International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:9, 2020

Deep Learning Based, End-to-End Metaphor Detection in Greek with Recurrent and Convolutional Neural Networks

Konstantinos Perifanos, Eirini Florou, Dionysis Goutsos

Abstract—This paper presents and benchmarks a number of end-to-end Deep Learning based models for metaphor detection in Greek. We combine Convolutional Neural Networks and Recurrent Neural Networks with representation learning to bear on the metaphor detection problem for the Greek language. The models presented achieve exceptional accuracy scores, significantly improving the previous state-of-the-art results, which had already achieved accuracy 0.82. Furthermore, no special preprocessing, feature engineering or linguistic knowledge is used in this work. The methods presented achieve accuracy of 0.92 and F-score 0.92 with Convolutional Neural Networks (CNNs) and bidirectional Long Short Term Memory networks (LSTMs). Comparable results of 0.91 accuracy and 0.91 F-score are also achieved with bidirectional Gated Recurrent Units (GRUs) and Convolutional Recurrent Neural Nets (CRNNs). The models are trained and evaluated only on the basis of training tuples, the related sentences and their labels. The outcome is a state-of-the-art collection of metaphor detection models, trained on limited labelled resources, which can be extended to other languages and similar tasks.

Keywords—Metaphor detection, deep learning, representation learning, embeddings.

I. Introduction

METAPHOR as a figure of speech has a widespread presence in any form of communication, either oral or written. According to Steen [1] data analysis suggests that, on average, one in every seven and a half lexical units in a corpus is related to metaphor. However, it is difficult to clearly define the boundaries that separate metaphorical from literal uses, as well as metaphor from other figures of speech.

The difficulty of clearly establishing a theoretical background for metaphor justifies the variety of NLP systems that aim at automatically distinguishing between metaphorical and literal meanings of a word or a phrase. This difficulty is further exacerbated if we take into account the limitations of Greek as regards resources and tools for metaphor detection. Thus, we can conclude that the development of neural language models is necessary for the automatic differentiation between the literal and the metaphorical meaning of phrases that are part of an authentic and non-annotated Greek corpus. For these reasons, our attempt to identify metaphors here is based on the principles of distributional semantics which focus on determining the relations of a word with its linguistic context and grouping semantic similarities between linguistic items based on distributional properties rather than connections of a certain term with its related concepts. Distributional semantics have been paramount in shifting research interest towards neural language models, which can attribute hidden statistical characteristics of distributed representations of word sequences in natural language. Therefore, a serious problem such as the automatic detection of metaphors and their differentiation from literal uses can be dealt with the development of neural language models.

Konstantinos Perifanos, Eirini Florou, and Dionysis Goutsos are with the Department of Linguistics, National and Kapodistrian University of Athens, Greece (e-mail: kperifanos@phil.uoa.gr, eirini.florou@gmail.com, dgoutsos@phil.uoa.gr).

II. Previous Work

The computational identification and interpretation of metaphors have been based on a variety of computational tools such as statistical models [2], word taxonomies [3], clustering [4], logistic regression [5], [6] or generative statistical models such as Latent Dirichlet Allocation (LDA) [7]. As has happened with many linguistic phenomena, computational approaches to metaphor are now based on neural models and take advantage of the benefits of representation learning [8], and more specifically distributed representations, also known as word embeddings [9], [10]. The neural models for metaphor detection include Long Short Term Memory (LSTMs) and Conditional Random Fields (CRFs) [11], which perform better with the contribution of linguistic features like the Wordnet, POS tags or clustering.

The omnipresence of metaphor in all types of Greek texts initially guided our research interest to an alternative approach to automatic metaphor detection, following the principles of distributional semantics and without the requirement of access to linguistic resources and tools or expensive and time-consuming manual annotation. This approach has been based on neural language models and has taken into account the context of each term in order to identify its function and uses without explicitly employing any connections between a word and its related concepts. Neural language models offer the opportunity to a language which is poor in linguistic resources and tools to overpass the problem of calculating the semantic relevance between phrases. By taking advantage of the benefits of distributional semantics we substituted the semantic comparison of terms with a numerical comparison of their distributional representation in vector space. By this comparison we were able to identify the literal or metaphorical function of words in a specific context. This first approach of metaphor detection in Greek texts is our baseline for this paper. Specifically, we aim at improving the procedure of metaphor detection and for this reason we take into account state-of-the-art Deep Learning based models such as Convolutional and Recurrent Neural Networks in order to achieve the prediction of metaphoricity for each word in running text.

