
The Interdisciplinary Center, Herzliya
Efi Arazi School of Computer Science
M.Sc. Program - Research Track

# Training Word Embeddings Jointly with Affect Information

by
**Yoav Goldman**

M.Sc. dissertation, submitted in partial fulfillment of the requirements
for the M.Sc. degree, research track, School of Computer Science
The Interdisciplinary Center, Herzliya
April 11, 2022

This work was carried out under the supervision of Dr. Kfir Bar from the Efi Arazi School of Computer Science, The Interdiciplinary Center, Herzliya.

# Abstract

We use affect lexicons to modify word vectors during training in order to capture emotional information. Our goal is to drag vectors of similar emotional connotation closer together, while dragging away vectors with an opposite one. We examine different approaches for incorporating emotional information into the vector-training process, including using emotions of context words, and compare them with vectors that were edited post training. In addition to some qualitative analysis, we use the modified vectors in a few downstream tasks and evaluate their performance. The results are encouraging; the vectors that were jointly trained with affect information show better performance across most of our experiments.

# Acknowledgements

First of all, I would like to thank Dr. Kfir Bar for all of his help, mentoring, teaching and for motivating me. I couldn't think of a better mentor and it was really a pleasure working with you.

# Contents

# 1 Introduction

Affect, in psychology, is an emotion that has a physical expression which can be observed by people. Emotions can be expressed in multiple ways, including facial expressions, hand gestures, body movements, prosody, and the content of speech. Processing the content of speech for detecting affective expressions has recently become an active research area of natural language processing (NLP). Emotions, in NLP, are measured either by assigning weights to a predefined close set of emotions (e.g., joy, sadness, anger) or by a system consisted of three principal dimensions: *valence*, *arousal* and *dominance*, typically referred to as VAD. Valence is an evaluation of a text or a word on a positive-to-negative spectrum. Arousal is the expression of calmness or excitement, and dominance is the degree of control that the author/reader of a text, has over their expressed affect. The three dimensions are typically measured on a scale of 1 and 9, accordingly to the standard forms that are used by people who are requested to assign VAD values to an object, a word, or a text. Flat affect is defined as a lack of signs of affective expressions. In our work, we will focus only on the valence-arousal dimensions. Figure 1 shows the two axes, populated with some known labeled emotions. For example, *satisfaction* is a positive emotion that has a neutral arousal level. On the other hand, *enthusiasm* has higher arousal levels while still being categorized as a positive emotion.

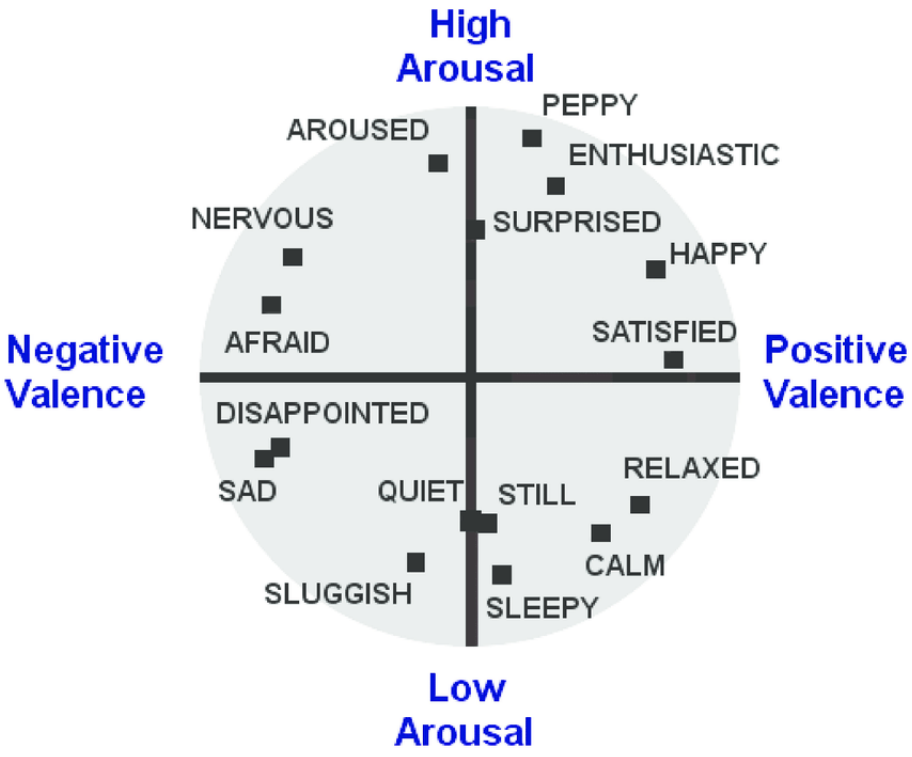


Figure 1: An example of the valence-arousal system, populated with examples of labeled emotions. The image was borrowed from Abdur-Rahim et al. (2016).

