How Pre-trained Word Representations Capture Commonsense Physical Comparisons

Pranav Goel
Computer Science
University of Maryland
pgoel1@cs.umd.edu

Shi Feng
Computer Science
University of Maryland
shifeng@umiacs.umd.edu

Jordan Boyd-Graber
Computer Science, iSchool,
UMIACS, and LSC
University of Maryland
jbg@umiacs.umd.edu

Abstract

Understanding common sense is important for effective natural language reasoning. One type of common sense is how two objects compare on physical properties such as size and weight: e.g., 'is a house bigger than a person?'. We probe whether pre-trained representations capture comparisons and find they, in fact, have higher accuracy than previous approaches. They also generalize to comparisons involving objects not seen during training. We investigate *how* such comparisons are made: models learn a consistent ordering over all the objects in the comparisons. Probing models have significantly higher accuracy than those baseline models which use dataset artifacts: e.g., memorizing some words are larger than any other word.

1 Introduction

Pre-trained word representations or embeddings (Mikolov et al., 2013) such as GloVe (Pennington et al., 2014) underpin modern NLP. To understand what information is encoded, supervised models *probe* (Adi et al., 2016; Linzen et al., 2016; Conneau et al., 2018) a particular property, for example, part-of-speech (Belinkov et al., 2017), morphology (Peters et al., 2018a), etc. in these representations. With the advent of contextualized word embeddings such as ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2018), efforts to understand the information encoded in representations learned by neural model have increased (Peters et al., 2018b; Tenney et al., 2019; Liu et al., 2019). Apart from linguistic properties, what do these representations learn about the world? Commonsense reasoning over language that incorporates world knowledge such as 'an elephant is heavier than a person' can help agents make better decisions and understand 'complex' phenomena like humor and irony. However, extracting common sense from text corpora is challenging since we rarely state obvious things directly (Van Durme, 2010; Gordon and Van Durme, 2013; Misra et al., 2016; Zhang et al., 2017).

This paper asks if pre-trained representations encode a specific type of common sense: physical comparisons between objects.[1] The supervised classification task takes a pair of words being compared on a physical attribute such as size or speed, with the system's objective to decide which is 'bigger' or 'faster' (§ 2.1). We use a linear or a one-layer fully-connected neural network probing model with only a combination (concatenation or subtraction) of the frozen pre-trained embeddings for the words to be compared as input (§ 2.2). This probing model achieves better accuracy than previous approaches (§ 2.3) which use extra information other than the words (such as the verbs connecting the words) on the Verb Physics dataset (Forbes and Choi, 2017) (§ 3): it encodes physical commonsense comparisons.[2] It generalizes to objects not present in the training set (§ 3.1) with higher accuracy than baselines exploiting dataset artifacts (§ 4). We use a 'simple' probing model since more complex models make it difficult to disentangle the major contributing factor to results - model or embeddings (as in other probing studies like Liu et al. (2019)). Our other major contribution is analyzing how models compare objects. The output logits for labels (indicating model confidence) order objects consistently across orderings or rankings built around different objects (§ 4.1.1). Models also learn an ordering over all the objects and use this learned ordering for comparisons (§ 4.1.2).

[1] Note: Concurrent work by Forbes et al. (2019) also finds neural representations are proficient at capturing physical properties of objects (focus of this work) but not at tackling the relationship with actions applicable to objects.

[2] This work aims to probe representations for physical commonsense comparisons; better accuracy is a byproduct.

