Emotion Enriched Retrofitted Word Embeddings

Sapan Shah[1,2], **Sreedhar Reddy**[1], **and Pushpak Bhattacharyya**[2]

[1]TCS Research, Tata Consultancy Services, Pune
[2]Indian Institute of Technology Bombay, Mumbai
{sapan.hs,sreedhar.reddy}@tcs.com
pb@cse.iitb.ac.in

Abstract

Word embeddings learned using the distributional hypothesis (e.g., GloVe, Word2vec) are good at encoding various lexical-semantic relations. However, they do not capture the emotion aspects of words. We present a novel retrofitting method for updating the vectors of emotion bearing words like fun, offence, angry, etc. The retrofitted embeddings achieve better inter-cluster and intra-cluster distance for words having the same emotions, e.g., the *joy* cluster containing words like fun, happiness, etc., and the *anger* cluster with words like offence, rage, etc., as evaluated through different cluster quality metrics. For the downstream tasks on sentiment analysis and sarcasm detection, simple classification models, such as SVM and Attention Net, learned using our retrofitted embeddings perform better than their pre-trained counterparts (about 1.5% improvement in F1-score) as well as other benchmarks. Furthermore, the difference in performance is more pronounced in the limited data setting.

1 Introduction

Word embedding models inspired from the distributional hypothesis (Harris, 1954) have one major limitation: they mix semantic similarity with other types of semantic relatedness (Hill et al., 2015). For instance, consider `cheap` and `expensive`. Though opposite in meaning, the distributional vectors of these words are similar since they occur in nearly identical contexts. This is problematic for many applications such as text simplification, dialogue state tracking, etc. To address this, researchers have proposed various models that leverage knowledge resources to improve word embeddings. At a high level, these models are categorized into two types: Joint specialization models (Yu and Dredze, 2014; Liu et al., 2015); and Retrofitting (post-processing) models (Faruqui et al., 2015; Mrkšić et al., 2016). Joint specialization models typically modify the optimization objective of distributional models by integrating external knowledge into the objective function. In contrast, retrofitting models first generate training data from knowledge resources in the form of constraints and then modify the pre-trained embeddings in a post-processing step so that they respect the constraints. These approaches focus mainly on constraints from relations such as synonymy, antonymy, hypernymy, etc., that are present in WordNet, Paraphrase database, etc.

word pair	GloVe	RETripletGBal
(angry, offence)	0.2339	0.3924
(angry, enjoy)	0.3400	0.2950
(fun, closeness)	0.2232	0.3688
(fun, miserable)	0.3105	0.2812

Table 1: Cosine similarity between words from same and different emotion categories: pre-trained GloVe vs. embeddings retrofitted by our method RETripletGBal

While pre-trained embeddings and their retrofitted versions are good at encoding various lexico-semantic relations, they do not consider the emotion content of words. For example, consider words such as `angry` and `offence` that evoke *anger* emotion and words such as `fun` and `enjoy` eliciting *joy*. Table 1 shows cosine similarity as computed using pre-trained GloVe embeddings. Even though `angry` and `offence` evoke the same emotion (*anger*), their cosine similarity is lower than that between `angry` and `enjoy`, a pair of words eliciting different emotions, pointing to the shortcomings of existing embedding models. Recently, a few attempts have used affective lexicons (Khosla et al., 2018; Seyeditabari et al., 2019) or task-dependent distant supervision (Tang et al., 2016; Agrawal et al., 2018) to induce emotion embeddings. While they work well for some tasks, they do not generalize well across tasks and have not been tested extensively for intrinsic quality.

Emotion	anger	joy	sadness	fear	anticipation	surprise	trust	disgust
#words	543	651	1153	772	623	318	1197	1024

Table 2: EmoLex statistics: number of words annotated with Plutchik's eight basic emotion categories

In this work, we present a novel retrofitting method to learn emotion enriched embeddings. For knowledge, it relies on word-level emotion annotations available in the NRC word-emotion association lexicon (known as EmoLex). The central idea is: if words w_a and w_p are associated with the same emotion category t and word w_n is not associated with t, then w_a is semantically closer to w_p than w_n in the context of the emotion category t. This can be stated as an inequality constraint: $sim_t(w_a, w_p) > sim_t(w_a, w_n)$. Such emotion inequality constraints containing word triplets (w_a, w_p, w_n) are generated for all emotion categories present in EmoLex. We use these constraints as training data to learn a non-linear transformation function that maps original word vectors to a vector space respecting these constraints. The transformation function is learned in a similarity metric learning setting using a multi-layer feed-forward network.

The embeddings retrofitted using our method achieve better clustering for emotion bearing words. For the downstream tasks on sentiment analysis and sarcasm detection, they perform better than their pre-trained counterparts and other benchmarks, with significant gains in limited data setting. The main contributions of this work are:

1. A novel retrofitting method to learn emotion enriched embeddings in a similarity metric learning setting (Section 3).

2. A detailed evaluation of word embeddings for their emotion content using clustering experiments (Section 4.1).

3. A detailed evaluation on sentiment analysis and sarcasm detection showing the efficacy of our retrofitting method (Section 4.2).

2 Constraints from NRC EmoLex

A large body of work has focussed on understanding and modelling human emotions. For instance, Plutchik's wheel of emotions (Plutchik, 1980), Ekman's model (Ekman, 1992), Parrot's tree-structured emotions (Parrott, 2001), etc. The model proposed by Plutchik arranges emotions in circles with the length of radius indicating the intensity of emotions. It proposes eight basic or primary emotions: joy, trust, fear, surprise, sadness, disgust, anger, and anticipation.

