NTU NLP Lab System at SemEval-2018 Task 10: Verifying Semantic Differences by Integrating Distributional Information and Expert Knowledge

Yow-Ting Shiue[1], Hen-Hsen Huang[1], and Hsin-Hsi Chen[1,2]

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[2]MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
orina1123@gmail.com,hhhuang@nlg.csie.ntu.edu.tw,hhchen@ntu.edu.tw

Abstract

This paper presents the NTU NLP Lab system for the SemEval-2018 Capturing Discriminative Attributes task. Word embeddings, pointwise mutual information (PMI), ConceptNet edges and shortest path lengths are utilized as input features to build binary classifiers to tell whether an attribute is discriminative for a pair of concepts. Our neural network model reaches about 73% F1 score on the test set and ranks the 3rd in the task. Though the attributes to deal with in this task are all visual, our models are not provided with any image data. The results indicate that visual information can be derived from textual data.

1 Introduction

Modern semantic models are good at capturing semantic similarity and relatedness. The widely-used distributional word representations, or word embeddings, have achieved promising performance on various semantic tasks. The word pair similarities calculated with these models are to some extent consistent with human judgments, and many downstream applications such as sentiment analysis and machine translation have benefited from word embeddings' ability to aggregate the information of lexical items with similar meaning but different surface forms.

However, the ability to *distinguish* one concept from another similar concept is also core to linguistic competence. Our knowledge about what is a "subway", for example, may contain "it is a kind of train that runs underground". Also, discriminating things is an important mechanism for teaching and learning. For example, if we would like to explain how a "plate" is different from a "bowl", we may use expressions like "a plate is flatter" or "a bowl is deeper". All these examples show that one form of semantic difference is a *discriminative attribute* which applies to one of the two concepts being compared but does not apply to the other.

In the SemEval-2018 Capturing Discriminative Attributes task (Krebs et al., 2018), participants need to put forward semantic models that are aware of semantic differences. A data instance consists of a triple and a label. In this paper, we denote a triple with $< w_1, w_2, a >$, in which w_1 and w_2 are the two words (concepts) to be compared, and a is an attribute. The label is either positive (1) or negative (0). In a positive example, a is an attribute of w_1 but not an attribute of w_2. For negative examples, there are two cases: 1) both w_1 and w_2 have attribute a ; 2) neither w_1 nor w_2 has attribute a. In this task, a is limited to visual ones such as color and shape. The evaluation metric is the macro-averaged F1 score of the positive and the negative classes.

Visual attribute learning has been investigated by past researchers. Silberer et al. (2013) build a dataset of concept-level attribute annotations based on images in ImageNet (Deng et al., 2009). For each attribute, they train a classifier to predict its presence or absence in the input image. Lazaridou et al. (2016) propose a model that does not learn visual attributes explicitly, but learns discriminativeness. Their model predicts whether an attribute can be used to discriminate a referent from a context. Both the referent and the context are represented by visual instances sampled from ImageNet. This setting is similar to that of this SemEval task. However, one critical difference is that in this task, the set of attributes is open. The dataset is partitioned so that all the attributes in the test set are unseen in the training set, which makes this task more challenging.

The use of word embeddings for detecting semantic properties is studied by Rubinstein et al. (2015). They focus on a fixed set of properties and train a binary classifier for each property. Their

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1027–1033
New Orleans, Louisiana, June 5–6, 2018.

results indicate that word embeddings capture taxonomic properties (e.g. "an animal") better than attributive properties (e.g. "is fast"), possibly because attributive signal is weak in text.

In this task, most visual attributes are attributive properties. The signal of "visual" attributes can be even weaker in text since they are not mainly communicated through language in human cognition. The word "red" in "I bought a red apple" sounds more like a linguistic redundancy than that in "I bought a red jacket" does, since "red" is a typical attribute of apples. However, these visual attributes may impose constraints on valid expressions. For instance, we can say "the bananas turned yellow", but it would be extremely difficult to find some context where "the bananas turned red" makes sense. Therefore, visual attributes can be signaled in some implicit and indirect ways. By utilizing several computational approaches, we reveal to what extent visual attributes can be acquired from text.

