GHH at SemEval-2018 Task 10: Discovering Discriminative Attributes in Distributional Semantics

Mohammed Attia
Google Inc.
New York City
NY, 10011
attia@google.com

Younes Samih
Dept. of Computational Linguistics
Heinrich Heine University,
Düsseldorf, Germany
samih@phil.hhu.de

Manaal Faruqui
Google Inc.
New York City
NY, 10011
mfaruqui@google.com

Wolfgang Maier
Independent Researcher
Tübingen, Germany
wolfgang.maier@gmail.com

Abstract

This paper describes our system submission to the SemEval 2018 Task 10 on Capturing Discriminative Attributes. Given two concepts and an attribute, the task is to determine whether the attribute is semantically related to one concept and not the other. In this work we assume that discriminative attributes can be detected by discovering the association (or lack of association) between a pair of words. The hypothesis we test in this contribution is whether the semantic difference between two pairs of concepts can be treated in terms of measuring the distance between words in a vector space, or can simply be obtained as a by-product of word co-occurrence counts.

1 Introduction

Equipped with their cognitive skills, encyclopedic knowledge and linguistic competence, humans generally can identify the lexical association or semantic relation between two words or concepts with relative ease. However, building a computational model for identifying fine-grained semantic relations (such as synonymy, antonymy, hyponymy, or hypernymy, meronymy, holonymy, metonymy, containment or causality) or even detecting binary relatedness has proven to be a challenging task.

Efforts to model semantic representation computationally are generally classified into statistical and knowledge-driven semantics. This classification depends on whether the assumption is that human knowledge is encapsulated in language manifestation or that explicit manual encoding of this knowledge is needed. The statistical approach to the encoding of semantic relations is referred to as "distributional semantics" or "distributed word representations" (Speer et al., 2017), and its theoretical appeal stems from the fact that it gives practical application to the Firthian dictum "You shall know a word by the company it keeps" (Firth, 1957) which has become commonsense wisdom in lexical semantics. Features of the statistical model are extracted from unstructured data, such as words embeddings, n-gram counts, or directly from raw data.

The basic idea with word embeddings is to formulate semantic relations in arithmetic fashion by creating a vector space in which words with similar contextual embeddings have closer vectors distance (Hinton et al., 1986; Rumelhart et al., 1986; Elman, 1990; Bengio et al., 2003; Kann and Schtze, 2008; Mikolov et al., 2013c). The public availability of word embedding training programs such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) allowed researchers to create models with different parameters and dimensionality sizes for different purposes including capturing semantic relations (Gladkova et al., 2016; Attia et al., 2016).

The Google n-gram corpus (Brants and Franz, 2006) is a collection of English word n-grams and their observed counts generated from 1 trillion words of texts from web pages. This corpus has been used in many different applications including estimating word-relatedness (Islam et al., 2012),

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 947–952
New Orleans, Louisiana, June 5–6, 2018.

comparison of semantic similarity (Joubarne and Inkpen, 2011), information retrieval (Tandon and De Melo, 2010; Klein and Nelson, 2009), lexical disambiguation (Bergsma et al., 2009), improving general purpose NLP classifiers (Bergsma et al., 2010), and improving parsing performance (Pitler et al., 2010).

Knowledge-driven approaches to the detection of semantic relations rely on manually constructed lexical and encyclopedic resources, such as ConceptNet (Speer et al., 2017), ImageNet (Russakovsky et al., 2015), WordNet (Miller and Fellbaum, 1998), Wiktionary, Open Mind Common Sense (Singh et al., 2002) and DBpedia (Mendes et al., 2012).

In this work we follow a statistical based approach and show the strengths and weakness of the distributional semantics of the word vectors and n-gram frequency counts in capturing the different types of discriminative attributes.

2 Task and Data Description

The goal of the shared task on Capturing Discriminative Attributes (Krebs et al., 2018) is to detect semantic difference between pairs of concepts, or in other words, determine whether a semantic property differentiates between two possibly related concepts. For example both 'bear' and 'goat' are animals, but only a 'bear' has 'claws'. Therefore 'claws' is considered as a discriminative feature.

The shared task data is formatted in triples that represent a ternary relation between two concepts ($Word_1$, $Word_2$) on one hand and an attribute ($Word_3$) on the other. $Word_3$ is considered as a discriminative attribute if, and only if, it characterizes $Word_1$ but not $Word_2$. For example, in the triple (*sailboat,yacht,mast*), 'mast' is discriminative as it is found in $Word_1$, 'sailboat', but not in $Word_2$. By contrast, in the triple (*goose,duck,flies*) the event 'flies' is not discriminative as it characterizes both entities. Similarly in the triple (*pickle,lemon,round*), 'round' is not a discriminative feature, as it characterizes $Word_2$, not $Word_1$.