2 Experimental Setup

2.1 Probing Task & Data

We use Verb Physics (Forbes and Choi, 2017) and follow their setup. Given a pair of words or objects, a system predicts if *word*$_1$ $</>/\approx$ *word*$_2$ when compared on an attribute, for example, *bed* $>^{weight}$ *hand* or *mouth* $\approx^{size}$ *fist*. Verb Physics consists of five different datasets comparing objects on *size*, *weight*, *strength*, *rigidness*, and *speed*.[3] The train:dev:test split is 5:45:50 resulting in about 100 and 1000 comparisons in the training and dev sets respectively, with similar statistics for all attributes. This is the split used in the previous works and hence we use the same split in order to benchmark results. To test generalization to words not seen during training, we also use a different evaluation set released by Bagherinezhad et al. (2016) with 486 size-based comparisons of objects (§ 3.1).[4]

2.2 Our Probing Model

The probing model is a simple setup to assess if pre-trained representations capture physical object comparisons. We concatenate or subtract the word embeddings for the two words and pass it to a fully-connected neural network with zero (in which case, linear) or one hidden layer. Our primary experiments use GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018a), and BERT (Devlin et al., 2018) embeddings. Training details (including the specific pre-trained models and training parameters) are presented in Appendix A.1. Following Yang et al. (2018), we pass the reversed combination of the two embeddings through the network, and align and combine the outputs for both input pairs (*word*$_1$ $-$ *word*$_2$ and *word*$_2$ $-$ *word*$_1$) for the final output. If *word*$_1$ $<$ *word*$_2$ then *word*$_1$ $>$ *word*$_2$ as well. Unlike Yang et al. (2018), we pass the reversed pair at training. This 'reversal' trick is visualized in Figure 2, and the empirical results showing its effect in increasing accuracy are discussed in Appendix A.2.

2.3 Baselines

Majority Class: This baseline predicts the label for a comparison on the dev set based on the highest-frequency label for both the words as per training set. If the two labels agree, e.g., *word*$_1$ is 'bigger' and *word*$_2$ is 'smaller' in most training comparisons, this baseline predicts *word*$_1$ $>$ *word*$_2$. If the two majority labels disagree (both words tend to be 'bigger' most of the times), this baseline uses the ratio of frequency of the majority label with the total number of comparisons involving the word to decide.

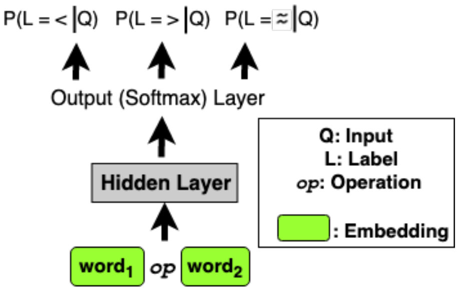

Figure 1: The Probing Model: We combine the pre-trained word embeddings of the two words being compared (via concatenation or subtraction) and pass it though zero (linear) or one hidden layer.

We also compare with the previous state-of-the-art approaches on the Verb Physics dataset:

Verb-centric Frame Semantics: (Forbes and Choi, 2017, **F&C**) use probabilistic graphical modeling for joint inference over objects as well as actions/verbs that can imply physical relationship their arguments (for example, 'x entered y' implies y is bigger than x).

Property Comparisons from Embeddings: (Yang et al., 2018, **PCE**) use a one-layer neural network over concatenated word embeddings and compare the projection with the embeddings of 'poles': words exemplifying a physical relation ('big', 'small' for size; 'fast', 'slow' for speed, etc.). Classification is the closest 'pole'. This use of poles is the main difference with our approach.

Apart from these baseline models, we devise additional baselines to test for possible artifacts in the dataset, such as using only one of the words as input to the model, in Section 4.

3 Results and Discussion

The probing model (Figure 1) with pre-trained representations has better accuracy than previous approaches which use extra information in addition to the words being compared (Table 1). This indicates that representations themselves capture physical commonsense comparisons.