In modern applications of NLP, words are typically represented by mathematical vectors, or *embeddings*, that capture their meaning as can be learned from the context in which the words are mentioned in, within a large corpus. This concept is also known as *distributional semantics*. The challenge with this approach is that words with opposite valence may appear in similar contexts, resulting in similar vectors. For example, in all common word embeddings collections, the word *happy* is found to be similar to both *sad* and *joy* almost on the same level. It happens because *happy* and *sad* are generic emotional adjectives, which may appear in similar contexts. For example, both expressions, "a happy song" and "a sad song" are likely to be mentioned in the same level of probability. On the other hand, the noun *joy* is similar to *happy* since they are both likely to occur in a positive sentence about happiness. This challenge may affect the performance of NLP sentiment-analysis models that use such embeddings as word representations.

A similar known challenge with word embeddings is related to gender bias. It seems like some word vectors are more related to one gender than to the other, and that poses an ethical question about models' ability to focus on content curation. Therefore, there is a continued attempt to edit word vectors for different purposes. For example, Ravfogel et al. (2020) propose a generic approach for removing bias, not necessarily gender related, from word embeddings; Gonen and Goldberg (2019) focus on removing gender bias from word vectors; and more related to this work, Khosla et al. (2018) modify an off-the-shelf word embeddings collection using some affect information that they collected about the words. Concluded from their work, word embeddings that were adjusted according to some

affect information, may improve state-of-the-art results of emotion-related downstream tasks, such as sentiment analysis. Another example of using such vectors is NLP systems that are designed to discover mental-health disorders in speech and text; for example Bar et al. (2019) use word embeddings intensively for capturing thought disorders of schizophrenics in transcribed speech. In this work, the authors measure the similarity of vectors of consecutive words in transcribed speech, in order to detect the situation of going out of topic. They also measure the similarity of vectors of adjectives that are being used by people with schizophrenia and control participants, when used to describe the same nouns. Intensifying the emotional signal of words in their corresponding vectors is likely to improve Bar et al. (2019)'s results, since most of the adjectives they describe in the paper are on either of the two extremes of the valence axis.

The goal of this study is to evaluate different approaches for editing word vectors, by incorporating emotional information from the surrounding context into the making procedure of word embeddings. To do that, we slightly modify the architecture of word2vec Mikolov et al. (2013a), a known word-embeddings training technique, to inject some information about the emotions of words, which was obtain from a large lexicon of words enriched with affect information. In the next step, we use our modified training technique to create emotionally aware word vectors, and use them to improve a data-driven model for sentiment analysis. We experiment with different affect lexicons, on two languages (English and Spanish), and on different hyperparameter settings.

We continue to describe our method in more details in Section 3.

# 2 Related Work

There is a growing body of research that examines ways of editing word vectors for different purposes. Specifically, there is a number of works that aim to incorporate affect information into word embeddings, and use them to improve state-of-the-art results of common NLP downstream tasks. Most of those work propose to edit an already trained vectors, where affect lexicons are widely used as an external resource for adding the affect information. Khosla et al. (2018) use the Warriner norm lexicon (Warriner et al., 2013), which assigns VAD scores to many English words, in order to edit GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013a) vectors. They append every individual word's VAD values to its corresponding vector, as additional dimensions, and then reduce their dimension back to the original one. The modified vectors are then being retrofitted (Faruqui et al., 2015) with WordNet (Miller, 1998) to drag similar words closer together. They report on state-of-the-art results in intrinsic word-similarity tasks, and outstanding performance in a few NLP downstream tasks, which employ models that use those modified embeddings as input. The caveat with this approach is that words that do not appear in the lexicon, such as proper nouns, are not edited at all. To handle that, in our study we examine a way to edit such words using the affect information of words that appear next to them in a large corpus. The inspiration for this idea came from other related works, such as (Recchia and Louwerse, 2015; Palogiannidi et al., 2015; Vankrunkelsven et al., 2015). In all of those works, the authors had developed algorithms for generat-

ing affective norms for out-of-vocabulary words, based on collocations. Similarly, Speer et al. (2017) use retrofitting with ConceptNet to improve semantic representation of pre-trained word vectors, and Mrkšić et al. (2016) use counterfitting to modify vectors with information about antonyms and synonyms. Yu et al. (2017) is using retrofitting on pre-trained vectors by making words with similar valence be close to each other.

Conceptually, our study falls under a different type of works in which the training process of word vectors is modified to include some external information, rather than editing pre-trained vectors as some sort of a post-processing step. One such example is RC-NET Xu et al. (2014), a system for adding some knowledge-base information to word vectors during training, by representing the information as a regularization function.

# 3 Our Method

In order to create our embeddings, we slightly modify the traditional word2vec skip-gram architecture and enrich it with some VAD information. The word2Vec neural architecture is essentially based on a two feed-forward layers: (1) a narrow hidden layer, which embeds the semantic context of the target word, and (2) an output layer which estimates the chances for each word from the vocabulary to a collocate with the target word given as an input. Figure 2 provides a visual representation of the classic neural word2vec architecture.

