Discriminator at SemEval-2018 Task 10: Minimally Supervised Discrimination

Artur Kulmizev
CLCG
University of Groningen
a.kulmizev@student.rug.nl

Mostafa Abdou
CLCG
University of Groningen
m.abdou@student.rug.nl

Vinit Ravishankar
Institute of Formal and Applied Linguistics
Charles University in Prague
vinit.ravishankar@gmail.com

Malvina Nissim
CLCG
University of Groningen
m.nissim@rug.nl

Abstract

We participated to the SemEval-2018 shared task on capturing discriminative attributes (Task 10) with a simple system that ranked 8th amongst the 26 teams that took part in the evaluation. Our final score was 0.67, which is competitive with the winning score of 0.75, particularly given that our system is a minimally supervised system that requires no training and minimal parameter optimisation. In addition to describing the submitted system, and discussing the implications of the relative success of such a system on this task, we also report on other, more complex models we experimented with.

1 Introduction and Background

Traditional evaluation tasks for semantic models have aimed to evaluate semantic relatedness (Bruni et al., 2014; Agirre et al., 2009) or similarity (Hill et al., 2014). More recently, analogy tasks (Mikolov et al., 2013; Gladkova et al., 2016; Abdou et al., 2018) have emerged in order to assess a model's ability to correctly answer questions in the form of "a is to b as c is to ?" using vector arithmetic. However, these approaches are not sufficient in evaluating the semantic competence of any given model: there are numerous flaws with similarity and analogy-based evaluation, the most pressing being the lack of correlation with downstream performance in real-world tasks (Schnabel et al., 2015). Furthermore, though analogy questions can assess how well certain semantic relations are modeled (*country : capital*, *country: language*), this is arguably more of a measure of context co-occurrence than it is a testament to any semantic understanding.

Capturing Discriminative Attributes is a novel semantic evaluation task which aims to assess the extent to which semantic models can capture semantic *differences* between words. Particularly, the task is concerned with identifying how well a given model represents attributes that discriminate between two related semantic concepts. For instance, given the terms *steak* and *salad*, **meat** would serve as a discriminative attribute, drawing a distinction between the former and latter as a quality that the two do not have in common.

2 Data

The data provided for this task was split between training and validation sections, with 17,500 and 2,721 samples comprising the former and latter, respectively. Every sample was composed of three terms (*pivot*, *comparison*, and *feature*) and their corresponding label $\in \{0, 1\}$. A sample was deemed as discriminative (label 1) if the *feature* served as an attribute that distinguished the *pivot* from the *comparison*. Otherwise, the sample was labeled as non-discriminative (0). Examples from the data are shown in Table 2. Please note that directionality is meaningful, i.e., swapping the *pivot* and *comparison* columns for the first two examples can change the label. For instance, *pink* is considered as discriminative for *pig* with regard to *sheep* (label is 1), but not the other way round (label would be 0). For a detailed description of the dataset, see (Krebs et al., 2018).

pivot	comparison	feature	label
sandwiches	breakfast	lunch	1
pig	sheep	pink	1
banana	raisin	round	0
uncle	father	male	0

Table 1: Samples of provided data. In the first two samples, the feature is discriminative (sandwiches are eaten at lunch; pigs are pink). In the last two, the feature is not discriminative (neither bananas nor raisins are round; an uncle and a father are both male).

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1008–1012
New Orleans, Louisiana, June 5–6, 2018.

It is important to note that only 5,000 instances were manually verified for consistency out of the combined training and validation datasets. Thus, whilst the entire validation set comprised of a subset of these verified samples (2,721), only 2,279 were represented in the training set (13%). Furthermore, the training set was heavily imbalanced towards negative ("non-discriminative") samples, accounting for a total 11,171 out of 17,500 (63.8%). As such, we largely focused our experiments on the validation set, since it was comprised entirely of manually curated data.

3 Models

In all models we describe, words not present in the vocabulary of a vector-space model were assigned the same 'unknown' vector drawn randomly from a normal distribution.

3.1 Baseline

The baseline we constructed was a simple support-vector machine classifier. We converted each word into a vector from a vector-space model. For every 4-tuple example, we passed the concatenation of vectors of the three words as a feature with the fourth as a label. This model failed to learn sufficiently, performing at near-chance levels on the validation set for all vector-space models.

3.2 Neural Models

Another model (**NN**) that we investigated was a feed-forward network with the concatenation of the word vector representations as an input layer, and a single binary output neuron as the output. Ultimately, we employed three hidden layers of sizes 450, 200 and 100, with ReLU as the activation function for each, and 20% dropout between each layer. Our output function was a sigmoid function. We used binary cross-entropy loss with a learning rate of 10^{-5} and Adam (Kingma and Ba, 2014) as our stochastic optimization function. This model outperformed the baseline for all vector-space models on the validation set.