GloVe is almost as accurate as ELMo and more accurate than BERT contrary to results seen on many NLP tasks (Peters et al., 2018a; Devlin et al.,

[3] https://github.com/uwnlp/verbphysics

[4] http://grail.cs.washington.edu/projects/size/

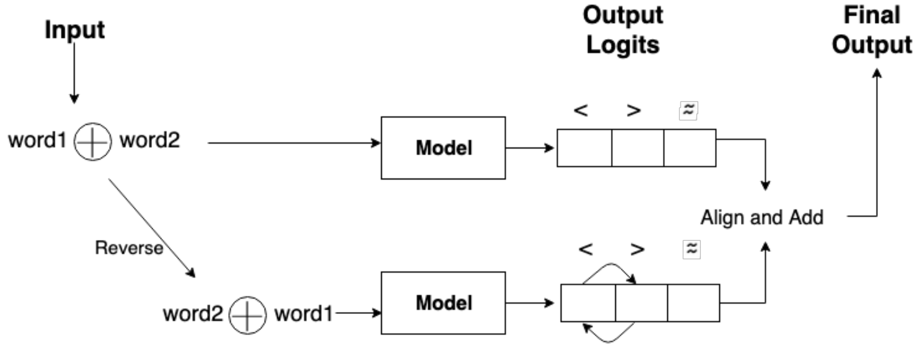

Figure 2: The Reversal Trick: As done by Yang et al. (2018) at test time, the reversed embedding is also passed through the network and the output logits for both pairs ($word_1$ concatenated with $word_2$ and $word_2$ concatenated with $word_1$) are aligned and combined for the final output. We try doing this at training time as well which leads to an improvement in accuracy.

2018). This task has no context to exploit and Tenney et al. (2019) also observe that contextualized embeddings win over non-contextual models on syntactic tasks but less for semantic tasks.

We also used **BERT-large** but saw similar accuracies as BERT-base. Concatenating word embeddings usually achieved slightly better accuracy (Appendix A.2) but subtracting gave more stable results across different random initializations. The reversed input pair embeddings (§ 2.2) at training and testing improves accuracy (Appendix A.2).

3.1 Generalization to New Objects

In Verb Physics, ∼99% of the words or objects involved in comparisons in the dev set are seen at training. If word embeddings capture common sense well, they should compare two words not seen during training. To test this, we use the Verb Physics training set for the 'size' attribute and evaluate on a different test set (Bagherinezhad et al., 2016): **EB evaluation set** (§ 2.1) where *only ∼33% of the words are seen during training.* Since this evaluation set contains only $<$ and $>$ comparisons, we use comparisons in Verb Physics training set with just these two labels. Unlike Bagherinezhad et al. (2016) who use visual and textual cues, our model use only pre-trained text representations. Yet the probing model achieves at least 4% higher accuracy (Table 2).

4 Analysis

Levy et al. (2015) find that in models for hypernymy detection: the accuracy gap between the full model using both the words as input and using just one of the words is less than 10%. Their training set contains prototypical hypernyms: single word in a pair that models can latch onto to detect hypernymy. The unsupervised method of using the cosine similarity of the two words is also a strong baseline in that work. We experiment with these same baselines for our task.

On the Verb Physics dataset: Only $word_2$ seems to be a strong baseline (much like the ma-

	Majority Class Baseline	F&C	PCE	Probing Model (GloVe)	Probing Model (ELMo)	Probing Model (BERT-base)
Size	0.66	0.75	0.80	**0.82**	**0.82**	0.80
Weight	0.67	0.74	0.81	**0.82**	**0.82**	0.80
Strength	0.66	0.71	0.77	0.78	**0.79**	0.75
Rigidness	0.60	0.68	0.71	0.71	**0.72**	0.71
Speed	0.59	0.66	0.72	0.72	**0.76**	0.71
Overall	0.64	0.71	0.76	0.77	**0.78**	0.75

Table 1: Accuracy of the probing model compared with the baselines including previous approaches on the attributes in the Verb Physics dataset. The simple probing model achieves better accuracy indicating that the frozen pre-trained representations capture commonsense physical comparisons.