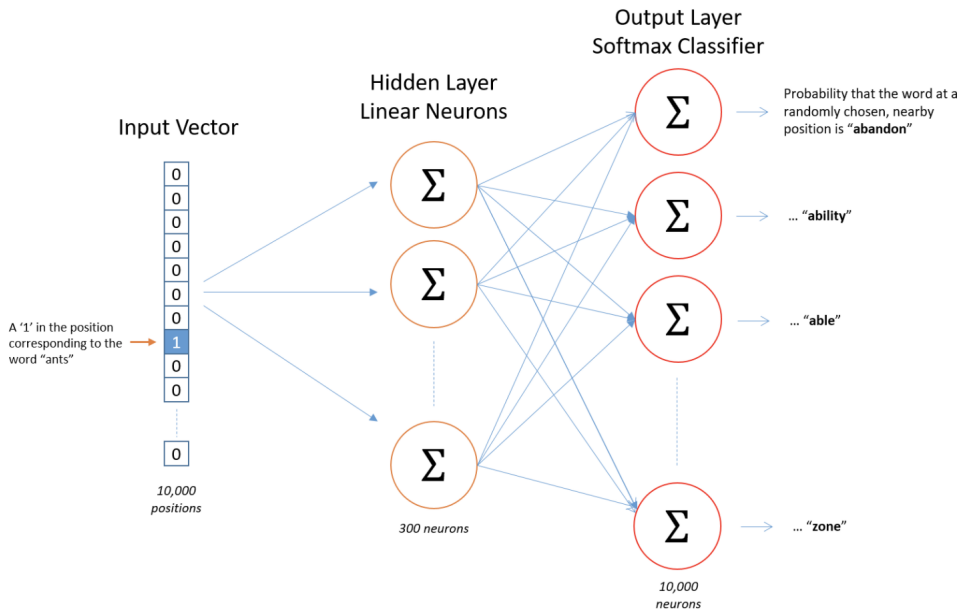


Figure 2: The word2Vec neural skip-gram architecture.

To train a neural word2vec architecture, we iterate over words taken from a

large collection text documents, and encode each word as a one-hot $v$-dimensional vector, with $v$ being the size of the entire vocabulary. That one-hot vector is passed as an input through the network, which has a single hidden layer of a configurable size (typically set to 300) connected with an output layer of the vocabulary size. Essentially, the network is trained to predict a distribution over the vocabulary, reflecting the chance for every word to appear in the surrounding context[1] of the word in focus. Negative sampling (NS) was introduced by Mikolov et al. (2013b) to handle the relatively large output-layer size being used along with the simple softmax-style classification. With NS, the classic cross-entropy classification loss function is replaced by an economical version of it, which essentially compares the true context word and only $K$ negative words, sampled from the vocabulary. The value of $K$ is typically small, around 10, which is sufficient for learning based on the assumption that in every natural language, the chances of a randomly chosen word being a collocation of the target word, are very low.

Once training is over, the network's latent space, that is the vectors that get calculated by the first layer, is then used as embeddings.

We modified the traditional word2Vec skip-gram architecture such that in addition to predicting the index of the context word, it is also predicting the *emotional weight* of the target word that has been passed in as an input, as well as the *emotional weight* of the context word that it tries to predict. The emotional weight is defined as a pair of numeric values, which we retrieve from an affect lexicon

[1] The surrounding context is typically defined by a window of $\pm k$ words; in all our experiments, we use $k = 5$.

that contains many words, as we further explain in what follows. Figure 3 is a visualization of our modified word2vec architecture.

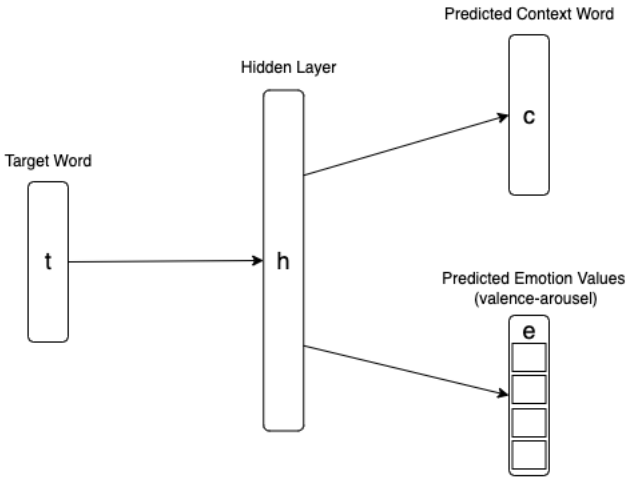


Figure 3: The *word2vec* skip-gram architecture, slightly modified to enrich word embeddings with affect information. $t$ is the target word, for which we predict $c$, the context word as well as the emotional weight $e$ of the target and context word, respectively. Therefore, the emotional weight is composed of two pairs of numbers, representing the valence and arousal values of the two words.