The size of the shared task data is described in Table 1. It is to be noted that there is no intersection between the discriminative attributes in any of the datasets. We think the purpose is to make sure that the participating systems are able to learn how to estimate the relations, regardless of the lexical items involved.

Dataset	# of triples	# of attributes
Training	17,547	1,292
Validation	2,722	576
Test	2,340	577

Table 1: Sizes of the shared task datasets.

3 System Description

In our system we use a deep neural network for the binary classification of discriminative attributes. The basic idea with deep learning is to use hidden layers of neural nets to automatically capture the underlying factors that lead from the input to the output, eliminating the need for feature engineering.

The system is trained on features extracted from two main publicly available resources that fall within the paradigm of unstructured data as no manual lexical or encyclopedic knowledge is encoded. The two resources are the Google n-gram counts and the Google News Word2Vec.

Google n-gram counts. We use the Google 5-gram counts as provided by Google Books ngrams[1] (Michel et al., 2011; Lin et al., 2012).

Google News Word2Vec. This is a publicly available pre-trained word vector[2], built with the word2vec architecture (Mikolov et al., 2013b) from a news corpus of 100B words (3M vocabulary entries) with 300 dimensions, negative sampling, using continuous bag of words and window size of 5.

3.1 Features Used

We describe the features used to train our DNN binary classifier to detect discriminative attributes. In this section we use the abbreviations W_1, W_2, and W_3 for $Word_1$, $Word_2$, and $Word_3$, respectively.

We use pre-trained word vectors in order to obtain similarity scores between words. This leads to the following features.

- $distW_1W_3$: Cosine distance between W_1 and W_3
- $distW_2W_3$: Cosine distance between W_2 and W_3
- *cosDiff*: Difference between $distW_1W_3$ and $distW_2W_3$

[1] https://books.google.com/ngrams/info
[2] https://goo.gl/tyVGqW

- *similarityCompare*: We compute the cosine similarity between two sets of words using the Gensim 'n_similarity' function. So it gives a single number for comparing the similarity between W_1 and W_3, and W_2 and W_3.

In order to capture all morphological variations of the words, we use word lemmas and then expand to all variants that share the same lemma.

- *lemmaDist*W_1W_3*Ex*: The average cosine distance between W_1 and all lemma expansions of W_3
- *lemmaDist*W_2W_3*Ex*: The average cosine distance between W_2 and all lemma expansions of W_3

We use to Google 5-gram counts to obtain the following features.

- *cnt*W_1W_3: counts of W_1 and W_3 co-occurring
- *cnt*W_2W_3: counts of W_2 and W_3 co-occurring
- *cnt*W_1W_3*Ex*: counts of W_1 and the lemma expansions of W_3 co-occurring
- *cnt*W_2W_3*Ex*: counts of W_2 and the lemma expansions of W_3 co-occurring

3.2 Machine Learning Models

We use a deep neural network model for the binary classification of attributes as either True or False (or discriminative or non-discriminative) based on the set of features described above.

We use a simple and straight-forward architecture consisting of 5 feed-forward fully-connected (or dense) layers with single dropout layer with a rate of 0.3. The network is narrow on the top and wide on the bottom. The function of the dropout layer (Hinton et al., 2012) is to mitigate overfitting and make sure that our model learns significant representations by randomly omitting a certain percentage of the neurons in the hidden layer for each presentation of the samples during training. This encourages each neuron to depend less on other neurons and to try to learn generalizations. Table 2 shows the layer configuration of the model.

4 Experiments and Results

We test our system on various combination of the features mentioned in subsection 3.1. We assume the baseline is 50% as this is what a random system would generate given that the validation set has an almost equal number of True's and False's. Table 3 shows the system results on the dev set, with the last row showing results on the test set using our best model, "all features". Surprisingly, using the cosine distance between pairs of words gives a low score (56.17%) which is slightly above the baseline, indicating the ineffectiveness of cosine distances in capturing this kind of relationships. Word counts alone were the most impactful of all the features.

Layer type	Output Shape	Param #
Dense_1	(None, 12)	132
Dropout_1	(None, 12)	0
Dense_2	(None, 12)	156
Dense_3	(None, 100)	1300
Dense_4	(None, 200)	20200
Dense_5	(None, 1)	201

Table 2: Neural Network Layout.

5 Error Analysis

In order to be able to analyze the performance of the system and identify where it is faring well and where it is failing, we first manually classify the relations between concepts and attributes in the validation set into 8 types.