Model	Accuracy
The Visual+Textual Model by Bagherinezhad et al. (2016)	0.835
Probing Model (GloVe)	0.879
Probing Model (ELMo)	**0.905**
Probing Model (BERT)	0.893

Table 2: The probing model trained on the Verb Physics size dataset and evaluated on (Bagherinezhad et al., 2016). Only $\sim$33% of the objects in this test set are present in training set: our model generalizes to new objects and gives better accuracy using the frozed pre-trained representations of the words alone.

jority class baseline for this dataset), but the drop in accuracy is higher than 10% for GloVe and ELMo (Table 3): Our model is *not* simply relying on lexical memorization. Randomly selecting a label gives $\sim$33% accuracy while using the majority label for all comparisons gives $\sim$50% accuracy. The unsupervised model gives low accuracy which suggests supervision is helpful.

On the EB Evaluation Set (Bagherinezhad et al., 2016): Using just one word when training and evaluating sees a drop of about 12 to 15% in accuracy (Table 4). This baseline is fairly strong in comparison to a random baseline (50% accuracy), but the difference in accuracy again indicates the model doing more than just lexical memorization.

4.1 Do Models Learn a Consistent Ordering?

Pre-trained representations encode commonsense physical comparisons, and do not rely on mere lexical memorization. One explanation is models could learn to rank or order the objects.

Using the given Verb Physics training set	***word***$_1$ - ***word***$_2$	**ONLY *word***$_2$ **Baseline**	**Unsupervised Baseline**
GloVe	0.78	0.66	0.49
ELMo	0.78	0.67	0.52
BERT	0.75	0.66	0.52

Table 3: Accuracy of probing models (averaged across the five attributes) on the Verb Physics dev sets. Unsupervised baseline takes cosine similarity of the embeddings and uses a threshold tuned on the dev set to classify. Using just one word when training and evaluating helps investigate possible lexical memorization.

On the Complete EB Evaluation Set; $\sim$33% 'overlap'	***word***$_1$ - ***word***$_2$	***word***$_1$	***word***$_2$
GloVe	0.88	0.74	0.73
ELMo	0.89	0.74	0.72
BERT	0.87	0.65	0.68

Table 4: Evaluation on Bagherinezhad et al. (2016). Accuracy drops by 15 to 20% when compared with the only one word baselines.

Examples of Orders Formed Around a Word
head < knee < meal < ***chair*** < back < place < street < world < *gate* < air < floor < *room*
eye < *chair* < child < king < daughter < wife < boy < messenger < father < coach < horse < door < house < ***gate*** < train < *room* < sun

Table 5: Two examples for orderings formed around the words ***chair*** and ***gate*** for the size attribute using GloVe. Comparisons between words occurring in both these orderings (italicized) are consistent.

4.1.1 Local Ordering formed via Logit Difference

A particular word gets compared with many other words in data. We can order those words to form a 'local' ordering, e.g., ordering around ***chair*** (Table 5). Orderings are *consistent* if the same pair of words in different local orderings hold the same relationship, e.g., chair < room in both orderings in Table 5. It is conceivable humans are more confident about a comparison when the difference in objects in terms of the property is large (a house is bigger than a chair). Larger difference in output logits (for label 0 (<) and 1 (>)) can indicate more model confidence and hence, objects being farther apart in an ordering. We form local orderings around a word using logit difference between the labels when compared with the other word.

All the local orderings formed around all words on Verb Physics are completely consistent for GloVe and BERT. For ELMo, more than 90% comparisons were usually consistent across any two orderings. Models seem to learn to arrange all the words in some sort of consistent ordering.

	Linear	Neural Net with 1 or 2 hidden layers
GloVe	0.76	0.77
ELMo	0.77	0.78
BERT	0.74	0.75

Table 6: The best accuracies obtained by a Linear Model compared with the best accuracies obtained by a shallow Neural Network. For all three representations, the linear model gives similar accuracy and hence we often use it for our analysis. Since good accuracy is achieved by a simple linear model from the frozen word representations alone, we can reasonably conclude that pre-trained embeddings encode information required to compare words for physical common sense.

