Bidirectional Attentional LSTM for Aspect Based Sentiment Analysis on Italian

Giancarlo Nicola
University of Pavia
giancarlo.nicola01@universitadipavia.it

Abstract

English. This paper describes the SentITA system that participated to the ABSITA task proposed in Evalita 2018. The system is based on a Bidirectional Long Short Term Memory network with attention that exploits word embeddings and sentiment specific polarity embeddings. The model also leverages grammatical information from POS tagging and NER tagging. The system participated in both the Aspect Category Detection (ACD) and Aspect Category Polarity (ACP) tasks achieving the 5^{th} place in the ACD task and the 2^{nd} in the ACD task.

Italiano. *Questo paper descrive il sistema SentITA valutato nel task ABSITA proposto all'interno di Evalita 2018. Il sistema è basato su una rete nuerale ricorrente con celle di memoria di tipo Long Short Term Memory e con implementato un meccanismo d'attenzione. Il modello sfrutta sia word embeddings generali sia polarity embeddings specifici per la sentiment analysis ed inoltre fa uso delle informazioni derivanti dal POS-tagging e dal NER-tagging. Il sistema ha partecipato sia nella sfida di Aspect Category Detection (ACD) sia in quella di Aspect Category Polarity (ACP) posizionandosi al quinto posto nella prima e al secondo posto nella seconda.*

1 Introduction

This paper describes the SentITA system that participated to the ABSITA task (Basile et al. 2018) proposed in Evalita 2018. In ABSITA the task consists in performing Aspect Based Sentiment Analysis (ABSA) on self-reliant sentences scraped from the "booking.com" website. The aspects are related to the accommodation reviews and comprehend topics like cleanliness, comfort, location, etc. The task is divided in two subtasks Aspect Category Detection (ACD) and Aspect Category Polarity (ACP). The fist, ACD consists in identifying the aspects mentioned in the sentence, while the second requires to associate a sentiment polarity label to the aspects evoked in the sentence. Both the tasks are addressed with the same architecture and the same data preprocessing. The system is based on a deep learning model, a Bidirectional Long Short Term Memory network with attention. The model exploits word embeddings, sentiment specific polarity embeddings and it leverages also grammatical and information from POS tagging and NER tagging.

Recently, deep learning has emerged as a powerful machine learning technique achieving state-of-the-art results in many application domains, including sentiment analysis. Among the deep learning frameworks applied to sentiment analysis, many employ a combination of semantic vector representations (Mikolov et al. 2013), (Pennington et al. 2014) and different deep learning architectures. Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997), (Socher et al. 2013), (Cho et al. 2014) have been applied to model complex and long term non-local relationships in both word level and character level text sequences. Recursive Neural Tensor Networks (RNTN) have shown great results for semantic compositionality (Socher et al. 2011), (Socher et al. 2013) and also convolutional networks (CNN) for both sentiment analysis (Collobert et al 2011) and sentence modelling (Kalchbrenner et al. 2014) have performed better than previous state of the art methodologies. All these methods in most of the applications receive in input a vector representation of words called word embeddings. (Mikolov 2012), (Mikolov et

al. 2013) and (Pennignton et al. 2014), further expanding the work on word embeddings (Bengio et al 2003), that grounds on the idea of distributed representations for symbols (Hinton et al 1986), have introduced unsupervised learning methods to create dense multidimensional spaces where words are represented by vectors. The position of such vectors is related to their semantic meaning and grammatical properties and they are widely used in all modern NLP. In fact, they allow for a dimensionality reduction compared to traditional sparse vectors space models and they are often used as pre-trained initialization for the first embedding layers of the neural networks in NLP tasks. In (Le and Mikolov 2014), expanding the previous work on word embeddings, is developed a model capable of representing also sentences in a dense multidimensional space. In this case too, sentences are represented by vectors whose position is related to the semantic content of the sentence with similar sentences represented by vectors that are close to each other.

