

EmotionX-AR: CNN-DCNN autoencoder based Emotion Classifier

Sopan Khosla

Big Data Experience Lab
Adobe Research, Bangalore, India
skhosla@adobe.com

Abstract

In this paper, we model emotions in EmotionLines dataset using a convolutional-deconvolutional autoencoder (CNN-DCNN) framework. We show that adding a joint reconstruction loss improves performance. Quantitative evaluation with jointly trained network, augmented with linguistic features, reports best accuracies for emotion prediction; namely joy, sadness, anger, and neutral emotion in text.

1 Introduction

Emotion recognition in content is an extensively studied area. It deals with associating words, phrases or documents with various categories of emotions. The importance of emotion analysis in human communication and interactions has been discussed by Picard (1997). Historically studied using multi-modal data, the study of human emotion from text and other published content has become an important topic in language understanding. Word correlation with social and psychological processes is discussed by Pennebaker (2011). Preotiuc-Pietro et al. (2017) studied personality and psycho-demographic preferences through Facebook and Twitter content. The analysis of emotion in interpersonal communication such as emails, chats and longer written articles is necessary for various applications including the study of consumer behavior and psychology, understanding audiences, and opinions in computational social science, and more recently for dialogue systems and conversational agents. This is an active research space today.

In contrast to sentiment analysis, emotion analysis in user generated content such as tweets (Dodds et al., 2011), blogs (Aman and Sz-

pakowicz, 2007) and chats remains a space less trodden. The WASSA-2017 task on emotion intensity (Mohammad and Bravo-Marquez, 2017) aims at detecting the intensity of emotion felt by the author of a tweet. Whereas (Alm et al., 2005; Aman and Szpakowicz, 2007; Brooks et al., 2013; Neviarouskaya et al., 2009; Bollen et al., 2011) provide discrete binary labels to text instances for emotion classification. Typical discrete categories are a subset of those proposed by Ekman (Ekman, 1992) namely anger, joy, surprise, disgust, sadness, and fear.

Paper Structure: The remainder of the paper is organized as follows. We summarize the EmotionLines dataset in Section 2. Section 3 describes different parts of our system. We present our experiments in Section 4. Section 5 discusses the results of our final system submitted to the EmotionX challenge. Finally, we present conclusion and future directions in section 6.

2 Data

EmotionLines dataset contains dialogues from the Friends TV series and EmotionPush chat logs. Both Friends TV scripts and EmotionPush chat logs contain 1,000 dialogues split into training(720), development(80), and testing(200) set separately. In order to preserve completeness of any dialogue, the corpus was divided by the dialogues, not the utterances. Refer to Chen et al. (2018) for details on the dataset collection and construction.

The EmotionX task on EmotionLines dialogue dataset tries to capture the flow of emotion in a conversation. Given a dialogue, the task requires participants to determine the emotion of each utterance (in that dialogue) among four label candidates: joy, sadness, anger, and neutral.

3 System Description

In this section, we provide the technical details of our model.

3.1 Architecture Overview

We propose a joint learning framework for emotion detection built on a convolutional encoder (CNN). We introduce a joint learning objective where the network needs to learn the (1) utterance text (the data itself) and the (2) emotion information from the labeled data (EmotionLines) together. The CNN along with a deconvolutional decoder (DCNN) provides the mechanism for text reconstruction, i.e. to learn the text sequences. On the other hand, the learned encoding, augmented with linguistic features, acts as the input feature space for emotion detection.