As can be seen from the figure, in our modified architecture, the hidden layer is connected individually to the original output layer, as well as to a new fully-connected layer that predicts the emotional weight of both, target and context words. The emotional weight of a word is represented by its valence and arousal values retrieved from an affect lexicon; therefore, this output layer is of size 4.[2] The valence and arousal values are both numbers in the range $[1,9]$. Our final

[2]We left the *dominance* value out, as among the three emotional values it has the worse inter-annotation agreement during manual annotation.

word vectors are composed only from the weights of the hidden layer.

**Loss functions.** We use two individual loss functions: The original cross-entropy word2vec loss, as well as a regression-style minimum square error (MSE) to predict the VAD values directly.

**Affect lexicons.** We experiment with two affect lexicons: 1) The Affective Norms for English Words (ANEW) Bradley and Lang (1999), covering about $3,200$ words, and 2) The Warriner norm lexicon Warriner et al. (2013), an extension to ANEW, covering about 14,000 words.[3] Words in both lexicons are provided with values of valence, arousal and dominance, ranging on a scale of $[1,9]$ (1 - low, 5 - neutral, 9 - high).

**Corpus.** To train the models, we use *Text8*, [4] a collection of 243,426 English Wikipedia pages of various domains.

**Training.** We train the network using the two loss functions sequentially. First we run back-propagation with the original word2vec loss, and then we use the MSE loss to optimize the network on valence-arousal (VA) value prediction. We take two different approaches for the second pass, and study the results. In one approach (refer to as *target affect*), we only use the VA weights of the target word.

[3] We still use ANEW since the values assigned to the same word in both lexicons are not always identical.

[4] http://mattmahoney.net/dc/textdata.html

In the second approach, we use the VA weights of the context word only if the target word is either missing from the lexicon or it does not express intense affect. In the latter case, we use both the target and the context VA weights. We refer to this approach as *target-context affect*. Intense affect is considered to be a pair of $[V,A]$ values outside of the square defined by the points $[4,4],[4,6],[6,6],[6,4]$. The second approach allows us to expand our coverage for words that are either missing from the lexicon (e.g., names of people, organizations) or words that have a flat affect (e.g., *a car*, *a table*, names of people that do appear in the lexicon). We believe that in addition to semantics, the context words may play an important role in capturing the emotional weight of the target word.[5]

Our modified word2vec skip-gram network is implemented in PyTorch.[6] We use a pair of embedding layers, one for the target word and one for the context word, and a single linear layer for predicting the four VA values, which we connect with the target-word embeddings layer. We use Adam Kingma and Ba (2015) with a learning rate of 0.003 for optimizing the original word2vec loss, and a plain stochastic gradient descent (SGD) with the same learning rate for the MSE loss. We use SGD to ensure that the learning rate remains constant across all epochs, so that the impact of the VA weights on the network's weights remain similar. We train every model with five epochs.

We've build the following four embeddings models (their aliases are provided in parenthesis):

[5]Due to the small size of ANEW, we only use Warriner for producing embeddings with the target-context affect approach.

[6]https://pytorch.org/

1) Target affect with ANEW (T-AN);

2) Target affect with Warriner (T-Wa);

3) Target-context affect with Warriner (TC-AN);

4) A skip-gram model, which we use as a control and does not use any of the VA weights during training (Baseline).

# 4 Results

## 4.1 Embeddings Evaluation

To assess the quality of the vectors, we look at the following word pairs:

*happy*;*sad*, *good*;*bad*, *surprised*;*relaxed*, *happy*;*joy emotional*;*society*

We measure their (cosine) distance under the four model spaces. Table 1 arranges their distances by model, and Figure 4 visualizes them using their 2 principle components calculated by running PCA.

To evaluate the general concept of editing vectors during training versus as a post-processing step, we repeat every experiment once again with the baseline model, which we run for the same number of epochs, optimizing the model only with the MSE loss for the Warriner's VA values of the target word. The results are in given in Table 1 in parentheses next to the T-Wa numbers, since both models are equivalent except the post-processing optimization. The distances that we get using the post-processed model are not very different than the baseline's, suggesting that the vectors that were jointly trained with the affect information are more sensitive to what we need.

| **Pair** | **Base** | **T-AN** | **T-Wa** | **TC-Wa** |
|---|---|---|---|---|
| *happy*;*sad* | 0.88 | 1.27 | 1.42 (0.87) | 0.85 |
| *happy*;*joy* | 0.91 | 0.43 | 0.37 (0.77) | 0.88 |
| *good*;*bad* | 0.78 | 1.23 | 1.42 (0.72) | 0.66 |
| *surprised*;*relaxed* | 0.92 | 1.11 | 1.25 (0.94) | 0.93 |
| *emotional*;*society* | 0.89 | 0.78 | 0.73 (0.84) | 0.89 |

Table 1: Word pairs, provided with their (cosine) distances, measured on the corresponding vectors given by different models. The distances in parentheses are from a model, that was post-processed to capture emotion information.

