AmritaNLP at SemEval-2018 Task 10: Capturing discriminative attributes using convolution neural network over global vector representation.

Vivek Vinayan, Anand Kumar M, Soman K P
Center for Computational Engineering and Networking (CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
cb.en.p2cen16018@cb.students.amrita.edu,
m_anandkumar@cb.amrita.edu

Abstract

The "Capturing Discriminative Attributes" sharedtask is the tenth task, conjoint with SemEval2018. The task is to predict if a word can capture distinguishing attributes of one word from another. We use GloVe word embedding, pre-trained on openly sourced corpus for this task. A base representation is initially established over varied dimensions. These representations are evaluated based on validation scores over two models, first on an SVM based classifier and second on a one dimension CNN model. The scores are used to further develop the representation with vector combinations, by considering various distance measures. These measures correspond to offset vectors which are concatenated as features, mainly to improve upon the F1score, with the best accuracy. The features are then further tuned on the validation scores, to achieve highest F1score. Our evaluation narrowed down to two representations, classified on CNN models, having a total dimension length of 1204 & 1203 for the final submissions. Of the two, the latter feature representation delivered our best F1score of 0.658024 (as per result[1].)

1 Introduction

As famously quoted by firth "You shall know a word by the company it keeps" that is, the semantic information embedded in a representation can only be described by the words surrounding it. This can only get you somewhere when, company itself is unambiguous and a representation goes through capturing "hypothetically" every sense of the word over a corpus. The capturing discriminative attributes sharedtask, conducted with SemEval(2018) is a task proposed by alicia kerbs and denis paperno (2016). It describes, how lexical similarity may not be enough to access qualitatively, the semantic information for a multitude of tasks. Wherein they propose that, with this task, a system can be modelled for effectively extracting certain semantic differences in the words for understanding the sense embedded within them. This is provided as a proof of concept dataset for this sharedtask, where a certain word is used to check if it can distinguish between a pair of words. The dataset in itself seems simple where, in the training set a label information for the two classes, positive or negative are provided making this a binary classification task.

The three words that are provided in each instance are given in the order as, a pivot word followed by a compare word and ending with a attribute or feature word, that may or may not be associated with the pivot word. Based on the last word it is decided, if that attribute word actually is a distinguishing feature that is able to discriminate the pivot word from that of the compare word. *e.g* (*apple,banana,red*) here apple is the pivot word, banana the compare word and red, the word which decides if this is a feature that can be associate with apple to distinguish it from banana. This is a rather oversimplified example to a human, as from a very young age we are taught to distinguish objects based on visual aid, which simplifies the task for us as we have embedded subconsciously to differentiate the fruits mainly based on their color or size. This information is seldom used to describe the fruits when illustrated in written form, thus lacking that visual form of information for a machine to make this judgment call, making it that much more difficult to take an informed decision. Their work is based on a method, that was presented by Lazaridou et al. (2016) for prediction of distinguishing feature with use of image as reference for visual discrimination attribute identification task, more prominently it was related to capturing of lexical information using offset vectors.

[1]Results/Evaluation under the team name "AmritaNLP"

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1003–1007
New Orleans, Louisiana, June 5–6, 2018.

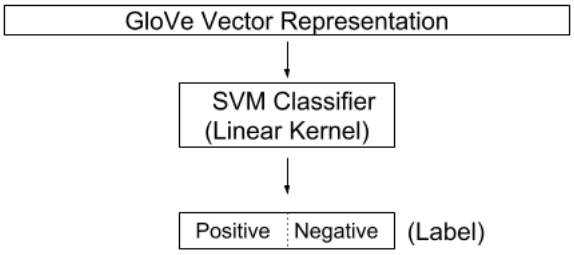

Figure 1: SVM architecture for feature representation.

2 Dataset

The dataset in the sharedtask2018 (Krebs and Paperno, 2018) is divided into three sets namely train test and validation. The training set contains automatically generated examples which are not manually curated. Whereas, the test and validation set are manually verified examples which include just over 5000 instances. The test set instances are made keeping in consideration that feature word overlap between the words in train and test are minimal. The validation set is similar to that of the test set and is used for parameter tuning of the models.

There are in total 17782 instances in the training set, 2722 in the validation set and 2340 in the test set. With the automated nature of the data, the training set is noisier in comparison to that of the validation and test set.

