# Detecting Most Insightful Parts of Documents Using a Regularized Attention-Based Model

Kourosh Modarresi[1]

kouroshm@alumni.stanford.edu

**Abstract.** Every individual text or document is generated for specific purpose(s). Sometime, the text is deployed to convey a specific message about an event or a product. Other occasions, it may be communicating a scientific breakthrough, development or new model and so on. Given any specific objective, the creators and the users of documents may like to know which part(s) of the documents are more influential in conveying their specific messages or achieving their objectives. Understanding which parts of a document has more impact on the viewer's perception would allow the content creators to design more effective content. Detecting the more impactful parts of a content would help content users, such as advertisers, to concentrate their efforts more on those parts of the content and thus to avoid spending resources on the rest of the document. This work uses a regularized attention-based method to detect the most influential part(s) of any given document or text. The model uses an encoder-decoder architecture based on attention-based decoder with regularization applied to the corresponding weights.

**Keywords:** Artificial Neural Networks, Natural Language Processing, Sparse Loss Function, Regularization, Transformer.

## 1 Motivation of This Work

### 1.1 Review

The main purpose of NLP (Natural Language Processing) and NLU (Natural Language Understanding) is to understand the language. More specifically, they are focused on not just to see the context of text but also to see how human uses the language in daily life. Thus, among other ways of utilizing this, we could provide an optimal online experience addressing needs of users' digital experience. Language processing and understanding is much more complex than many other applications in machine learning such as image classification as NLP and NLU involve deeper context analysis than other machine learning applications. This paper is written as a short paper and focuses on explaining only the parts that are contribution of this paper to the state-of-the art. Thus, this paper does not describe the state-of-the-art works in details and uses those works [2, 4, 5, 8, 53, 59, 65, 69, 73, 83] to build its model as a modification and

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

extension of the state of the art. Therefore, a comprehensive set of reference works have been added for anyone interested in learning more details of the previous state of the art research [3, 5, 10, 17, 32, 47, 48, 60, 61, 62, 66-72, 75, 76, 89, 90, 92 ].

### 1.2 Attention Based Model

Deep Learning has become a main model in natural language processing applications [6, 7, 11, 22, 37, 55, 63, 70, 74, 77-80, 84, 87, 93] . Among deep learning models, often RNN-based models like LSTM and GRU have been deployed for text analysis [9, 13, 16, 23, 31, 38, 39, 40, 41, 49-51, 58]. Though, modified version of RNN like LSTM and GRU have been improvement over RNN ( recurrent neural networks) in dealing with vanishing gradients and long-term memory loss, still they suffer from many deficiencies. As a specific example, a RNN-based encoder-decoder architecture uses the encoded vector (feature vector), computed at the end of encoder, as the input to the decoder and uses this vector as a compressed representation of all the data and information from encoder (input). This ignores the possibility of looking at all previous sequences of the encoder and thus suffers from information bottleneck leading to low precision, especially for texts of medium or long sequences. To address this problem, global attention-based model [2, 5] where each of the encoder sequence uses all of the encoder sequences. Figure 1 shows an attention-based model.