This paper aims at capturing semantic difference by incorporating information from both corpus statistics and expert-constructed knowledge bases. We build a rule-based system and a learning-based system for the binary classification problem, i.e., to tell whether an attribute is discriminative for two concepts. The learning-based system achieved F1 score of 0.7294, which is the third best in the official evaluation period of SemEval-2018 Task 10. Our approach is purely based on textual data, without access to image instances, which indicates that it is possible to figure out substantial visual information from text.

2 Distributional Information

We utilize two kinds of computational approaches to derive information from co-occurrence statistics in large copora. The first one is word embedding, which has been shown to encode semantic information in low-dimensional vectors. The second one is pointwise mutual information (PMI), which is a commonly-used measurement of the strength of association between two words. We analyze the performance of rule-based or learning-based models with different sets of features to reflect their effectiveness.

2.1 Concatenation of Word Embeddings

A very straight-forward approach is concatenating the embedding of w_1, w_2 and a into a feature vector to train a binary classifier. We use the pre-trained 300-dimensional Word2vec embeddings (Mikolov et al., 2013) trained on Google News[1] as input features. We construct a multilayer perceptron (MLP) model with two hidden layers of size 1,024 to conduct preliminary experiments. The activation function is ReLU and the dropout rate is 0.5. The model is implemented with Keras (Chollet, 2015). We train for 20 epochs and report the best validation scores.

Embeddings			**Train**		**Validation**	
w_1	w_2	a	Acc.	Macro F1	Acc.	Macro F1
V	V	V	**0.7468**	**0.6484**	**0.5184**	0.3409
		V	0.6379	0.5216	0.5180	0.2908
V	V		0.7017	0.5040	0.4996	0.2748
V		V	0.6790	0.5938	0.4945	**0.3558**
	V	V	0.6733	0.5421	0.5029	0.3170

Table 1: Training and validation scores of MLP model with embeddings of different subsets of the triple.

However, we find out that there is a serious issue of overfitting. As shown in Table 1, the gap between training and validation scores is large. We also experimented simpler models such as Logistic Regression and Random Forest, and got similar results. A possible cause of overffiting is that the model does not learn to extract and compare attributes, but learns the "pattern" of some combination of words in the triples.

To verify the above speculation, we train similar MLP models which only take "partial" triples as input. Theoretically, the label cannot be determined correctly with an incomplete triple. However, according to the results shown in Table 1, the models considering solely a part of every triple can still "learn" some information from the training set (majority-class baseline accuracy on the training set: 0.6383). Some models with partial information even achieve better validation scores than that with complete information. This indicates that the models overfit to the vocabulary of the training set. At the test time, all the attributes are unknown, so the model cannot make effective predictions. In fact, these results are similar to the lexical memorization phenomenon reported by Levy et al. (2015) on the hypernym detection task.

2.2 Embeddings Similarity Difference

Because "raw" word embedding features do not work, we turn to more abstract features. Let sim_1 and sim_2 be the cosine similarity of the vector of a

[1] https://code.google.com/archive/p/word2vec/

to the vector of w_1 and w_2 respectively. We compare the values sim_1 and sim_2. The rationale is that if a word w has an attribute a, then it tends to, though not necessarily, be more similar to a than other words without a.

The following six embedding models are experimented with. The embedding size is fixed to 300.