Various lexical resources have been proposed in the literature to capture the emotion aspect of words, e.g. (Mohammad, 2018a,b). In this work, we focus on NRC EmoLex (Mohammad and Turney, 2013). It contains a list of English words and their associations with Plutchik's eight basic emotions (Plutchik, 1980). Since words are ambiguous in their meaning and may evoke multiple emotions, each word in EmoLex has been associated with a set of emotions. For example, `playful` is associated with three emotion categories: *trust*, *surprise* and *joy*. Table 2 shows the total number of words annotated with each emotion category[1].

We obtain a set of inequality constraints from EmoLex in the form of triplets. Each triplet contains three words (w_a, w_p, w_n) in which we refer to w_a, w_p and w_n as the *anchor*, *positive* and *negative* words, respectively. The corresponding inequality constraint is: similarity between w_a and w_p shall be greater than the similarity between w_a and w_n, by at least a margin m. The margin m is set in the range $[0, 2]$ corresponding to a minimum versus maximum separation on cosine distance. For example, consider the following word-emotion pairs in EmoLex: (`lonely`, *sadness*), (`playful`, *joy*), and (`sorrow`, *sadness*). With `lonely` as the anchor word, `sorrow` can be considered the positive word since both these words belong to the same emotion category *sadness*. The word `playful` is then considered as a negative word since it is annotated with a different emotion category *joy*. This gives rise to the following constraint in the context of *sadness* category: $sim_{sadness}(lonely, sorrow) > sim_{sadness}(lonely, playful) + m_{sadness}$. Such constraints are obtained in the context of all the eight emotion categories by considering each word from the corresponding emotion category as the potential anchor and then generating the positive and negative words.

[1] EmoLex contains emotion and sentiment annotations for 14,182 words. Out of these, it has a total of 4,463 emotion bearing words i.e. words that are marked with at least one emotion category.

3 Retrofitting Method

Our goal is to learn a transformation function that maps pre-trained word embeddings to a vector space that respects the emotion inequality constraints. As explained earlier, an inequality constraint is created using a word triplet (w_a, w_p, w_n) and the corresponding emotion category-specific *margin* m. Thus, a natural way to create training data for this task is to generate a set of four tuples (w_a, w_p, w_n, m) using all possible inequality constraints. However, training data generation using all inequality constraints has a drawback. We first explain this drawback and ways to mitigate it, followed by our retrofitting model to learn the transformation function.

3.1 Training: Batch of triplets To Batch of words

Let's define the set of triplets that satisfy inequality constraints (hence zero loss) as *easy triplets*, and conversely, the set of triplets that do not satisfy the constraints (hence leading to non-zero loss) as *active triplets*. The constraint generation method described in Section 2 produces $O(n^3)$ triplets (n = #words in EmoLex), which by construction leads to training data explosion. Moreover, many of these triplets may trivially satisfy the inequality constraint (i.e., easy triplets) in pre-trained input vector space. In fact, the set of active triplets keeps on changing as the training progress, and just after a few batch updates in stochastic gradient descent, a majority ($> 99\%$) of the triplets become easy triplets resulting in zero loss. The gradients from these inactive triplets start vanishing at this point, leading to considerably slow training.

The stagnant-training problem described above is well studied in the computer vision community, where triplet loss has been successfully applied in metric learning settings for applications such as face verification (Schroff et al., 2015), person re-identification (Hermans et al., 2017), etc. Various approaches for selecting the right set of triplets (referred as triplet mining or sampling) are broadly categorized into offline (Gordo et al., 2016) and online mining (Hermans et al., 2017). In this work, we focus only on online mining as it generally leads to better training convergence than the offline approach. In online mining, we first sample a mini-batch consisting only of raw images (words in our case). The set of active triplets is then generated on-the-fly from the mini-batch. Various policies to sample active triplets from a given batch include BatchHard, BatchHardNegative, and BatchAll. For a given anchor image a from class X, the BatchHard (**BH**) policy selects the hardest positive image p (farthest from a in terms of distance metric) from among the rest of the images of X in the batch. It then selects the hardest negative image n (closest to a in terms of distance metric) from the set of images belonging to classes other than X. The BatchHardNegative (**BHN**) policy relaxes positive image mining by considering all possible in-batch positive images p and then selects the hardest negative n. The BatchAll (**BA**) policy considers all possible in-batch positive images and in-batch negative images for the given anchor a and then selects active triplets from the complete set.

In a nutshell, to mitigate the stagnant-training problem, instead of sampling a mini-batch of triplets from a huge set created offline, we first sample mini-batch of individual words and then generate triplets from the mini-batch on-the-fly using online triplet mining policies.

3.2 Retrofitting model

Our retrofitting model takes pre-trained word embeddings as input and updates them using a non-linear transformation function $\mathbf{T}(\boldsymbol{x_w})$ such that the emotion aspects of words, as induced by the inequality constraints, are respected. The transformation function is learned in a similarity metric learning setting. Figure 1 shows the architecture for learning our retrofitting model.

1. Training data generation: A training instance for our model consists of a word and its emotion category. The data generation component samples words from EmoLex to create a mini-batch b of size n for training. We experiment with two variants: (1) *Uniform* variant samples an equal number of words from all the eight emotion categories; (2) *Weighted* variant, on the other hand, samples words from a category proportional to the number of words annotated with that category in Emolex.