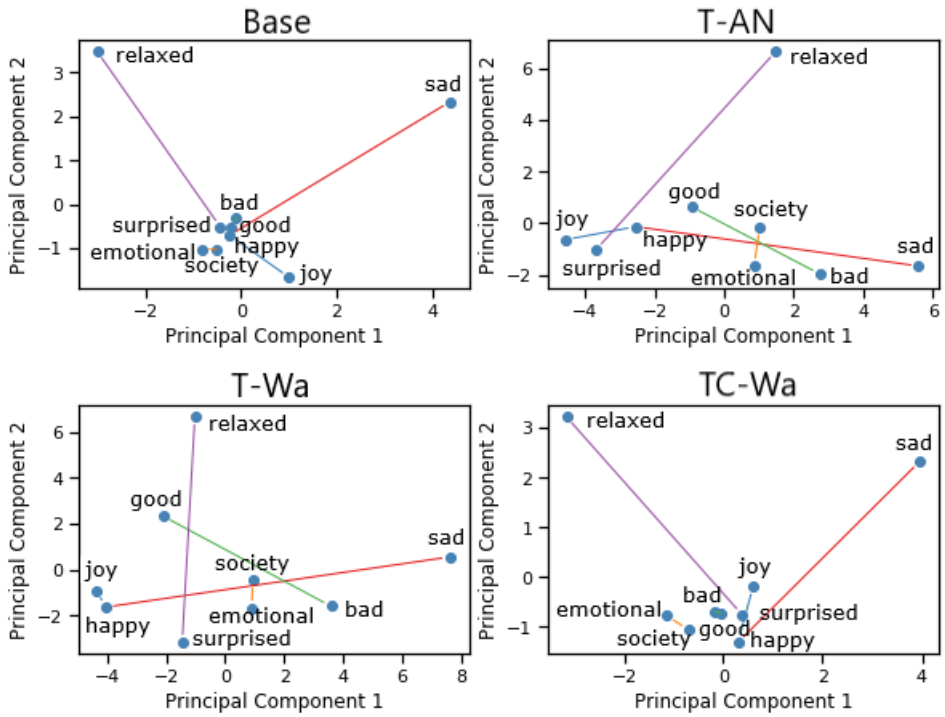


Figure 4: PCA representation for selected word pairs. Visualization shows that the new proposed models improve the emotional distance between word vectors. The distance between ”joy” and ”happy” got smaller, while the distance of ”happy” and ”sad” grew larger.

Clearly, the target-affect models increase the distances of *happy*;*sad* and *good*;*bad*. This is an ideal result, since each word pair expresses a non-flat opposite valence, which is incorporated into their vectors. The same rational works for *happy* and *joy*, both words have a similar valence, so their distance decreases under the target-affect approach, bringing them closer then before. The *surprised*;*relaxed* pair demonstrates a case of opposite arousal weights, which make their corresponding target-affect vectors become slightly different than before. On

the other hand, the target-context affect model does not change the distances significantly for those pairs, and for *good*;*bad* it even puts them closer to each other in the vector space. However, it does seem to properly handle the pair *emotional* and *society*, both express no emotions, keeping them in the same distance as before, which is what we would want as a result (The target-affect model does worse by dragging them closer together, since they have a similar emotional weight). This is a nice behaviour; we relate this result to the fact that both *emotional* and *society* appear in different contexts expressing diverse emotions, which probably cause the target-context affect network predict a similar average emotional weight for both words. We provide another visualization in Figures 5-6. We visualize the vectors of several emotional words (e.g., aroused, joy, relaxed, quiet, sad, surprised, happy, calm, sleepy, bad, good), alongside some other words, this time using t-SNE (Van der Maaten and Hinton, 2008) for reducing vector dimensionality to a visualization level. Figure 5 shows the words vectors that were generated by the Baseline model, while Figure 6 shows the same words using vectors that were generated by the T-AN model. It can be seen that the Baseline model groups the words together, while T-AN separates the words according to their emotional context.

We believe that with words that appear in a more homogenic emotional context, such as political figures and parties, we would observe the distances derived by the target-context affect model, change accordingly. Since Wikipedia does not qualify as an emotional chatter platform, we plan to use other corpora, where emotions are likely to be expressed, and in a more consisted way. Finally, we evaluate

our approach on a different language, as we train a skip-gram model as well as a target-affect model in Spanish,[7] and compare distances of word pairs, equivalent to the English ones we use. The results are encouraging, as can be seen in the following table (translations are in Table 1).

| **Pair** | Baseline | T-AN(Spanish) |
|---|---|---|
| *feliz*;*triste* | 0.70 | 1.21 |
| *feliz*;*alegria* | 0.77 | 0.52 |
| *bien*;*mala* | 0.82 | 1.44 |
| *sorpendido*;*relajada* | 0.95 | 1.05 |
| *emocional*;*sociedad* | 0.74 | 0.71 |

Table 2: Distances of equivalent Spanish word pairs.