When working with isolated and short sentences, often with a specific writing style, like tweets or phrases extracted from internet reviews many long term text dependencies are lost and not exploitable. In this situation it is important that the model learns both to pay attention to specific words that have key roles in determining the sentence polarity like negations, magnifiers, adjectives and to model the discourse but with less focus on long term dependencies (due to the text brevity). For this reason, deep learning word embedding based models augmented with task specific gazettes (dictionaries) and features, represent a solid baseline when working with these kind of datasets (Nakov et al. 2016)(Attardi et al. 2016)(Castellucci et al. 2016)(Cimino et al. 2016)(Deriu et al. 2016). In this system, a polarity dictionary for Italian has been included as feature to the model in addition to the word embeddings. Moreover every sentence during preprocessing is augmented with its NER tags and POS tags which then are used as features in the model. Thanks to the inclusion of these features relevant for the considered task in combination with word embeddings and an attentional bidirectional LSTM recurrent neural network, the model achieves useful results with some thousands labelled examples.

The remainder of the paper presents the model and the experiments on the ABSITA task. in Section 2 the model and its features are explained; in Section 3 the model training and its performances are discussed; in Section 4 a conclusion with the next improvement of the model is given.

2 Description of the system

The model implemented is an Attentional Bidirectional Recurrent Neural Network with LSTM cells. It operates at word level and therefore each sentence is represented as a sequence of words representations that are sequentially fed to the model one after another until the sequence has been entirely used up. One sentence sequence coupled with its polarity scores represent a single datapoint for the model.

The input to the model are sentences, represented as sequence of word representations. The maximum sequence length has been set to 35, with shorter sentences left-padded to this length and longer sentences cut to this length. Each word of the sequence is represented by five vectors corresponding to 5 different features that are: high dimensional word embedding, word polarity, word NER tag, word POS tag, custom low dimensional word embedding. The high dimensional word embeddings are the pretrained Fastext embeddings for Italian (Grave et al. 2018). They are 300-dimensional vectors obtained using the skip-gram model described in (Bojanowski et al. 2016) with default parameters. The word polarity is obtained from the OpeNER Sentiment Lexicon Italian (Russo et al. 2016). This freely available Italian Sentiment Lexicon contains a total of 24.293 lexical entries annotated for positive/negative/neutral polarity. It was semi-automatically developed using a propagation algorithm starting from a list of seed key-words and manually reviewing the most frequent.

Both the NER tags and POS tags are obtained from the Spacy library Tagger model for Italian (Spacy 2.0.11 - https://spacy.io/). The custom low dimensional word embeddings are generated by random initialization and are inserted to provide an embedding representation of the words that are missing from the Fastext embeddings, which otherwise would all be represented by the same out of vocabulary token (OOV token). In general, it could be possible to train and fine-tune these custom embeddings on specific datasets to let the model learn the usage of words in specific cases. The information extracted from the OpeNER Sen-

