themselves.

As we see, existing techniques have focused either on learning fairly low resolution relative physical comparisons between objects [Forbes and Choi, 2017, Yang et al., 2018, Tandon et al., 2014], or on real valued, high resolution data [Elazar et al., 2019]. Both of these approaches may have virtues or drawbacks, but our approach contrasts with them both in using a medium-resolution landmark based measure for physical attributes, introduced by [Wang et al., 2018]. These landmark values relate to some prototypical example of some object in an attribute spectrum, and new objects are labelled with respect to their best similarity to the landmark objects. The full resolution table with landmarks is shown in Table 2.1. We like this landmark-based view because it seems a bit more manageable and approximate than the very high-resolution descriptions in [Elazar et al., 2019], yet it should pick up additional nuance in relative sizes over the lower resolution examples. Additionally, it provides a global, non-relative concrete value for each object, without the relative context pair, which seems more valuable as well.

	Attribute Values						
Attribute	0	1	2	3	4	5	6
sentience	rock	tree	ant	cat	chimp	man	-
masscount	milk	sand	legos	car	-	-	-
phase	smoke	milk	wood	-	-	-	-
size	-watch	watch-book	book-cat	cat-person	person-jeep	jeep-stadium	stadium-
weight	-watch	watch-book	book-dumbbell	dumbbell-person	person-jeep	jeep-stadium	stadium-
rigidity	water	skin	leather/plastic	wood	metal	-	-

Table 2.1: Physical Attribute Landmarks

3 Methods

3.1 Overview

The goal of this work is to learn the physical attribute values for new words using dependency-based contextual information. Our approach involves training machine learning models on data from a corpus, and assigning new words attribute values from an approximate, medium-resolution value range, as defined in [Wang et al., 2018]. Figure 3.1 provides an illustrated overview of our approach: first processing each sentence in the corpus into a tuple of dependency components, aggregating dependency patterns co-occuring with seed word nouns, extracting new nouns co-occuring with dependency patterns, performing dimensionality reduction, and training a classifier to assign attribute values to novel words.

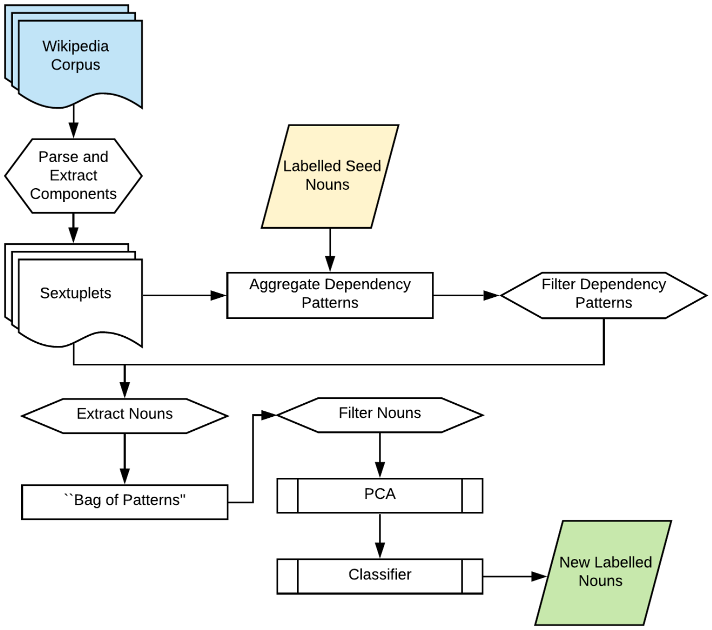


Figure 3.1: Method Summary

3.2 Corpus Processing

We start by processing each sentence in English Wikipedia [1] into a sextuplet representation with components (*subject, subject-pre-modifiers, root verb, object, object-pre-modifiers, prepositional phrase*). We use spaCy's[2] English dependency parser and the WordNetLemmatizer [Miller, 1995] for parsing and lemmatization respectively. The *subject* component is defined as a head noun with a dependency relation 'nsubj', 'nsubjpass', or 'csubj' to the *root verb*. The *object* component is defined as a head noun with a dependency relation 'dobj', 'dative', 'attr', 'oprd', 'acomp' or 'agent'→'pobj'[3] to the *root verb*. The *pre-modifiers* are defined by their 'compound', 'nmod', or 'amod' dependency relations to either the *subject* or *object* components. The *prepositional phrase* is defined by its 'prep'→'pobj' dependency relation to the *root verb*. The *root verb* is concatenated with its 'particle' token, if one is present. Each component of the sextuplet is defined by its dependency relation to the *root verb* or another component. A component then is a token that is defined by a class of dependents or subtrees, summarized in Figure 3.2. Note that we only extract a single phrase for *subject*, *verb*, *object*, and *prepositional phrase* components, but we may extract multiple *pre-modifier* components.

- subject $\in$ {'nsubj', 'nsubjpass', 'csubj'}
- object $\in$ {'dobj', 'dative', 'attr', 'oprd', 'acomp', 'agent'→'pobj'}
- root verb $\in$ {'root', 'root'→'prt'} with pos 'verb'.
- pre-modifier $\in$ {'compound', 'nmod', 'amod'}
- prepositional phrase $\in$ {'prep'→'pobj'}

Figure 3.2: Component Classes for Sextuplet Representation

3.3 Dependency Patterns

After processing the sextuplets from our corpus, we will extract a variety of contextual elements, we will call them dependency patterns, which are defined by a set of fixed components, and a single variable component (either the *subject* or *object* component). The collection of sextuplets extracted from the corpus represents a body of sentence instances. By extracting dependency patterns, we can aggregate the distributional information of the dependency pattern's variable component to determine the physical preference that the dependency pattern may have, in order to inform our classifier. In Figure 3.3 we enumerate the *types* of dependency patterns that we compile distributions for. The fixed components are shown in black, while the variable component is shown in blue and parenthesized to emphasize its variable status.

[1] Wikimedia dumps service enwiki dumped 1 March, 2020

[2] https://github.com/explosion/spaCy

[3] Here 'agent'→'pobj' semantically represents the agent of the sentence. Active/passive voice is not normalized.