2. Transformation function: We take the d-dimensional pre-trained embeddings of words present in the mini-batch b as input and pass them through the transformation function to compute retrofitted embeddings, i.e., $x_w^t = \mathrm{T}(x_w)$. This function is realized using a multi-layer feed-forward neural network with a corresponding set of network weights W_T.

3. Triplet mining: The retrofitted embeddings

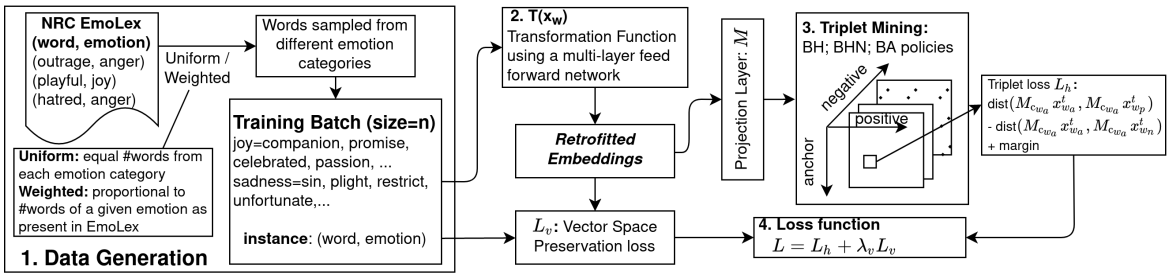


Figure 1: Architecture for our retrofitting model

computed by $\mathrm{T}(x_w)$ are passed to the triplet mining component, which samples the set of active triplets A from batch b according to the selected online triplet mining policy.

4. Loss function: Active triplets obtained from the triplet mining component are used to compute triplet loss from the mini-batch b. It is defined in terms of a margin based hinge loss function,

$$L_h = \sum_{A} \Big(dist(M_{c_{w_a}} x^t_{w_a}, M_{c_{w_a}} x^t_{w_p}) - dist(M_{c_{w_a}} x^t_{w_a}, M_{c_{w_a}} x^t_{w_n}) + margin \Big)_+ \quad (1)$$

Here, $dist$ is a cosine-distance function; $(x)_+ = \max(0, x)$; and $margin$ is a hyper-parameter, set in $[0, 2]$. In Emolex, a word may be tagged with multiple emotion categories e.g. `lonely` is tagged with both *sadness* and *anger*. Thus, while generating *sadness* related constraints from the word `lonely` (e.g. the constraint in $(lonely, sorrow, playful)$), we need a way to extract the sadness aspect. Similarly, when generating *anger* related constraints, we need to consider the anger aspect. To account for this, we first project the retrofitted embeddings to an emotion category-specific vector space using a linear transformation matrix $M_{c_{w_a}} \in \mathbb{R}^{d \times d}; c_{w_a} \in \{1, 2, .., 8\}$ (learned jointly with T). The $dist$ function in Eq. 1 is then applied to the projected retrofitted embeddings.

Vector Space Preservation: Pre-trained embeddings contain useful semantic relations between words as captured by the distributional hypothesis. The transformation function learned by our model should preserve these relations while also respecting the emotion inequality constraints. To address this, we use a regularization term which penalizes vector space transformations that drastically change the topology of input vector space, similar to (Mrkšić et al., 2016; Glavaš and Vulić, 2018). It measures the Euclidean distance between the pre-trained vector x_i and its transformed version $\mathrm{T}(x_i)$ for all words present in batch b,

$$L_v = \sum_{w \in b} \|x_w - \mathrm{T}(x_w)\|_2 \quad (2)$$

The final loss function used by our model is then: $L = L_h + \lambda_v L_v$, where λ_v is a hyper-parameter that determines how strictly the topology of original vector space is preserved. The loss function also includes weight decay for parameters W_T and M.

Since the retrofitting function $\mathrm{T}(x_w)$ is formulated as a representation learning problem (similarity metric learning setting), it can be used to transform pre-trained embeddings of all words present in a given vocabulary post training.

4 Experimental Results

To evaluate our retrofitting method, we experimented with 300-dimensional pre-trained embeddings in GloVe[2] (Pennington et al., 2014) and Word2vec[3] (Mikolov et al., 2013). Due to space constraints, we discuss only GloVe results here (Word2vec results are present in Appendix B). As explained earlier, we used triplet constraints extracted from EmoLex to learn retrofitted embeddings. We refer to our method as **RETriplet** hereafter. Although we report the complete hyper-parameter grid search details in Appendix A, the hyper-parameter λ_v for the vector space preservation loss in Eq. 2 needs special attention. Setting the right value for λ_v is extremely important to learn a meaningful retrofitting model. If we set it very high, RETriplet may not focus on the inequality constraints in triplet loss, thereby learning retrofitted embeddings nearly identical to their pre-trained version. Conversely, a low value of λ_v may produce embeddings that largely satisfy emotion

[2]https://nlp.stanford.edu/data/glove.42B.300d.zip
[3]https://code.google.com/archive/p/word2vec/

constraints but may not preserve the topology of input vector space, possibly leading to degraded performance on downstream tasks. To account for this trade-off, we devise the following scheme and select two configurations: (1) We use adjusted rand index (ARI, a clustering evaluation metric, described in Section 4.1) to measure the quality of retrofitted embeddings and select the configuration that gives the highest value for ARI (referred as **RETripletG**); (2) we compute the average cosine distance between pre-trained and retrofitted embeddings for words in EmoLex and filter configurations having distance < 0.15. We then choose the configuration with the highest ARI from the filtered list (referred as **RETripletGBal**).