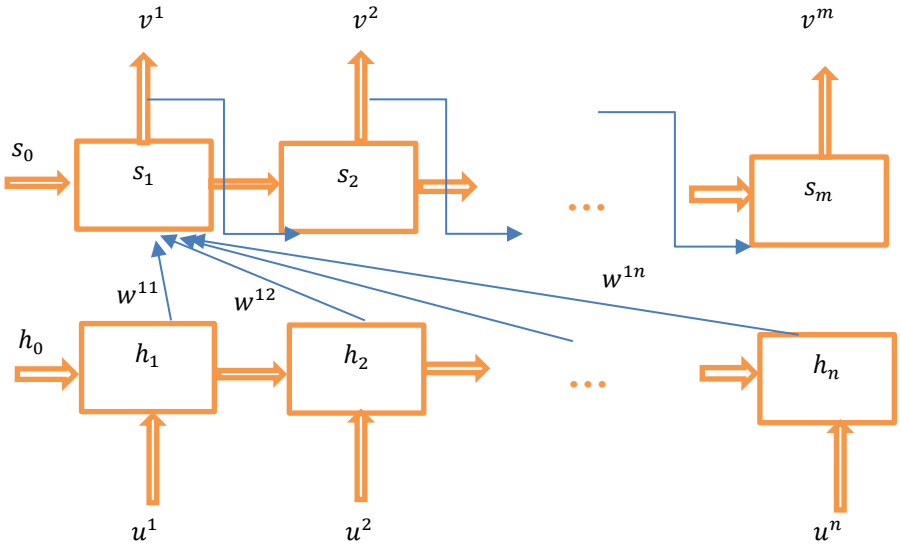


**Fig. 1.** A description of attention-based encoder-decoder architecture. The attention weights for one of the decoder sequences (the first decoder sequence) are displayed.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

Where $i$=1:n is the encoder sequences and , $t$=1:m represents the decoder sequences. Each of the encoder states looks into the data from all the encoder sequences with specific attention measured by the weights. Each weight, $w^{ti}$, indicates the attention decoder network $t$ pays for the encoder network $i$. These weights are dependent on the previous decoder and output states and present encoder state as shown in figure 2.
Given the complexity of these dependencies, a neural network model is used to compute these weights. Two layers (1024) of fully connected layers and ReLU activation function is used.

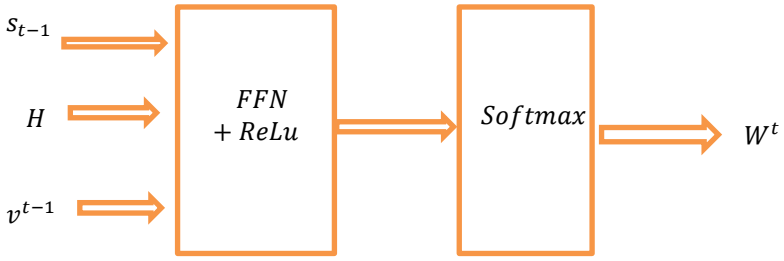


**Fig. 2.** The computation model of weights using fully-connected networks and SoftMax layer**.**

Where $H$ is the state of the encoder networks, $s_{t-1}$ is the previous state of the decoder and $v^{t-1}$is the previous decoder output. Also, $W^t$ is the weights of the encoder state $t$.

$$W^t = \begin{bmatrix} w^{1t} \\ w^{2t} \\ . \\ . \\ . \\ w^{nt} \end{bmatrix}, \qquad H = \begin{bmatrix} h_i \\ h_2 \\ . \\ . \\ . \\ h_n \end{bmatrix}, \tag{1}$$

Since $W^t$ are the output from softmax function, then,

$$\sum_{i=1}^{n} w^{it} = 1 \tag{2}$$

## 2 Sparse Attention-Based Model

This section overviews of the contribution of this paper and explains the extension made over the state-of-the-art model.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

### 2.1 Imposing Sparsity on the Weight Vectors

A major point of attention for many texts related analysis is to determine which part(s) of the input text has had more impact in determining the output. he length of input text could be very long combining of potentially hundreds and thousands of words or sequences, i.e., n could be very large number. Thus, there are many weights ($w^{ti}$) in determining any part of output $v^t$, and also since many of these weights are correlated, it's difficult to determine the significance of any input sequence in computing any output sequence $v^t$. To make these dependencies clearer and to recognize the most significant input sequences for any output sequence, we apply a zero-norm penalty to make the corresponding weight vector to become a sparse vector. To achieve the desired sparsity, zero-norm ($L_0$ ) is applied to make any corresponding $W^t$ vector very sparse as the penalty leads to minimization of the number of non-zero entries in $W^t$. The process is implemented by imposing the constraint of,

$$\|W^t\|_0 \leq \vartheta \tag{3}$$

Since $L_0$ is computationally intractable, we could use surrogate norms such as $L_1$ norm or Euclidean norm, $L_2$. To impose sparsity, the $L_1$ norm, LASSO [8, 14, 15, 18, 21] is used in this work,

$$\|W^t\|_1 \leq \vartheta \tag{4}$$

Or,

$$\beta\|W^t\|_1 \tag{5}$$

As the penalty function to enforce sparsity on the weight vectors.

