Extracting Commonsense Properties from Embeddings with Limited Human Guidance

Yiben Yang [1], Larry Birnbaum[2], Ji-Ping Wang[1], Doug Downey[2]

[1]Department of Statistics, Northwestern University, Evanston, IL, 60208, USA
[2]Department of Electrical Engineering & Computer Science, Northwestern University, Evanston, IL, 60208, USA
[1]{yiben.yang, jzwang}@northwestern.edu
[2]{l-birnbaum, d-downey}@northwestern.edu

Abstract

Intelligent systems require common sense, but automatically extracting this knowledge from text can be difficult. We propose and assess methods for extracting one type of commonsense knowledge, object-property comparisons, from pre-trained embeddings. In experiments, we show that our approach exceeds the accuracy of previous work but requires substantially less hand-annotated knowledge. Further, we show that an active learning approach that synthesizes common-sense queries can boost accuracy.

1 Introduction

Automatically extracting common sense from text is a long-standing challenge in natural language processing (Schubert, 2002; Van Durme and Schubert, 2008; Vanderwende, 2005). As argued by Forbes and Yejin (2017), typical language use may reflect common sense, but the commonsense knowledge itself is not often explicitly stated, due to reporting bias (Gordon and Van Durme, 2013). Thus, additional human knowledge or annotated training data are often used to help systems learn common sense.

In this paper, we study methods for reducing the amount of human input needed to learn common sense. Specifically, we focus on learning relative comparisons of (one-dimensional) object properties, such as the fact that a cantaloupe is *more round* than a hammer. Methods for learning this kind of common sense have been developed previously (e.g. Forbes and Choi, 2017), but the best-performing methods in that previous work requires dozens of manually-annotated frames for each comparison property, to connect the property to how it is indirectly reflected in text—e.g., if text asserts that "x carries y," this implies that x is probably *larger* than y.

Our architecture for relative comparisons follows the zero-shot learning paradigm (Palatucci et al., 2009). It takes the form of a neural network that compares a projection of embeddings for each of two objects (e.g. "elephant" and "tiger") to the embeddings for the two poles of the target dimension of comparison (e.g., "big" and "small" for the size property). The projected object embeddings are trained to be closer to the appropriate pole, using a small training set of hand-labeled comparisons. Our experiments reveal that our architecture outperforms previous work, despite using less annotated data. Further, because our architecture takes the property (pole) labels as arguments, it can extend to the zero-shot setting in which we evaluate on properties not seen in training. We find that in zero-shot, our approach outperforms baselines and comes close to supervised results, but providing labels for both poles of the relation rather than just one is important. Finally, because the number of properties we wish to learn is large, we experiment with active learning (AL) over a larger property space. We show that synthesizing AL queries can be effective using an approach that explicitly models which comparison questions are nonsensical (e.g., is Batman taller than Democracy?). We release our code base and a new commonsense data set to the research community.[1]

2 Problem Definition and Methods

We define the task of comparing object properties in two different ways: a three-way classification task, and a four-way classification task. In the three-way classification task, we want to estimate the following conditional probability:

$$P(\mathbf{L}|\mathbf{O_1}, \mathbf{O_2}, \textbf{Property}), \mathbf{L} \in \{\boxed{<}, \boxed{>}, \boxed{\approx}\}.$$

[1] https://github.com/yangyiben/PCE

For example, $Prob($An elephant is larger than a dog) can be expressed as $P(\mathbf{L} = \boxed{>} | \mathbf{O_1} = "elephant", \mathbf{O_2} = "dog", \mathbf{Property} = "size")$. The three-way classification task has been explored in previous work (Forbes and Choi, 2017) and is only performed on triples where both objects have the property, so that the comparison is meaningful. In applications, however, we may not know in advance which comparisons are meaningful. Thus, we also define a four-way classification task to include "not applicable" as the fourth label, so that inference can be performed on any object-property triples. In the four-way task, the system is tasked with identifying the nonsensical comparisons. Formally, we want to estimate the following conditional probability:

$$P(\mathbf{L}|\mathbf{O_1}, \mathbf{O_2}, \mathbf{Property}), \mathbf{L} \in \{\boxed{<}, \boxed{>}, \boxed{\approx}, \boxed{\text{N/A}}\}.$$