Retrofitting approaches proposed in the literature use *attract* and *repel* constraints, extracted from WordNet, Paraphrase database, etc., to update pre-trained embeddings. The attract constraints pull similar (e.g., synonyms, hypernyms, etc.) word pairs close together. While the repel constraints push non-similar (e.g., antonyms) word pairs away from each other. We compare RETriplet with the following,

Counterfit (Mrkšić et al., 2016): It defines the loss function as a weighted sum of terms that brings attract word pairs closer and pushes repel word pairs apart. It also includes a vector space regularization term.

Attract-Repel (AR) (Mrkšić et al., 2017): The counterfit method updates embeddings of attract and repel words without considering their relations to other words. AR addresses this problem by performing context-sensitive vector updates. For each word in attract pairs, it finds the closest (in terms of cosine distance) in-batch word to generate negative examples (conversely farthest for repel words). It then uses these negative examples to form a hinge loss function for context-sensitive updates.

Post-specialization: The methods described above *locally* update vectors of only those words that are present in constraints (i.e., seen words), whereas vectors for all other words remain intact. To address this, post-specialization methods use retrofitted embeddings of seen words to learn a global specialization function which then updates vectors of unseen words. We use the generative adversarial network architecture proposed by Ponti et al. (2018) for post specialization with AR as the local method (referred to as **AR+PS**).

We also learn emotion enriched embeddings using the methods described above by extracting attract and repel constraints from EmoLex. Two words annotated with the same emotion category in EmoLex are added to the attract set, e.g. *(angry, offence)* since both `angry` and `offence` are marked with the *anger* category. In contrast, two words, when annotated with different emotion categories, are added to the repel set, e.g. *(fun, miserable)* since `fun` is marked with *joy* and `miserable` with *sadness*. The generated attract and repel sets are then used to learn retrofitted embeddings. They are referred by appending **+EL** to the retrofitting method, e.g., AR+EL for embeddings retrofitted using AR with EmoLex constraints.

We also compare our method with the following emotion enriched embeddings: (1) **EWE** (Agrawal et al., 2018): It first creates noisy emotion labelled data using distant supervision and then applies recurrent neural network to learn emotion embeddings; (2) **Aff2vec** (Khosla et al., 2018): It appends valence (V), arousal (A) and dominance (D) values of words as present in Warriner's VAD lexicon (Warriner et al., 2013) to the counterfitted GloVe embeddings, resulting in 303-dimensional affective embeddings; (3) **EEArmin** (Seyeditabari et al., 2019): It applies counterfit method directly on the *(word, emotion)* pairs in EmoLex; (4) **SentiEmbs** (Yu et al., 2017): embeddings refined for sentiment using valence values present in Warriner's lexicon.

4.1 Clustering Experiments

Since our main objective is to investigate word embeddings for their emotion content, it is natural to ask, do words that evoke the same emotion have similar embeddings? In other words, are words with similar emotion content clustered together in the vector space? To study this, we extract all words present in EmoLex and their emotion labels to create a dataset for clustering. The embeddings of words are then used as features to perform K-means clustering with the number of means (k) set to 8. Since the true labels are available, we apply various external cluster validity indices to measure clustering quality. In particular, we use adjusted rand index (ARI), Fowlkes Mallows score (FMS), adjusted mutual information score (AdjustedMIS), V-measure, and entropy (refer Scikit-learn user guide). In addition to good cluster quality, retrofitted embeddings shall also preserve the topology of pre-trained vector space. To quantify this, we compute the average cosine distance between

Embeddings	ARI↑	FMS↑	AdjustedMIS↑	V-measure↑	Entropy↓	VDist↓
GloVe	0.0456	0.1542	0.0863	0.0888	1.8092	0
counterfit	0.0897	0.1969	0.1634	0.1657	1.6404	0.1740
AR	0.0749	0.1802	0.1479	0.1502	1.6717	0.0977
AR+PS	0.0853	0.1911	0.1607	0.1630	1.6444	0.1257
counterfit+EL	0.1530	0.2532	0.1953	0.1976	1.5680	0.0308
AR+EL	0.2071	0.3126	0.3966	0.3984	1.1594	0.3068
AR+PS+EL	0.1567	0.2689	0.2579	0.2600	1.4495	0.2029
EWE	0.0556	0.1630	0.1083	0.1108	1.7605	0.0085
Aff2vec	0.0824	0.1877	0.1574	0.1598	1.6517	NA
EEArmin	0.3764	0.4566	0.5501	0.5514	0.7856	1.0152
SentiEmbs	0.0009	0.2974	0.0135	0.0176	1.9817	0.4329
RETripletGBal	0.0951	0.2000	0.1639	0.1662	1.6373	0.0946
RETripletG	0.1616	0.2602	0.3031	0.3050	1.3271	0.4445

Table 3: External cluster validity indices (with k=8) for pre-trained GloVe and its retrofitted versions (↓: lower values are better; ↑: higher values are better) - Overall, RETripletGBal and counterfit+EL provide substantially good clustering while preserving the topology of pre-trained vector space. The embeddings in red are not desirable as they drastically change the pre-trained vector space (high VDist) and may not perform well on affective end-tasks.

pre-trained and retrofitted embeddings for words in EmoLex. It is referred to as VDist (lower values are better).