[7]We use a Spanish adaptation of ANEW (Redondo et al., 2007) for retrieving emotional weights; we train the models using a Spanish Wikipedia corpus (Reese et al., 2010).

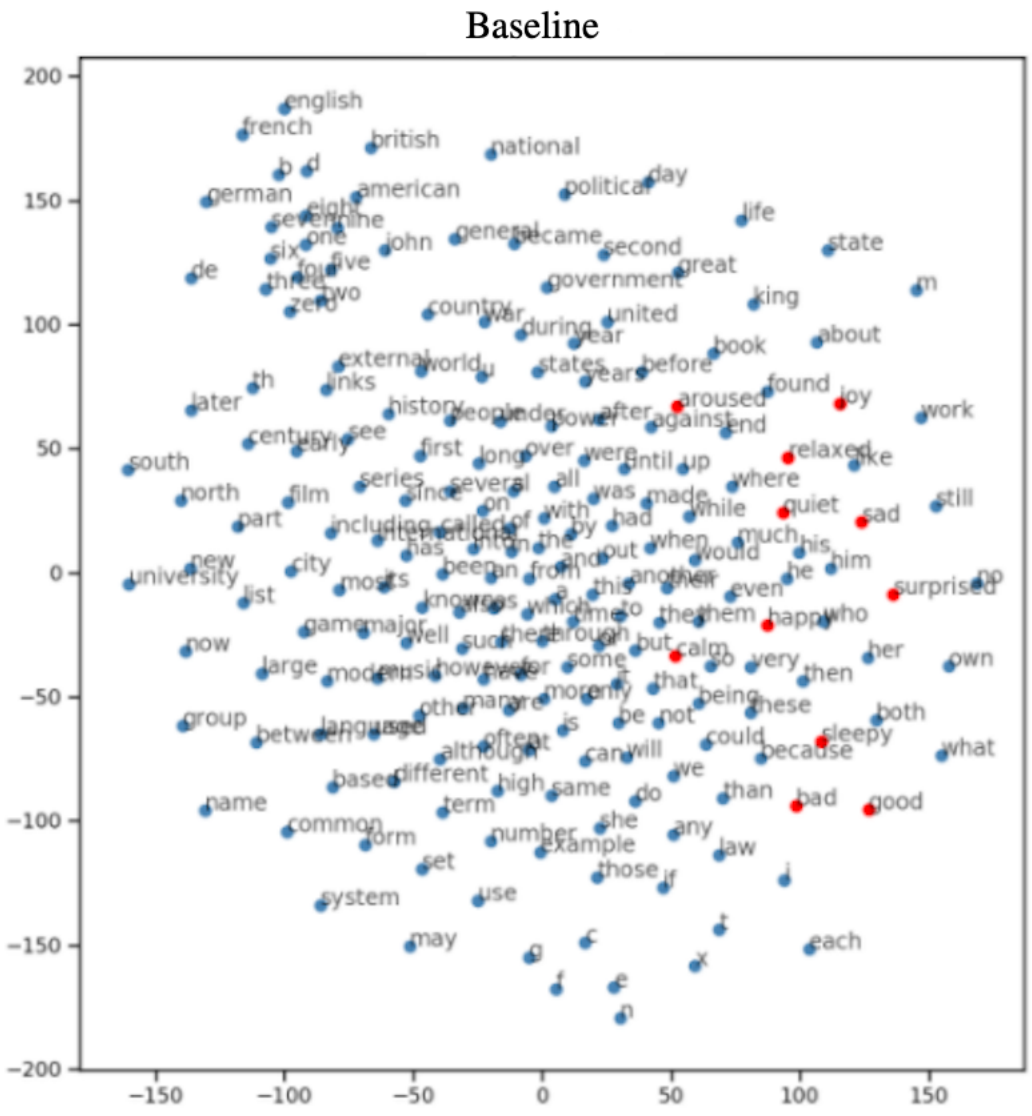


Figure 5: t-SNE visualization of word vectors generated by the Baseline model. It can been seen that the examined emotional words (in Red) are grouped together.