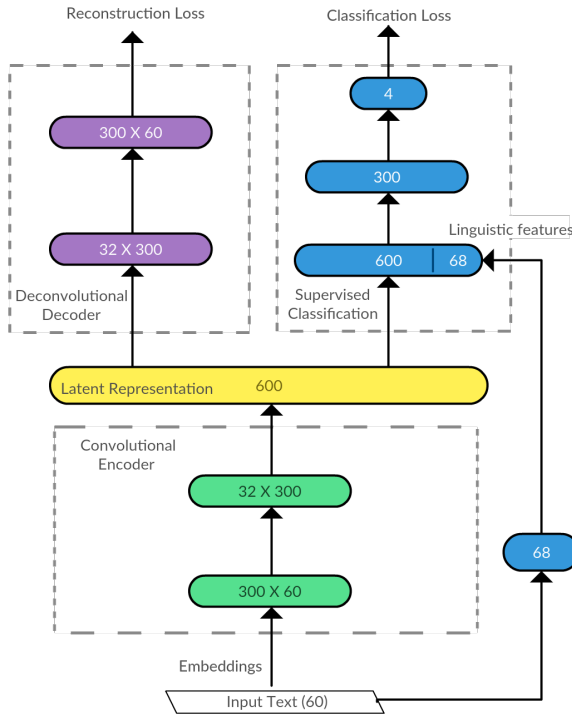


Figure 1: Architecture Overview

The architecture diagram is shown in figure 1. The network aims at emotion classification and in turn also learns the reconstruction objective. Key components of this approach are: **(1) Convolutional Autoencoder (CNN-DCNN)**, **(2) Linguistic features**, and **(3) Joint-learning objective**.

Consider a text input d to the model. Each word w_d^t in d is embedded into a k -dimensional representation $e_t = \mathbf{E}[w_d^t]$ where \mathbf{E} is a learned matrix. The embedding layer is passed through a

CNN encoder to create a fixed-length vector \mathbf{h}_L for the entire input text d . This latent representation, appended with linguistic features is then sent to a fully connected layer with a softmax classifier on top. Along with this, \mathbf{h}_L is also fed to a deconvolutional decoder which attempts to reconstruct d from the latent vector. Therefore, the final loss function: $\alpha_{ae}L_{ae} + (1 - \alpha_{ae})L_c$ for the model is a combination of the classification error L_c and the reconstruction error L_{ae} explained in the following subsections.

3.2 CNN-DCNN Autoencoder

Zhang et al. (2017) introduce a sequence-to-sequence convolutional encoder followed by a deconvolutional decoder (CNN-DCNN) framework for learning latent representations from text data. Their proposed framework outperforms RNN-based networks for text reconstruction and semi-supervised classification tasks. We leverage their network in our work.

Convolutional Encoder. CNN with L layers, inspired from Radford et al. (2015) is used to encode the document into a latent representation vector, \mathbf{h}_L . Former $L - 1$ convolutional layers create a feature map which is fed into a fully-connected layer implemented as a convolutional layer. This final layer produces the latent representation \mathbf{h}_L which acts as a fixed-dimensional summarization of the document.

Deconvolutional Decoder. We leverage the deconvolutional decoder introduced by Zhang et al. (2017) as is for our model. The reconstruction loss is defined as,

$$L_{ae} = \sum_{d \in D} \sum_t \log p(\hat{w}_d^t = w_d^t), \quad (1)$$

where D is the set of observed sentences. w_d^t and \hat{w}_d^t correspond to the words in the input and output sequences respectively.

3.3 Linguistic Features

Here, we explain the various linguistic features used in our network. Inspired from Chhaya et al, (2018), we use 68 linguistic features further divided into 4 sub-groups: Lexical, Syntactic, Derived and Affect-based. The lexical and syntactic features include features such as ‘averageNumberOfWords per sentence’ and ‘number of capitalizations’. Features that can help quantify readability of text are the part of derived features.

Thus, this set contains features like Hedges, Contractions, and Readability scores. The fourth group of features are the Affect-related features. These features are lexica-based and quantify the amount of affective content present in the text. All features used by Pavlick et al. (2016) for formality detection and by Danescu et al. (2013) for politeness detection are included in our analysis. We use Stanford CoreNLP¹ and TextBlob² feature extraction and pre-processing.

Lexical and Syntactic Features: The lexical features capture various counts associated with the content like '#Question Marks', 'Average Word Length' etc. Syntactic features include NER-based features, Number of blank lines, and text density which is defined as follows:

$$\rho = \frac{\#(sentences)}{1 + \#(lines)}$$

where ρ is the text density, $\#(sentences)$ denotes number of sentences in the text content and $\#(lines)$ number of lines including blank lines in the text message. Prior art in NLP extensively relies on these features for their analysis.

Derived: Readability Features: The derived features capture information such as readability of text, existence of hedges, subjectivity, contractions and sign-offs. Subjectivity, contractions and hedges are based on the TextBlob implementation. Readability is measured based on Flesh-Kincaid readability score. This score is a measure of ease of reading of given piece of text. We use the textstat package³ in Python for implementation.