As shown in Table 3, the scores for the pre-trained GloVe baseline are lowest across all clustering indices. This indicates that there is a scope of improvement for injecting emotion content into pre-trained embeddings. The embeddings from pair-wise retrofitting methods with synonymy and antonymy constraints (i.e., counterfit, AR, AR+PS) reasonably improve clustering quality while maintaining a fairly good VDist (< 0.18). When used with the attract and repel constraints from EmoLex (+EL setting), both AR+EL and AR+PS+EL embeddings achieved extremely good clustering. However, their VDist is very high, pointing to the fact that they did not maintain semantic relations present in GloVe. The EWE embeddings perform poorly on clustering indices as they are identical to their pre-trained version (VDist=0.0085). The SentiEmbs embeddings do not provide good clustering since they are optimized only for coarse-grained sentiments. On the other hand, the EEArmin embeddings have completely overfitted for clustering, with extremely poor VDist. Though Aff2vec embeddings achieve reasonably good clustering, we could not compute VDist due to the three extra dimensions appended for VAD. The embeddings in counterfit+EL and RETripletGBal provide the right balance overall with substantially good cluster quality along with low values for VDist. The counterfit+EL embeddings, however, do not perform well on downstream tasks, as reported later in Table 5.

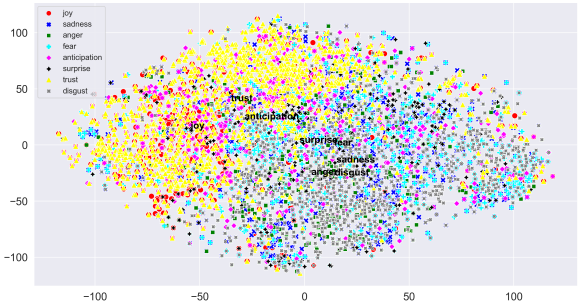
Figure 2 shows t-SNE plots for EmoLex words using pre-trained GloVe, RETripletGBal, and RETripletG, marking the median point for each emotion category. These points are very close to each other for pre-trained GloVe as it only uses the distributional hypothesis to learn embeddings, not considering the emotion content of words. On the other hand, RETripletG (selected based only on clustering quality) provides extremely good separation but at the expense of losing semantic relations present in GloVe. RETripletGBal embeddings not only provide reasonably good separation but also preserve the topology of the input vector space. The cosine similarity values computed using RETripletGBal are well-calibrated for emotion content, as evident for the exemplar pairs in Table 1.

4.2 Evaluation on Downstream tasks

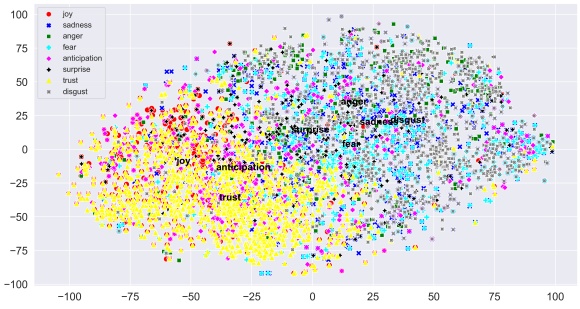
We evaluate emotion enriched embeddings on two affective end-tasks: (1) Sentiment analysis using binary (SST2) and graded (SST5) Stanford sentiment treebank, and tweet messages from SemEval 2017 (task 4A); (2) Sarcasm detection using sit-com utterances in Mustard++. Table 4 reports statistics for these datasets. Similar to EWE (Agrawal et al., 2018), we use a probing framework (Conneau et al., 2018; Eichler et al., 2019) to evaluate embeddings for their performance on the downstream tasks. In particular, we apply two classification models: sup-

Task	Dataset	#class	size	#token	Type	Vocab	Source
Sentiment analysis	SST2	2	9613	162783	sentence	17630_1	(Socher et al., 2013)
	SST5	5	11855	199120	sentence	19631_1	(Socher et al., 2013)
	SemEval	3	61854	1174626	tweet	23005_2	(Rosenthal et al., 2017)
Sarcasm detection	Mustard++	2	1202	14219	utterance	2632_1	(Ray et al., 2022)

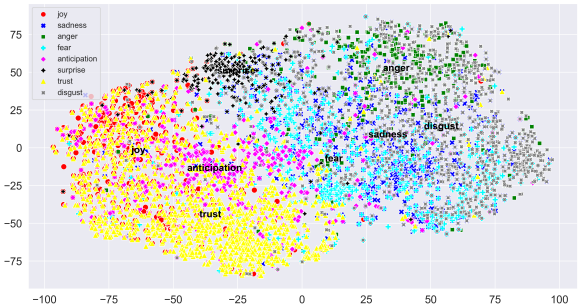
Table 4: Dataset statistics for downstream tasks (subscript in **Vocab** indicate minimum frequency threshold)



(a) Pre-trained GloVe



(b) RETripletGBal embeddings



(c) RETripletG embeddings

Figure 2: t-SNE plots for emotion bearing words

port vector machine (SVM), and attention network (AttnNet). The embeddings of tokens present in a given sentence/utterance/tweet are averaged to compute input features for SVM. Whereas the token embeddings as a sequence are passed as input to an attention layer followed by softmax to compute cross-entropy loss for AttnNet.