1. **W2V(GNews):** The standard Word2vec model as described in Section 2.1.
2. **fastText:** fastText (Bojanowski et al., 2017) is a modification of Word2vec that takes subword information into account. We adopt the pre-trained vectors trained on 6B tokens[2].
3. **Numberbatch:** Numberbatch embeddings are built upon several corpus-based word embeddings and improved by retrofitting on ConceptNet, a large semantic network containing an abundance of general knowledge (Speer et al., 2017). We use the pre-trained embeddings of English concepts[3].
4. **GloVe(Common Crawl)**: The GloVe model (Pennington et al., 2014) obtains word representation according to global co-occurrence statistics. We use the pre-trained vectors trained on 840B tokens of Common Crawl[4].
5. **Sense(enwiki)-*c*:** Sense vectors may encode more fine-grained semantic information than word vectors do, so we also experimented with sense vectors. We perform word sense disambiguation (WSD) on the English Wikipedia corpus to get a sense-annotated corpus, using the Adapted Lesk algorithm implemented in pywsd[5]. The sense inventory is based on synsets in WordNet. We train a Word2vec Skip-gram (SG) model with this corpus to obtain sense vectors. To apply sense vectors to words and attributes in this SemEval task, we propose the following *closest* sense-selection method (denoted by -*c*) to choose a sense for each of w_1, w_2 and a. $S(w)$ denotes the set of synsets that a word w belongs to and emb(s) denotes the vector of synset (sense) s.

$$s_1*, s_a* = \underset{\substack{s_1 \in S(w_1) \\ s_a \in S(a)}}{\operatorname{argmax}} \cos(\text{emb}(s_1), \text{emb}(s_a))$$

$$s_2* = \underset{s_2 \in S(w_2)}{\operatorname{argmax}} \cos(\text{emb}(s_1*), \text{emb}(s_2))$$

Since a might be an attribute of w_1, we choose the closest pair of senses for them. Then, we choose the sense of w_2 that is closest to s_1*, the selected sense for w_1. The reason is that a semantic difference is more likely to be meaningful for two similar concepts. Finally, we use the vector of the selected senses to compute similarities.

6. **Sense(enwiki)-*f*:** We use the same sense embeddings as described previously but directly select the *first* sense (predominant sense) in WordNet for w_1, w_2 and a respectively, without performing WSD. This method is denoted by -*f*.

We first use these similarities in a simple rule-based model: if $sim_1 > sim_2$ then output 1; otherwise output 0. The results are summarized in Table 2. In general, this similarity comparison rule performs better on the positive class than on the negative class. GloVe results in the highest negative F1, while Numberbatch results in the best macro-averaged F1. We show the confusion matrix for this rule with Numberbatch in Table 3. As can be seen, similarity differences are helpful for discriminating the positive examples, but they are not good indicators of negative examples.

We use $sim_1 - sim_2$ of different kinds of embeddings as features and train MLP models as described in the previous section. The results of different combinations of embeddings are shown in Table 4. However, there is only slight macro-

Embeddings	Acc.	Pos. F1	Neg. F1	Macro F1
1. W2V	0.6128	0.6512	0.5648	0.6080
2. fastText	0.6047	0.6435	0.5565	0.6000
3. Numberbatch	**0.6653**	**0.7142**	0.5964	**0.6553**
4. GloVe	0.6330	0.6594	**0.6022**	0.6308
5. Sense-*c*	0.5981	0.6609	0.5068	0.5838
6. Sense-*f*	0.5816	0.5597	0.6013	0.5805

Table 2: Performance of the $sim_1 > sim_2$ rule with different embeddings on the validation set.

True label	$sim_1 > sim_2$	otherwise
1	1138	226
0	685	673

Table 3: Confusion matrix of the $sim_1 > sim_2$ rule with Numberbatch embeddings on the validation set.

Embeddings	Acc.	Pos. F1	Neg. F1	Macro F1
[**sim x3**] 1. – 3.	**0.6598**	**0.6640**	**0.6555**	**0.6598**
[**sim x4**] 1. – 4.	0.6547	0.6572	0.6521	0.6546
[**sim x6**] 1. – 6.	0.6565	0.6609	0.6520	0.6564

Table 4: Performance of MLP models with different combinations of word vector similarity differences.

[2]https://fasttext.cc/docs/en/english-vectors.html
[3]https://github.com/commonsense/conceptnet-numberbatch
[4]https://nlp.stanford.edu/projects/glove/
[5]https://github.com/alvations/pywsd

F1 improvement over the rule-based models. On the other hand, though including the last three embedding models does not yield better result in this setting, we find them useful when combined with other kinds of features. Therefore, they are included in one of our submitted systems.