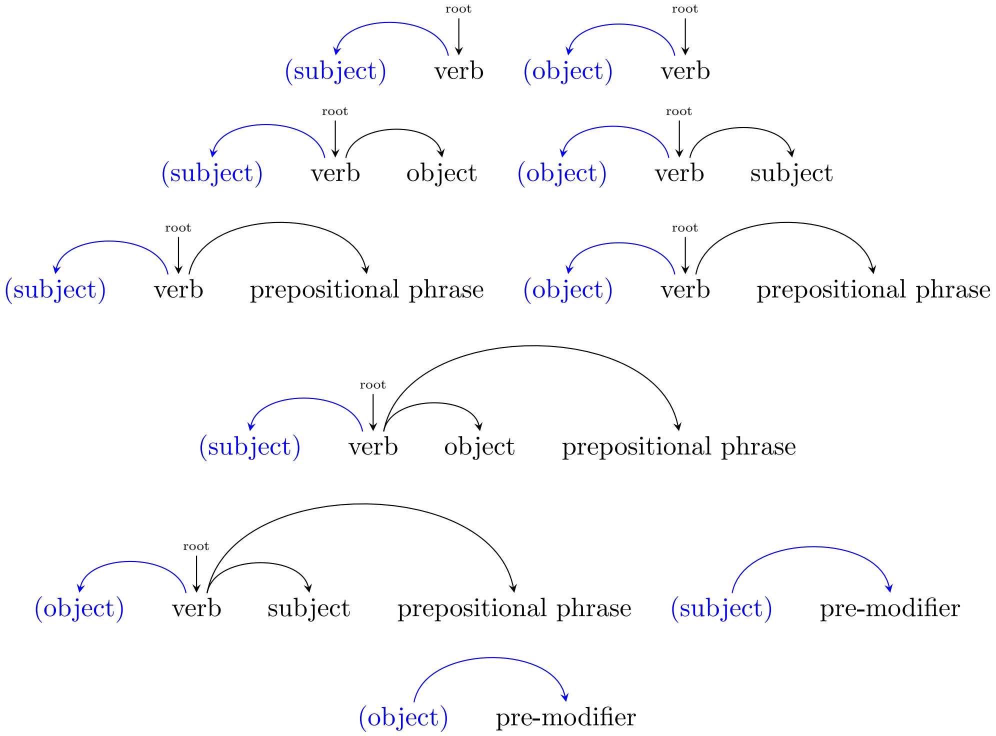

root
(subject) verb
root
(object) verb
root
(subject) verb object
root
(object) verb subject
root
(subject) verb prepositional phrase
root
(object) verb prepositional phrase
root
(subject) verb object prepositional phrase
root
(object) verb subject prepositional phrase
(subject) pre-modifier
(object) pre-modifier

Figure 3.3: Dependency Pattern Types

Notice that some dependency patterns are simpler, with fewer fixed components, and thus the distribution of the variable component for a simpler dependency pattern forms a superset over sub-patterns with additional fixed components. Not all sextuplets will fill every component. In those cases, the sextuplet simply doesn't contribute to certain dependency pattern types.

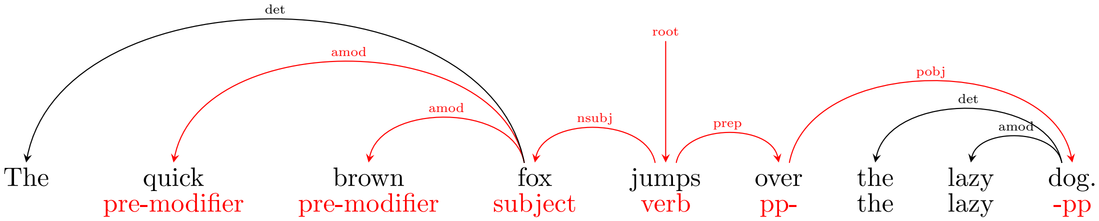


Figure 3.4: Dependency Parse with Sextuplet Components Highlighted

As a concrete example, consider the sentence parse in Figure 3.4. This sentence produces the sextuplet (*fox*, [*brown*, *quick*], *jumps*, *(empty)*, *(empty)*, *over dog*). From this sextuplet, we can consider all the dependency patterns generated of the types specified in Figure 3.3. This sentence generates the patterns displayed in Figure 3.5.

- (subject) verb[‘jump’]
- (subject) verb[‘jump’]-pp[‘over dog’]
- (subject) pre-modifier[‘brown’]
- (subject) pre-modifier[‘quick’]

Figure 3.5: Dependency patterns generated by the sentence in Figure 3.4. We show the patterns “flattened” for concision.

Since the sentence lacks an *object* component, each of the generated dependency patterns is defined only with the *subject* variable component. Note again, that the *subject* component appearing in the dependency patterns identified in Figure 3.5 is the variable, so while the word ‘fox’ forms the *subject* component of the corresponding sextuplet, it isn’t fixed when the *subject* component is the variable of a dependency pattern. Figure 3.6 highlights the pattern **(subject) verb[‘jump’]-pp[‘over dog’]** with the variable component name in blue and fixed components’ names in green; red highlighted components form part of the sextuplet, but are ignored by this dependency pattern.

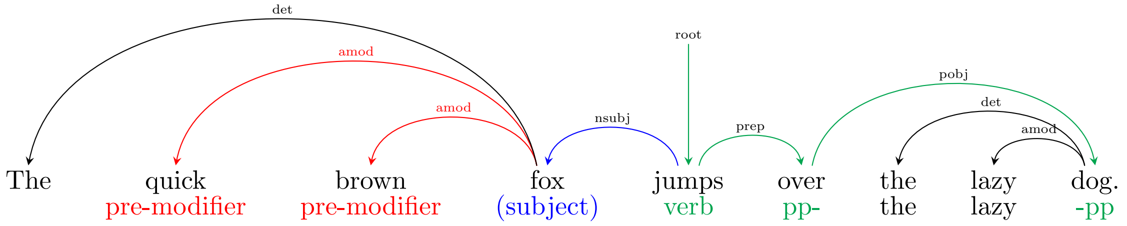
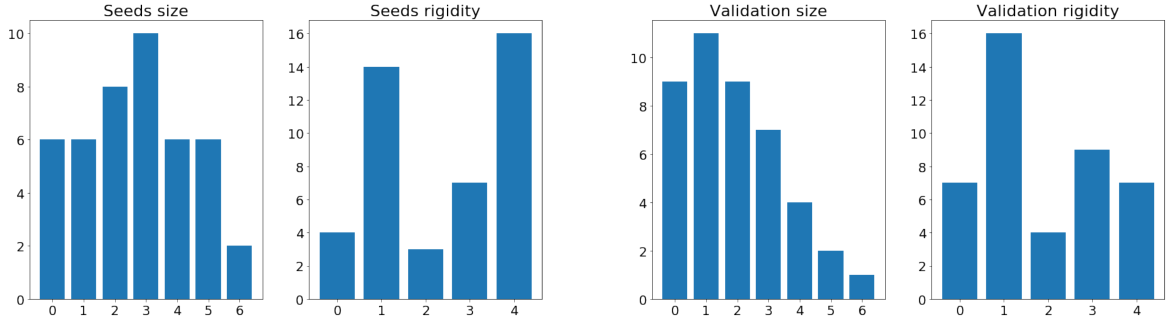