This penalty, $\beta\|W^t\|_1$, is the first extension to the attention model [2, 5 ]. Here, $\beta$ is the regularization parameter which is set as a hyperparameter where its value is set before learning. Higher constraint leads to higher sparsity with higher added regularization biased error and lower values of the regularization parameter leads to lower sparsity and lesser regularization bias.

### 2.2 Embedding Loss Penalty

The main goal of this work is to find out which parts of encoder sequences are most critical in determining and computing any output. The output could be a word, a sentence or any other subsequence. The goal is critical especially in application such as machine translation, image captioning, sentiment analysis, topic modeling and predictive modeling such as time series analysis and prediction.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

To add another layer of regularization, this work imposes embedding error penalty to the objective function (usually, cross entropy). This added penalty also helps to address the "coverage problem" (the phenomenon of often observed dropping or frequently repeating words - - or any other subsequence - - by the network). The embedding regularization is,

$$\alpha \| Embedding\ Error \|_2 \tag{6}$$

Input to any model has to be a number and hence the raw input of words or text sequence needs to be transformed to continuous numbers. This is done by using one-hot encoding of the words and then using embedding as shown in fig.3.

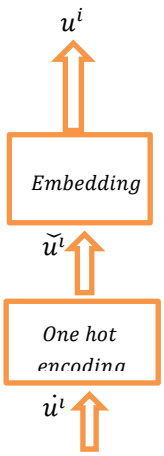


**Fig. 3.** The process of representation of input words by one-hot encoding and embedding**.**

Where $\dot{u}^i$ is the raw input text, $\breve{u}^i$ is the one-hot encoding representation of the raw input and $u^i$ is the embedding of the i-th input or sequence. Also, $\alpha$ is the regularization parameter.

The idea of embedding is based on that embedding should preserve word similarities, i.e., the words that are synonyms before embedding, should remain synonyms after embedding. Using this concept of embedding, the scaled embedding error is,

$$L(U)=\sum_{i=1}^{n}\sum_{j=1}^{n}\left(L(\dot{u}^i,\dot{u}^j)-L(u^i,\ u^j)\right)^2 \tag{7}$$

Or, after scaling the embedding error,

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

$$L(U)=\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(L(\dot{u^i},\dot{u^j})-L(u^i,\ u^j)\right)^2 \tag{8}$$

Which could be re-written, using a regularization parameter ($\alpha$), as,

$$L(U)=\alpha\left(\frac{1}{2n}\sum_{i,j=1}^{n}\left(L(\dot{u^i},\dot{u^j})-L(u^i,\ u^j)\right)^2\right) \tag{9}$$

Where L is the measure or metric of similarity of words representations. Here, for all similarity measures, both Euclidean norm and cosine similarity (dissimilarity) have been used. In this work, the embedding error using the Euclidean norm is used,

$$L(U)=\alpha\left(\frac{1}{2n}\sum_{i,j=1}^{n}\left(L_2(\dot{u^i},\dot{u^j})-L_2(u^i,\ u^j)\right)^2\right) \tag{10}$$

Alternatively, we could include the embedding error of the output sequence in equation (10). When the input sequence (or the dictionary) is too long, to prevent high computational complexity of computing similarity of each specific word with all other words, we choose a random (uniform) sample of the input sequences to compute the embedding error. The regularization parameter, $\alpha$, is computed using cross validation [26-30]. Alternatively, adaptive regularization parameters [81, 82] could be used.

### 2.3 Results and Experiments

This model was applied on Wikipedia datasets for English-German translation (one-way translation) with 1000 sentences. The idea was to determine which specific input word (in English) is the most important one for the corresponding German translation. The results were often an almost diagonal weight matrix, with few non-zero off diagonal entries, indicating the significance of the corresponding word(s) in the original language (English). Since the model is an unsupervised approach, it's hard to evaluate its performance without using domain knowledge. The next step in this work would be to develop a unified and interpretable metric for automatic testing and evaluation of the model without using any domain knowledge and also to apply the model to other applications such as sentiment analysis.