4.1.2 Global Ordering over all Words Using Learned Weights

We use a *linear* model (0 hidden layers in Figure 1) to order all the objects in one of the Verb Physics dev sets. Per Table 6, linear modela are almost at par (accuracy within 1%) with shallow fully connected neural networks on the Verb Physics dev set. A score for a word is its embedding multiplied with the weight learned for mapping the input to the label 1 which would be higher if $word_1 > word_2$. We use this score to rank the objects. Appendix A.3 shows an example of a learned ordering over all the words in the dev set using GloVe. Using this ordering to classify the comparisons of pair of words achieves accuracy at par with the original models on a subset of the dev set containing only 0/1 labels. **This suggests the models assign an absolute value to every word to rank all the objects and then *use this global ranking* to compare any two objects.** Using the weight corresponding to the label 0 achieves similar results. An ordering can be used directly for $>$ or $<$ comparisons but is not that indicative for $\approx$ comparisons. This might explain the relative struggles GloVe, ELMo, and BERT face classifying comparisons labeled 2 (Table 7).

5 Conclusion

A linear or a small fully connected neural network probing model can compare two words on commonsense physical attributes using frozen pre-trained representations (GloVe, ELMo, and BERT) of the words alone with higher accuracy than previous approaches which use extra information in addition to the objects being compared. They also generalize to objects not seen during training and get significantly higher accuracy than using just one word: embeddings encode physical common sense. Models learn an ordering over of all the words involved in the comparisons and embeddings could be using this ordering to compare any two objects. The difference in the output logit values corresponding to the labels serves as a surprisingly good proxy to form completely consistent orderings around different words. One direction of future work would be to move beyond comparisons or relative information towards directly probing for size estimates for various physical properties for objects (without the setting being relative), using the recently released large-scale resource containing 'distributions over physical quantities associated with objects, adjectives, and verbs' (Elazar et al., 2019).

	0 ($<$)	**1** ($>$)	**2** ($\approx$)
GloVe	0.79	0.77	0.33
ELMo	0.81	0.80	0.18
BERT	0.77	0.78	0.12

Table 7: Label-Wise Accuracy: The GloVe, ELMo, and BERT representations (fed to a linear model) struggle to capture the relationship $word_1 \approx word_2$ (label 2). This is likely due to the class imbalance in the dataset, with the rough distribution of the labels across all attributes in the Verb Physics training set being 41% for the label 0, 49% for the label 1, and just 10% for the label 2. The representations seem to learn an ordering over all the words and use it to compare objects (§4.1.2). This is also one possible explanation for comparatively poor accuracy on the label 2 since judging $\approx$ relationship between words is hard while the $<$ or $>$ relation can be inferred directly from an ordering. Accuracies here are averaged across the results for all the five attributes.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. 2016. Are elephants bigger than butterflies? reasoning about sizes of objects. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology?

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. *arXiv preprint arXiv:1906.01327*.

Maxwell Forbes and Yejin Choi. 2017. Verb physics: Relative physical knowledge of actions and objects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 266–276.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *arXiv preprint arXiv:1908.02899*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

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.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2939.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Benjamin Van Durme. 2010. Extracting implicit knowledge from text.

Yiben Yang, Larry Birnbaum, Ji-Ping Wang, and Doug Downey. 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.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395.

A Supplementary Material

A.1 Training Detail

We specify the variation of each pre-trained representation we used:

	Concatenation			*Subtraction*		
	No Reversal	**Reversal at Test Only**	**Reversal at Train & Test**	**No Reversal**	**Reversal at Test Only**	**Reversal at Train & Test**
GloVe	0.72	0.74	0.77	0.75	0.74	0.76
ELMo	0.73	0.76	0.78	0.77	0.77	0.77
BERT	0.7	0.74	0.75	0.73	0.74	0.75

Table 8: Effect of using the reversal trick (Figure 2): Accuracy increases for all the representations when using reversal at the test time (as noted by (Yang et al., 2018), and a further increase when used at both training and testing. The effect is more pronounced when the embeddings are combined via concatenation.