Figure 3.6: Color-coded Representation for **(subject) verb[‘jump’]-pp[‘over dog’]**

Both **(subject) verb[‘jump’]** and **(subject) verb[‘jump’]-pp[‘over dog’]** are generated by the sentence in Figure 3.4. This is an example of the case discussed before, where **(subject) verb[‘jump’]-pp[‘over dog’]** is a sub-pattern of **(subject) verb[‘jump’]**, since every sentence that generates the former, necessarily generates the latter. The sentence ‘Kangaroos can jump long distances.’ also generates the pattern **(subject) verb[‘jump’]** (this time with ’kangaroo’ under the variable element), but it does not generate the more specific form **(subject) verb[‘jump’]-pp[‘over dog’]**. Clearly the simpler dependency patterns will typically be much more common than one of its sub-patterns. This is largely why we limit the number of dependency templates, as more sophisticated or arbitrary structures are considered, the more sparse and redundant the data becomes.

3.4 Seed Selection

We start by splitting the dataset of annotated words form [Wang et al., 2018] into a validation and test set with 44 and 393 words respectively; approximately a 10%-90% split. We then begin to compile a list of seed nouns from the list of words rated by concreteness in [Brysbaert et al., 2014]. We first select each non-compound word from [Brysbaert et al., 2014] that was given the maximum concreteness rating. We determine the total frequency of each word in our corpus, and rank the resulting candidates from our list by their frequency in the corpus. From this list, we select 44 words to form the seed vocabulary, chosen to produce a relatively even distribution of attribute values for each attribute. Each of the seed words are then manually annotated using the same resolution and attribute types considered in [Wang et al., 2018]. When performing the manual annotation we consulted the validation set annotations in order to promote a consistency between the validation and seed words' labels. The full list of seed words with their physical attribute annotations is shown in Appendix A.



(a) Seed Set

(b) Validation Set

Figure 3.7: Distribution of Validation Set Attribute Values for Size and Rigidity

3.5 Features Representation

Our goal is to learn the physical attribute values of new words. We start off by looking at the sextuplets, and their dependency patterns, whose *subject* or *object* components contain words that are in our annotated seed vocabulary. Let DP be the set of all dependency patterns that co-occurred with at least one seed words as its variable argument. Then for each dependency pattern $dp_i \in DP$, we can consider the discrete function $P(A|dp_i)$ where A is the random variable representing the value of a particular physical attribute, and dp_i is a dependency pattern co-occuring with that word. We approximate this function by observing the frequency distribution of the attribute values of seeds that co-occur as the variable component of dp_i. We can do this because we know the physical attribute values for the annotated seed words. After this process, we then have a sample to approximate $P(A|dp_i)$ for each dependency pattern $dp_i \in DP$.

We want to eventually assign a specific attribute value to each new word for each physical attribute under consideration. That is, for a non-seed word that appears in the corpus, we want to learn some of its physical attributes; its approximate size or rigidity for example. After compiling the attribute value frequency distributions for each dependency pattern $dp_i \in DP$, we can extract new words that co-occur as the variable argument of each dependency pattern, but for which we have no physical attribute annotations. For each new word w, we can now calculate the frequency with which w occurs as the variable argument for each $dp_i \in DP$; we'll call this frequency c_i. Then for each new word $w \in W$, we define a vector $< c_0, c_1, ...c_N >$ representing a "bag of patterns" counting the co-occurrences of w with each $dp_i \in DP$. This "bag of patterns" vector, for each word $w \in W$, is the basic datatype we use to perform the following steps for feature selection and data filtering.

3.6 Feature Selection and Data Filtering

As one might expect, the bag of patterns representation for each word results in a sparse, high-dimensional feature space. We compile real distributional data for approximately 2 million unique dependency patterns. Additionally we extract about 2 million unique non-seed words that co-occurred at least once with a valid dependency pattern. In order to make this data and feature space more workable, we can filter the set of extracted words and we can perform feature selection.

We can filter the extracted word set in three ways: filtering lexically by excluding words with lemmas not contained in WordNet's vocabulary [Miller, 1995], filtering by frequency by excluding words with lower frequenncy counts, and filtering by variance by excluding words whose bag of patterns representation suggests high variance in expected attribute values. By excluding the list of words under consideration to those whose lemmas are contained by WordNet's vocabulary, we reduce the number of words extracted to about 40 thousand; a great reduction compared to the nearly 2 million initially considered. Upon inspection, seemingly, many of the words filtered out by this method include proper nouns, obscure, scientific, and technical terms, and more abstract words without physical properties. Since physical attribute data is not as meaningful for these words, we can exclude them to promote a greater semantic consistency of the candidate words. For frequency filtering, we are specifically considering the total number of times that a word occurs under the variable element of those dependency patterns in our feature space. For variance filtering, we compute a word's sample variance by modeling its "predictions" $X = \underset{a \in A}{\operatorname{argmax}} P(a|dp_i)$ for $dp_i \in DP$, weighted by c_i, and then filter out words for which $\operatorname{Var}(X)$ is greater than a specified threshold.

For feature selection, we can also reduce the number of features based upon frequency and variance. When performing filtering based on variance for patterns, we simply compute the variance of a pattern's variables' attribute value for the attribute under consideration. Even after such filtering though, we typically end up with hundreds or thousands of dependency patterns comprising the feature space. Since the number of labelled instances is so few, we perform z-score normalization with respect to each feature before performing dimensionality reduction on the entire feature set, after filtering, using scikit-learn's incremental PCA [Pedregosa et al., 2011] with batch size 2048 before training models for classification. Notice then, that the variance and frequency parameters for the feature set and words considered have an effect on the resulting reduced representation.

