

# A Hybrid Deep Learning Network for Modelling Opinionated Content

Pantelis Agathangelou

University of Nicosia

Nicosia, Cyprus

agathangelou.p@live.unic.ac.cy

Ioannis Katakis

University of Nicosia

Nicosia, Cyprus

katakis.i@unic.ac.cy

## ABSTRACT

The ability to accurately understand opinionated content is critical for a large set of applications. Models targeting at learning from such content should overcome the inherent difficulties of the data. We propose a novel hybrid neural network embedded in a deep learning framework that can be used for sentiment classification. Our method consists of an independent set of feed forward learning models that are able to identify rich linguistic patterns through recurrent semantic trees. We evaluate our method in four sentiment classification problems that include both binary and multi-class classification tasks. Moreover, we compare our model's prediction accuracy with state-of-the-art methods. We observe that our method outperforms the alternative approaches. The strengths of the proposed approach are due to i) a novel Convolutional Neural Network which can be employed autonomously or as part of a greater framework, ii) a hybrid framework which consists of a set of independent blocks that propagates information and improve the classification task.

## KEYWORDS

Sentiment Analysis, Opinion Mining, Deep Learning

### ACM Reference Format:

Pantelis Agathangelou and Ioannis Katakis. 2019. A Hybrid Deep Learning Network for Modelling Opinionated Content. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, April 8–12, 2019, Limassol, Cyprus. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3297280.3297570>

## 1 INTRODUCTION

Online content like user opinionated reviews, contains information that if exploited appropriately may provide commercial and research value. Assigning sentiment to sentences (or phrases) is a trivial task in most cases. Trying to assess the sentiment for a large piece of text (like a product review) however, might be a little bit more challenging. This is mainly because language might include a mixed distribution of sentiment.

In this paper, in order to alleviate the above limitation, we introduce a convolution to the widely used recurrent deep learning model. The **contributions** of this paper can be summarized in

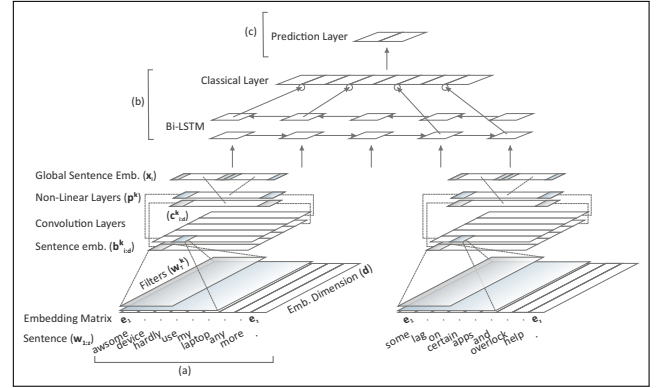


Figure 1: Model Architecture.

the following points. a) We introduce a Deep Hybrid Neural Network for modeling opinionated content. b) We introduce a Neural Network Hyper-Parameter that joins mixed content motifs. c) We experiment over a set of known datasets and demonstrate how our proposed approaches outperform the state-of-the-art. d) We share our deep learning model in Python/Tensorflow with the community in order to promote the research efforts in the field.

## 2 OUR METHOD (HYCOR)

Some principles of our method (Hybrid Convolution Recurrent Neural Network) were inspired by the work of [1], [2], and [3].

**Overview.** HyCoR is organized into three feed forward blocks. In the first “sentence level” block (see Fig 1(a)) we use a multi-filter convolution layer to transform word embeddings into sentence embeddings. In the second “document level” block (Fig 1(b)) we feed the sentence embeddings into a modified Bi-directional LSTM recurrent neural network. Next a classical layer employs a number of output predictions. These predictions are the partial sentiment classifications of a candidate opinion. Finally in the third block “prediction level” (Fig 1(c)) we forward the partial predictions into an appropriate discrete classifier that merges all partial predictions. Fig 1 illustrates the model architecture and the corresponding blocks.