Psycholinguistic Features: The affect features used in our analysis include:

1. **Valence-Arousal-Dominance (VAD) Model (Mehrabian, 1980):** We use the Warriner’s lexicon (Warriner et al., 2013) for these features. This lexicon contains real-valued scores for Valence, Arousal, and Dominance (VAD) on a scale of 1–9 each for 13915 English words. 1, 5, 9 correspond to the low, moderate (i.e. neutral), and high values for each dimension respectively.

2. **Ekman’s Emotions (Ekman, 1992):** Ekman introduced six fundamental emotions namely anger, joy, surprise, disgust, sadness, and fear. In this work, we use the NRC lexicon (EMOLEX) (Mohammad et al., 2013) which provides a measure for the existence of the emotion as well as the intensity of the detected emotion on word level.
3. **PERMA Model (Seligman, 2011):** The PERMA model is a scale to measure positivity and well-being in humans (Seligman, 2011). This model defines the 5 dimensions: Positive Emotions, Engagement, Relationships, Meaning, and Accomplishments as quantifiers and indicators of positivity and well-being. Schwartz et al. (Schwartz et al., 2013) published a PERMA lexicon. We use this lexicon in our work.

Formality Lists: We use the formality list, provided by Brooke et al. (2010), for our experiments. It contains a set of words usually used to express formality or informality in text.

3.4 Supervised Classification

Traditional affective language studies focus on analyzing features including lexical (Pennebaker et al., 2001), syntactic, and psycholinguistic features to detect emotions. We augment the latent vector produced by CNN encoder with the set of linguistic features (Section 3.3) to capture emotions.

Let \mathbf{h}' denote the representation vector for linguistic features extracted from the input data d . \mathbf{h}' is normalized and concatenated with \mathbf{h}_L to derive $\mathbf{h}'' = \mathbf{h}_L \frown \mathbf{h}'$. \mathbf{h}'' , producing a probability p_n for each neuron in the softmax layer, where y_n denotes the ground-truth for corresponding class n .

We use cross-entropy based classwise loss as given below:

$$loss_n = -\left[y_n \log(p_n) + (1 - y_n) \log(1 - p_n) \right]$$

Since, EmotionLines suffers from class imbalance, we give higher weight (w_n) to the losses incurred on data samples of minority classes.

$$w_n = \frac{a_n}{\sum_{i=1}^N a_i}$$

where a_n denote the number of samples of class n in the training set. Finally, we use a weighted

¹<https://stanfordnlp.github.io/CoreNLP/>

²<https://textblob.readthedocs.io/en/dev/>

³<https://pypi.python.org/pypi/textstat/0.1.6>

Features	Feature list
Lexical	Average Word Length, Average Words per Sentence, # of Upper Case Words, # Ellipses, # Exclamation marks, # Question Mark, # Multiple Question Marks, # Words, # Lower Case words, First word upper case, # NonAlphaChars, # Punctuation Chars
Syntactic	# BlankLines, NER-Person, NER-Location, NER-PersonLength, NER-Organization, TextDensity
Derived	# Contractions, ReadabilityScore- FKgrade, FirstPerson, Hedge, Subjectivity, Sentiment, ThirdPerson, SignOff
Psycholinguistic Features	ANEW-arousal, ANEW-dominance, ANEW-valence, EmolexIntensity-anger, EmolexIntensity-fear, EmolexIntensity-joy, EmolexIntensity-sadness, Emolex-anger, Emolex-anticipation, Emolex-disgust, Emolex-fear, Emolex-joy, Emolex-negative, Emolex-positive, Emolex-sadness, Emolex-surprise, Emolex-trust, Perma-NEG-A, Perma-NEG-E, Perma-NEG-M, Perma-NEG-P, Perma-NEG-R, Perma-POS-A, Perma-POS-E, Perma-POS-M, Perma-POS-P, Perma-POS-R
Formal Words	formal-words, informal-words (Brooke et al., 2010)

Table 1: Summary of feature groups used in our model.