Our next model (**NN-WN**) was built on the intuition that representations of more descriptive elements than just words could prove to be more helpful. We therefore converted each word into its first matching definition in WordNet (Miller, 1995), and condensed this definition into a dimension 4096 representation using Conneau et al. (2017)'s BiLSTM-max pooling encoder which is pre-trained on the *Stanford Natural Language Inference* dataset (Bowman et al., 2015). Similar to the previous model, this representation was passed to a feed-forward network, albeit with two hidden layers of sizes 1024 and 128 and sigmoid non-linearities. We did not evaluate this model with every set of embeddings due to time constraints and disappointing initial results; evaluated on standard GloVe (840B) embeddings, the performance of this model on the validation set was slightly lower than the feed-forward network (**NN**) (when utilizing the same vector-space model).

3.3 Discriminator

Our final submitted system (**Discriminator**), unlike any of our other models, consisted of a surprisingly simple set of rules which were designed to leverage the information encoded in distributional semantic vector-space models (VSM) for the purpose of classifying an attribute as discriminative or non-discriminative. We relied on the widely-used metric of cosine similarity; we measured the cosine similarities between the vector assigned to the *pivot* (word 1), *comparison* (word 2), and *feature* (word 3) in a given VSM. Words not found in the VSM were assigned the vector for the UNK token. Our algorithm is summarized in Algorithm 1.

Algorithm 1 Classification algorithm

```
procedure CLASSIFY(w)
    s12 ← SIM(w1, w2)
    s13 ← SIM(w1, w3)
    s23 ← SIM(w2, w3)
    if s13 − s23 > 0.015 & s12 > 0.30 & s13 > 0.1 & s23 < 0.54 then
        return 1
    else
        return 0
```

These thresholds were obtained via grid-search over both the training data and validation data, per VSM. The range of evaluated thresholds was between 0 and 0.50, with strides of 0.02. Besides choice of VSM, these thresholds were the only variable parameters in our model. The VSM used in our final submission consisted of an average of three sets of embeddings: GloVe word embeddings trained on Common Crawl (840B tokens) (Pennington et al., 2014), the same GloVe embeddings post counter-fitting (Mrkšić et al., 2016) using data from the training and valida-

tion sets, and Paragram$_{sl999}$ embeddings provided by Wieting et al. (2015), also post counter-fitting. Counter-fitting is detailed in Section 4.2. The VSM obtained by averaging these three models (**AvgVSM**) outperformed each individual VSM, and was therefore submitted as our official system.

4 Model Variations

4.1 Distributional Vector-Space models

In the course of our investigation we tested a large number of vector-space models which were generated using different methods. All VSMs were evaluated with each of our models and with our final system in order to determine the model which best encodes the discriminative information required for this task. Below is a description of all VSMs we tried:

(a) Skipgram embeddings trained on the Google News Corpus (Mikolov et al., 2013).

(b) Glove embeddings trained on Common Crawl (6B and 840B tokens).

(c) LexVec embeddings trained on Common Crawl (58B tokens) (Salle et al., 2016).

(d) Paragram embeddings trained on English Wikipedia[1] and tuned on SimLex-999.

(e) Items (b) (840B), (c), and (d) counter-fitted using the training and validation sets.

(f) AvgVSM: Average of three models cf. Section 3.3.

4.2 Counter-fitting

Counter-fitting is a method of post-processing VSMs to adapt them to certain linguistic constraints such as information from semantic lexicons or ontologies. Mrkšić et al. (2016) for instance, successfully used counter-fitting with semantic lexicons to achieve a new state of the art on the SimLex-999 similarity judgment dataset.

We employed counter-fitting to move pivot and comparison vectors (from the training set)[2] closer together in the vector-space for all training and validation examples with label 1, under the rationale that pivot and comparison words should be related for a feature to be considered discriminative.

[1]December 2, 2013 snapshot

[2]The lexicon used for counter fitting can be found at https://github.com/rutrastone/discrimemb

4.3 kNN Averaging

In evaluating VSMs, we experimented with words outside of the *pivot*, *comparison*, *feature* triples that occurred in the data. For all three words, we extracted the respective k-nearest neighbors ($k \in \{5, 10, 20\}$) and averaged their corresponding vectors. This was motivated by the intuition that it is not just the *pivot*, *comparison* pairs that determine a discriminative feature distance threshold, but also their general semantic neighborhoods. As in the main approach, we computed cosine difference thresholds via grid search for a variety of *term-neighborhood*, *neighborhood-neighborhood* combinations. Unfortunately, this approach brought marginal improvements (when tested on the validation set) at best under any configuration of thresholds, models, etc. and was thus discarded in favor of the much more lightweight **Discriminator** model.