1. **Part-whole**. This is when the attribute denotes an entity that can be part or whole of $concept_1$, e.g. *tractor, wheels*; *moose, legs*; *cat, eyes*; *iguana, tongue*; *condos, rooms*.
2. **Container-contained**. This is when the entity attribute can be located/situated physically or temporally in $concept_1$, e.g. *oven, kitchen*; *fort, cannons*; *mouse, house*; *priest, parish*; *surfboard, water*.
3. **Made-of**. This is when the entity attribute is a material of which $concept_1$ can be made, e.g. *cart, wood*; *wire, metal*; *rum, sugarcane*; *scarf, wool*; *wine, grape*; *roof, clay*.
4. **Agent-patient**. This is when the attribute is a topic or theme on which $concept_1$ can act on, e.g. *politician, politics*; *physiotherapist, muscles*; *dermatologist, skin*; *mammals, milk*.
5. **HasAttribute**. This is when the attribute is an adjective that can be used to describe *con-*

Features	Accuracy
baseline	50.00
$distW_1W_3$, $distW_2W_3$	56.17
$distW_1W_3$, $distW_2W_3$, $lemmaDistW_1W_3Ex$, $lemmaDistW_2W_3Ex$	55.79
cosDiff, *similarityCompare*	59.12
$cntW_1W_3$, $cntW_2W_3$	65.27
$cntW_1W_3$, $cntW_2W_3$, $cntW_1W_3Ex$, $cntW_2W_3Ex$	65.45
all features	**66.50**
result on the test set	65.17

Table 3: System results with different feature combinations.

class	Total	%	correct	%
event	346	12.71	260	75.14
containment	228	8.38	167	73.25
made-of	158	5.80	113	71.52
relates-to	164	6.02	115	70.12
agent-patient	121	4.45	84	69.42
part-whole	524	19.25	361	68.89
hasAttribute	850	31.23	515	60.59
hyper-hypo	331	12.16	196	59.21
Total	2,722		1,811	66.53

Table 4: Discriminative classes sorted by system performance.

cept$_1$, e.g. *garlic, white*; *girl, virgin*; *alligator, long*; *tuna, large*; *honey, sweet*; *pumpkin, round*.

6. **Hyper-hypo**. That is when the attribute is a hyponym or hypernym of the concept, e.g. *rum, alcohol*; *orthodontist, profession*; *steak, meat*; *mother, female*; *lorry, vehicle*; *lavender, plant*.

7. **Event**. That is when the attribute is a verb that is associated with the concept/entity, e.g. *woman, talk*; *educator, teaches*; *knee, bend*; *tuna, swims*; *frog, jumps*; *shirt, wear*; *seabirds, fly*; *novelist, write*.

8. **Relates-to**. This is when the relationship cannot stated with any of the aforementioned types, e.g. *bus, passengers*; *knee, pads*; *lung, transplant*; *widow, death*; *brother, sister*; *uncle, nephew*.

Table 4 shows our manual classification of the discriminative attributes in the validation set. It is to be noted that the majority of relations (62.64%) are of three types: hasAttribute, part-whole and hyper-hypo.

The types of discriminative features in Table 4 are sorted by system performance, highlighting strengths and weaknesses of the system. The deep learning algorithm assumes that the attribute is discriminative for *concept*$_1$ if it has considerably higher n-gram counts with *concept*$_1$ than with *concept*$_2$. In the upper end n-gram counts shows strength in dealing with events and container-contained relationships, where co-occurrence statistics showed to be very helpful. The examples below shows frequency counts that indicate stronger relation between *Word*$_1$ and *Word*$_2$ than between *Word*$_2$ and *Word*$_3$. Gold answers are the numbers (0 or 1) following the triples.
(*shoulder, cheek, carry*, 1), $cntW_1W_3$: 104620, $cntW_2W_3$: 498
(*teacher, pupil, teaches*, 1), $cntW_1W_3$: 134656, $cntW_2W_3$: 0
(*albums, music, picture*, 1), $cntW_1W_3$: 3937564, $cntW_2W_3$: 374572

It is to be mentioned that in the validation set, there were 246 (9%) instances where no frequency counts were found for either concepts.

In the lower end of our system performance there were the classes of *hasAttribute*, *part-whole* and *hyper-hypo*. As these classes constitute the majority of the data, the overall system performance is compromised. We make further detailed analysis of our top losses with *hasAttribute* and *part-whole*.

Analysis of Errors with *hasAttribute*
Most of the errors in this class can be identified with one of three reasons.

- N-gram counts are not aware of the qualification scope. For example, in the tuple below, 'large' has equally high frequency with 'brick', not because a brick can be large, but

they co-occur in phrases like, "large brick house/ranch"
(*garage, brick, large*, 1), $cntW_1W_3$: 245802, $cntW_2W_3$: 193816

- Contrary to common sense knowledge, data could prove the association between a concept and attribute that might not be readily perceived. The example below shows that "green tomato" is not a rarity. This could indicate an error with manual annotation of the data.
(*zucchini, tomato, green*, 1), $cntW_1W_3$: 29280, $cntW_2W_3$: 179646

- The collocation between the attribute and $concept_2$ could be higher than with $concept_1$.
(*drizzle, rain, light*, 1), $cntW_1W_3$: 231348, $cntW_2W_3$: 4108548

Analysis of Errors with *part-whole*
Similarly the errors in this class can be attributed to one of three causes.