In the dataset, positive examples are annotated with the label '1', signifying that the attribute/feature word is a positive association only to the pivot word in the order presented and not vice verse. *e.g.* (*airplane,helicopter,wings*) here 'wings' is an attribute only associated to 'airplane', whereas *(helicopter,airplane,wings)* is an invalid entry. The combination of (*helicopter,airplane*) in this order will only be added if the concept 'helicopter' has a feature that airplane does not have in this set.

On the other hand, the negative examples are annotated with label '0' at the end. These are considered when the attribute/feature words are either similar to both pivot and the compare word or are dissimilar to them, *e.g. (Tractor,scooter,wheels), (Spider,elephant,legs) e.t.c*.

In the training dataset, there is a total of 508 unique concepts (pivot) words, of which 375 words have positive attributes and 505 of these have negative attributes, seeing the big contrast between the two labeled attributes we can infer that not every concept word has an equal proportion of labeled instances.

W_p	Pivot word
W_c	Compare word
W_a	Attribute word
Cos_p	cosine_similarity (W_pW_a)
Cos_c	cosine_similarity (W_cW_a)
Dis_p	kulsinski(W_pW_a)
Dis_c	kulsinski(W_cW_a)
Min_p	minkowski(W_pW_ap=1)
Min_c	minkowski(W_cW_ap=1)
Coref_p	corrcoef(W_pW_a)
Coref_c	corrcoef(W_cW_a)
Sqeu	sqeuclidean(W,W_a)

Table 1: Nomenclature for feature representation.

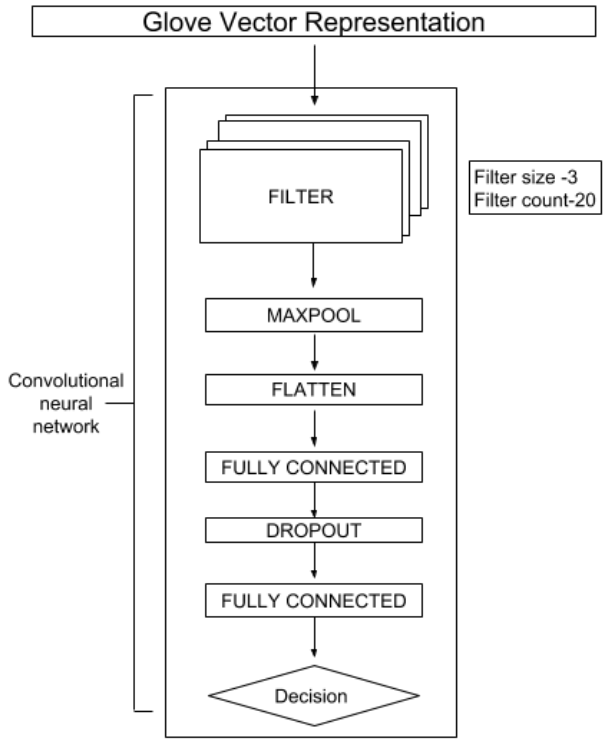

Figure 2: Convolution neural network architecture for feature representation.

3 Methodology

Here discussed are methods which are considered in our implementation. On a cursory look at the dataset, we decided to go with a pre-trained representation of the words, rather than preparing a word embedded representation of the dataset. This is devised with a notion that, word pair associated models on this dataset would not help educate the

SN	GloVe Pre-trained word Vectors	Representation	Representation Length	Validation	
				Accuracy	F1score
1	6B_50d	W_p,W_c,W_a	150	51.16	42.20
2		W_p,W_a,W_c,W_a	200	51.08	42.10
3		W_p,W_a,W_c,W_a, Cos_p	201	51.48	51.10
4		W_p,W_a,W_c,W_a, Cos_p, Dis_p	202	51.37	50.90
5		CR 1	200	50.28	41.90
6		CR 2,Cos	202	47.59	46.00
7		CR 2,Cos,Min_c,Min_p	204	49.61	46.80
8	6B_300d	W_p,W_c,W_a	900	52.04	45.80
9		W_p,W_a,W_c,W_a	1200	51.64	45.57
10	840B_300d	W_p,W_c,W_a	900	51.10	41.00
11		W_p,W_a,W_c,W_a	1200	51.41	40.34

Table 2: Validation accuracy for varied dimension GloVe representation using SVM.