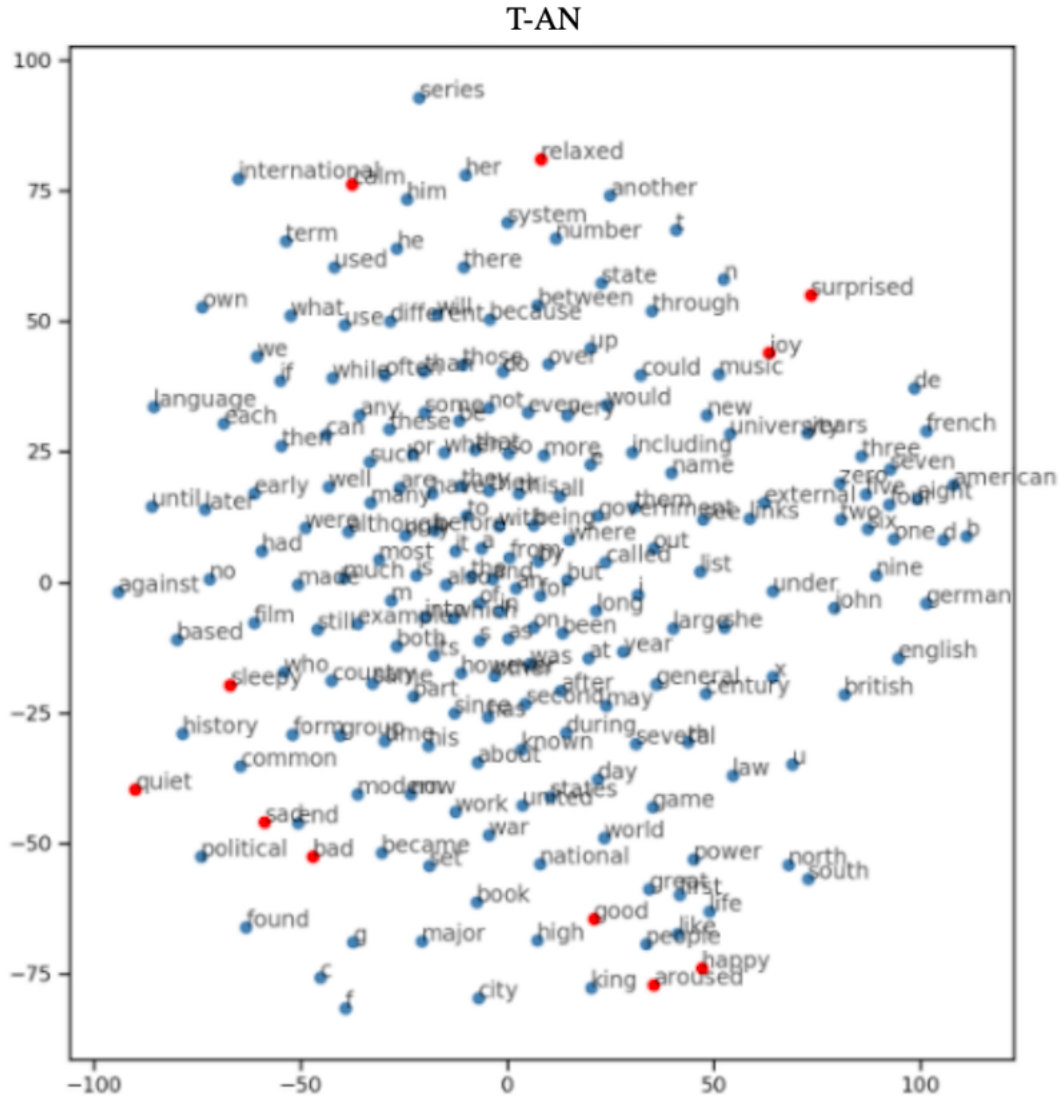


Figure 6: t-SNE visualization of word vectors generated by the T-AN model. It can been seen that the examined emotional words (in Red) are separated according to their emotional context.

## 4.2 Downstream Tasks

We evaluate the contribution of our vectors to the following two affect-related downstream tasks: (1) The Stanford Sentiment Treebank 2 (SST-2) Socher et al. (2013), containing about 12K movie reviews written in English, tagged with one of five labels, and (2) EmoBank Buechel and Hahn (2017), containing 10K English short texts from different sources (but mostly from SST), annotated with VAD values. For both tasks, we employ a flavour of a recurrent neural network (RNN), connected with an array of fully-connected linear layers for predicting the required target value. We design a simple architecture, so that we can focus on measuring the contribution of the vectors.

For SST-2, we use a 2-layer bidirectional Gated Recurrent Unit (GRU) Cho et al. (2014) RNN architecture, and take the concatenation of the final two vectors (of the two directions) and simply forward it to a linear layer for a final prediction of the label. We use a cross-entropy loss for optimizing the network. The hidden size that we use for this network is 700. For EmoBank, we use a bidirectional 2-layer Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997) architecture, expanded with a simple additive attention mechanism for generating one context vector, which we forward into an array of two linear layers. The output layer generates the three valence-arousal-dominance numeric values, which we optimize with a regression-style MSE loss function. The hidden size that we use for this network is 800. We train and evaluate both networks using the train/test splits that were defined originally by the data owners, with the word vectors that we generate with the four models.

| | SST-2 | | EmoBank | |
|---|---|---|---|---|
| **Model** | **Static** | **Dynamic** | **Static** | **Dynamic** |
| Baseline | 69.4% | 78.0% | 0.438 | 0.421 |
| T-AN | 71.8% | 80.8% | 0.377 | 0.400 |
| T-Wa | 71.5% | 80.5% | 0.388 | 0.397 |
| TC-Wa | 70.3% | 79.9% | 0.414 | 0.380 |

Table 3: SST-2 and EmoBank evaluation results.