III. Deep Learning for Text Classification

Recent advances in Neural Networks and Transfer learning have been successfully applied to Natural Language Processing. More specifically, Convolutional Neural Networks (CNNs, ConvNets) and Recurrent Neural Network (RNNs) architectures have been applied to text classification problems, such as Named Entity Recognition, Part-of-Speech tagging, Semantic Role Labelling etc. [12], [13], [14], [15]

Training models with RNNs and CNNs from scratch typically requires a vast amount of labelled data, which is generally a time-consuming and expensive process. We tackle this by using transfer learning, and more specifically by using pre-trained word embeddings, and by allowing the model to fine-tune the first layer of the network (the embedding layer) as part of the training process. The term *embeddings* refers to compact, continuous representations of words in a D-dimensional space and has emerged from representation learning [8]. Based on this compact representation, we can measure the semantic similarity of words by using geometrical properties of the word vector representation, typically the cosine distance between word vectors.

Continuing on the work of [16], we use fastText [17] embeddings trained in the Corpus of Greek Texts [18].

FastText[1], as described in [17] is an efficient library for representation learning and text classification. Similar to word2vec [19], it produces word embeddings by training a neural language model that is trying to predict words, given a certain context (CBOW architecture) or context given words (SkipGram architecture). As in word2vec, fastText operates as a neural language model. The key difference with fastText, however, is that it is taking into account morphology in the form of ngram representations. The representation of a word is calculated as the sum of the embeddings of its ngrams. The ability of fastText to capture morphological information in the produced representations seems to be more efficient, compared to other models in downstream tasks such as text classification.

IV. Data and Methodology

We trained fastText embeddings on the Corpus of Greek Texts [18], for dimensions ranging from $D = 50$ to $D = 500$, in steps of 50. The Corpus of Greek Texts consists of approximately 28 million words, a reasonable corpus size to produce meaningful embeddings, able to capture semantic similarity.

FastText is using sub-word information to learn distributed word representation and empirically performs better in downstream NLP tasks compared to word2vec [19] or Glo.Ve. [20]. It also naturally tackles the problem of spelling errors, as word-level embeddings are essentially averages of n-gram level embeddings, while words with simple spelling errors still produce very similar embeddings to the intended word.[2]

The metaphor training set consists of 1145 labelled sentences, 563 metaphoric and 582 literal ones. The median length of words in the training set is 12, the minimum number of words is 2 and the maximum is 225.

To customize the training set we distinguished between literal and metaphorical phrases according to the Metaphor Identification Procedure (MIP) as suggested by the Pragglejaz Group [21]. Based on MIP, we created two lists of phrases, one with literal and one with metaphorical ones, from the Corpus of Greek Texts. Both lists had the same verbs as a kernel but each could take various objects as predicates. Our training set included a few cases of intransitive verbs but did not include collocations, auxiliary, linking, modal or delexical verbs. Furthermore, the phrases of our training corpus did not have any metaphor markers, which could signal the metaphorical use of a term. Finally, it must be noted that in many cases the metaphorical interpretation is based on the comparison between a human activity and the implementation of the same activity by a non-human.

In [16] the classification of phrases into literal and metaphorical is performed by locating the verb in the sentence and averaging the embeddings of a small, fixed-size window centred on the verb, to produce a fixed size input vector for the machine learning algorithm. The averaged context representation is then fed to a Support Vector Machine, which results to 0.83 classification accuracy. The idea of averaging context embeddings in small window sizes comes from [19] and the window size is empirically determined.

Here, we extend the fixed-size contextual representation by passing the entire sentence to the classifier. The classifier then models the probability of a sentence being a metaphor, e.g. $p(label = metaphor|sentence)$ and we accordingly optimize the model.

Both CNNs and RNNs are utilizing the learned (or fine-tuned) representations of all words in a sentence. This is done by the convolution operator in CNNs and the hidden states in LSTM and GRU reccurent neural networks. Eventually, in both cases, a representation of all words in the sentence is passed to a fully connected layer of the classifier. This improves classification quality, whereas in the simple window-based averaging method contextual information is distorted for context size larger than 3 or 4 words.

We evaluate all our models with 10-fold cross validation and we report average accuracy and f1-score.

A. CNN Architecture

The CNN architecture is based on the work of [12]. More specifically, we use kernel heights with sizes $k = \{3, 4, 5\}$ and out channel size of 32. The convolution channels are then max-pooled, concatenated and passed into a fully connected layer. The network is regularized to prevent overfitting by using dropout [22], e.g. dropping units from the network to prevent overfitting, with dropout probability 0.5.