## References

1. Anger, G., Gorenflo, R., Jochum, H., Moritz, H., Webers, W. (eds.): Inverse Problems: principles and Applications in Geophysics, Technology, and Medicine. Akademic Verlag, Berlin (1993).
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: Attention Is All You Need, arXiv:1706.03762 (2017).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

3. Axelrod, A., He, X., and Gao, J. : Domain adaptation via pseudo in-domain data selection. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 355–362. Association for Computational Linguistics (2011).
4. Ba, Jimmy Lei, Mnih, Volodymyr, and Kavukcuoglu, Koray.: Multiple object recognition with visual attention. arXiv:1412.7755, December (2014).
5. Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, September (2014).
6. Baldi, Pierre and Sadowski, Peter.: The dropout learning algorithm. Artificial intelligence, 210:78–122 (2014).
7. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y.: Theano: new features and speed improvements. Deep Learning an Unsupervised Feature Learning NIPS 2012 Workshop (2012).
8. Becker, S., Bobin, J., Candès, E.J.: NESTA, a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. 4(1), 1–39 (2009).
9. Bengio, Y., Simard, P., and Frasconi, P.: Learning long-term de-pendencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166 (1994).
10. Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res., 3, 1137–1155 (2003).
11. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., WardeFarley, D., and Bengio, Y.: Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy). Oral Presentation (2010).
12. Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P.: Audio chord recognition with recurrent neural networks. In ISMIR, (2013).
13. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 20(4), 1956–1982 (2008).
14. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. 9, 717–772 (2008).
15. Candès, E.J.: Compressive sampling. In: Proceedings of the International Congress of Mathematicians, Madrid, Spain (2006).
16. Cheng, J., Li Dong, and Mirella Lapata: Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733 (2016).
17. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 2, 12, 13, 14 (2014).
18. D'Aspremont, A., El Ghaoui, L., Jordan, M.I., Lanckriet, G.R.G.: A direct formulation for sparse PCA using semidefinite programming. SIAM Rev. 49(3), 434–448 (2007).
19. Davies, A.R., Hassan, M.F.: Optimality in the regularization of ill-posed inverse problems. In: Sabatier, P.C. (ed.) Inverse Problems: An Interdisciplinary Study. Academic Press, London (1987).
20. DeMoor, B., Golub, G.H.: The restricted singular value decomposition: properties and applications. SIAM J. Matrix Anal. Appl. 12(3), 401–425 (1991).
21. Donoho, D.L., Tanner, J.: Sparse nonnegative solutions of underdetermined linear equations by linear programming. Proc. Natl. Acad. Sci. 102(27), 9446–9451 (2005).
22. Dozat, Timothy, Peng Qi, and Christopher D. Manning: Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task. In CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 20-30 (2017).
23. Dyer, Chris, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith: Recurrent neural network grammars. In Proc. of NAACL (2016).


ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

24. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. 32, 407–499 (2004).
25. Elden, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT 17, 134–145 (1977).
26. Elden, L.: A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems. BIT 24, 467–472 (1984).
27. Engl, H.W., Hanke, M., Neubauer, A.: Regularization methods for the stable solution of inverse problems. Surv. Math. Ind. 3, 71–143 (1993).
28. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Dordrecht (1996)..
29. Engl, H.W., Kunisch, K., Neubauer, A.: Convergence rates for Tikhonov regularisation of non-linear ill-posed problems. Inverse Prob. 5, 523–540 (1998)., H.W., Groetsch, C.W. (eds.): Inverse and Ill-Posed Problems. Academic Press, London (1987).
30. Gander, W.: On the linear least squares problem with a quadratic Constraint. Technical report STAN-CS-78–697, Stanford University (1978).
31. Gers, F., Schraudolph, N., & Schmidhuber, J.: Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, 3, 115–143 (2002).
32. Goldwater, Sharon , Dan Jurafsky, and Christopher D. Manning: Which words are hard to recognize? Prosodic, lexical, and disfluency factors that increase speech recognition error rates. Speech Communication 52: 181-200 (2010).
33. Golub, G.H., Van Loan, C.F.: Matrix Computations. Computer Assisted Mechanics and Engineering Sciences, 4th edn. Johns Hopkins University Press, US (2013).
34. Golub, G.H., Van Loan, C.F.: An analysis of the total least squares problem. SIAM J. Numer. Anal. 17, 883–893 (1980).
35. Golub, G.H., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. SIAM J. Numer. Anal. Ser. B 2, 205–224 (1965).
36. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21, 215–223 (1979).
37. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y.: Maxout networks. In Proceedings of The 30th International Conference on Machine Learning, pages 1319–1327 (2013).
38. Graves, A.: Sequence transduction with recurrent neural networks. In Proceedings of the 29th International Conference on Machine Learning, ICML (2012).
39. Graves, A., Fern´andez, S., & Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. Proceedings of the 2005 International Conference on Artificial Neural Networks. Warsaw, Poland (2005).
40. Graves, A., & Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18, 602–610 (2005).
41. Graves, A.: Generating sequences with recurrent neural networks. arXiv:1308.0850 [cs.NE], (2013).
42. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning; Data mining, Inference and Prediction. Springer, New York (2001).
43. Hastie, T.J., Tibshirani, R.: Handwritten Digit Recognition via Deformable Prototypes. AT&T Bell Laboratories Technical report (1994).
44. Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L., Botstein, D.: 'Gene Shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol. 1, 1–21 (2000).
45. Hastie, T., Mazumder, R.: Matrix Completion via Iterative Soft-Thresholded SVD (2015).
46. Hastie, T., Tibshirani, R., Narasimhan, B., Chu, G.: Package 'impute'. CRAN (2017).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

47. Hermann, K. and Blunsom, P.: Multilingual distributed representations without word alignment. In Proceedings of the Second International Conference on Learning Representations, ICLR, (2014).
48. Hirschberg, Julia and Christopher D. Manning: Advances in natural language processing. Science 349(6):261-266 (2015).
49. Hochreiter, Sepp and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, (1997).
50. Hochreiter, S., & Schmidhuber, J.: Long ShortTerm Memory: Neural Computation, 9, 1735–1780 (1997).
51. Hochreiter, Sepp, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, (2001).
52. Hofmann, B.: Regularization for Applied Inverse and Ill-Posed problems. Teubner, Stuttgart, Germany (1986).
53. Hudson, Drew A. and Christopher D. Manning: Compositional Attention Networks for Machine Reasoning. In International Conference on Learning Representations, ICLR (2018).
54. Jeffers, J.: Two case studies in the application of principal component. Appl. Stat. 16, 225–236 (1967).
55. Jolliffe, I.: Principal Component Analysis. Springer, New York (1986).
56. Jolliffe, I.T.: Rotation of principal components: choice of normalization constraints. J. Appl. Stat. 22, 29–35 (1995).
57. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the LASSO. J. Comput. Graph. Stat. 12(3), 531–547 (2003).
58. Kalchbrenner, N. and Blunsom, P.: Recurrent continuous translation models. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1700–1709. Association for Computational Linguistics. Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press, New York, NY,USA (2013).
59. Kim, Yoon, Carl Denton, Luong Hoang, and Alexander M.: Rush. Structured attention networks. In International Conference on Learning Representations (2017).
60. Koehn, P., Och, F. J., and Marcu, D.: Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics (2003).
61. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio: Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs/1406.1078 (2014).
62. Lafferty, J., McCallum, A., & Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. 18th International Conf. on Machine Learning (pp. 282–289). Morgan Kaufmann, San Francisco, CA (2001).
63. LeCun, Y., Bottou, L., Orr, G., & Muller, K.: Efficient backprop. Neural Networks: Tricks of the trade. Springer (1998).
64. Lin, Zhouhan, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017).
65. Luong, Hieu Pham, Christopher D. Manning: Effective Approaches to Attention-based Neural Machine Translation EMNLP (2015).
66. MacCartney Bill, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning: Learning to recognize features of valid textual entailments. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006), pp. 41-48 (2006).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