The values for each frequency and variance threshold in the data filtering and feature selection preproocessing step produces its own parameter space for which we perform a grid search over the following values:

- Pattern frequency threshold ($\geq$) $\in \{10, 25, 50, 100, 150\}$
- Pattern variance threshold ($\leq$) $\in \{0.1, 0.5, 1.0, 1.5, 2.0\}$
- Word frequency threshold ($\geq$) $\in \{5, 25, 50, 100\}$

- Word variance threshold $(\leq)$ $\in \{0.1, 0.5, 1.0, 1.5, 3.0, 4.0, 5.0\}$
- PCA components $\in \{25, 60, 100, 200\}$

As performing PCA is a space intensive operation, some combinations of feature selection and data filtering parameters which result in a very large feature matrix are omitted from consideration.

The final choice of parameters can be seen in Table 3.1. These values were attained during parameter tuning, with the best model being chosen by F1 micro average score on the validation set.

Model	Pattern Freq.	Pattern Var.	# Patterns	Word Freq.	Word Var.	# Words	PCA Components
MLP	75	1.0	1,145	25	5.0	16,812	60
KNN	75	0.5	769	5	5.0	**20,460**	60
SVM	75	0.5	769	25	5.0	10,396	100

(a) Size

Model	Pattern Freq.	Pattern Var.	# Patterns	Word Freq.	Word Var.	# Words	PCA Components
MLP	75	0.5	1,035	50	4.0	14,204	60
KNN	75	1.0	1,525	25	5.0	**20,924**	25
SVM	75	0.5	1,035	50	4.0	17,924	60

(b) Rigidity

Table 3.1: Data Filtering, Feature Selection Thresholds, and PCA Components Selected for Evaluation.

3.7 Machine Learning Models

We proceed to perform attribute prediction using vectors from the transformed matrix. We try out 3 different machine learning models: a k-nearest neighbors classifier, a multi-layer perceptron, and a support vector machine classifier using scikit-learn's implementations [Pedregosa et al., 2011].

Using the validation set of the 44 annotated words from [Wang et al., 2018], we perform minimal parameter tuning for each model's particular hyper parameters; leaving unspecified parameters as their default values in scikit-learn. For the k-nearest neighbors classifier, we compare values of $k \in \{1, 2\}$; for the multi-layer perceptron classifier, we compare values of the learning rate in $\{1e-2, 1e-3, 1e-4\}$ with 1, 2, or 3 hidden layers with sizes of $\{(64,), (64, 32), (128, 64, 32)\}$ respectively; for the support vector machine classifier we use the 'rbf' kernel and compare values of $C \in \{0.5, 1.0, 2.0, 3.0, 5.0, 8.0, 10.0, 14.0\}$. The final models' parameter values, shown in Table 3.2, were chosen based upon performance of their F1 Micro Average, as shown in Table 4.1.

Model	Parameters
MLP	learning rate: 1e-2, hidden layers: {64, 32}
KNN	k: 1
SVM	C: 14.0

(a) Size

Model	Parameters
MLP	learning rate: 1e-2, hidden layers: {64, 32}
KNN	k: 2
SVM	C: 14.0

(b) Rigidity

Table 3.2: Evaluation Model Parameter Specification

4 Evaluation

To evaluate the attribute predictions produced by each model, we compare the learned physical attributes to the gold data from [Wang et al., 2018] not in the validation set used for parameter tuning. This work focused on learning words' size and rigidity attributes, though the same procedure could be performed for any of the attributes considered in [Wang et al., 2018], as shown in Table 2.1.

After parameter tuning, an interesting result is that the top performing parameters for the data filtering and feature selection methods have a common pattern across the different models; stricter thresholds for patterns, and less strict thresholds for words. For the models trained to classify for size, for example, all of the evaluation models selected a 75 pattern frequency threshold, 0.5 or 1.0 pattern variance threshold, and more liberal thresholds for filtering words. This specification filters the patterns to a relatively small number, and suggests that the models may rely significantly on the data filtering process for patterns, and that low variance patterns are inherently informative.

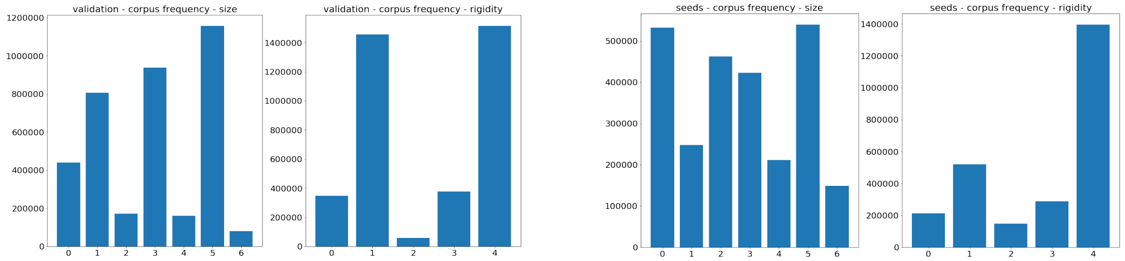
Observing the patterns that do get through, it is clear that some number are arbitrary and likely uninformative. On the size attribute with pattern filtering parameters of 75 and 0.5 for frequency and variance thresholds respectively, the pattern **(SUBJECT, subject-pre-modifier[‘cylinder’])** is attested with the seed words *bank, trophy, anchor, crown, cabin, lighthouse, bull, barrel,* and *motorcycle*, all contributing to a total of 77 instances for size attributes for this pattern. Yet, *barrel* contributes the majority of these instances, producing a low variance pattern with a fairly definite expected value. Analyzing the words extracted by this pattern however led to a somewhat different picture; some of the words most commonly co-occurring with this pattern include *locomotive, car, submarine*, and *engine*, suggesting a potentially different real variance and expected value. Of course the pre-modifier ‘cylinder’ has a very common and definite meaning with respect to engines and vehicles, but the choice of seed words may have biased the feature distribution in some arbitrary way.

