Can you tell?

SSNet - a Sagittal Stratum-inspired Neural Network Framework for Sentiment Analysis

Apostol Vassilev
Munawar Hasan

National Institute of Standards and Technology

apostol.vassilev@nist.gov, munawar.hasan@nist.gov

Abstract

When people try to understand nuanced language they typically process multiple input sensor modalities to complete this cognitive task. It turns out the human brain has even a specialized neuron formation, called sagittal stratum, to help us understand sarcasm. We use this biological formation as the inspiration for designing a neural network architecture that combines predictions of different models on the same text to construct a robust, accurate and computationally efficient classifier for sentiment analysis. Experimental results on representative benchmark datasets and comparisons to other methods[1]show the advantages of the new network architecture.

Keywords: natural language processing, machine learning, deep learning, artificial intelligence

Introduction

Applications of deep learning to natural language processing represent attempts to automate a highly-sophisticated human capability to read and understand text and even generate meaningful compositions. Language is the product of human evolution over a very long period of time. Scientists now think that language and the closely related ability to generate and convey thoughts are unique human traits that set us apart from all other living creatures. Modern science describes two connected but independent systems related to language: inner thought generation and sensor modalities to express or take them in for processing [7]. For example, human sensory modalities are speaking, reading, writing, etc. This allows homo sapiens to express an infinite amount of meaning using only a finite set of symbols. e.g. the 26 letters in the English language. The result is a very powerful combination that has resulted in the vast amount of knowledge and information amassed in the form of written text today.

Over the course of the long evolutionary development and especially in the modern era, the sapiens have mastered the ability to generate and convey sophisticated and nuanced thoughts. Consequently, the texts deep learning is tasked with processing, known as natural language processing (NLP), range from the simple ones that say what they mean to those that say one thing but mean another. An example of the latter is sarcasm. To convey or comprehend sarcasm the sapiens typically invoke more than one sensory modality, e.g. combining speech with gestures or facial expressions, or adding

[1] DISCLAIMER: This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

nuances to the speech with particular voice tonalities. In written text, comprehending sarcasm amounts to what colloquially is known as reading between the lines.

With the emergence of A.M. Turing's seminal paper [33], the research in NLP kicked off. Initially, it revolved around handwritten rules, later obsoleted by statistical analysis. Until the advent of deep learning, the decision tree based parsing techniques [19, 20] were considered as the state of the art methods and linguistic performance was measured on the basis of the *Turing Test* [33].

Fast forward to present day, deep learning and the enormous increase in available computational power allowed researchers to revisit the previously defined as computationally intensive family of recurrent neural networks [14, 31] and produced several groundbreaking NLP results [6, 15, 35, 36]. There are also numerous other application-specific architectures [4, 8, 10] for NLP problems. We refer the reader to a recent comprehensive survey [24] for a detailed review of a large number of deep learning models for text classification developed in recent years and a discussion of their technical contributions, similarities, and strengths. This survey also provides a large list of popular datasets widely used for text classification along with a quantitative analysis of the performance of different deep learning models on popular benchmarks.

Through the evolution of their brain, sapiens have acquired a polygonal crossroad of associational fibers called sagittal stratum (SS), cf. Figure 1[2], to cope with this complexity. Researchers have reported [29] that the bundle of nerve fibers that comprises the SS and connects several regions of the brain that help with processing of information enables people to understand sarcasm through sensory modalities – both visual information, like facial expressions, and sounds, like tone of voice. Moreover, researchers have shown that the patients who had the most difficulty with comprehending sarcasm also tended to have more extensive damage in the right SS. In other words, the ability to understand sophisticated language nuances is dependent on the ability of the human brain to successfully take in and combine several different types of sensory modalities.

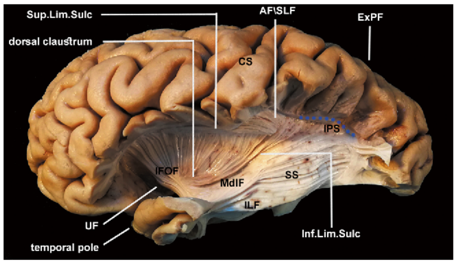

Figure 1. This image shows the sagittal stratum (SS). The SS is situated deep on the lateral surface of the brain hemisphere, medial to the arcuate/superior longitudinal fascicle complex, and laterally to the tapetal fibers of the atrium [5]. The SS is a bundle of nerve fibers that connects many different parts of the brain and helps with processing sensory modalities (visual and sound) and thus enables people to understand nuanced language such as sarcasm.