67. MacCartney Bill and Christopher D. Manning: Natural logic for textual inference. ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 193-200 (2007).
68. Manning Christopher D.: Ergativity. In Keith Brown, ed., Encyclopedia of Language & Linguistics, Second Edition, volume 4, pp. 210-217. Oxford: Elsevier (2006).
69. Manning Christopher D., Prabhakar Raghavan, Hinrich Schütze: Introduction to information retrieval. Computer Science (2008).
70. Manning, Christopher D, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, David McClosky: The Stanford CoreNLP Natural Language Processing Toolkit. Computer Science, ACL (2014).
71. Manning, Christopher D.: Computational Linguistics and Deep Learning. Computational Linguistics 41(4): 701-707 (2015).
72. McFarland, Daneil A., Daniel Ramage, Jason Chuang, Jeffrey Heer, and Christopher D. Manning, and Daniel Jurafsky: Differentiating Language Usage through Topic Models Poetics 41(6): 607-625 (2013).
73. Minh-Thang Luong, Hieu Pham, and Christopher D Manning: Effective approaches to attention based neural machine translation. arXiv preprint arXiv:1508.04025 (2015).
74. Modarresi, Kourosh : Application of DNN for Modern Data with two Examples: Recommender Systems & User Recognition. Deep Learning Summit, San Francisco, CA, Jan. 25-26 (2018).
75. Modarresi, Kourosh, Abdurrahman Munir: Standardization of Featureless Variables for Machine Learning Models Using Natural Language. Journal of Computational Science –Lecture Notes in Computer Science, vol 10861. Springer, Cham, pp 234-246 (2018).
76. Modarresi, Kourosh, Abdurrahman Munir: Generalized Variable Conversion Using K-means Clustering and Web Scraping. Journal of Computational Science –Lecture Notes in Computer Science, vol 10861. Springer, Cham, pp 247-258 (2018).
77. Modarresi, Kourosh, Jamie Diner: An Efficient Deep Learning Model for Recommender Systems., Journal of Computational Science, Lecture Notes in Computer Science, vol 10861. Springer, Cham, pp 221-233 (2018).
78. Modarresi, Kourosh: Effectiveness of Representation Learning for the Analysis of Human Behavior. American Mathematical Society, San Francisco State University, San Francisco, CA, Oct. 27 (2018).
79. Modarresi , Kourosh, Jamie Diner: An Evaluation Metric for Content Providing Models, Recommendation Systems, and Online Campaigns. Computational Science, Springer Lecture Notes in Computer Science (LNCS) Series, June (2019).
80. Modarresi, Kourosh: Combined Loss Function for Deep Convolutional Neural Networks. American Mathematical Society, University of California, Riverside, Riverside, CA, Nov. 9-10 (2019).
81. Modarresi, Kourosh and Gene H Golub: A Randomized Algorithm for the Selection of Regularization Parameter. Inverse Problem Symposium, Michigan State University, MI, June 11-12 (2007).
82. Modarresi:, Kourosh: A Local Regularization Method Using Multiple Regularization Levels. PhD. Thesis, Stanford University, Stanford, CA (2007).
83. Parikh, Ankur, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model: In Empirical Methods in Natural Language Processing (2016).
84. Pascanu, R., Mikolov, T., and Bengio, Y.: On the difficulty of training recurrent neural networks. In ICML (2013).
85. Pascanu, R., Mikolov, T., and Bengio, Y.: On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, ICML (2013).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20

86. Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y.: How to construct deep recurrent neural networks. In Proceedings of the Second International Conference on Learning Representations, ICLR (2014).
87. Schraudolph, N. N.: Fast Curvature MatrixVector Products for Second-Order Gradient Descent. Neural Comp., 14, 1723–1738, (2002).
88. Schuster, M., & Paliwal, K. K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45, 2673–2681 (1997).
89. Schwenk, H.: Continuous space translation models for phrase-based statistical machine translation. In M. Kay and C. Boitet, editors, Proceedings of the 24th International Conference on Computational Linguistics (COLIN), pages 1071–1080. Indian Institute of Technology Bombay (2012).
90. Schwenk, H., Dchelotte, D., and Gauvain, J.-L.: Continuous space language models for statistical machine translation. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 723–730. Association for Computational Linguistics (2006).
91. Sutskever, I., Vinyals, O., and Le, Q.: Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems , NIPS (2014).
92. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016).
93. Zeiler, M. D.: ADADELTA: An adaptive learning rate method. arXiv:1212.5701 (2012).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_20