There are however examples of much more intuitive examples of useful and informative patterns. Again with the same pattern filtering parameters, one attested pattern is **(subject[‘–pron–’], verb[‘drink’], OBJECT)**, with the most common seed words contributing being *wine* and *coffee* with common physical attributes. The extracted words most commonly co-occurring with this pattern include other liquids, such as *water, alcohol, blood,* and *beer.*

4.1 Results

The final results for each model are shown in Table 4.1, shown as the F1 micro average. With more in depth analysis of the results for precision, recall, and F1 score per attribute value for the validation and evaluation sets on the size attribute are shown in Tables 4.2 and 4.3. For the rigidity attribute, results are shown in Tables 4.4 and 4.5. Upon inspection, the results of predictions for the size attribute, we see that the resulting F1 scores suggest moderate performance, with better results for the rigidity attribute than size. This may be related to an imbalance in both the attribute values distributions of the validation and seed sets, shown in Figure 3.7, as well as the distribution of attribute values attested in the corpus, as seen in Figure 4.1. In particular, the very poor performance on attributes values 2 in rigidity, and 6 in size, may be due to the relatively low frequencies that these attribute values are attested to in the corpus.

Performance on the validation set outpaces that on the evaluation set by a significant margin, though relative performance on the validation set during parameter tuning was highly correlated with performance on the evaluation set. Overall, it would seem that the best performing model was the multi layer perceptron on both the size and rigidity attributes, based on final testing F1 micro average scores. The support vector machine also performed fairly well, especially on the validation set, though it seemed to not generalize quite as well as the multi layer perceptron. Unfortunately, precision and recall differ significantly between attribute values for all models, meaning that the error per class is inconsistent, a somewhat undesirable trait.



(a) Validation Set

(b) Seed Set

Figure 4.1: Attribute Value Frequency in Wikipedia, Based on Word Count

	Size		Rigidity	
Model	Validation	Testing	Validation	Testing
MLP	0.51	0.33	0.56	0.43
KNN	0.48	0.29	0.53	0.40
SVM	0.50	0.31	0.58	0.37

Table 4.1: Final Model F1 Micro Average

	Attr. Val	0	1	2	3	4	5	6
MLP	Precision	0.71	0.56	0.45	0.56	0.50	0.25	0.00
	Recall	0.56	0.42	0.56	0.71	0.25	0.50	0.00
	F1	**0.63**	0.48	0.50	**0.63**	0.33	0.33	0.00
KNN	Precision	1.00	0.75	0.44	0.27	0.40	0.00	0.00
	Recall	0.33	0.50	0.44	0.43	0.50	0.00	0.00
	F1	0.50	0.60	0.44	0.33	0.44	0.00	0.00
SVM	Precision	0.33	0.67	0.56	0.36	1.00	0.50	0.00
	Recall	0.11	0.67	0.56	0.57	0.75	0.50	0.00
	F1	0.17	**0.67**	**0.56**	0.44	**0.86**	**0.50**	0.00
	# words	9	12	9	7	4	2	1

Table 4.2: Results for Size Attribute on Validation Set

	Attr. Val	0	1	2	3	4	5	6
MLP	Precision	0.37	0.33	0.25	0.56	0.21	0.37	0.33
	Recall	0.21	0.40	0.24	0.38	0.14	0.42	0.22
	F1	**0.27**	**0.37**	0.25	**0.46**	0.16	**0.39**	**0.27**
KNN	Precision	0.28	0.48	0.23	0.40	0.15	0.22	0.06
	Recall	0.24	0.22	0.27	0.36	0.20	0.31	0.11
	F1	0.26	0.30	0.25	0.38	0.17	0.25	0.08
SVM	Precision	0.38	0.41	0.25	0.44	0.30	0.35	0.00
	Recall	0.13	0.29	0.40	0.48	0.14	0.23	0.00
	F1	0.20	0.34	**0.30**	0.46	**0.19**	0.30	0.00
	# words	67	99	62	86	44	26	9

Table 4.3: Results for Size Attribute on Test/Evaluation Set

	Attr. Val	0	1	2	3	4
MLP	Precision	0.50	0.74	1.00	0.63	0.40
	Recall	0.14	0.88	0.25	0.56	0.50
	F1	0.22	0.80	**0.40**	**0.59**	**0.44**
KNN	Precision	0.63	0.80	0.33	0.36	0.33
	Recall	0.71	0.75	0.25	0.56	0.13
	F1	0.67	0.77	0.29	0.43	0.18
SVM	Precision	0.71	0.92	0.00	0.57	0.29
	Recall	0.71	0.75	0.00	0.44	0.50
	F1	**0.71**	**0.83**	0.00	0.50	0.36
	# words	7	16	4	9	8

Table 4.4: Results for Rigidity Attribute on Validation Set

	Attr. Val	0	1	2	3	4.
MLP	Precision	0.41	0.74	0.33	0.20	0.30
	Recall	0.29	0.66	0.14	0.35	0.23
	F1	**0.34**	**0.70**	**0.20**	0.25	0.26
KNN	Precision	0.25	0.78	0.00	0.22	0.37
	Recall	0.52	0.54	0.00	0.34	0.31
	F1	**0.34**	0.64	0.00	**0.26**	0.34
SVM	Precision	0.28	0.65	0.00	0.21	0.32
	Recall	0.37	0.54	0.00	0.22	0.40
	F1	0.31	0.59	0.00	0.21	**0.36**
	# words	52	149	49	59	84