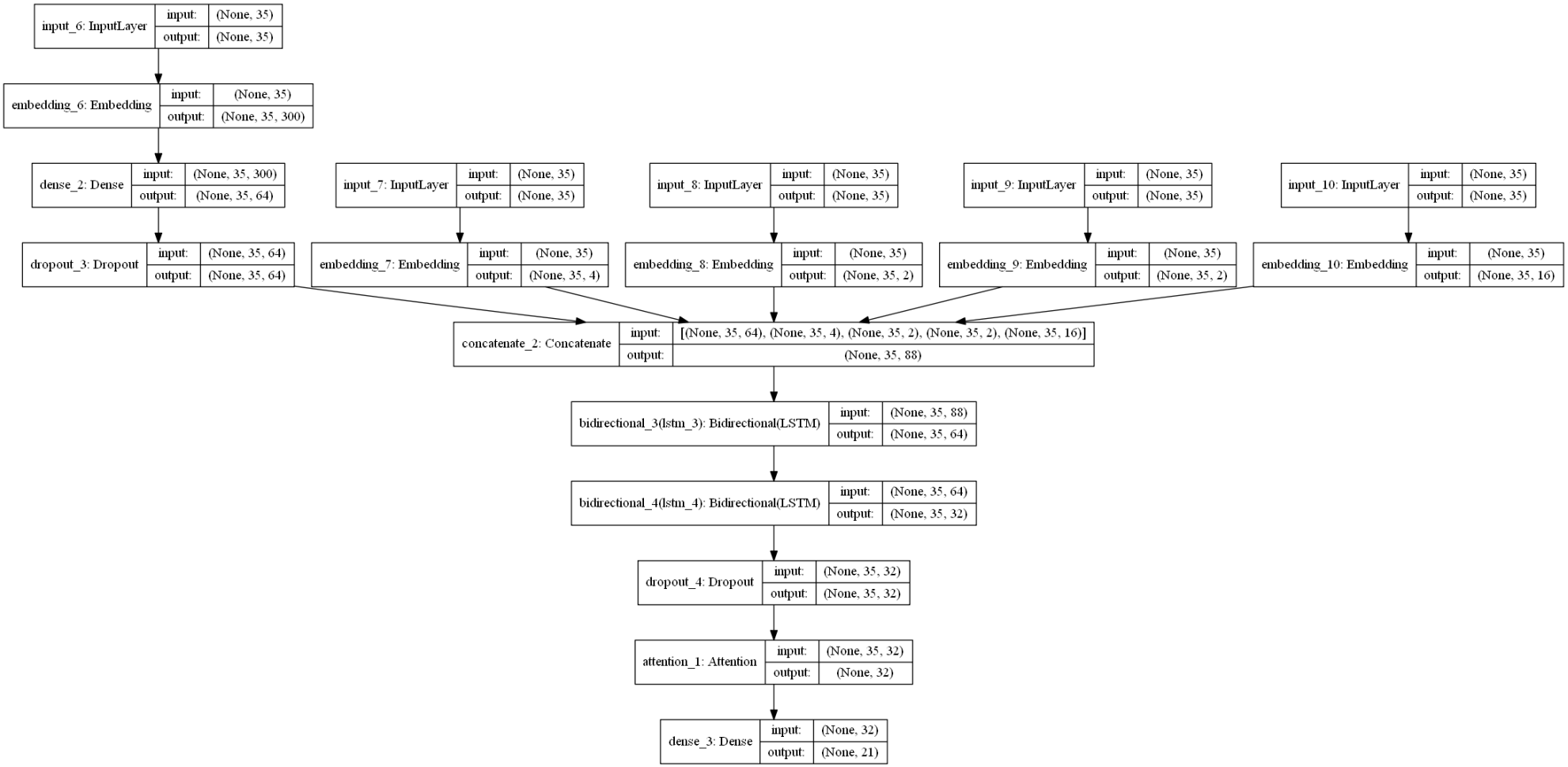

Figure 1: Model architecture

timent Lexicon Italian are the word polarity with its confidence and they are concatenated in a vector of length 2 that is one of the input to the first layer of the network. The NER tags and POS tags instead are mapped to randomly initialized embeddings of dimensionality respectively 2 and 4 that are not trained during the model training for the task submission. With more data available it would probably be beneficial to train all the NER, POS and custom embeddings but for this specific dataset the results were comparable and slightly better when not training the embeddings.

The model, whose architecture is schematized in fig. 1, performs in its initial layer a dimensionality reduction on the Fastext embeddings and then concatenates them with the rest of the embeddings (polarity, NER tag, POS tag, and custom word embeddings) for each each timestep (word) of the sequence. The concatenation of the embeddings is fed in a sequence of two bidirectional recurrent layers with LSTM cells. The result of these recurrent layers is passed to the attention mechanism presented in (Raffel et al. 2016) and finally to the dense layers that outputs the aspect detection and aspect polarity signals. The attention mechanism in this formulation, produces a fixed-length embedding of the input sequence by computing an adaptive weighted average of the sequence of states (normally denoted as "h") of the RNN. This form of integration is similar to the "global temporal pooling" described in (Sander 2014), which is based on the "global average pooling" technique of (Min et al. 2014). The non linear activations used in the model are Rectified Linear Units (ReLU) for the internal dense layers, hyperbolic tangent (tanh) in the recurrent layers and sigmoid in the output dense layer. In order to contrast overfitting the dropout mechanism has been used after the Fastext embedding dimensionality reduction with rate 0.5, in both the recurrent layers between each timestep with rate 0.5 and on the output of the recurrent layers with rate 0.3.

The model has 61,368 trainable parameters and a total of 45,233,366 parameters, the majority of them representing the Fastext embedding matrix (45,000,300). Compared to many NLP models used today the number of trainable parameters is quite small to reduce the possibility of overfitting the training dataset (6,337 examples is small compared to many English sentiment datasets) and also because is compensated by the addition of engineered features like polarity dictionary, NER tag and POS tag that help in classifying the examples.