Friends TV Series Script	WA	UWA	Joy	Sad	Ang	Neu
CNN + MLP (S)	67.67	57.61	66.67	38.70	38.82	76.58
S + Joint Learning (J)	67.40	58.47	63.41	45.16	38.82	76.17
S + Linguistic Features (L)	65.30	59.48	66.67	43.55	47.06	70.88
S + J + L	60.97	59.39	59.35	58.06	44.71	64.56

EmotionPush Chat Logs	WA	UWA	Joy	Sad	Ang	Neu
CNN + MLP (S)	68.89	59.22	69.37	76.31	22.22	68.97
S + Joint Learning (J)	70.44	59.58	68.75	76.31	22.22	71.03
S + Linguistic Features (L)	67.54	64.03	70.00	63.16	55.56	67.39
S + J + L	65.69	65.08	71.88	68.42	55.56	64.48

Table 2: Weighted (WA) and Unweighted (UWA) accuracies(%) on Friends and EmotionPush validation sets provided by the challenge authors. **S**: Supervised learning using CNN encoder trained on labeled data only, **J**: Joint learning with reconstruction task using DCNN decoder, **L**: Linguistic features.

cross entropy loss defined by

$$L_c = -\frac{1}{N} \sum_{n=1}^N w_n * loss_n \quad (2)$$

Table 1 provides a summary of the features considered. Ngrams and other semantic features are ignored as they introduce domain-specific biases. Word-embeddings are treated separately and considered as raw features to train a supervised model.

3.5 Joint learning

The CNN-DCNN network learns the text information i.e. sequences, the linguistic features learn the emotional aspect. Joint learning introduces the

mechanism to learn shared representations during the network training. We implement joint learning using simultaneous optimization for both sequence reconstruction (CNN-DCNN) and emotion detection (linguistic features). The combined loss function is given by,

$$L = \alpha_{ae} L_{ae} + (1 - \alpha_{ae}) L_c. \quad (3)$$

where α_{ae} is a balancing hyperparameter with $0 \leq \alpha_{ae} \leq 1$. Higher the value of α_{ae} , higher is the importance given to the reconstruction loss L_{ae} while training and vice versa.

Friends	WA	UWA	Joy	Sad	Ang	Neu
S	64.90	59.09	69.10	53.22	45.88	68.24
S + J	69.54	60.54	71.54	51.67	45.88	75.20
S + L	62.78	59.16	62.60	54.10	56.47	64.75
S + J + L	65.83	60.48	68.29	48.33	58.82	68.44

EmPush	WA	UWA	Joy	Sad	Ang	Neu
S	68.89	64.62	80.62	52.63	66.67	58.54
S + J	70.44	60.53	85.00	58.97	33.33	63.27
S + L	67.54	62.95	76.87	56.41	44.44	72.60
S + J + L	65.69	64.89	75.62	63.16	55.56	65.21

Table 3: Accuracy(%) for models trained on Friends + EmotionPush data, tested on individual validation sets.

4 Experiments

In this section, we show the experimental evaluation of our system on the EmotionLines dataset.

4.1 Experimental Setup

CNN encoder with MLP Classifier: We use 300-dimensional pre-trained glove word-embeddings (Pennington et al., 2014) as input to the model. The encoder contains two convolutional layers. Size of the latent representation is set to 600. The MLP classifier contains one fully-connected layer followed by a softmax layer.

Joint Training: We set $\alpha_{ae} = 0.5$ as this gives equal importance to both objectives and reports best results.

Linguistic Features: We concatenate a full set of 68 linguistic features with the latent representation for emotion detection.

Friends	UWA	Joy	Sad	Ang	Neu
Our Model	62.5	71.1	55.3	55.3	68.3
Highest	62.5	71.1	55.3	55.3	99.5

EmPush	UWA	Joy	Sad	Ang	Neu
Our Model	62.5	76.0	51.7	45.9	76.3
Highest	62.5	76.0	54.0	45.9	99.0

Table 4: Results on the EmotionX challenge test sets for Friends and EmotionPush datasets. Accuracy(%) rounded off to one decimal point.

4.2 Results

Table 2 shows the results for models trained on individual training sets using our weighted loss function. The performance is evaluated using both, the weighted accuracy (WA) and the unweighted accuracy (UWA), as defined by the chal-

lenge authors (Chen et al., 2018).

$$WA = \sum_{c \in C} p_c a_c \quad (4)$$

$$UWA = \frac{1}{|C|} \sum_{c \in C} a_c \quad (5)$$

where a_c denotes the accuracy of emotion class c and p_c denotes the percentage of utterances in emotion class c .