### 2.1 Global Sentence Embedding

The NN layers that we use are the following:

**Look-up Table Layer** A look-up table is created after we define the vocabulary size  $|V|$  of the corpus and the embedding size  $d$  of the word vectors. This table  $W_{(d \times |V|)}$  is a matrix of parameters to be learned [4].

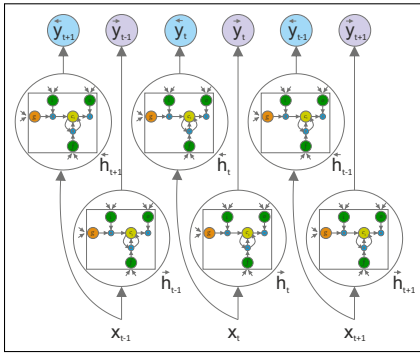
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SAC '19, April 8–12, 2019, Limassol, Cyprus

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5933-7/19/04.

<https://doi.org/10.1145/3297280.3297570>



**Figure 2: The BLSTM implementation and the separate output predictions that we exploit.**

**Convolution Layer** We begin in this layer with a tokenized sentence of length  $z$  which has previously been transformed into a matrix of word vectors of dimensionality  $d$ .

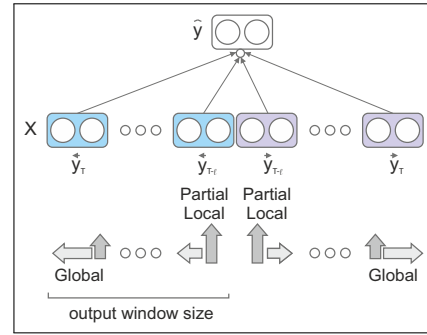
**Non-Linear Layer** We employ the partial global sentence embedding  $c^k$  and apply the  $\tanh$  non linear function in order to explore more complicated classification patterns. The result produces the vector  $p^k$  with dimension  $d$ . Finally we concatenate all  $p^k$  vectors to produce a global sentence embedding  $x_i$ . Vector  $x_i$  with size  $n \cdot d$  ( $n$  the number of convolutional filters) stands for the global sentence embedding which along with the  $\tau$  sentences an opinion consists of we forward to the next step of the neural network. This block is responsible for converting a sequence of word embeddings into a sentence embedding for every sentence in an opinion. Since sentence embeddings include language patterns that are explored by the convolution filters  $w_T^k$  we force the optimization algorithm through back-propagation to best tune up these filters over all the map of patterns the convolution layers include. Thus our convolution architecture becomes sensitive to dynamically exploring language patterns that apart from the classification task it also provides better word embeddings.

## 2.2 Exploiting Semantic Embeddings

In this phase our network receives a sequence of global sentence embeddings. We forward them to a modified Bi-directional recurrent structure that creates two independent trees.

**Bi-Directional Unit** Given an input sequence  $x = (x_1, \dots, x_T)$  a standard recurrent neural network (RNN) computes the hidden vector sequence  $h = (h_1, \dots, h_T)$  and the output vector sequence  $y = (y_1, \dots, y_T)$ . We exploit each layer’s sequential output separately. Next we construct a vector consisting of an array of predictions from these outputs which form a classical layer. We employ a hyper-parameter named “output window size” that immediately affects the classical layer’s vector length. Fig 2 presents the Bi-Directional [5] (LSTM [6]) implementation in our network.

**Classical Layer** A number of output predictions from the RNN structure consists the input basis of a classical layer. We exploit these output predictions evenly from both directions. We refer to this number as “output window size”  $\in [0, 1]$ . This layer produces the final vector  $\hat{y} \in \Delta$  that is forwarded next for normalization and



**Figure 3: The Implementation of the Classical Layer and its role to the overall classification task. Each output prediction layer ( $\hat{\mathbf{y}}_t, \vec{\mathbf{y}}_t$ ) encompasses two parts of information. One local and one global. As we move on to the terminal states, global information increases, and local diminishes.**

prediction. Fig 3 depicts how the classical layer is implemented and portrays its role at the overall classification task. In this block we materialize two operations. First we explore the sequential dependency sentences provide via the “global sentence embeddings”. Second via the classical layer and the “output window size” we grasp both “local” and “global” predictions all together.