3 Training and results

The only preprocessing applied to the text is the conversion of each character to its lower case form. Then, the vocabulary of the model is limited to the first 150,000 words of the Fastext embeddings trough a cap on the max number of embeddings, due to memory constraints of the GPU used for training the model. The Fastext embeddings are sorted by descending frequency of appearance in their training corpus, thus the vocabulary comprises the 150,000 most frequent words in Italian. The other words that remain out of this cut are represented in the model high dimensional embeddings (Fastext embeddings) by an out of vocabulary token. However, all the training set words are anyhow included in the custom low dimensional word embeddings; this is done since both our training text and in general users text (specially when working with reviews, tweets, social network platforms) could be quite different from the one on which Fastext embeddings are trained. In addition the NER-tagging and POS-tagging models for Italian included in the Spacy library are applied to the text to compute the additional NER-tags and POS-tags features for each word.

To train the model and generate the challenge submission a k-fold cross validation strategy has been applied. The dataset has been divided in 5 folds and 5 different instantiations of the same model (with the same architecture) have been trained picking each time a different fold as validation set (20%) and the remaining 4 folds as training set (80%). The number of training epochs is defined with the early stopping technique with patience parameter equal to 7. Once the training epochs are completed, the model snapshot that achieved the best validation loss is loaded. At the end the predictions from the 5 models have been averaged together and thresholded at 0.5. The training of five different instantiations of the same model and the averaging of their predictions overcomes the fact that in each k^{th}-fold the model selection based on the best validation loss is biased on the validation fold itself.

Each of the five models is trained minimizing the crossentropy loss on the different classes with the Nesterov Adam (Nadam) optimizer (Dozat

Ranking	Micro Precision	Micro Recall	Micro F1-score
1	0.8397	0.7837	0.8108
2	0.8713	0.7504	0.8063
3	0.8697	0.7481	0.8043
4	0.8626	0.7519	0.8035
5	0.8819	0.7378	0.8035
6	0.898	0.6937	0.7827
7	0.8658	0.697	0.7723
8	0.7902	0.7181	0.7524
9	0.6232	0.6093	0.6162
10	0.6164	0.6134	0.6149
11	0.5443	0.5418	0.5431
12	0.6213	0.433	0.5104
baseline	0.4111	0.2866	0.3377

Table 1: Task ACD (Aspect Category Detection) ranking. This system score is reported between dashed lines

2015) with default parameters ($\lambda = 0.002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, schedule_decay $= 0.004$). The Nesterov Adam optimizer is similar to the Adam optimizer (Kingma et al. 2014) but were momentum is replaced with Nesterov momentum (Nesterov 1983). Adam in fact, combines two algorithms known to work well for different reasons: momentum, which points the model in a better direction, and RMSProp, which adapts how far the model goes in that direction on a per-parameter basis. However, Nesterov momentum which can be viewed as a simple modification of the former, increases stability, and can sometimes provide a distinct improvement in performance, superior to momentum (Sutskever et al. 2013). For this reason the two approaches are combined in the Nadam optimizer.

This system obtained the 5^{th} place in the ACD and the 2^{nd} place in the ACP task as reported respectively in Table 1 and Table 2. In these tables the performances of the systems participating to the challenge have been ranked by F1-score from the task organizers. In particular, it is interesting the second place in the ACP since the model is more oriented towards polarity classification for which it has specific dictionaries more than aspect detection. This is confirmed also from the high precision score obtained from the model in the ACP task, the 2^{nd} highest among the participating systems.

4 Discussion

The results obtained by the SentITA system at ABSITA 2018 are promising, as the system placed 2^{nd} in the ACP and 5^{th} in the ACD task but not very far from the 1^{st} in terms of F1-score. The model in general shows a high precision but in general a lower recall compared to the other systems. The proposed architecture makes use of different features that is easy to obtain through other models like POS and NER tags, polarity and word embeddings, for this reason, the human effort in the data preprocessing is very limited. One important direction to further improve the model would be to rely more on unsupervised learning, which at the moment is used only for the word embeddings. It could be possible to integrate in the model features based on language models or encoder-decoder networks, for example. More unsupervised learning would better ensure the model generalization to cover most of the argument and lexical content of the Italian language due to the large quantity of text available and thus improving also the model recall.