SN	GloVe Pre-trained word Vectors	Representation	Representation Length	Validation	
				Accuracy	F1score
1	6B_300d	W_p,W_c,W_a	900	51.3	41.1
2		W_p,W_a,W_c,W_a	1200	50.9	43.8
3		CR 1	1200	50.8	42.2
4	840B_300d	W_p,W_c,W_a	900	52.0	51.0
5		W_p,W_a,W_c,W_a	1200	52.0	45.3
6		CR 1	1200	53.8	48.4

Table 3: Validation accuracy for varied dimension GloVe representation using CNN.

embedding. Further, using the pre-trained embeddings, the representation are evaluated based on validation accuracy with machine learning techniques like SVM, where we use ten fold ten cross with linear kernel for validation. This algorithm was earlier explored for sense disambiguation of a native language (Tamil), having rich feature representation presented in his work by Anand Kumar et al. (2014a), and is also implemented in his work (2014b). A simple one dimension convolution neural networks model is also illustrated upon, based on the works by Vinayakumar et al. (2017). The CNN model is fixed on an empirical method where the representation is convoluted with *twenty* filters, of size *three*, on a batch size of *sixty-four*, with activation ReLU over a wayward *ten* epochs, which are flattened and reduced to *thirty-two* and later to one at the final layer for evaluation. The architectures for the models, are as shown in Figure 1 & 2 respectively.

Moving ahead, a GloVe pre-trained word embedding (Pennington et al., 2014) of various dimensions are considered, which is learned over public data, available under the PDDL.[2] (100, 300 dimension word representation, embedded over 6B, 840B sizes common crawl corpus are considered). The focus is on using one of these representations for our base method. Upon these embedding, various distance, dissimilarity and similarity measures are considered, to provide a measure between vectors or in our case between the words.

In Table 1, provided are abbreviations that we used through out the upcoming discussion regarding the methods and the representations. With the implementation of pre-trained vectors, we refer few vector measurement technique that could be used to measure a sense of semantic similarity among them. These vector carry within them a spacial correlation between words which has be discussed in their work by (Pennington et al.,

[2] Public Domain Dedication and License v1.0. http://www.opendatacommons.org/licenses/pddl/1.0/.

SN	Conditional representation (CR)	
	If :	Else :
1	W_p,(W_p + W_a),W_c,W_a	W_p,(W_p – W_a),W_c,(W_c – W_a)
2	W_p,(W_p+ W_a),W_c,W_a,(Dis_c – Dis_p)	W_p,(W_p– W_a),W_c,(W_c – W_a),(Dis_c – Dis_p)

Table 4: Various feature representation taken for the classification task.

SN	GloVe Pre-trained word Vectors	Representation	Representation length	Validation	
				Accuracy	F1score
1	840B_300d	CR 1, Dis	1201	53.1	47.3
2		CR 1, Cos	1201	50.3	50.1
3		CR 1, Dis, Cos	1202	55.1	53.1
4		CR 1, Min, Dis_p, Dis_c	1203	50.9	51.6
5		CR 1, (Coref_p – Coref_c), Min_c, Min_p	1203	51.1	56.8
6		CR 1, Cos, Min_c, Min_p	1203	54.6	58.9
7		CR 2	1201	51.3	45.6
8		CR 2, (Coref_p – Coref_c), Min_c, Min_p	1204	55.7	56.4
9		CR 2, (Min_c – Min_p)	1202	61.5	55.8
10		CR 2, Cos, Min_c, Min_p	1204	**60.1**	**56.1**
11		CR 2, Min_c, Min_p	1203	**61.1**	**60.2**
12		CR 2, (Cos_p – Cos_c), Min_c, Min_p	1204	58.9	51.3
13		CR 2, Cos, Min_c, Min_p, Sqeu	1205	54.4	54.9

Table 5: Validation accuracy of 300 dimension, GloVe representations on 840B common crawl tokens using CNN.

2014).

Initially, a simple concatenation of the three words is considered as an instance, which are the pivot(W_p), compare(W_c) and attribute(W_a) words, for the entire dataset. The same representation is taken of two different dimensions lengths as mentioned earlier. Based on the model fit across training data, the validation accuracy and F1score are measured, these are as shown in Table 2. Similarly, these representations are also passed on to a convolution neural network, where their respective accuracy and F1scores are measured and shown in Table 3.