The evolution of language and the resulting increased sophistication of expressing human thoughts has created a challenging problem for deep learning. How to capture and process the full semantics in a text is still an open problem for machine learning. This is partly manifested by the facts that first, there are many different ways of encoding the semantics in a text, ranging from simple encoding relying on treating words as atomic units represented by their rank in a vocabulary [3], to using word embeddings or distributed representation of words [23], to using sentence embeddings and even complete language models [13, 32]; second, there is no established dominant neural network type capable of successfully tackling natural language processing in most of its useful for practice applications to the extent required by each specific application domain.

Based on this observation, we explored the extent to which it is possible to utilize a simple encoding of semantics in a text and define an optimal neural network for that encoding [34] for sentiment analysis. Our study showed that although each of these encoding types and corresponding neural network architecture may yield good results, they are still limited in accuracy and robustness when taken by themselves.

The main thrust of NLP research is based on the idea of developing computationally intensive neural network architectures intended to produce better results in the form of accuracy on representative benchmark datasets. In contrast, the research on simulating

[2] Reprinted from [5] with permission by Springer Nature, order #4841991468054.

the decision making capabilities of our brain related to perception or past experiences with machine learning has lagged. Thus, the computed probability of any linguistic sample predicted by any individual model is not grounded in a state or function of a biological mind. As we mentioned above, the anatomy of the human brain allows processing of multiple input sensor modalities to make a decision. Inspired by this, this paper seeks to establish a novel approach to sentiment analysis for NLP.

The primary goal of this paper is to explore the problem from a different perspective and study ways to combine different types of encoding intended to capture better the semantics in a text, along with a corresponding neural network architecture inspired by the SS in the human brain. To do this, we introduce a new architecture for neural network for sentiment analysis and draw on the experiences from using it with several different types of word encoding in order to achieve performance better than that of each individual participating encoding. The main contribution of this paper is the design of the SS-inspired framework for neural networks for sentiment analysis in Section 2.

The authors would like to emphasize that this paper does not try to improve the metrics achieved by aforementioned papers but presents an approach to simulate certain decision making scenarios in the human brain. Any model referenced in this section can be used as a plug and play module in our framework.

1 Limitations of existing standalone NLP approaches to machine learning

As indicated above, there are multiple different types of encoding of semantics in text, each of varying complexity and suitability for purpose. The polarity-weighted multi-hot encoding [34], when combined with appropriately chosen neural network, is generic yet powerful for capturing the semantics of movie reviews for sentiment analysis. Even though the overall accuracy reported in [34] is high, the approach quickly reaches a ceiling should higher prediction accuracy be required by some application domains.

Encoding based on word embeddings or distributed representation of words [23] is widely used. For example, the approach in [27] has been influential in establishing semantic similarities between the words for a given corpus, by projecting each word of the corpus to a high dimensional vector space. While the dimension of the vector space itself becomes a hyperparameter to tweak around, the vectors can be further processed or utilized using a recurrent neural network (RNN). When tackling NLP problems, a variant of RNN, namely the long short term memory (LSTM) variant and its bidirectional version (BLSTM) are known to perform better than other neural networks. Through our experiments on various datasets [1, 16], we found that certain vocabulary provides deeper semantics to the sentence from the corpus based on the receiver's perception and context. In such situations, the idea of attention [2, 12, 18] plays an important role and provides the network with an additional parameter called the context vector, which can make convergence slower, but the model overall is robust. It is also possible to use a learnable word embedding, where the first layer of the neural network architecture is the embedding followed by one or more RNN's.

Although intuitively one may think that word embeddings should help to increase the accuracy of the model to any desirable level because word embeddings do capture the semantics contained in the text in a way that mimics how people perceive language structure, the available empirical test evidence in terms of reported accuracy rates is inconclusive. Our own experiments with word embeddings used by themselves revealed an accuracy ceiling similar to that of the polarity-weighted multi-hot encoding. Attempts to utilize sentence embeddings have been even less successful [9].

All these different types of encoding can be challenged further by varying style of writing or level of mastering the language. Examples of the former are nuanced language such as sarcasm. Some reviewers choose to write a negative review using many positive words yet an experienced reader can sense the overall negative sentiment conveyed between the lines while the polarity-weighted multi-hot encoding [34] and word embeddings [27] may struggle with it. Examples of the latter are primitive use of the language by non-native speakers resulting in sentences with broken syntax and inappropriate terminology. Other difficult cases are reviews that contain a lot of narrative about the plot of the movie but very little of the reviewer's opinion about how she feels about the movie. Yet another problematic class are movie reviews that rate a movie excellent for one audience, e.g. children, but not good for another, e.g. adults. Careful analysis of the data in [1, 16] reveals examples of all these kinds of reviews, often confusing models based on the encodings described here. Such complications represent significant challenges to each of these types of encoding when used by themselves, no matter the power of the neural network.