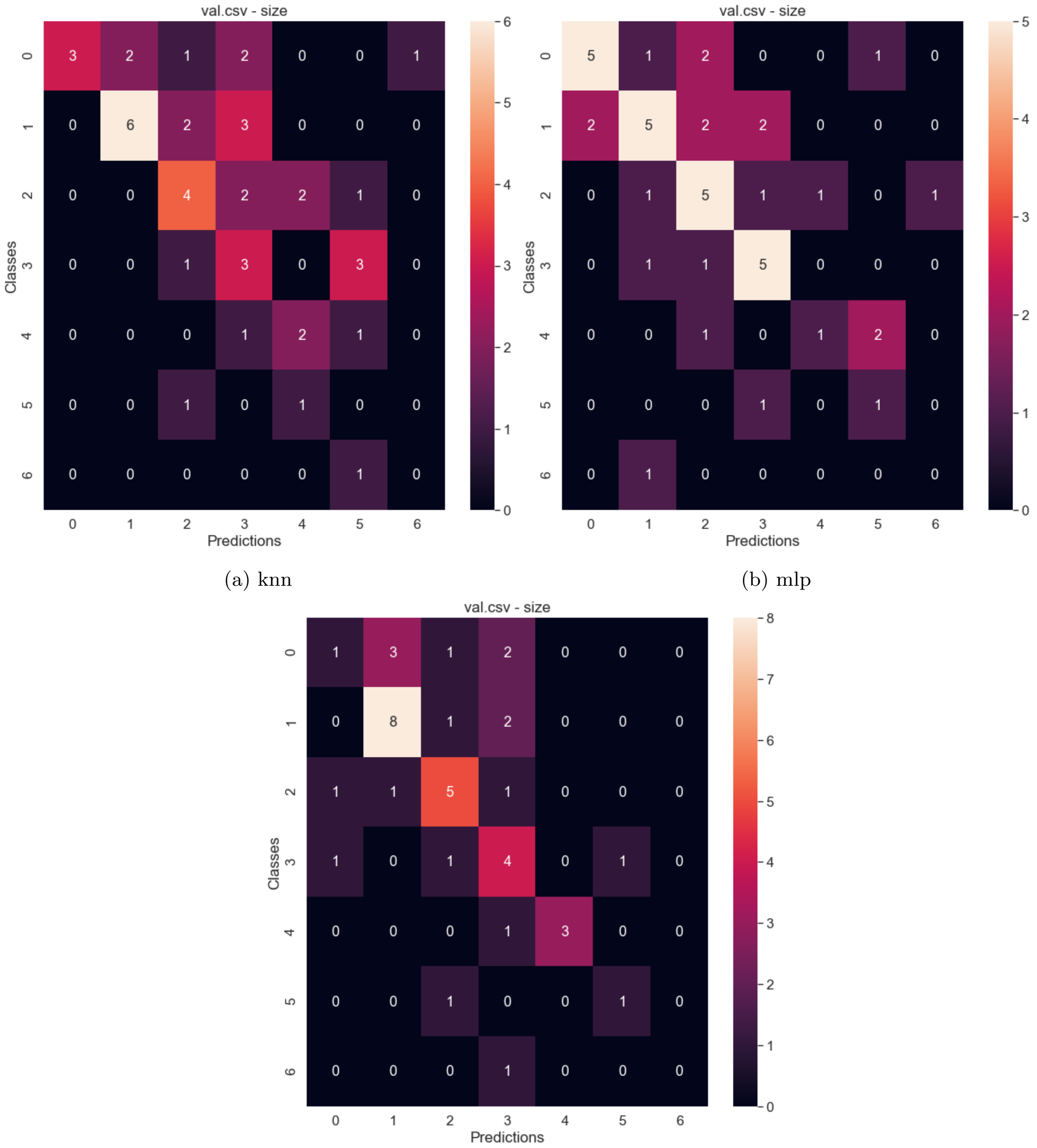
Table 4.5: Results for Rigidity Attribute on Test/Evaluation Set

We additionally evaluate the quality of the learned attributes by comparing the results to the relative object pair dataset from [Forbes and Choi, 2017], consisting of 3,656 pairs of objects labelled along the physical attributes *size, weight, strength, rigidness,* and *speed*, and split into a (183 / 1645 / 1828) seed, development, and test set. This dataset considers the relation between an object pair, whether one object is bigger, heavier, stronger, more rigid, or faster than other. We can see how our learned size and rigidity attribute values compare to the relative attribute object pairs in Table 4.6. Here the multi layered perceptron performed best for the size attribute, with slightly better results on rigidity. Interestingly the KNN model performs relatively well, and outperfoms other models on the rigidity attribute, despite performing slightly worse on average for the [Wang et al., 2018] test set. The performance of each model is again moderate, with models producing somewhat noisier results on this dataset than the method used in [Forbes and Choi, 2017].

	Test Acc.	
Model	Size	Rigidity
[Forbes and Choi, 2017]	0.75	0.75
MLP	0.62	0.48
KNN	0.53	0.54
SVM	0.57	0.43
Random	0.33	0.33

Table 4.6: Relative Attribute Comparison

To get a better idea of how our models are behaving, the confusion matrices in Figures 4.2 and 4.3 of our models show mostly issues with precision, while usually capturing the general tendencies of words' physical attributes. For most attribute values, the plurality of word predictions are correct, and most mislabels fall within an adjacent value bin, suggesting that though the labels are somewhat noisy, their utility in downstream tasks will depend on how important the resolution is for that task. Nonetheless, these results may be particularly useful for downstream tasks that utilize a larger vocabulary, since manually annotating thousands or tens of thousands of words is infeasible.