Table 5 reports the micro F1-scores for SVM and AttnNet. The pre-trained GloVe embeddings seems to be a hard baseline to beat on the sentiment analysis task. While the pair-wise retrofitting methods (counterfit, AR, AR+PS) have been shown to improve tasks such as dialogue state tracking, text simplification, etc., they have not been extensively tested for sentiment analysis. Surprisingly, embeddings from these methods could not beat the baseline even though they are updated to respect the synonymy and antonymy constraints. When retrofitted using attract and repel constraints from EmoLex, their (+EL variants) performance degraded even further. This degradation is partly attributable to the in-batch sampling of negative examples. Unlike synonym constraints where distinct word pairs are not interrelated, word pairs in Emolex attract constraints are interrelated due to their emotion labels. For example, consider in-batch attract pairs such as (enjoy, fun), (happy, thankful), and (loving, delightful), having the common emotion label *joy*. While generating negative examples for `enjoy`, pairs such as (enjoy, happy) and (enjoy, delightful) may be inappropriately considered as candidates, leading to spurious training data. The EWE embeddings trained using distant supervision are nearly identical to their pre-trained version (VDist=0.009), leading to no improvement in end-task. Though retrofitted for emotions, both Aff2vec and EEArmin embeddings could not beat the pre-trained baseline, possibly due to drastic changes to the topology of input vector space (high VDist). SentiEmbs, though optimized for sentiments, unexpectedly could not perform well on any datasets. RETripletGBal embeddings learned using our method achieved the highest F1-score for both SVM and AttnNet on the sentiment analysis task. For the sarcasm detection task (Mus++ in Table 5), the embeddings learned using our method performed better than their pre-trained counterparts (about 1.5% improvement in F1-score) and achieved the highest F1-score with SVM.

4.2.1 Limited data experiments

To evaluate embeddings in a low resource setting, we sample datasets of various sizes, such as 10%,

Embeddings	SVM				AttnNet			
	SST2	SST5†	SemEval†	Mus++	SST2	SST5†	SemEval†	Mus++
GloVe	**0.8034**	0.4122	0.6131	0.5333	0.7705	0.4072	0.6375	0.5125
counterfit	0.7996	**0.4181**	**0.6236**	0.5105	0.7419	0.4005	0.6274	0.5375
AR	0.8029	0.3846	0.5782	0.5063	0.721	0.3937	0.635	0.4833
AR+PS	0.8018	0.4041	0.6031	0.4979	**0.7853**	**0.4204**	0.6306	0.5417
counterfit+EL	0.7985	0.4032	0.6112	0.5021	0.7326	0.3842	**0.6391**	0.5125
AR+EL	0.7902	0.405	0.607	0.4979	0.7348	0.3923	0.6365	0.5042
AR+PS+EL	0.7628	0.3842	0.5711	0.4979	0.7721	0.3624	0.6171	0.4875
EWE	0.7974	0.402	0.6049	0.5523	0.7738	0.4068	0.6237	0.5292
Aff2vec	0.7831	0.3893	0.5725	0.523	0.7381	0.4023	0.6241	**0.5457**
EEArmin	0.7644	0.3805	0.5604	0.5397	0.76	0.3928	0.6226	**0.5458**
SentiEmbs	0.7397	0.3633	0.5511	0.5356	0.6985	0.3543	0.5409	0.5125
RETripletGBal	**0.816**	**0.4339**	**0.6305**	**0.5542**	**0.7946**	**0.4267**	**0.6405**	0.5292
RETripletG	0.7705	0.3946	0.6101	**0.5667**	0.7787	0.3973	0.6288	0.4792

Table 5: Micro F1-scores for SVM and AttnNet with various embeddings as input (**Bold+Underline**: highest; **Bold**: next highest); †: Wilcoxon's signed rank test with $\alpha = 0.5$ indicates RETripletGBal is better than GloVe

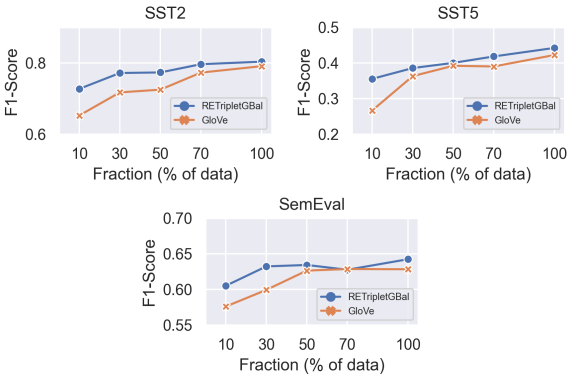


Figure 3: Data size vs. micro F1-score for Pre-trained GloVe and RETripletGBal in limited data setting

30%, etc., from the original sentiment analysis datasets. We then compare pre-trained GloVe with RETripletGBal in terms of micro F1-score across the data sizes. As we can see in Figure 3, RETripletGBal performs significantly better than pre-trained GloVe in a low data regime ($< 60\%$ data). The difference in performance reduces nearly after 80% data size. This points to the fact that the external knowledge from EmoLex as captured by our retrofitting method helps improve the end-task, especially in the limited data scenario.

5 Related Work

Large language models with contextualized word embeddings (e.g., BERT and its variants) have lately received a lot of attention in the NLP community. Nonetheless, their static counterparts are still actively explored, e.g., combining static and contextualized embeddings to improve end-tasks (Alharbi and Lee, 2021; Alghanmi et al., 2020), inducing knowledge bases (Dufter et al., 2021), bilingual lexicon induction (Zhang et al., 2021), etc. In this work, we focus on static word embeddings that are learned primarily using the distributional hypothesis. A major limitation with these embeddings is that they do not differentiate semantic similarity from other types of relatedness (Hill et al., 2015). This problem is addressed by borrowing semantic relations from resources such as WordNet, Paraphrase Database, etc., in the form of constraints. These constraints are then used by joint specialization (Yu and Dredze, 2014; Liu et al., 2015) or retrofitting models (Faruqui et al., 2015; Mrkšić et al., 2016; Shah et al., 2020) to improve word embeddings. These models, however, focus mainly on synonymy, antonymy, and hypernymy relations. Recently, a few attempts, such as Aff2vec from Khosla et al. (2018) and emotion embeddings from Seyeditabari et al. (2019), incorporate knowledge present in affective lexicons to learn emotion enriched embeddings.