This observation raises a question: if one is interested in obtaining a more robust and versatile representation of the semantics of text would an approach that combines different types of encoding yield a better result than attempting to just improve each of them within their envelopes?

2 Sagittal stratum-inspired neural network

We now turn to the design of a neural network that aims to simulate the way SS in the human brain operates. Recall that the SS brings information from different parts of the brain, each responsible for processing different input sensory modalities, to enable a higher order of cognition, such as comprehension of sarcasm. Our context here is NLP and one way to map the functioning of the SS to it is to consider combining different representations of the semantic content of a text. To do this, one first has to pick the types of representations of the text. Because we aim at computing different perspectives on the same text, it is natural to seek representations that are independent. For example, the polarity-weighted multi-hot encoding [34] is based on the bag-of-words model of the language and has nothing in common with word embeddings that rely on language structure [27]. But if independence of representation is adopted, how does one combine the two models computed from each of them?

Unlike image processing where each model is computed over the pixel grid of the image, in NLP there is no common basis onto which to map and combine the different models. Instead, we again use a hint from how the human brain performs some cognitive tasks. When a person hears another person utter a phrase, to comprehend what the speaker is trying to convey the brain of the listener first processes the words in the phrase, then the listener assesses if the speaker rolled her eyes, for example, when uttering the words, to decide if she spoke sarcastically. The brain of the listener combines these two assessments with the help of the SS to arrive at a final conclusion if the speaker spoke sarcastically or not. This suggest we can combine the resulting assessments from each model on a particular review, e.g., the probability of classifying it as positive or negative, to decide on the final classification.

The two neural networks based on the different language models discussed above are shown side by side in Figure 2, on the left is the A-network based on the polarity-weighted multi-hot encoding [34] and on the right is the B-network based on the word embedding language model [27]. Both networks are trained on the same data but subject to different Accuracy/Loss constraints and number of training epochs because the different models and neural network designs are unlikely to attain the same accuracy and loss targets. When each of them converges to a model ($Model_A$ or $Model_B$) that satisfies the corresponding target, the network outputs it for later use in