2.3 PMI Difference

Similar to word embedding, PMI reflects the co-occurrence tendencies of words. It has been shown that the Skip-gram with Negative Sampling (SGNS) algorithm in Word2vec corresponds to implicit factorization of the PMI matrix (Levy and Goldberg, 2014). Nevertheless, PMI should be interpreted differently from word vector similarity. Since PMI is calculated in an exact matching manner, there is no propagation of similarity as in the case of word vectors. That is, suppose that both PMI("red", "yellow") and PMI("apple", "banana") are high, this does not imply that PMI("red", "banana") will be high. Thus, PMI might be less prone to confusion of similar concepts.

We calculate PMI on the English Wikipedia corpus. We first experimented with a $\underline{PMI_1 > PMI_2}$ rule that is similar to the one for vector similarities. In Table 5, we report the results of PMI calculated with different sizes of context window within which a pair of words is considered to be a co-occurrence. 20-word context window yields the best performance so we show its corresponding confusion matrix in Table 6. As can be seen, PMI performs slightly better in discriminating the negative class, compared to word similarities (Table 3).

Context window	Acc.	Pos. F1	Neg. F1	Macro F1
10 words	0.6550	0.6986	**0.5968**	0.6477
20 words	**0.6561**	**0.7013**	0.5948	**0.6481**
30 words	0.6506	0.6959	0.5896	0.6427
whole sentence	0.6447	0.6906	0.5830	0.6368

Table 5: Performance of the $PMI_1 > PMI_2$ rule with different context windows on the validation set.

True label	$PMI_1 > PMI_2$	otherwise
1	1099	265
0	671	687

Table 6: Confusion matrix of the $PMI_1 > PMI_2$ rule with context window size 20 on the validation set.

True label	$sim_1 > sim_2$ & $PMI_1 > PMI_2$	otherwise
1	964	400
0	429	929

Table 7: Confusion matrix of the $sim_1 > sim_2$ & $PMI_1 > PMI_2$ rule on the validation set.

Features	Acc.	Pos. F1	Neg. F1	Macro F1
PMI(10+20+30)	0.6492	0.7026	0.5723	0.6375
sim x6 + PMI x3	**0.6763**	**0.7039**	**0.6432**	**0.6735**

Table 8: Performance of MLP models with combinations of word vector similarity differences and sign of PMI differences.

Based on the above observation, we propose a heuristic rule of combining vector similarity and PMI: if $\underline{sim_1 > sim_2}$ **and** $\underline{PMI_1 > PMI_2}$ then output 1. We use the Numberbatch embeddings and PMI of 20-word context. This majority-voting model is more reliable and achieves macro-F1 above 0.69. It is one of our submitted systems so the result is shown in Table 14. According to the confusion matrix in Table 7, both the positive and the negative classes can be discriminated well with the combination of distributional vectors and PMI.

We also build learning-based models with combinations of PMI of different context window sizes. Since the range of PMI can be large, we only consider the sign of the difference. The sign of zero is defined to be negative. In addition, we also combine vector similarities to train the MLP model. The results are all shown in Table 8. However, none of the results show improvement over the corresponding rule-based models.

3 Expert Knowledge from ConceptNet

3.1 Edge Connection

ConceptNet can be regarded as a directed graph of concepts (vertices) connected by different relations (edges). There are 47 relation types in ConceptNet. Some of them, such as `HasProperty` and `CapableOf`, are directly related to attributes. Other relations such as `RelatedTo` can also reflect some kinds of attributes.

We experiment with a simple rule-based model that outputs 1 if there exists a relation from w_1 to a and there is no relation from w_2 to a. Additionally, we augment the ConceptNet graph with reverse edges and apply the rule again. The results of both versions are shown in Table 9. The

Graph	Acc.	Pos. F1	Neg. F1	Macro F1
ConceptNet edges	0.5996	0.4593	0.6820	0.5707
+ reverse edges	**0.6297**	**0.5140**	**0.7009**	**0.6074**

Table 9: Performance of the $w_1 \rightarrow a$ & $w_2 \nrightarrow a$ rule with the ConceptNet graph and its extension on the validation set.