Adding a reconstruction loss with classification loss improves performance. We attribute this to improved generalizability provided by a semi-supervised loss. Concatenating linguistic features improves minority class accuracies for both Friends TV dialogues and EmotionPush chats. The improvements due to joint loss and linguistic features are more significant for EmotionPush chat log dataset. Accuracies of majority class (Neutral) take a considerable hit with the addition of **J** and **L** for both datasets, whereas minority emotions like Sadness and Anger consistently benefit from addition of linguistic features.

Table 3 contains results for models trained on both Friends and EmotionPush training data. Increase in training data, even though from a different domain, improves performance for Joy and Anger emotions. Accuracy on sadness dips significantly for EmotionPush. Overall WA and UWA also increase slightly for Friends dataset.

5 EmotionX Submission and Analysis

We implement an ensemble of the four model variants trained on the Friends + EmotionPush data as our final submission for the EmotionX challenge. We arrive at the final class predictions using the algorithm explained in Algorithm 1. For each test

Algorithm 1 Ensemble Algorithm

```
1: procedure FILTER( $p_x, threshold$ )
2:    $candidates \leftarrow []$ 
3:   for  $p \in p_x$  do ▷ For each base model
4:     if  $max(p) > threshold$  then
5:        $candidates.add(arg\ max(p))$ 
6:     end if
7:   end for
8:   return  $candidates$ 
9: end procedure
10:
11: procedure ENSEMBLE( $data, p\_softmax$ ) ▷  $p\_softmax.shape = (\#test\ samples, \#models, \#classes)$ 
12:    $ensemble\_pred \leftarrow []$ 
13:   for  $x \in data$  do
14:      $candidate\_classes \leftarrow FILTER(p\_softmax[x], 0.75)$  ▷ High Confidence
15:     if  $len(candidate\_classes) > 0$  then
16:        $ensemble\_pred.add(most\_common(candidates))$ 
17:     else
18:        $candidate\_classes \leftarrow FILTER(p\_softmax[x], 0.50)$  ▷ Moderate Confidence
19:       if  $len(candidate\_classes) > 0$  then
20:          $ensemble\_pred.add(most\_common(candidate\_classes))$ 
21:       else
22:          $candidate\_classes \leftarrow FILTER(p\_softmax[x], 0.00)$  ▷ Low Confidence
23:          $ensemble\_pred.add(most\_common(candidate\_classes))$ 
24:       end if
25:     end if
26:   end for
27:   return  $ensemble\_predictions$ 
28: end procedure
```

sample, we find models for which the maximum output probability associated with a class is greater than a threshold of 0.75 (High Confidence). Predictions from this subset are considered as the candidate high confidence classes. The most common class in this subset is taken as the final prediction for EmotionX submission. If the subset is empty, a similar approach is followed but with a reduced threshold of 0.50 (Moderate Confidence). Predictions for samples which do not satisfy any of the above thresholds are termed as Low Confidence Predictions.

The results on the test-set for both datasets are shown in Table 4. Comparison with the best results in each class shows that for Friends dataset, our model tops for all emotions except Neutral. Whereas, for the EmotionPush dataset, we perform well on Joy and Anger. Our model had the best unweighted accuracy (UWA) for both datasets in the EmotionX challenge.

Text	Prediction	Label
Come on, Lydia, you can do it.	Neutral	Neutral
Push!	Anger	Joy
Push 'em out, push 'em out, harder, harder.	Anger	Joy
Push 'em out, push 'em out, way out!	Anger	Joy
Let's get that ball and really move, hey, hey, ho, ho.	Joy	Joy
Let's- I was just-yeah, right.	Joy	Joy
Push!	Anger	Joy
Push!	Anger	Joy

Table 5: An example dialogue from Friends dataset with corresponding predictions and labels.

5.1 Error Analysis

Our model does not explicitly import contextual information from other utterances in the conversation. Therefore, quite expectedly, we found that

most of the utterances misclassified by our model occur in dialogues where the current utterance does not exhibit the emotion it is tagged with.