With an empirical approach, the representations are further extended by appending (W_a) to (W_p) and (W_c) sequentially and passing it to the two models(As shown by representation two in the Table 2). The SVM model did not show any significant improvement in the score, over the representations. In comparison, the CNN model observed a slight improvement in scores on the same representation. Word embedding being a vector representation in higher dimensional space, has proved (Pennington et al., 2014) to captures spatial information, that can be employed to use as features for the representation. This is exerted by using certain measures between the (W_p), (W_a) and (W_c), (W_a). These measures are calculated using Scipy libraries (Jones et al., 2001) and Sklearn library (Pedregosa et al., 2011) to find the distance, similarity and dissimilarity measure between the two 1-D array words. The similarity of the two words indicates how similarly associated these words are, this measure is calculated using the cosine distance which is a scalar representation that signifies, larger the number between the two words the more similar they are. Whereas, the dissimilarity is the vice-versa of this measure. Of the various distance measures explored, we considered euclidean, chebyshev, sqeuclidean, minkowski and for dissimilarity measures jaccard, kulsinski, Hamming and these are implemented using the Scipy (Jones et al., 2001) library. Amongst the measures considered, kulsinski dissimilarity gave the nearest disambiguation between the comparison of W_p, W_a and W_c, W_a, thus we chose it as the threshold measured for differentiating the representations between a positive and a negative instances i.e if the dissimilarity of W_c, W_a is greater than that of the W_p, W_a then the W_a were added to the W_P and concatenated to form a representation. Otherwise the second representation is considered where the W_a is subtracted from both the words. This is as shown in the first conditional representation (CR) of the Table 4.

The CR based representation accuracy decreased for SVM models. Whereas, the F1score and accuracy increased for the CNN model over the initial representations shown in Table 3. Thus, the further representation where improved on the CNN model to achieve better F1score with good accuracy. Comparing the two GloVe pre-trained vectors of 300 dimension for varied corpus size shown in Table 3, the 840B_300d trained model has achieved better F1score and accuracy compared to the other, thus moving along further with word embedding.

In Table 5 we see that subsequent representations, built upon the simple representation of CR1 are concatenated with kulsinski(Dis[3]) distance and Cosine similarity (Cos[3]) have improved the F1score. As show in the third representation, where the F1score increased to 53.1% with a considerable accuracy over the previous iteration. Further improvisation on CR1 representation with different features like correlation coefficient have increased the F1score to 56.8% but brought down the accuracy. Representation six is the next feature representation for which the accuracy, as well as the F1score, increases to 54.6% and 58.9% respectively. After many iterations of adding features, the representation eleven is the one that gave the highest F1score with the best accuracy, and this representation based model is submitted along with the representation ten, which also had good F1score, but a lower accuracy on the validation dataset.

4 Results & Conclusion

The tenth and eleventh representation of Table 5 are the two feature set based on CNN models, which are predicted on the test set and submitted for the competition. The results published for our models showed that the first set was scored at 0.52, where as the second set was scored at 0.66 for F1score. Comparing the predicted labels of the two systems with that of the gold standard, we see that our system fit over the tenth representation predicted correctly only 399 of 1293 as negative example and 855 of 1047 as the positive example. On the eleventh representation it gave 857 of 1293 and 687 of 1047 for negative and positive example respectively. Comparing the outcomes of the systems we see that majority of negative sample are mis-classified for system ten, on the other hand, the eleventh system improved upon this classification of the negative samples which increased the F1score for the system.

[3]These representations are same as earlier mentioned in Table 1 wherein here the pivot word based measure is taken for the 'if' condition and the compare word based on the other

References

M Anand Kumar, S Rajendran, and KP Soman. 2014a. Tamil word sense disambiguation using support vector machines with rich features. *International Journal of Applied Engineering Research*, 9(20):7609–20.

M Anand Kumar and KP Soman. 2014b. Amrita-cen@ fire-2014: Morpheme extraction and lemmatization for tamil using machine learning. In *ACM International Conference Proceeding Series*, pages 112–20.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001. SciPy: Open source scientific tools for Python.

Alicia Krebs and Denis Paperno. 2016. Capturing discriminative attributes in a distributional space: Task proposal. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 51–54.

Alicia Krebs and Denis Paperno. 2018. Semeval-2018 Task 10: Capturing discriminative attributes. In *Proceedings of 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 their discriminative properties. *arXiv preprint arXiv:1603.02618*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

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.

R. Vinayakumar, K. P. Soman, and Prabaharan Poornachandran. 2017. Applying convolutional neural network for network intrusion detection. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1222–1228.