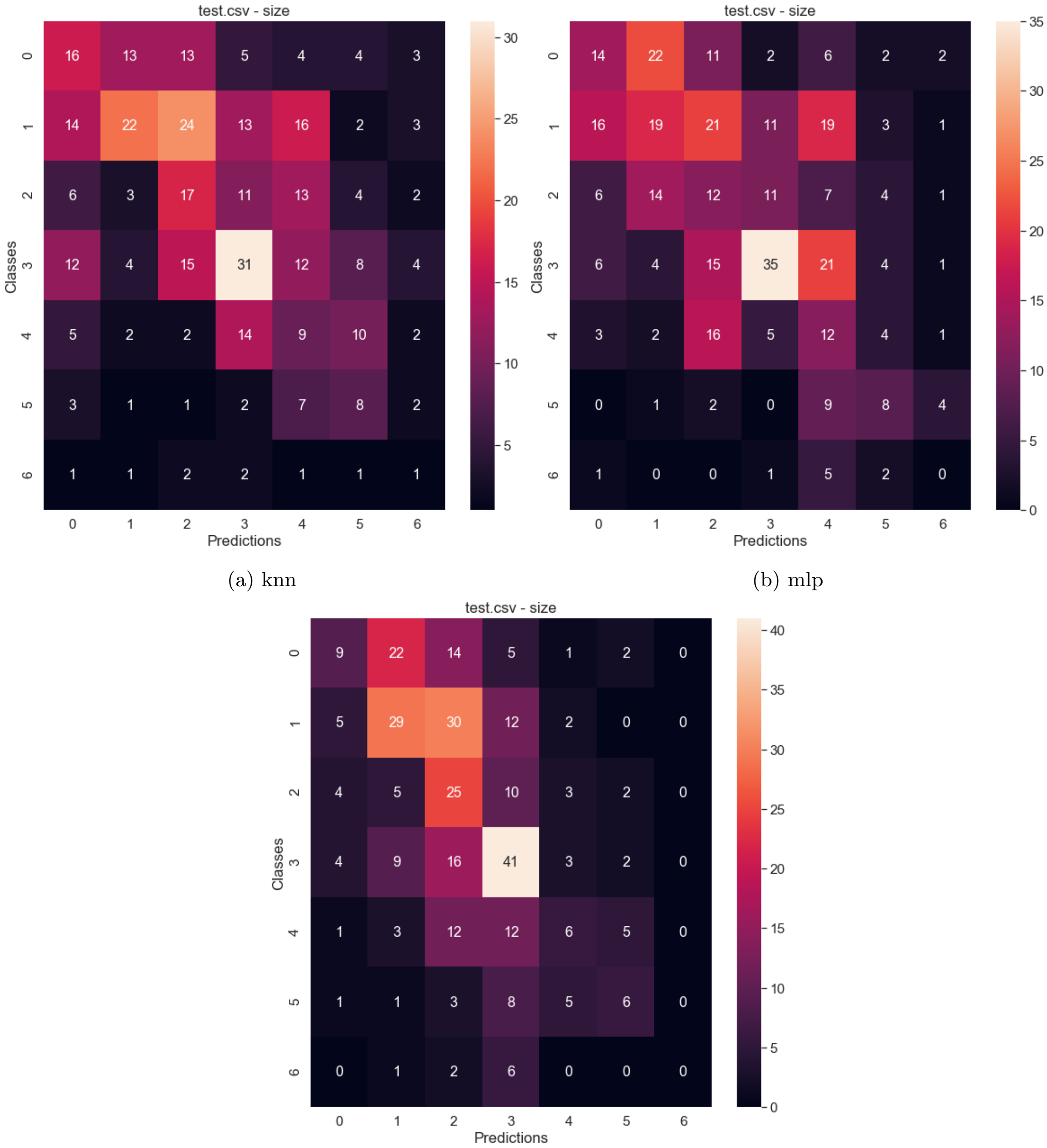


(a) knn

(b) mlp

(c) svc

Figure 4.2: Confusion Matrix for Validation Set on Size Attribute



Classes \ Predictions	0	1	2	3	4	5	6
0	16	13	13	5	4	4	3
1	14	22	24	13	16	2	3
2	6	3	17	11	13	4	2
3	12	4	15	31	12	8	4
4	5	2	2	14	9	10	2
5	3	1	1	2	7	8	2
6	1	1	2	2	1	1	1

(a) knn

Classes \ Predictions	0	1	2	3	4	5	6
0	14	22	11	2	6	2	2
1	16	19	21	11	19	3	1
2	6	14	12	11	7	4	1
3	6	4	15	35	21	4	1
4	3	2	16	5	12	4	1
5	0	1	2	0	9	8	4
6	1	0	0	1	5	2	0

(b) mlp

Classes \ Predictions	0	1	2	3	4	5	6
0	9	22	14	5	1	2	0
1	5	29	30	12	2	0	0
2	4	5	25	10	3	2	0
3	4	9	16	41	3	2	0
4	1	3	12	12	6	5	0
5	1	1	3	8	5	6	0
6	0	1	2	6	0	0	0

(c) svc

Figure 4.3: Confusion Matrix for Evaluation Set on Size Attribute

5 Conclusions and Future Work

This work contributes a means for acquiring large amounts of medium-resolution physical attribute knowledge using a small amount of seed data. Analysis of results suggests that the produced attribute knowledge is reasonable, if somewhat noisy, and can likely be useful in downstream tasks, for which manual annotation would be unreasonably time and resource consuming.

5.1 Potential Improvements

In retrospect to this study, a few potential means for improvement identified include: normalizing active/passive voice, which was not done in this study when extracting sextuplet representations, performing seed selection to guarantee a more balanced extraction of dependency patterns and words from the corpus, using more powerful machine learning models to make attribute value predictions from contextual representations alone.

We might also want to consider the role that different resolutions for each attribute plays in defining the quality of acquired knowledge. The attribute resolution is fundamental to how dependency patterns are selected, and likely have a significant effect on the functions learned by each machine learning model. The resolution scheme we used was chosen in part to take advantage of the dataset produced by [Wang et al., 2018], and though we like a medium resolution scheme in general, it is not clear exaclty how specific resolutions affect the knowledge acquisition process.

5.2 Weak Labelling for Semantic Plausibility

A recent solution to the semantic plausibility task of [Wang et al., 2018], involves finetuning a BERT model [Devlin et al., 2019], a large pretrained neural model, on the plausibility dataset [Porada et al., 2019]. This paper trains the large uncased BERT model on the plausibility dataset with 10-fold cross validation of the 3,062 event examples. The training procedure is identical to the models proposed and evaluated by [Wang et al., 2018], with results in Table 5.1. The NN model refers to the selectional preference-based neural network model proposed in [de Cruys, 2014], while NN+WK refers to the model discussed in Table 2.1 which enriches the NN model with explicit world knowledge [Wang et al., 2018]. The nature of the plausibility dataset, and the method of training in the supervised example should be detailed more explicitly. Specifically, as the data has a limited vocabulary (150 verbs, and 450 nouns), the folds are organized such that the vocabulary is effectively spread somewhat evenly across each of the folds, so that the vocabulary of each 9-fold training set spans that of the test set as well. In addition to having a fairly high training:test size ratio, this detail would seem to make the task a bit easier. Nevertheless, the BERT model obviously outperforms the prior models significantly.

A more interesting datum from this [Porada et al., 2019], perhaps, is another approach to finetuning the BERT model, which uses no data from the plausibility task at all. In this implementation, they create a ‘self supervised’ dataset of (*subject, verb, object*) events by extracting events from English Wikipedia. The extracted events represent the physically plausible events in the self supervised paradigm. They then create an equal number of pseudo-negative examples by sampling *subjects, verbs,* and *objects* based upon frequency to generate events that do not occur in the corpus (i.e., implausible events). After this process, they generate a dataset with 12 million events (6 million positive, and 6 million pseudo-negative) for finetuning the BERT model. For evaluation, the plausibility dataset from [Wang et al., 2018] is split into two 1,531 sets of triples for validation and testing, and training on the self supervised dataset for the BERT model, as well as the NN model (without world knowledge). Results are shown in Table 5.2. Clearly the results for the self supervised dataset suggest that both the BERT and NN model fail is some significant way to learn plausibility as derived in this scheme from natural text. As such, the attributes we have learned could be used as weak labels to enrich the great majority of the events in the self supervised dataset, hopefully providing a better measure of the impact of physical world knowledge for plausibility tasks from natural text.