2.1 Three-way Model

For each comparison property, we pick an adjective and its antonym to represent the $\{\boxed{<}, \boxed{>}\}$ labels. For example, for the property *size*, we pick "big" and "small". The adjective "similar" serves as the label for $\boxed{\approx}$ for all properties. Under this framework, a relative comparison question, for instance, "Is a dog bigger than an elephant?", can be formulated as a quintuple query to the model, namely {dog, elephant, small, similar, big}. Denoting the word embeddings for tokens in a quintuple query as X, Y, $R_<$, $R_\approx$, $R_>$, our three-way model is defined as follows:

$$P(\mathbf{L} = s|Q) = softmax(R_s \cdot \sigma((X \oplus Y)W)),$$

for $s \in \{<, >, \approx\}$, where Q is an quintuple query, $\sigma(\cdot)$ is an activation function and W is a learnable weight matrix. The symbol $\oplus$ represents concatenation. We refer to this method as **PCE** (**P**roperty **C**omparison from **E**mbeddings) for the 3-way task. We also experiment with generating label representations from just a single adjective (property) embedding $R_<$, namely $R_\approx = \sigma(R_< W_2)$, $R_> = \sigma(R_< W_3)$. We refer to this simpler method as **PCE(one-pole)**.

We note that in both the three- and four-way settings, the question "A>B?" is equivalent to "B<A?". We leverage this fact at test time by feeding our network a reversed object pair, and taking the average of the aligned network outputs before the softmax layer to reduce prediction variance. We refer to our model without this technique as **PCE(no reverse)**.

The key distinction of our method is that it learns a projection from the object word embedding space to the label embedding space. This allows the model to leverage the property label embeddings to perform zero-shot prediction on properties not observed in training. For example, from a training example "dogs are smaller than elephants", the model will learn a projection that puts "dogs" relatively closer to "small," and far from "big" and "similar." Doing so may also result in projecting "dog" to be closer to "light" than to "heavy," such that the model is able to predict "dogs are lighter than elephants" despite never being trained on any weight comparison examples.

2.2 Four-way Model

Our four-way model is the same as our three-way model, with an additional module to learn whether the comparison is applicable. Keeping the other output nodes unchanged, we add an additional component into the softmax layer to output the probability of "N/A":

$$h_x = \sigma(XW_a),\ h_y = \sigma(YW_a),$$
$$A_i = h_i \cdot R_> + h_i \cdot R_<,$$
$$P(\mathbf{L} = \boxed{\text{N/A}}|Q) \propto exp(A_x + A_y).$$

2.3 Synthesis for Active Learning

We propose a method to synthesize informative queries to pose to annotators, a form of active learning (Settles, 2009). We use the common heuristic that an informative training example will have a high uncertainty in the model's predictive distribution. We adopt the confidence measure (Culotta and McCallum, 2005) to access the uncertainty of a given example:

$$Uncertainty(x) = 1 - \max_y P(y|x, D_{train}).$$

Good candidates for acquisition should have high uncertainty measure, but we also want to avoid querying outliers. As the vocabulary is finite, it is possible to evaluate the uncertainty measures for all possible inputs to synthesize the most uncertain query. However, such a greedy policy is expensive and prone to selecting outliers. Hence, we adopt a sampling based synthesis strategy: at each round, we generate one random object pair per property, and query the one that achieves the highest uncertainty measure.

A classical difficulty faced by synthesis approaches to active learning is that they may produce unnatural queries that are difficult for a human to label (Baum and Lang, 1992). However, our task formulation includes ”similar” and ”N/A” classes that encompass many of the more difficult or confusing comparisons, which we believe aids the effectiveness of the synthesis approach.

3 Experiments

We now present our experimental results on both the three-way and four-way tasks.