### 3 EXPERIMENTS

### 3.1 Preprocessing & Datasets

**Reproducibility Note.** All source code that is required to run the following experiments is available at the following link: <https://github.com/unic-ailab/hycor-sac>

For the experiments, we focus on the following corpora.

SST-1: Stanford Sentiment Treebank<sup>1</sup> - Movie reviews with one sentence per review provided with fine-grained labels (very positive, positive, neutral, negative, very negative). For the purposes of this study we assembled sentences into opinions and created custom train/test datasets.

SST-2, SST-3: The SST-1 dataset transformed into binary labels with the neutral labels removed and the respective three classes dataset (positive, neutral, negative).

PG-1: A Pricegrabber<sup>2</sup> dataset used in [2]. The dataset consists of customer reviews on consumer products. Each opinion is evaluated upon fine-grained labels (see above).

PG-1b, PG-2b, PG-3b: The equivalent balanced PG-1, binary balanced and three classes balanced dataset respectively.

PG-2, PG-3: The equivalent binary PG-1 dataset and three classes dataset respectively. CR: Customer reviews<sup>3</sup> of various products (phones, cameras e.t.c.). The task is to predict positive/negative labels [7]

MR: Movie reviews with one opinion per review. Classification involves detecting positive/negative reviews<sup>4</sup>.

<sup>1</sup><https://nlp.stanford.edu/sentiment/>

<sup>2</sup><https://github.com/ailabunic/datasets>

<sup>3</sup><https://www2.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

<sup>4</sup><http://www.cs.cornell.edu/people/pabo/movie-review-data/>

**Table 1: Datasets characteristics and Preprocessing settings**

	CR	MR	PG.2	PG.2b	SST.2	PG.3	PG.3b	SST.3	PG.1	PG.1b	SST.1
Opinions	1724	10662	3682	498	9614	3860	650	11855	3860	650	11855
Sentences / Opin	2	7	4	3	4	4	4	4	4	4	5
Words / Sentence	100	52	163	137	52	163	137	52	163	137	52
Vocabulary	4736	19661	8958	3027	17162	9273	3729	19073	9273	3729	19073
Input Size	Avg	Max	Avg	Avg	Max	Avg	Avg	Max	Avg	Avg	Max

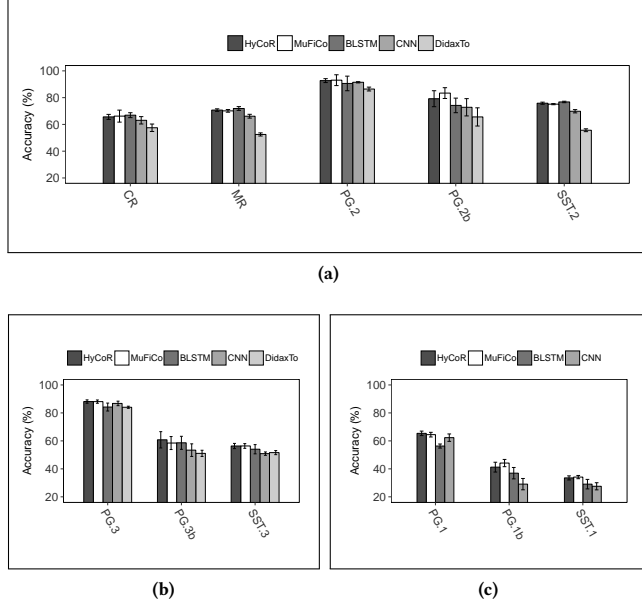
**Figure 4: Generalization of the proposed HyCoR, MuFiCo and the competitor BLSTM, CNN, DidaxTo (A pattern-based unsupervised method [2]) models on the experimental datasets. (a) binary evaluation, (b) three classes evaluation, (c) Fine grained evaluation (multiple classes)**

Table 1 presents the basic characteristics of each dataset along with preprocessing settings.