Another set of errors occur where the whole conversation is not able to explain the respective emotions of each utterance. Table 5 shows an example conversation where it might be difficult for even a human to classify the utterances without the associated multi-modal cues.

6 Conclusion and Future Work

We propose a CNN-DCNN autoencoder based approach for emotion detection on EmotionLines dataset. We show that addition of a semi-supervised loss improves performance. We propose multiple linguistic features which are concatenated to the latent encoded representation for classification. The results show that our model detects emotions successfully. The network, using a weighted classification loss function, tries to handle the class imbalance in the dataset.

In future, we plan to include results of modeling emotion on the whole dialog using an LSTM layer over our network. We would experiment with concatenating subsets of linguistic features to better estimate the contribution of each feature group. We also plan to use data-augmentation techniques such as backtranslation and word substitution using Wordnet and word-embeddings in order to handle class-imbalance in the dataset.

References

- Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 579–586.
- Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*. Springer, pages 196–205.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. [Enriching word vectors with subword information](http://arxiv.org/abs/1607.04606). *CoRR* abs/1607.04606. <http://arxiv.org/abs/1607.04606>.
- Johan Bollen, Huina Mao, and Alberto Pepe. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Icwsn* 11:450–453.
- Julian Brooke, Tong Wang, and Graeme Hirst. 2010. Automatic acquisition of lexical formality. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pages 90–98.
- Michael Brooks, Katie Kuksenok, Megan K Torkildson, Daniel Perry, John J Robinson, Taylor J Scott, Ona Anicello, Ariana Zukowski, Paul Harris, and Cecilia R Aragon. 2013. Statistical affect detection in collaborative chat. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, pages 317–328.
- Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Lun-Wei Ku, et al. 2018. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*.
- Niyati Chhaya, Kushal Chawla, Tanya Goyal, Projjal Chanda, and Jaya Singh. 2018. Frustrated, polite or formal: Quantifying feelings and tone in emails. In *Second Workshop on Computational Modeling of Peoples Opinions, Personality, and Emotions in Social Media, NAACL HLT*.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078*.
- Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. 2011. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PLoS one* 6(12):e26752.
- Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion* 6(3-4):169–200.
- Albert Mehrabian. 1980. Basic dimensions for a general psychological theory implications for personality, social, environmental, and developmental studies.
- Saif M Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.
- Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2009. Compositionality principle in recognition of fine-grained emotions from text. In *ICWSM*.
- Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics* 4:61–74.

- James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 71:2001.
- J.W. Pennebaker. 2011. *The Secret Life of Pronouns: What Our Words Say About Us*. Bloomsbury USA. <https://books.google.com/books?id=Avz4rthHySEC>.
- 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.
- Rosalind W. Picard. 1997. *Affective Computing*. MIT Press, Cambridge, MA, USA.
- Daniel Preotiuc-Pietro, Ye Liu, Daniel J. Hopkins, and Lyle Ungar. 2017. Personality Driven Differences in Paraphrase Preference. In *Proceedings of the Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*. ACL.
- Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. [Ultradense word embeddings by orthogonal transformation](http://arxiv.org/abs/1602.07572). *CoRR* abs/1602.07572. <http://arxiv.org/abs/1602.07572>.
- H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one* 8(9):e73791.
- João Sedoc, Jean Gallier, Lyle H. Ungar, and Dean P. Foster. 2016. [Semantic word clusters using signed normalized graph cuts](http://arxiv.org/abs/1601.05403). *CoRR* abs/1601.05403. <http://arxiv.org/abs/1601.05403>.
- Martin EP Seligman. 2011. Flourish: a visionary new understanding of happiness and well-being. *Policy* 27(3):60–1.
- Ivan Vulic, Nikola Mrksic, Roi Reichart, Diarmuid Ó Séaghdha, Steve J. Young, and Anna Korhonen. 2017. [Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules](http://arxiv.org/abs/1706.00377). *CoRR* abs/1706.00377. <http://arxiv.org/abs/1706.00377>.
- Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods* 45(4):1191–1207.
- Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. [Deconvolutional paragraph representation learning](http://arxiv.org/abs/1708.04729). *CoRR* abs/1708.04729. <http://arxiv.org/abs/1708.04729>.