3.1 Data Sets

We test our three-way model on the VERB PHYSICS data set from (Forbes and Choi, 2017). As there are only 5 properties in VERB PHYSICS, we also develop a new data set we call PROPERTY COMMON SENSE. We select 32 commonsense properties to form our property set (e.g., value, roundness, deliciousness, intelligence, etc.). We extract object nouns from the McRae Feature Norms dataset (McRae et al., 2005) and add selected named entities to form a object vocabulary of 689 distinct objects. We randomly generate 3148 object-property triples, label them and reserve 45% of the data for the test set. We further add 5 manually-selected applicable comparison examples per property to our test set, in order to make sure each property has some applicable testing examples. To verify the labeling, we have a second annotator redundantly label 200 examples and find a Cohen’s Kappa of 0.64, which indicates good annotator agreement (we analyze the source of the disagreements in Section 4.1). The training set is used for the passive learning and pool-based active learning, and a human oracle provides labels in the synthesis active learning setting.

3.2 Experimental Setup

We experiment with three types of embeddings: **GloVe**, normalized 300-dimensional embeddings trained on a corpus of 6B tokens (Pennington et al., 2014) (the F&C method (Forbes and Choi, 2017) uses the 100-dimensional version, as it achieves the highest validation accuracy for their methods); **Word2vec**, normalized 300-dimensional embeddings trained on 100B tokens (Mikolov et al., 2013); and **LSTM**, the normalized 1024-dimensional weight matrix from the softmax layer of the Google 1B LSTM language model (Jozefowicz et al., 2016).

For training PCE, we use an identity activation function and apply 50% dropout. We use the Adam optimizer with default settings to train the models for 800 epochs, minimizing cross entropy loss. For zero-shot learning, we adopt a hold-one-property-out scheme to test our models’ zero-shot performance.

Finally, for active learning, we use Word2vec embeddings. All the models are trained on 200 random training examples to warm up. We train for 20 epochs after each label acquisition. To smooth noise, we report the average of 20 different runs of **random** (passive learning) and *least confident* (**LC**) pool-based active learning (Culotta and McCallum, 2005) baselines. We report the average of only 6 runs for an *expected model change* (**EMC**) pool-based active learning (Cai et al., 2013) baseline due to its high computational cost, and of only 2 runs for our synthesis active learning approach due to its high labeling cost. The pool size is 1540 examples.

3.3 Results

In Table 1, we compare the performance of the three-way PCE model against the existing state of the art on the VERB PHYSICS data set. The use of LSTM embeddings in PCE yields the best accuracy for all properties. Across all embedding choices, PCE performs as well or better than F&C, despite the fact that PCE does not use the annotated frames that F&C requires (approximately 188 labels per property). Thus, our approach matches or exceeds the performance of previous work using significantly less annotated knowledge. The lower performance of ”no reverse” shows that the simple method of averaging over the reversed object pair is effective.

Table 2 evaluates our models on properties not seen in training (zero-shot learning). We compare against a random baseline, and an Emb-Similarity baseline that classifies based on the cosine similarity of the object embeddings to the pole label embeddings (i.e., *without* the projection layer in PCE). PCE outperforms the baselines. Although the one-pole method was shown to perform similarly to the two-pole method for properties seen in training (Table 1), we see that for zero-shot learning, using two poles is important.

In Table 3, we show that our four-way models with different embeddings beat both the majority and random baselines on the PROPERTY

Model	Development						Test					
	size	weight	stren	rigid	speed	overall	size	weight	stren	rigid	speed	overall
Majority	0.50	0.54	0.51	0.50	0.53	0.51	0.51	0.55	0.52	0.49	0.50	0.51
F&C	0.75	0.74	0.71	0.68	0.66	0.71	0.75	0.76	0.72	0.65	0.61	0.70
PCE(LSTM)	0.79	**0.81**	0.75	**0.71**	**0.72**	**0.76**	**0.80**	**0.79**	0.76	**0.71**	0.71	**0.76**
PCE(GloVe)	0.75	0.75	0.71	0.67	0.69	0.71	0.76	0.75	0.71	0.68	0.68	0.72
PCE(Word2vec)	0.76	0.76	0.73	0.70	0.68	0.73	0.76	0.76	0.73	0.68	0.66	0.72
PCE(one-pole)	**0.80**	**0.81**	**0.77**	0.65	**0.72**	0.75	0.79	**0.79**	**0.77**	0.65	**0.72**	0.75
PCE(no reverse)	0.72	0.74	0.71	0.67	0.67	0.70	0.73	0.75	0.72	0.65	0.68	0.71

Table 1: Accuracy on the VERB PHYSICS data set. PCE outperforms the F&C model from previous work. PCE(one-pole) and PCE(no reverse) use LSTM embeddings.