Features	Acc.	Pos. F1	Neg. F1	Macro F1
$w_1/w_2 \overset{r}{\leftrightarrow} a$ for **each** r	0.5724	0.4785	0.6376	0.5581
$w_1/w_2 \overset{r}{\leftrightarrow} a$ for **any** r	**0.5974**	**0.4931**	**0.6661**	**0.5796**

Table 10: Performance of MLP models with ConceptNet edge features on the validation set.

version with reverse edges performs competitively with the vector similarity rule (macro F1 about 0.6), but the behavior is quite different. As can be seen, the ConceptNet features help achieve better negative F1. The relatively low performance on the positive class might be due to the sparseness of the knowledge graph. Some w_1 might have attribute a but it is not directly connected to a on the graph.

To encode edge connection information for training learning-based models, we compute the following four binary features:

- Is there an edge from w_1 to a?
- Is there an edge from a to w_1?
- Is there an edge from w_2 to a?
- Is there an edge from a to w_2?

We also experimented with two versions. In the first version, each type of relations are considered separately, so the total dimensionality is 4 * 47 = 188. In the second version, we set a binary feature to 1 if there is at least one edge that satisfies its condition, so the feature dimensionality is only 4. The results are shown in Table 10. Although different types of relations have different semantics and should be treated differently, the version considering relation type does not perform better. A possible reason is that it can suffer from the data sparseness problem, since some dimensions are zero for almost all the instances.

3.2 Shortest Path Length

To include connections between words and attributes that take more than one step, we calculate the shortest path lengths. Let $dis(w_i, a)$ be the shortest path length between w_i and a on the ConceptNet graph. We first experiment with a simple rule-based model that outputs 1 when $dis(w_1, a) < dis(w_2, a)$, that is, when w_1 is closer to a. The results are reported in Table 11. Including reverse edges slightly improves the accuracy but does not improve the macro F1 score. A confusion matrix is presented in Table 12, showing that this rule is a strong indicator for the negative class. Compared to the ones with edge connection features, however, these rule-based classifiers

Graph	Acc.	Pos. F1	Neg. F1	Macro F1
ConceptNet edges	0.6308	**0.5740**	0.6742	**0.6241**
+ reverse edges	**0.6315**	0.5622	**0.6819**	0.6220

Table 11: Performance of the $dis(w_1, a) < dis(w_2, a)$ rule with the ConceptNet graph and its extension on the validation set.

True label	$dis(w_1, a) < dis(w_2, a)$	otherwise
1	644	720
0	283	1075

Table 12: Confusion matrix of the $dis(w_1, a) < dis(w_2, a)$ rule (reverse edges considered) on the validation set.

Graph	Acc.	Pos. F1	Neg. F1	Macro F1
ConceptNet edges	0.6532	0.6629	**0.6430**	0.6529
+ reverse edges	**0.6646**	**0.6984**	0.6223	**0.6603**

Table 13: Performance of MLP models with one-hot representation of ConceptNet shortest path lengths on the validation set.

achieve slightly lower negative F1 but higher positive F1.

Since the maximum shortest path distance between a word and an attribute in the training set is 5 (when reverse edges are included), we encode $dis(w_i, a)$ into 6-dimensional discrete binary features as follows.

- No path from w_i to a
- $dis(w_i, a) = 1$
- $dis(w_i, a) = 2$
- $dis(w_i, a) = 3$
- $dis(w_i, a) = 4$
- $dis(w_i, a) \geq 5$

We build similar MLP models that take these features as input. The features for w_1 and w_2 are computed separately and then concatenated., There are clear improvements of learning-based models (Table 13) over rule-based ones (Table 11). The improvements are mostly contributed by the higher positive F1 scores. On the other hand, in general it is helpful to include a separate set of features calculated on the graph with reverse edges.