The contrastive learning approach similar to our work has recently been applied to learn transformer based sentence embeddings in SBERT (Reimers and Gurevych, 2019) and zero-shot image classification in CLIP (Radford et al., 2021). However, these methods are not specialized to learning emotion enriched embeddings.

There is a large body of work that focuses on

learning task-specific *affective* embeddings. These methods first use distant supervision to create a noisy labelled dataset and then use it to update word embeddings or learn them from scratch. For instance, sentiment-aware embeddings using tweet data (Tang et al., 2014, 2016); affective embeddings using tweet emojis (Felbo et al., 2017); emotion enriched embeddings using product reviews data (Agrawal et al., 2018). Since embeddings learned from these methods are tied to the dataset used for distant supervision, they may not work well for other related affective end-tasks. Moreover, they are not very accurate due to noisy labelling.

The emotion-enriched embeddings learned by our method are not only accurate compared to the methods described above, as evident from the clustering experiments, they also work well on the related affective end-tasks.

6 Summary and Future work

We present a novel retrofitting method to learn emotion enriched embeddings using triplet constraints from EmoLex. These constraints are used as training data to learn a retrofitting function in a similarity metric learning setting. The embeddings learned by our method perform better than their pre-trained counterparts and other benchmarks in both intrinsic clustering evaluation and the extrinsic downstream tasks in sentiment analysis and sarcasm detection. As future work, we plan to extend our triplet constraint-based approach to other resources such as VAD lexicon (Warriner et al., 2013; Mohammad, 2018a). We also plan to develop a similar approach for contextualized word embeddings.

References

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2018. Learning emotion-enriched word representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 950–961, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Israa Alghanmi, Luis Espinosa Anke, and Steven Schockaert. 2020. Combining BERT with static word embeddings for categorizing social media. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 28–33, Online. Association for Computational Linguistics.

Abdullah I. Alharbi and Mark Lee. 2021. Multi-task learning using a combination of contextualised and static word embeddings for Arabic sarcasm detection and sentiment analysis. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 318–322, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Alexis Conneau, German 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, Melbourne, Australia. Association for Computational Linguistics.

Philipp Dufter, Nora Kassner, and Hinrich Schütze. 2021. Static embeddings as efficient knowledge bases? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2353–2363, Online. Association for Computational Linguistics.

Max Eichler, Gözde Gül Şahin, and Iryna Gurevych. 2019. LINSPECTOR WEB: A multilingual probing suite for word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 127–132, Hong Kong, China. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6:169–200.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2018. Explicit retrofitting of distributional word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–45, Melbourne, Australia. Association for Computational Linguistics.

Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. 2016. Deep image retrieval: Learning

global representations for image search. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pages 241–257. Springer.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, abs/1703.07737.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Sopan Khosla, Niyati Chhaya, and Kushal Chawla. 2018. Aff2Vec: Affect–enriched distributional word representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2204–2218, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501–1511, Beijing, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Saif M. Mohammad. 2018a. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Saif M. Mohammad. 2018b. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

W. Gerrod Parrott. 2001. Emotions in social psychology : essential readings.

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.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion*.

Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Adversarial propagation and zero-shot cross-lingual transfer of word vector specialization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 282–293, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Anupama Ray, Apoorva Nunna, and Pushpak Bhattacharyya. 2022. A multimodal corpus for emotion recognition in sarcasm. In *Proceedings of the 13th Edition of the Language Resources and Evaluation Conference (LREC-2022)*, Marseille, France.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society.

Scikit-learn user guide. Clustering performance evaluation. Online; accessed 01-February-2022.

Armin Seyeditabari, Narges Tabari, Shafie Gholizadeh, and Wlodek Zadrozny. 2019. Emotional embeddings: Refining word embeddings to capture emotional content of words. *ArXiv*, abs/1906.00112.

Sapan Shah, Sreedhar Reddy, and Pushpak Bhattacharyya. 2020. A retrofitting model for incorporating semantic relations into word embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1292–1298, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45:1191–1207.

Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539, Copenhagen, Denmark. Association for Computational Linguistics.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, Baltimore, Maryland. Association for Computational Linguistics.

Jinpeng Zhang, Baijun Ji, Nini Xiao, Xiangyu Duan, Min Zhang, Yangbin Shi, and Weihua Luo. 2021. Combining static word embeddings and contextual representations for bilingual lexicon induction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2943–2955, Online. Association for Computational Linguistics.

A Training details

This section details the hyper-parameters in our retrofitting method and the best combinations selected thereof. The transformation function $\mathrm{T}(x_w)$ in RETriplet is implemented using a multi-layer feed-forward neural network. The hyper-parameters are:- number of hidden layers: $\{2, 3, 4\}$, size of hidden layer: $\{300, 400, 500\}$, activations: $\{tanh, ReLU\}$, dropout: 0.2 and L2 regularization: 0.0005. We use Adam (Kingma and Ba, 2014) optimization algorithm with, learning rate: $\{0.001, 0.0005\}$ and batch size: $\{64, 128, 256\}$. We experiment with two data generation schemes described earlier: Uniform and Weighted. For the hinge loss function in Eq. 1, we use cosine as the distance metric with two margin values, i.e. $\{0.2, 0.6\}$. The margin is set to the same value for all emotion categories. For triplet mining, during initial experiments, we observed that both the BatchHard and BatchAll mining policies performed equally well, with BatchAll having better convergence during initial epochs. Hence, we primarily experimented with the BatchAll mining policy. The hyper-parameter for vector space preservation loss λ_v is varied as $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5\}$. We set aside 10% words in EmoLex for validation and use early stopping with patience 10. For experimentation, we used CPU machines with 64GB RAM and 20 core CPUs. Each configuration on an average took 80 minutes to run.