Model	Development					Test				
	size	weight	stren	rigid	speed	size	weight	stren	rigid	speed
Random	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
Emb-Similarity	0.43	0.55	0.51	0.43	0.35	0.37	0.53	0.48	0.43	0.35
PCE(one-pole)	0.73	0.71	0.67	0.53	0.34	**0.74**	0.72	0.68	0.53	0.32
PCE	**0.76**	**0.72**	**0.71**	**0.62**	**0.60**	**0.74**	**0.73**	**0.70**	**0.62**	**0.58**

Table 2: Accuracy of zero-shot learning on the VERB PHYSICS data set(using LSTM embeddings). PCE outperforms the baselines, and using both poles is important for accuracy.

Model	Test
Random	0.25
Majority Class	0.51
PCE(GloVe)	0.63
PCE(Word2vec)	**0.67**
PCE(LSTM)	**0.67**

Table 3: Accuracy on the four-way task on the PROPERTY COMMON SENSE data.

COMMON SENSE data. Here, the LSTM embeddings perform similarly to the Word2vec embeddings, perhaps because the PROPERTY COMMON SENSE vocabulary consists of less frequent nouns than in VERB PHYSICS. Thus, the Word2vec embeddings are able to catch up due to their larger vocabulary and much larger training corpus.

Finally, in Figure 1, we evaluate in the active learning setting. The synthesis approach performs best, especially later in training when the training pool for the pool-based methods has only uninformative examples remaining. Figure 2 helps explain the relative advantage of the synthesis approach: it is able to continue synthesizing informative (uncertain) queries throughout the entire training run.

4 Discussion

4.1 Sources of annotator disagreement

As noted above, we found a "good" level of agreement (Cohen's Kappa of 0.64) for our PROPERTY COMMON SENSE data, which is lower than one might expect for task aimed at common sense. We analyzed the disagreements and found that they stem from two sources of subjectivity in the task. The first is that different labelers may have different thresholds for what counts as similar—a spider and an ant might be marked similar in size for one labeler, but not for another labeler. In our data, 58% of the disagreements are cases in which one annotator marks similar while the other says not similar. The second is that different labelers have different standards for whether a comparison is N/A. For example, in our data set, one labeler labels that a toaster is physically stronger than alcohol, and the other labeler says the comparison is N/A. 37% of our disagreements are due to this type of subjectivity. The above two types of subjectivity account for almost all disagreements

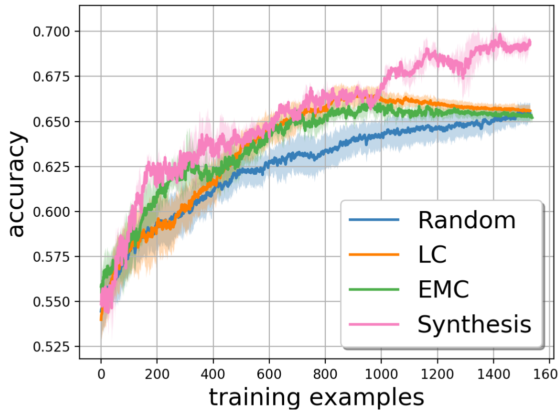

Figure 1: Test accuracy as a function of the number of queried training examples. The synthesis approach performs best.

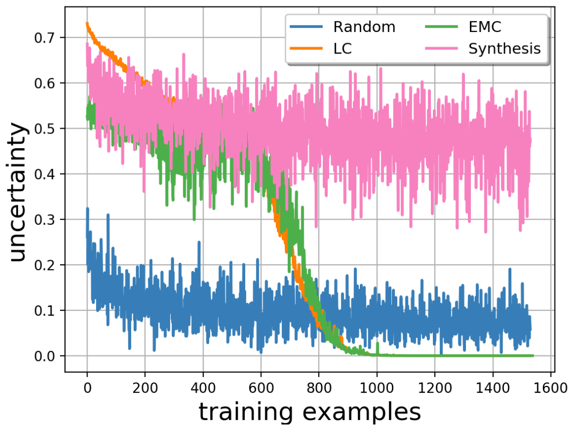

Figure 2: The uncertainty measure of each queried training example. As training proceeds, the synthesis approach continues to select more uncertain examples.