- Disproportionate frequency count, which could be tied to the disparity in the individual frequency of the concepts themselves. This might be solved by taking the n-gram count as a function of the unigram counts of the concepts themselves.
(*car, taxi, wheels*, 0), $cntW_1W_3$: 504848, $cntW_2W_3$: 2734

- There could be an association of different kind between $concept_2$ and the attribute that yield higher frequency counts. For instance in the example below, 'garlic' and 'wings' have higher frequency, not because garlic has wings, but because they co-occur in phrases like "garlic chicken wings".
(*pheasant, garlic, wings*, 1), $cntW_1W_3$: 500, $cntW_2W_3$: 11136

- Either of the two concepts has no n-gram co-occurrence with the given attribute leading to missing information.
(*owl, buzzard, eyes*, 0), $cntW_1W_3$: 10088, $cntW_2W_3$: 0

6 Conclusion

In this paper we have presented our system for detecting discriminative features using distributional semantics. We have shown that, without resort to human knowledge, a great deal of encyclopedic knowledge can be captured from unstructured data. We also conducted a detailed error analysis which shows the strengths and weaknesses of the system.

In its quest to approximate the distance between words with similar contexts, the cosine distance becomes oblivious to the internal intrinsic relationship between words and their immediate neighbors, and this is why many relations that are induced from co-occurrence counts are not captured by cosine distance.

While n-gram counts from raw data can present a great wealth for mining for lexical information and inducing semantic knowledge, co-occurrence counts can suffer from considerable constraints when two or more adjacent words have different scope of predication or qualification. For example, while "wood spoon" has a high frequency due to the semantic relation of 'made-of', "wood pepper" has an even higher frequency count, not due to any semantic relationship, but because 'wood' is scoped to a subsequent word, "wood pepper mill". If syntactic information related to the head of noun compounds and scope of modification, more meaningful assumptions can be made.

References

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. Cogalex-v shared task: Ghhh - detecting semantic relations via word embeddings. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 86–91.

Y. Bengio, R. Ducharme, and P. Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Shane Bergsma, Dekang Lin, and Randy Goebel. 2009. Web-scale n-gram models for lexical disambiguation. In *IJCAI*, volume 9, pages 1507–1512.

Shane Bergsma, Emily Pitler, and Dekang Lin. 2010. Creating robust supervised classifiers via web-scale n-gram data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 865–874. Association for Computational Linguistics.

Thorsten Brants and Alex Franz. 2006. The google web 1t 5-gram corpus version 1.1. In *LDC2006T13*.

J. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

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.

G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. 1986. Distributed representations. *In: Parallel distributed processing: Explorations in the microstructure of cognition*, 1.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Aminul Islam, Evangelos Milios, and Vlado Keselj. 2012. Comparing word relatedness measures based on google n-grams. *Proceedings of COLING 2012: Posters*, pages 495–506.

Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the google n-gram corpus and second-order co-occurrence measures. In *Canadian Conference on Artificial Intelligence*, pages 216–221. Springer.

Katharina Kann and Hinrich Schtze. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Martin Klein and Michael L Nelson. 2009. Correlation of term count and document frequency for google n-grams. In *European Conference on Information Retrieval*, pages 620–627. Springer.

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)*.

Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, William Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google Books Ngram Corpus. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics Volume 2: Demo Papers (ACL '12)*.

Pablo N Mendes, Max Jakob, and Christian Bizer. 2012. Dbpedia: A multilingual cross-domain knowledge base. In *LREC*, pages 1813–1817. Citeseer.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, William Brockman, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Aiden Lieberman. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *In Proceedings of International Conference on Learning Representations (ICLR) 2013. arXiv:1301.3781v3*, pages 746–751, Scottsdale, AZ.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT 2013*, pages 746–751, Atlanta, Georgia.

George Miller and Christiane Fellbaum. 1998. Wordnet: An electronic lexical database.

Jeffrey Pennington, Richard Socher, and Christopher D. 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, Doha, Qatar.

Emily Pitler, Shane Bergsma, Dekang Lin, and Kenneth Church. 2010. Using web-scale n-grams to improve base np parsing performance. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 886–894. Association for Computational Linguistics.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. Learning internal representations by back-propagating errors. *Nature. 323:533.536.*

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Niket Tandon and Gerard De Melo. 2010. Information extraction from web-scale n-gram data. In *Web N-gram Workshop*, volume 7.