4 Submitted Systems

We submitted the predictions of a rule-based system and a learning-based system. The evaluation results are summarized in Table 14. Run 1 system is a rule-based combination of similarity differences of the Numberbatch embedding and the sign of PMI differences (window size 20). Run 2 is an MLP model with three size-2048 hidden layers that takes input features of the similarity dif-

Model	Validation				Test			
	Acc.	Pos. F1	Neg. F1	Macro F1	Acc.	Pos. F1	Neg. F1	Macro F1
[1] Rule: $sim_1 > sim_2$ & $PMI_1 > PMI_2$	0.6954	0.6993	0.6915	0.6954	0.7047	0.6944	0.7143	0.7044
[2] MLP: sim x6 + PMI(10,20,30) + ConceptNet	**0.7175**	**0.7213**	**0.7136**	**0.7174**	**0.7303**	**0.7138**	**0.7451**	**0.7294**

Table 14: Evaluation results of the two submitted systems.

ference of the six kinds of embeddings, the sign of PMI differences of three different context window sizes and the ConceptNet edge and shortest path length features.

Our run 2 system performed the third best among all 26 participants with macro-F1 0.7294, showing that the features we proposed are highly effective. On the other hand, our run 1 system got an only slightly lower macro-F1 of 0.7044 and would get a rank between 5 (0.69) and 4 (0.72) if it was considered. This again proves the complementary effect of word vector similarity and PMI.

5 Error Analysis

Since even the top system in this task did not achieve macro-F1 above 75%, we think that there might be some cases that are very difficult to handle. Based on the test ground-truth released officially, we analyze the errors of our best system. We find out that the difficulties mainly arise from the following cases.

Ambiguous concept: Word ambiguity is not considered in this task. However, this may be problematic in some cases such as the positive example <mouse, squirrel, plastic>. According to the answer, we know that the word "mouse" is interpreted as a "computer device" instead of an "animal". Therefore, sometimes the answer is dependent on which sense is selected.

Vague or ambiguous attribute: Since the attribute is expressed only with a single word in this task, sometimes it is hard to tell what the attribute means, even from a human's perspective. For example, the triple <philanthropist, lawyer, active> is labeled 0 in the gold answer. Nevertheless, a positive interpretation also makes sense: philanthropists usually engage in philanthropy actively, while lawyers usually handle matters under the authorization of someone.

Relative attribute: In some positive examples, w_1 does not necessarily have a, but only more likely to have it. In the positive example <father, brother, old>, "father" might be "old" when being compared to "brother", but not necessarily so when considered isolatedly. It is even more difficult to determine when to evaluate the absence of an attribute relatively, given that we also encounter cases such as <banker, lawyer, rich>, whose gold label is 0.

6 Conclusions

We propose several approaches to tackle the SemEval 2018 Capturing Discriminative Attributes task in this paper. We utilize information derived from both corpus distribution statistics and expert knowledge in ConceptNet to build our systems. According to the experimental results, word embedding and PMI, though both based on co-occurrence, can complement each other in a simple heuristic rule-based system. Moreover, the ConceptNet features with high sensitivity to the negative class can complement the corpus-based features, which are more sensitive to the positive class. Our best learning-based system achieved F1 score of 0.7294 and got the 3rd place in the official run. We did not adopt image features, which suggests that it is possible to learn substantially about visual attributes solely from text.

Given the limited advancement of the learning-based model over the rule-based one, it is worth studying how to design some mechanism in machine learning models that can guide them to "compare" the features of the two concepts and determine the discriminativeness.

Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-107-2634-F-002-011-, MOST-106-2923-E-002-012-MY3, and MOST-105-2221-E-002-154-MY3.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 248–255.

Alicia Krebs, Alessandro Lenci, and Denis Paperno. 2018. Semeval-2018 task 10: Capturing discriminative attributes. In *Proceedings of the 12th international workshop on semantic evaluation (SemEval 2018)*.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2016. The red one!: On learning to refer to things based on discriminative properties. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 213–218. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

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. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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. Association for Computational Linguistics.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 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. Association for Computational Linguistics.

Carina Silberer, Vittorio Ferrari, and Mirella Lapata. 2013. Models of Semantic Representation with Visual Attributes. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 572–582. Association for Computational Linguistics.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 4444–4451.