[1] fastText, https://fasttext.cc/

[2] fastText embeddings for Greek can be downloaded at http://sek.edu.gr

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:9, 2020

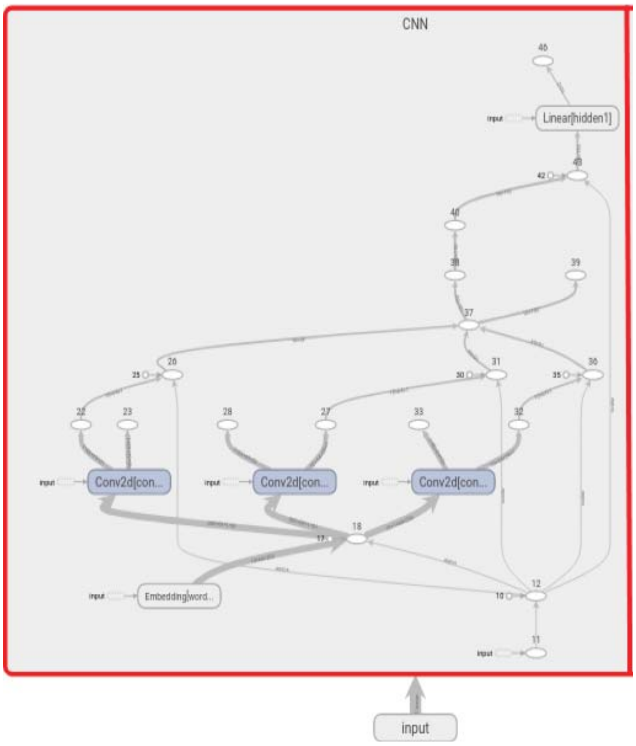

Fig. 1 CNN architecture

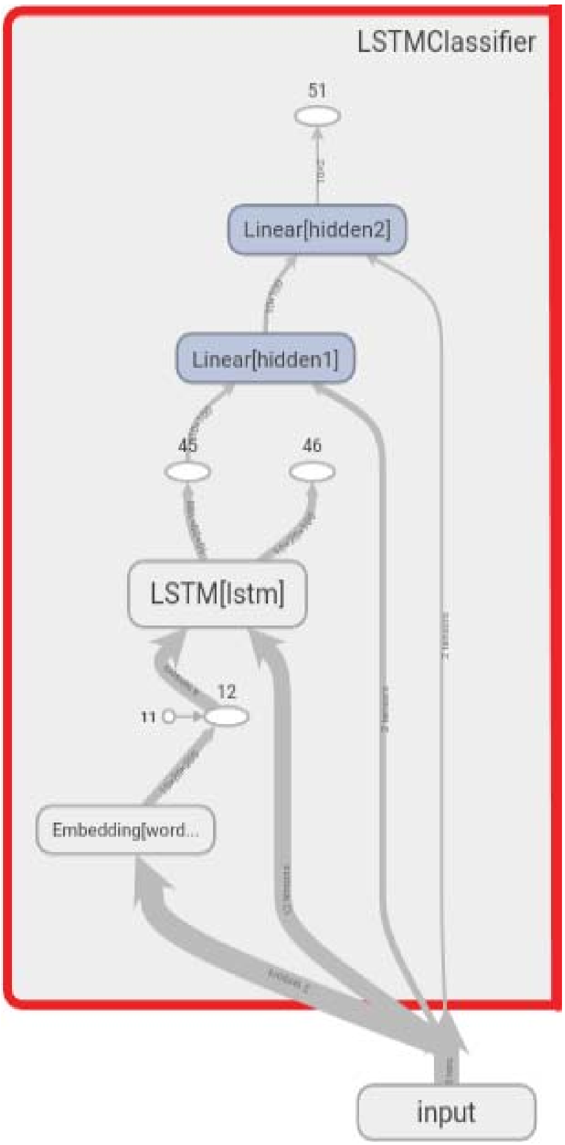

Fig. 2 LSTM architecture

B. RNN Architecture

In our experiments we tested Gated Recurrent Units (GRUs, [23]) and Long Short Term Memory architectures (LSTMs, [24]), using both unidirectional and bidirectional [25] architectures.

Bidirectional recurrent neural networks are essentially trained on the same sequence of data in forward and backward directions simultaneously and so the output state at every step encodes information about the past (forward direction) and the future (backward direction).

The architecture is exactly the same in both GRU and LSTM configurations, with the recurrence mechanism as the only difference. We feed a fixed size, zero padded sentence into the network, followed by the recurrence unit. We then apply 1-max-pooling[3] over the intermediate hidden layers, followed by a fully connected layer of 100 units and finally the output sigmoid unit.