4.4 Hierarchical Vector-Space models

Poincaré embeddings (Nickel and Kiela, 2017) are a new approach to learning representations for datasets with a latent hierarchal structure. By learning embeddings in hyperbolic spaces instead of euclidean vector spaces using an algorithm based on Riemannian optimization, this method has been shown to outperform Euclidean embeddings on datasets with latent hierarchies, such as HYPERLEX (Véronis, 2004), a dataset used to evaluate if semantic models can capture hyponymy-hypernymy or lexical entailment relationships. This can be seen as closely related to capturing concept attribute relationships, as is required in this task. We used two different sets:

- Size 50 embeddings, trained on the all WordNet common-noun hypernyms, provided by Nickel and Kiela (2017)[3];
- Size 50 embeddings trained on all feature norm derived concept-attribute pairs (i.e. all pivot-feature pairs when label is 1).

Since there was no overlap between the features in the training and testing sets, this method was not of immediate use for the task, as it could not account for the features in the test set. Therefore, our objective was solely to measure the embedding method's effectiveness in modeling the dataset's

[3]https://github.com/TatsuyaShirakawa/poincare-embedding

VSM	NN (T)	Dsc. (T)	Dsc. (V)
Skipgram	55.00	65.42	64.62
GloVe (6B)	55.38	66.06	64.30
GloVe (840B)	58.50	65.85	64.96
GloVe-cf (840B)	57.74	64.82	63.47
LexVec	59.66	65.98	64.71
LexVec-cf	**60.94**	66.29	64.92
Paragram$_{sl999}$	58.60	66.32	62.68
Paragram$_{sl999}$-cf	57.80	57.86	62.27
Poincare (Wnet)	51.71	53.56	51.78
Poincare (FN)	57.22	62.29	56.91
AvgVSM	58.63	**67.01**	**68.21**

Table 2: Performance of different Vector-space models on the official test set (T) and the validation set (V).

concept-attribute relations. To achieve this, we trained it on attribute concept pairs extracted from the feature norms which were used to build the task's dataset (McRae et al., 2005; Krebs et al., 2018). Since the use of feature norms was not permitted in this shared task, the results from this method were not submitted.

5 Results and Discussion

We report the performance of all vector-space models for each classification system in Table 2. The **NN** model is trained on the validation set[4] and all results for both models are reported on the official test set. Validation set results are only reported for the Discriminator models as the **NN** model fails to learn when training on the training set and testing on the validation set.

5.1 Discriminator

Considering its simple architecture, our system's performance was remarkable. Using nothing but cosine similarity, it performed on a level that is well above our far more complex models, and competitive with the task's best performing systems. This is demonstrative of two things: a) the information required to classify an attribute as discriminative or not with respect to two concepts is (to different extents) present in the distributional vector-space models, and b) the concatenation of the vectors associated with each word is not a sufficient feature for our trained models to learn the simple thresholds which our final model uses. We hypothesize that this is because concatenation fails to account for the interactions between the *pivot*, *comparison*, and *feature*. Further investigation is needed to assert this.

5.2 Vector-space models

Examining the degree to which different VSM could capture whether an attribute is discriminative was one of our main goals. Our initial intuition was that the relationship between concept and attribute is too specific to be adequately captured by distributional vector-space models which are, after all, based only on co-occurrence. Our results, however, contradict this expectation, showing that they are, to a certain extent, successful. For Discriminator, the best performing vector-space model was AvgVSM.

Furthermore, we found that counter-fitting using the training and validation sets did not prove effective, leading to a degradation in performance, with the exception of LexVec-cf when it did lead to improvements and AvgVSM when it was averaged with non-counter-fit models. Further investigation is required to determine exactly under what conditions counter-fitting works well.

Finally, we note that the WordNet-trained Poincaré hierarchical vector-space model had low coverage and performed poorly. However, the model trained on feature norms showed promise, particularly as it required far less space, training time, and data in order to model the dataset when compared to the distributional models.

6 Conclusion and Future work

In this paper we present Discriminator, our contribution to *SemEval 2018 Task 10: Capturing Discriminative Attributes*. Though this model is simple and does not require any training, our minimally supervised thresholding system achieved a score of 0.67, which was 0.08% below the top submitted system. We found the average of GloVe (840B), GloVe counter-fitted, and Paragram$_{sl999}$ counter-fitted vector-space models to achieve best performance in our system, out of a set of 8 different models. Future work will explore leveraging image-processing inspired models, given the intuition that such methods have the ability to capture attributes-concept relations. Preliminary work with a 2D convolutional architecture, where different sets of word embeddings serve as channels in the feature space, has shown promise.

[4]While larger, the training set is very noisy.

References

Mostafa Abdou, Artur Kulmizev, and Vinit Ravishankar. 2018. MGAD: Multilingual Generation of Analogy Datasets. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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)*.

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, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6341–6350.

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.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*.

Tobias Schnabel, Igor Labutov, David M Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *EMNLP*, pages 298–307.

Jean Véronis. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.