1. GloVe: 300 dimensional vectors trained on the Common Crawl dataset with representations for 840B tokens.[5]

2. ELMo: The 'original' pre-trained ELMo model using a 2-layer LSTM.[6] The word embedding dimension is 1024.

3. BERT: The fully pre-trained BERT-base model (uncased) using a 12-layer bidirectional transformer.[7]

We take the mean of the multi-layer BERT/ELMo representations. The probing model (Figure 1) used 500 dimensional hidden layer(s) with ReLU non-linearities. We train with SGD (learning rate = 0.005) with momentum = 0.7 for 25 epochs with a patience of 20. Our models are implemented using the AllenNLP framework (Gardner et al., 2018).

A.2 The Reversal Trick

Figure 2 visualizes the reversal trick. Passing the reversed embedding ($word_1 - word_2$ as well as $word_2 - word_1$) at test time helped Yang et al. (2018) achieve better accuracy on the Verb Physics dev set. With our setup being basically the same as their approach, except for their use of poles, our results that the reversal trick was the major reason for their accuracy and not the use of poles (Table 8). We tried using the reversal trick at training time as well which further increased the scores, making the accuracy of the model better than the 'PCE' model (§2.3) without the use of poles (Table 1)!

A.3 Global Ordering over all Words: Example

The process is described in §4.1.2: the trained weight of a linear model corresponding to label 1 or 0 is multiplied with the pre-trained representation of a word to get a 'score' corresponding to every word. This score can then be used to form an ordering over all the words in a particular set. Table 9 contains an example of such an ordering obtained using GloVe for the size attribute over all the words in the corresponding Verb Physics dev set using the trained weight corresponding to the label 1.

[5] https://nlp.stanford.edu/projects/glove/

[6] https://allennlp.org/elmo

[7] https://github.com/huggingface/pytorch-pretrained-BERT

scissors < beard < spoon < hair < knife < finger < lip < purse < chin < goose < vial < eye < nose < bow < fist < piece < ash < glass < chair < skirt < grass < picture < head < face < hat < gulp < bag < ear < hand < strap < dress < bottle < torso < elbow < edition < mouth < pocket < arm < shoulder < rope < magazine < tear < seal < hedge < effect < violin < tree < knee < lamp < cup < pedestal < throat < book < coal < object < suit < button < ball < chest < magistrate < newspaper < fox < ice < candidate < harlot < basin < mosquito < meal < bower < foot < shirt < step < child < stone < body < anchor < clothes < seed < exile < shippe < dinner < trench < element < lung < light < block < poet < sink < king < stair < breath < fool < phone < coward < banker < result < base < response < sip < bench < end < lock < victim < source < torrent < brick < sail < daughter < master < watch < gully < cross < scene < disciple < lady < food < direction < teacher < boy < middle < boat < messenger < parent < precipice < person < call < window < shore < wife < vessel < horse < temple < servant < piano < bed < patient < side < something < parcel < back < way < position < wall < place < lover < wind < state < corner < office < father < prison < worker < volunteer < street < abode < coach < flood < doorway < anything < someone < ground < front < brother < world < horseback < shop < current < city < energy < reservation < friend < camp < store < bank < factory < gentleman < rain < lad < deck < soul < home < beach < everything < floor < clock < car < house < door < ship < heaven < truck < air < system < barn < stream < mountain < restaurant < road < river < sea < bay < gate < hill < coast < farm < town < train < sun < room

Table 9: **Example of an Ordering Over all Words in a Set:** The trained weights of the linear probing model multiplied with the embedding of a word can help form a 'global' ordering over all the words (§4.1.2). This particular example is when the weight corresponding to the label 1 is used for the words in the Verb Physics size dev set with GloVe embeddings.