C. CRNN

Finally, we also evaluated a combination of Convolutional Neural Nets and Recurrent Neural Nets, and more specifically the architecture described in [27]. Here, the architecture utilises recurrent structure to capture contextual information as far as possible when learning word representations, followed by a max-pooling layer. Essentially, max-pooling determines the most significant words in the underlying text classification problem. Bi-directional architectures consistently outperform uni-directional so we omit results. This is in agreement with [28].

All network architectures presented in this paper are optimised by the Adam optimizer [29] under the Maximum Likelihood principle and Negative Log Likelihood as loss function. The implementation is based in PyTorch [30]. The results of the experiments are summarized in table 1

V. Discussion

We presented a collection of state-of-the art metaphor detection models achieving accuracy higher than 90% for the Greek language. This extends the work of [16] and, to the best

[3]In our experiments we also tried average pooling, with good but inferior results compared to max-pooling, a result consistent with [26]

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:9, 2020

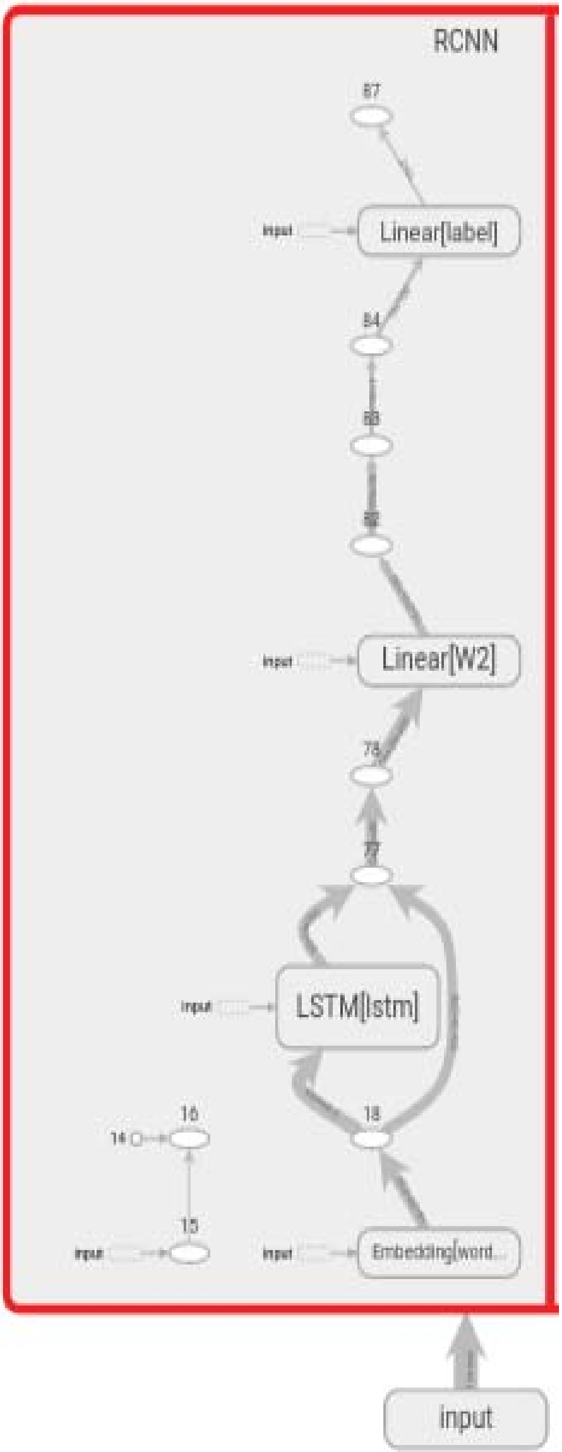

Fig. 3 RCNN architecture

of our knowledge, sets a new state-of-the-art for metaphor detection in Greek, dealing simultaneously with the lack of linguistic resources for the language. We aim at continuing our work by exploring the performance of contextual embeddings such as ELMO [31] and BERT [32]. Another recent promising direction, especially for small datasets, is Graph Neural Networks (GNNs) [33], [34]. In this specific variation of graph neural networks, the entire training set is represented as a graph $G = (V, E)$ and the task of the model is node representation and classification, even with potentially few training examples. This is achieved by exploiting the graph structure and the representation of adjacent nodes in the graph.