### 3.2 Sentiment Classification

In this part we evaluate our model's performance on a sentiment analysis task. From Figs 4(a), (b) and (c), we notice that the proposed models, HyCoR and MuFiCo, present better generalization over the rest of the approaches in almost all cases (8 out of 11). We attribute this superior performance of the neural models over the DidaxTo in the inherent nature of neural models to discover language patterns that correlate with one of the classes. Regarding the comparison against the BLSTM and CNN we observe that only the BLSTM presented slightly better performance at the MR and SST.2 datasets. In all other datasets and experiments the proposed models outperformed their neural counterparts. Moreover as we move on from binary to fine-grained classification we notice that the generalization difference between the proposed methods and the alternatives is more evident. This observation indicates

that HyCoR and MuFiCo present better sensitivity as we move to fine-grained classification.

### 3.3 Holistic vs Cumulative Content

At this section we explore how the dynamic characteristics of the classical layer (see related paragraph at section 2.2) may provide additional features to the operability of the HyCoR Model. In order to accomplish the above task we employ the binary and the fine-grained datasets of Table 1, the proposed HyCoR model and we experiment on the "output window size" hyper-parameter.

**Table 2: Holistic/Cumulative content identification over a set of different datasets & output window size values.**

output window size	CR	MR	PG.2	PG.2b	SST.2	SST.1	PG.1	PG.1b
0	<b>69.60%</b>	<b>71.07%</b>	93.16%	75.20%	<b>74.16%</b>	32.81%	64.90%	38.92%
0.25	65.80%	69.72%	93.16%	75.20%	74.16%	31.22%	64.90%	38.92%
0.5	66.20%	68.40%	92.38%	77.40%	72.14%	31.13%	65.08%	37.85%
0.75	66.79%	71.00%	93.11%	<b>79.40%</b>	73.25%	<b>33.94%</b>	<b>66.74%</b>	<b>39.85%</b>
1	66.73%	68.82%	<b>93.30%</b>	74.00%	72.95%	32.39%	66.22%	38.77%

Table 2 provides insight over the cumulative/holistic nature of a content, since arithmetic values can help locate the exact point that reveals the nature of that content. We observe the following: CR dataset present cumulative content since the best performance is observed when output window is 0. Similar conclusions can be drawn on the MR and SST.2 datasets. What we observe over the binary PG.2, PG.2b datasets and the fine-grained PG.1, PG.1b is that the content is holistic in most cases, since the best performance is performed in values that are > 0.5.

**Acknowledgement** This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825171.

### REFERENCES

- [1] Dimitrios Kotzias, Misha Denil, Phil Blunsom, and Nando de Freitas. Deep multi-instance transfer learning. *CoRR*, abs/1411.3128, 2014. URL <http://arxiv.org/abs/1411.3128>.
- [2] Pantelis Agathangelou, Ioannis Katakis, Ioannis Koutoulakis, Fotis Kokkoras, and Dimitrios Gunopulos. Learning patterns for discovering domain-oriented opinion words. *Knowledge and Information Systems*, Jun 2017. ISSN 0219-3116. doi: 10.1007/s10115-017-1072-y. URL <https://doi.org/10.1007/s10115-017-1072-y>.
- [3] Pantelis Agathangelou, Ioannis Katakis, Fotis Kokkoras, and Konstantinos Ntonas. *Mining Domain-Specific Dictionaries of Opinion Words*, pages 47–62. Springer International Publishing, Cham, 2014. ISBN 978-3-319-11749-2. doi: 10.1007/978-3-319-11749-2\_4. URL [https://doi.org/10.1007/978-3-319-11749-2\\_4](https://doi.org/10.1007/978-3-319-11749-2_4).
- [4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=944919.944966>.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [6] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014. URL <http://arxiv.org/abs/1409.2329>.
- [7] Mingqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014073. URL <http://doi.acm.org/10.1145/1014052.1014073>.