(95%), and the remaining 5% are due to annotation errors (one of the annotators makes mistake).

4.2 Model Interpretation

Since we adopt an identity activation function and a single layer design, it is possible to simplify the mathematical expression of our model to make it more interpretable. After accounting for model averaging, we have the following equality:

$$
\begin{aligned}
&P(\mathbf{L} =< |Q) \propto \\
&\quad exp(R_{<} \cdot ((X \oplus Y)W) + R_{>} \cdot ((Y \oplus X)W)) \\
&= exp(R_{<}^T(XW_1 + YW_2) + R_{>}^T(YW_1 + XW_2)) \\
&\propto exp((R_{<} - R_{>})^T(XW_1 + XW_2)),
\end{aligned}
$$

where $W = W_1 \oplus W_2$. So we can define a *score* of "$R_<$" for a object with embedding X as the following:

$$score(X, R_<) = (R_< - R_>)^T(XW_1 + XW_2).$$

An object with a higher score for $R_<$ is more associated with the $R_<$ pole than the $R_>$ one. For example, score("elephant","small") represents how small an elephant is—a larger score indicates a smaller object. Table 4 shows smallness scores for 5 randomly picked objects from the VERB PHYSICS data set. PCE tends to assign higher scores to the smaller objects in the set.

4.3 Sensitivity to pole labels

PCE requires labels for the poles of the target object property. Table 5 presents a limited sensitivity analysis to pole labels, evaluating the test accuracy of PCE as the pole label varies among different combinations of synonyms for the size and speed relations. We evaluate in both the trained setting (comparable to the results in Table 1) and the zero-shot setting (comparable to Table 2). We see that the trained accuracy remains essentially unchanged for different pole labels. In the zero-shot setting, all combinations achieve accuracy that beats the baselines in Table 2, but the accuracy value is somewhat sensitive to the choice of pole label. Exploring how to select pole labels and experimenting with richer pole representations such as textual definitions are items of future work.

Object	Smallness
restaurant	0.077
gully	0.416
lung	1.182
bow	4.036
scissors	14.492

Table 4: Scores of smallness for 5 randomly picked objects in VERB PHYSICS data set

Word choice	Trained	Zero
fast vs. slow	0.71	0.58
speedy vs. slow	0.71	0.56
fast vs. plodding	0.72	0.48
speedy vs. plodding	0.72	0.51
big vs. small	0.80	0.74
large vs. small	0.80	0.76
big vs. little	0.80	0.71
large vs. little	0.80	0.69

Table 5: Trained and zero-shot accuracies for different word choices

5 Conclusion

In this paper, we presented a method for extracting commonsense knowledge from embeddings. Our experiments demonstrate that the approach is effective at performing relative comparisons of object properties using less hand-annotated knowledge than in previous work. A synthesis active learner was found to boost accuracy, and further experiments with this approach are an item of future work.

Acknowledgments

This work was supported in part by NSF Grant IIS-1351029. We thank the anonymous reviewers for helpful comments.

References

Eric B Baum and Kenneth Lang. 1992. Query learning can work poorly when a human oracle is used. In *International joint conference on neural networks*, volume 8, page 8.

W. Cai, Y. Zhang, and J. Zhou. 2013. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th International Conference on Data Mining*, pages 51–60.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, pages 746–751. AAAI Press / The MIT Press.

Maxwell Forbes and Yejin Choi. 2017. Verb physics: Relative physical knowledge of actions and objects. *arXiv preprint arXiv:1706.03799*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30. ACM.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. 2009. Zero-shot learning with semantic output codes. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Lenhart Schubert. 2002. Can we derive general world knowledge from texts? In *Proceedings of the second international conference on Human Language Technology Research*, pages 94–97. Morgan Kaufmann Publishers Inc.

B. Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Benjamin Van Durme and Lenhart Schubert. 2008. Open knowledge extraction through compositional language processing. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 239–254. Association for Computational Linguistics.

Lucy Vanderwende. 2005. Volunteers created the web. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 84–90.