TABLE 1
EXPERIMENT RESULTS, WITH MODEL ARCHITECTURE AND FASTTEXT EMBEDDING DIMENTIONALITY

Results		
Model	Accuracy	F1-score
Florou et.al. 2018	0.83	0.83
CNN ($D = 500$)	0.90	0.89
CNN, fine-tuning ($D = 150$)	**0.92**	**0.92**
b-LSTM ($D = 350$)	0.90	0.91
b-LSTM, fine-tuning ($D = 200$)	**0.92**	**0.92**
b-GRU, fine-tuning ($D = 450$)	0.91	0.91
b-GRU ($D = 200$)	0.86	0.83
CRNN, fine-tuning ($D = 450$)	0.91	0.91
CRNN, ($D = 450$)	0.90	0.91

Both CNNs and bi-directional LSTMs with fine-tuning achieve accuracy higher than 90%. If we disable fine-tuning, classification accuracy is still high, although overall fine-tuning appears to consistently outperform non-fine-tuning configurations, which is also consistent with the results presented in [28].

There are several factors that can explain the performance achieved with neural networks. First, the full sentence is passed into the classifier and thus the model can benefit by exploiting potential long-term semantic dependencies. These dependencies are captured by the LSTM cells and the convolutional operators. Additionally, in the case of LSTMs and GRUs, bidirectional architectures appear to consistently outperform unidirectional architectures.

Finally, transfer learning, in the form of pre-trained embeddings such as fastText is extremely useful in the sense that the learned representations capture semantic properties of words in an unsupervised learning fashion; we also allow fine-tuning, which is proven to further enhance the accuracy of models [35]. Fast-text's ability to implicitly utilize morphological structure in the form of sub-word representations is also proven to help the overall downstream architecture to significantly improve. We conjecture that this property holds in languages with a rich morphological structure like Greek.

Keeping in mind that it is possible to distinguish between different kinds of metaphor and even between levels of metaphoricity of a term in a sentence, our effort is solely aimed at distinguishing between the literal and the metaphorical use of a term in a specific linguistic context. In this regard, we have not checked at all whether neural language models have the appropriate properties in order to discriminate pure metaphor from other kinds of figurative speech such as personification, metonymy, synecdoche etc. In addition, our approach to metaphor detection is not able to classify metaphorical phrases into categories like direct and indirect, or implied and extended. Of course, such an endeavour constitutes a particularly interesting and demanding research challenge, even though the main goal of our specific approach is metaphor detection and its discrimination from literal cases by the use of machine learning algorithms.

REFERENCES

[1] Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna A Kaal, and Tina Krennmayr. Metaphor in usage. *Cognitive Linguistics*, 21(4):765–796, 2010.
[2] M. Schulder and E. D. Hovy. Metaphor detection through term relevance. In *Proceedings of the Second Workshop on Metaphor in NLP. Association for Computational Linguistics*, pages 18–26, Baltimore, MD, USA, 2014.
[3] T. B. Sardinha. *Metaphor probabilities in corpora. In Zanotto, Mara Sophia, Cameron, Lynne and Cavalcanti, Marilda do Couto (eds.)* Confronting metaphor in use. John Benjamins, Amsterdam/Philadelphia, 2008.
[4] J. Birke and A. Sarkar. A clustering approach for the nearly unsupervised recognition of nonliteral language. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 329–336, Trento, Italy, 2006.
[5] J. E. Dunn. Evaluating the premises and results of four metaphor identification systems. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 2 (CICLing13)*, pages 471–486, Samos, Greece, 2013.
[6] Jonathan Dunn. Measuring metaphoricity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 745–751, 2014.
[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
[9] Maximilian Köper, Evgeny Kim, and Roman Klinger. IMS at EmoInt-2017: Emotion intensity prediction with affective norms, automatically extended resources and deep learning. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–57, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
[10] Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. Grasping the finer point: A supervised similarity network for metaphor detection. *arXiv preprint arXiv:1709.00575*, 2017.
[11] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
[12] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
[13] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.
[14] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2428–2437, 2016.
[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.
[16] Eirini Florou, Konstantinos Perifanos, and Dionysis Goutsos. Neural embeddings for metaphor detection in a corpus of greek texts. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–4. IEEE, 2018.
[17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
[18] Dionysis Goutsos. The corpus of greek texts: A reference corpus for modern greek. *Corpora*, 5(1):29–44, 2010.
[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.
[21] Gerard J Steen. *Finding metaphor in grammar and usage: A methodological analysis of theory and research*, volume 10. John Benjamins Publishing, 2007.
[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
[23] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
[25] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
[26] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.
[27] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
[28] Alex Graves. Supervised sequence labelling with recurrent neural networks. 2012. *URL http://books. google. com/books*, 2012.
[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
[31] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[33] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
[34] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.
[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.