Model	10-fold CV Mean Acc.
Random	0.50
NN [de Cruys, 2014]	0.68
NN+WK [Wang et al., 2018]	0.76
Fine-tuned BERT	**0.89**

Table 5.1: Supervised Results for Plausibility Task

Model	Valid	Test
Random	0.50	0.50
NN [de Cruys, 2014]	0.53	0.52
Fine-tuned BERT	**0.65**	**0.63**

Table 5.2: Self Supervised Results for Plausibility Task

Appendices

A Seed Words

word	sentience	masscount	phase	size	weight	rigidity
gold	0	3	2	0	1	4
bank	0	3	2	5	5	4
crown	0	3	2	2	2	4
factory	0	3	2	6	6	4
trophy	0	3	2	2	2	4
priest	5	3	2	3	3	1
restaurant	0	3	2	5	5	4
wine	0	0	1	1	1	0
salt	0	1	2	0	0	0
ferry	0	3	2	4	3	3
timber	0	2	2	4	4	3
volleyball	0	3	2	2	1	2
grandfather	5	3	2	3	3	1
beetle	2	3	2	0	0	2
dogs	3	3	2	2	2	1
eagle	3	3	2	2	1	1
telephone	0	3	2	1	1	3
coffee	0	0	1	1	1	0
clock	0	3	2	2	1	3
bull	3	3	2	4	4	2
photographer	5	3	2	3	3	1
jet	0	3	2	5	5	4
mirror	0	3	2	3	2	4
sheep	3	3	2	3	3	1
dancer	5	3	2	3	3	1
anchor	0	3	2	3	4	4
deer	3	3	2	3	3	1
violin	0	3	2	3	2	3
runner	5	3	2	3	3	1
motorcycle	0	3	2	4	4	4
lighthouse	0	3	2	5	5	4
cabin	0	3	2	5	5	4
casino	0	3	2	6	6	4
snail	2	3	2	0	0	1
cottage	0	3	2	5	5	4
coin	0	3	2	0	0	4
barrel	0	3	2	4	4	3
toy	0	3	2	1	1	3
dish	0	3	2	2	1	4

photograph	0	3	2	1	0	1
frog	3	3	2	1	1	1
duck	3	3	2	2	2	1
smell	0	0	0	0	0	0
chimp	4	3	2	4	4	1

References

[Brysbaert et al., 2014] Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46:904–911.

[de Cruys, 2014] de Cruys, T. V. (2014). A neural network approach to selectional preference acquisition. In *EMNLP*, Doha, Qatar.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Durme, 2010] Durme, B. V. (2010). *Extracting implicit knowledge from text.* PhD thesis, University of Rochester, Rochester, NY 14627.

[Elazar et al., 2019] Elazar, Y., Mahabal, A., Ramachandran, D., Bedrax-Weiss, T., and Roth, D. (2019). How large are lions? inducing distributions over quantitative attributes. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.*

[Forbes and Choi, 2017] Forbes, M. and Choi, Y. (2017). Verb physics: Relative physical knowledge of actions and objects. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).*

[Forbes et al., 2019] Forbes, M., Holtzman, A., and Choi, Y. (2019). Do neural language representations learn physical commonsense? *ArXiv*, abs/1908.02899.

[Grice, 1975] Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York.

[Havasi and Alonso, 2007] Havasi, C. and Alonso, J. B. (2007). Conceptnet 3 : a flexible , multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing (RANLP-07)*, Borovets, Bulgaria. Association for Computational Linguistics.

[Hermann et al., 2012] Hermann, K. M., Dyer, C., Blunsom, P., and Pulman, S. (2012). Learning semantics and selectional preference of adjective-noun pairs. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 70–74, Montréal, Canada. Association for Computational Linguistics.

[Katz and Fodor, 1963] Katz, J. J. and Fodor, J. A. (1963). The structure of a semantic theory. *Language*, 39(2):170–210.

[Levesque et al., 2012] Levesque, H. J., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561, Rome, Italy. AAAI Press.

[Lucy and Gauthier, 2017] Lucy, L. and Gauthier, J. (2017). Are distributional representations ready for the real world? evaluating word vectors for grounded perceptual meaning. In *RoboNLP@ACL*, Vancouver, Canada.

[Miller, 1995] Miller, G. A. (1995). Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41.

[Ó Séaghdha, 2010] Ó Séaghdha, D. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444, Uppsala, Sweden. Association for Computational Linguistics.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Porada et al., 2019] Porada, I., Suleman, K., and Cheung, J. C. K. (2019). Can a gorilla ride a camel? learning semantic plausibility from text. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*. Association for Computational Linguistics.

[Resnik, 1997] Resnik, P. (1997). Selectional preference and sense disambiguation. In *Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, DC. Association for Computational Linguistics.

[Rubinstein et al., 2015] Rubinstein, D., Levi, E., Schwartz, R., and Rappoport, A. (2015). How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China. Association for Computational Linguistics.

[Saba, 2019] Saba, W. (2019). On the winograd schema: Situating language understanding in the data-information-knowledge continuum. In *Florida Artificial Intelligence Research Society Conference*, Melbourne, Florida. AAAI Press.

[Tandon et al., 2014] Tandon, N., De Melo, G., and Weikum, G. (2014). Acquiring comparative commonsense knowledge from the web. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 166–172, Québec City, Québec, Canada. AAAI Press.

[Wang et al., 2018] Wang, S., Durrett, G., and Erk, K. (2018). Modeling semantic plausibility by injecting world knowledge. In *NAACL-HLT*, Minneapolis, Minnesota.

[Yang et al., 2018] Yang, Y., Birnbaum, L., Wang, J.-P., and Downey, D. (2018). Extracting commonsense properties from embeddings with limited human guidance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 644–649, Melbourne, Australia. Association for Computational Linguistics.