For both GloVe and Word2vec, we select two configurations to generate retrofitted embeddings. One configuration is selected only on the basis of clustering quality metric (ARI). Whereas, the second configuration takes vector space preservation into account in addition to the clustering quality. Table 6 reports these configurations.

	GloVe		Word2vec	
hyperparameter	RETripletGBal	RETripletG	RETripletWBal	RETripletW
#layers	2	3	2	3
#hidden units	300	300	300	200
activation	ReLU	ReLU	ReLU	ReLU
dropout	0.2	0.2	0.2	0.2
L2-regularization	0.0005	0.0005	0.0005	0.0005
batch-size	128	128	128	128
learning rate	0.0005	0.0005	0.0005	0.0005
data generation	Weighted	Uniform	Weighted	Weighted
triple mining	BatchAll	BatchAll	BatchAll	BatchAll
λ_v	0.5	0.1	0.7	0.4

Table 6: Selected hyper-parameter configurations for retrofitted embeddings (1) GloVe:- RETripletG has the best ARI; RETripletGBal has the best ARI with VDist < 0.15 (2) Word2vec:- RETripletW has the best ARI; RETripletWBal has the best ARI with VDist < 0.15

B Experimental results for Word2vec

Table 7 reports clustering experiments for Word2vec pre-trained baseline and their retrofitted versions. Table 8 reports results for sentiment analysis and sarcasm detection tasks for SVM and Attention network with Word2vec as the base embeddings.

Embeddings	ARI↑	FMS↑	AdjustedMIS↑	V-measure↑	Entropy↓	VDist↓
Word2vec	0.0553	0.1641	0.1019	0.1044	1.7753	0.0
counterfit	0.0762	0.1814	0.1495	0.1518	1.6682	0.1803
AR	0.0794	0.186	0.1538	0.1561	1.6601	0.2556
AR+PS	0.0913	0.2051	0.159	0.1613	1.6559	0.1326
counterfit+EL	0.1628	0.2618	0.2029	0.2051	1.5507	0.0232
AR+EL	0.2008	0.3066	0.39	0.3917	1.1729	0.37
AR+PS+EL	0.1228	0.2527	0.3349	0.3369	1.3078	0.1931
EWE	-	-	-	-	-	-
Aff2vec	0.0914	0.1978	0.1567	0.1591	1.6548	NA
EEArmin	0.3655	0.4468	0.5495	0.5507	0.7964	0.9986
SentiEmbs	0.0007	0.3000	0.0085	0.0126	1.9896	0.4382
RETripletWBal	0.1493	0.2545	0.2086	0.2109	1.5448	0.1371
RETripletW	0.1768	0.2764	0.2784	0.2804	1.3885	0.3633

Table 7: External cluster validity indices for pre-trained Word2vec and its retrofitted versions (↓: lower values are better; ↑: higher values are better) - Overall, RETripletWBal and counterfit+EL provide substantially good clustering while preserving the topology of pre-trained vector space. The embeddings in red are not desirable as they drastically change the pre-trained vector space (high VDist) and may not perform well on affective end-tasks. *EWE embeddings not available for Word2vec.

Embeddings	SVM				AttnNet			
	SST2	SST5	SemEval	Mus++	SST2	SST5	SemEval	Mus++
Word2vec	0.8144	0.4262	0.6209	0.5481	**0.7985**	0.4136	**0.6342**	0.525
counterfit	0.8127	**0.4281**	**0.6298**	0.5063	0.7408	0.4023	0.6277	0.5125
AR	0.8018	0.409	0.5995	0.5105	0.7842	0.3787	0.6307	0.5083
AR+PS	0.8023	0.4176	0.5995	0.5397	0.7924	**0.4249**	0.6281	0.5667
counterfit+EL	0.816	0.4262	0.6245	0.5272	0.7567	0.3778	**0.6385**	0.5458
AR+EL	0.8127	0.4208	0.6243	0.5314	0.7776	0.3697	0.6306	0.5375
AR+EL+PS	0.7968	0.3959	0.6012	0.523	0.7452	0.4072	0.6186	0.5125
EWE	-	-	-	-	-	-	-	-
Aff2vec	**0.8166**	0.407	0.6119	0.5146	0.7414	0.3692	0.6299	**0.575**
EEArmin	0.771	0.3887	0.5964	**0.5523**	0.7479	0.3566	0.6197	0.5542
SentiEmbs	0.7567	0.3656	0.5716	**0.5649**	0.7205	0.3661	0.5462	0.4958
RETripletWBal	**0.8221**	**0.438**	**0.6323**	**0.5523**	**0.8051**	**0.419**	0.6323	**0.5792**
RETripletW	0.7979	0.4145	0.6153	0.5105	0.7979	0.3982	0.6307	0.5

Table 8: Micro F1-scores for SVM and AttnNet with various embeddings as input: Experiments with Word2vec as baseline (**Bold+Underline**: highest; **Bold**: next highest) *EWE embeddings not available for Word2vec