We choose whether to freeze the vector weights during training (referred to as *static*), or to continue updating them (referred to as *dynamic*). We report on the accuracy values (the higher, the better) for SST-2, and regression-style MSE (the lower, the better) for EmoBank in Table 3.

Overall, the results clearly show an improvement in both tasks when the modified vectors are used. For the most part, when we keep optimizing the vectors for the downstream task (i.e., *dynamic*), the results get better, except for the target-affect models when trained on EmoBank, which may be related to the low agreement rate between annotators (we plan to continue investigating that part in the near future).

# 5 Conclusion

The method proposed here has demonstrated its potential for augmenting vectors with affect information during training. Similar to other works, we have used affect lexicons as a resource for retrieving word-level emotional weights. Our qualitative study suggests that words, which express intense affect are likely to get moved to a better position in the vector space with respect to other words with

a similar or an opposite affect. A similar conclusion has been drawn by evaluating a neural network that uses our vectors to address two emotion-related downstream tasks. In addition to English, we got some preliminary results for Spanish, which look similar to the ones we got got English. In the next stage of this research, we could expand our experiments on other languages, as well as testing our approach for updating vectors based on emotional weights of surrounding words, in a more dynamic corpus where emotions toward specific words and entities are expressed in a more consistent way.

# 6 References


Abdur-Rahim J, Morales Y, Gupta P, Umata I, Watanabe A, Even J, Suyama T, Ishii S. 2016. Multi-sensor based state prediction for personal mobility vehicles. *PloS one*. 11:e0162593.

Bar K, Zilberstein V, Ziv I, Baram H, Dershowitz N, Itzikowitz S, Vadim Harel E. 2019. Semantic characteristics of schizophrenic speech. In: Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology (NAACL). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 84–93.

Bradley MM, Lang PJ. 1999. Affective norms for English words (ANEW): Instruction manual and affective ratings. *Technical report, Technical Report C-1, the Center for Research in Psychophysiology, University of Florida*. .

Buechel S, Hahn U. 2017. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 578–585.

Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on

Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734.

Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA. 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 (HLT-NAACL). Denver, Colorado: Association for Computational Linguistics, pp. 1606–1615.

Gonen H, Goldberg Y. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, pp. 609–614.

Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Computation*. 9:1735–1780.

Khosla S, Chhaya N, Chawla K. 2018. Aff2vec: Affect–enriched distributional word representations. *arXiv preprint arXiv:1805.07966*. .

Kingma DP, Ba J. 2015. Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Mikolov T, Chen K, Corrado G, Dean J. 2013a. Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations (ICLR).

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. 2013b. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119.

Miller GA. 1998. WordNet: An electronic lexical database. MIT press.

Mrkšić N, Ó Séaghdha D, Thomson B, Gašić M, Rojas-Barahona L, Su PH, Vandyke D, Wen TH, Young S. 2016. Counter-fitting word vectors to linguistic constraints. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL).

Palogiannidi E, Iosif E, Koutsakis P, Potamianos A. 2015. Valence, arousal and dominance estimation for English, German, Greek, Portuguese and Spanish lexica using semantic models. In: Sixteenth Annual Conference of the International Speech Communication Association.

Pennington J, Socher R, Manning CD. 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543.

Ravfogel S, Elazar Y, Gonen H, Twiton M, Goldberg Y. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In: Jurafsky D, Chai J, Schluter N, Tetreault JR, editors, Proceedings of the 58th Annual Meeting of

the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, pp. 7237–7256.

Recchia G, Louwerse MM. 2015. Reproducing affective norms with lexical co-occurrence statistics: Predicting valence, arousal, and dominance. *Quarterly journal of experimental psychology*. 68:1584–1598.

Redondo J, Fraga I, Padrón I, Comesaña M. 2007. The Spanish adaptation of ANEW (affective norms for English words). *Behavior Research Methods*. 39:600–605.

Reese S, Boleda G, Cuadros M, Rigau G. 2010. Wikicorpus: A word-sense disambiguated multilingual Wikipedia corpus. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA).

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1631–1642.

Speer R, Chin J, Havasi C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI Press, AAAI'17, p. 4444–4451.

Van der Maaten L, Hinton G. 2008. Visualizing data using t-sne. *Journal of machine learning research*. 9.

Vankrunkelsven H, Verheyen S, De Deyne S, Storms G. 2015. Predicting lexical norms using a word association corpus. In: Proceedings of the 37th Annual Conference of the Cognitive Science Society. Cognitive Science Society; Austin, TX, pp. 2463–2468.

Warriner AB, Kuperman V, Brysbaert M. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior Research Methods*. 45:1191–1207.

Xu C, Bai Y, Bian J, Gao B, Wang G, Liu X, Liu TY. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 1219–1228.

Yu LC, Wang J, Lai KR, Zhang X. 2017. Refining word embeddings for sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 534–539.