Ranking	Micro Precision	Micro Recall	Micro F1-score
1	0.8264	0.7161	0.7673
2	0.8612	0.6562	0.7449
3	0.7472	0.7186	0.7326
4	0.7387	0.7206	0.7295
5	0.8735	0.5649	0.6861
6	0.6869	0.5409	0.6052
7	0.4123	0.3125	0.3555
8	0.5452	0.2511	0.3439
baseline	0.2451	0.1681	0.1994

Table 2: Task ACP (Aspect Category Polarity) ranking. This system score is reported between dashed lines

References

Giuseppe Attardi, Daniele Sartiano, Chiara Alzetta, Federica Semplici. 2016. Convolutional Neural Networks for Sentiment Analysis on Italian Tweets. CLiC-it/EVALITA (2016).

Pierpaolo Basile and Valerio Basile and Danilo Croce and Marco Polignano. 2018. Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA). Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), CEUR.org, Turin

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin (2003) A neural probabilistic language model. The Journal of Machine Learning Research, 3:1137–1155, 2003.

P. Bojanowski, E. Grave, A. Joulin, T. Mikolov (2016) Enriching Word Vectors with Subword Information. arXiv:1607.04606v2

Giuseppe Castellucci, Danilo Croce, Roberto Basili. 2016. Context–aware Convolutional Neural Networks for Twitter Sentiment Analysis in Italian. CLiC-it/EVALITA (2016).

K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, 2014.

Andrea Cimino, Felice Dell'Orletta. 2016. Tandem LSTM–SVM Approach for Sentiment Analysis. Castellucci, Giuseppe et al. CLiC-it/EVALITA (2016).

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12:2493- 2537, 2011.

Jan Deriu, Mark Cieliebak. 2016. Sentiment Detection using Convolutional Neural Networks with Multi–Task Training and Distant Supervision. CLiC-it/EVALITA (2016).

Timothy Dozat (2015) Incorporating Nesterov Momentum into Adam. http://cs229.stanford.edu/proj2015/054_report.pdf.

E. Grave*, P. Bojanowski*, P. Gupta, A. Joulin, T. Mikolov (2018) Learning Word Vectors for 157 Languages. Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)

G. E. Hinton, J. L. McClelland, and D. E. Rumelhart (1986) Distributed representations. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. 1986. Volume 1: Foundations, MIT Press, Cambridge, MA. pp 77-109.

S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. Neural Computation 9(8):1735-1780, 1997

N. Kalchbrenner, E. Grefenstette, P. Blunsom. (2014) A Convolutional Neural Network for Modelling Sentences. In Proceedings of ACL 2014.

Kingma, Diederik and Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations. https://arxiv.org/pdf/1412.6980.pdf

Q. Le, T. Mikolov. Distributed Representations of Sentences and Documents. Proceedings of the 31 st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP, volume 32.

T. Mikolov. (2012) Statistical Language Models Based on Neural Networks. PhD thesis, PhD Thesis, Brno University of Technology, 2012.

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

Min Lin, Qiang Chen, and Shuicheng Yen. Network in network. arXiv preprint arXiv:1312.4400, 2014.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. Proceedings of SemEval-2016, pages 1–18, San Diego, California, June 16-17, 2016.

Y. Nesterov (1983) A method of solving a convex programming problem with convergence rate o (1/k2). In Soviet Mathematics Doklady, volume 27, pages 372-376, 1983.

J. Pennington, R. Socher, and C. 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, Doha, Qatar, October. Association for Computational Linguistics.

Colin Raffel, Daniel P. W. Ellis (2016) Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. https://arxiv.org/abs/1512.08756

Russo, Irene; Frontini, Francesca and Quochi, Valeria, 2016, OpeNER Sentiment Lexicon Italian - LMF, ILC-CNR for CLARIN-IT repository hosted at Institute for Computational Linguistics "A. Zampolli", National Research Council, in Pisa, http://hdl.handle.net/20.500.11752/ILC-73.

Sander Dieleman. Recommending music on Spotify with deep learning. http://benanne.github.io/2014/08/05/spotify-cnns.html, 2014.

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and Christopher D. Manning. (2011) Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP).

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. 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, Stroudsburg, PA, October. Association for Computational Linguistics.

Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton (2013) Proceedings of the 30th International Conference on Machine Learning, PMLR 28(3):1139-1147, 2013.