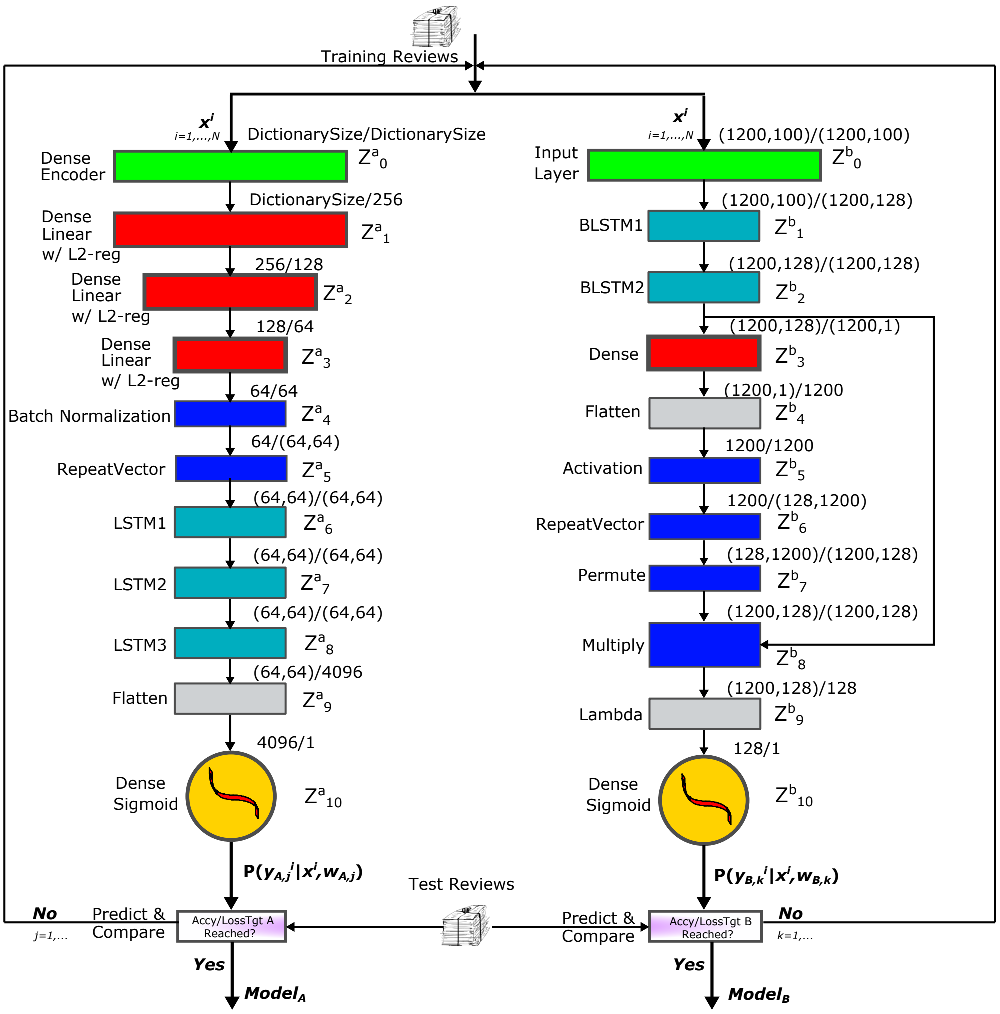

Figure 2. Participating neural network training.

combination to make predictions on data of interest - see Figure 3.

With $Model_A$ and $Model_B$ defined, the next step is to assemble the SS-inspired classifier. Based on our understanding for how SS works in the human brain to enable interpretation of language that can only be resolved correctly when multiple sensor modalities are used together, we construct the network shown in Figure 3.

Note that the computed probabilities for review classification from the two participating models are combined in a way that favors identifying the most probable case, which is analogous to the way humans assess multiple sensor modalities in order to process ambiguous speech and deduce the most plausible interpretation. Note also that the particular models we have considered so far to provide different perspectives on the semantics in the reviews are not necessarily unique for this task and the combined network. Other networks based on different language models may be possible to use with the understanding that each participating model should be sufficiently robust and accurate. We discuss this further in the next section.

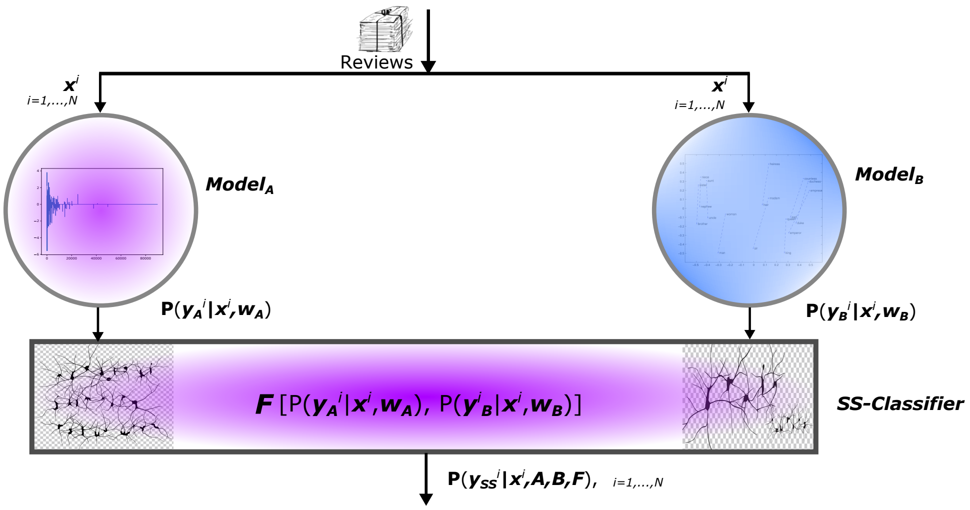

Figure 3. The SS classifier. Here F is some appropriately defined function that combines the input from both models. See Section 3 for details.

3 Computational results

In this section, we present the computational results obtained with our proposed architecture. We begin with an overview of the individual components of the architecture and then show its performance. All the experiments were performed on TensorFlow 2.1.0 [11] library. The models were trained on a professional Graphics Processing Unit (GPU) cluster having 8 NVIDIA Tesla V100 (32 GB each). The inference was carried on a 2015 MacBook Pro with 2.5 GHz Intel Core i7 and 16 GB RAM *without* Graphics Processing Unit (GPU) acceleration.

3.1 Datasets

Our experiments can be divided into two parts, following the approach in [34]. This allows for a clear and objective assessment of the advantages of the proposed sagittal stratum inspired neural network (SSNet) offers compared to BowTie. The first part is the training of the two models from Figure 2. We used the Stanford Large Movie Review dataset (SLMRD) [1] for training. The SLMRD dataset contains 25 000 labeled train reviews and 25 000 labeled test reviews.

The second part is the calculation of transfer accuracy of the trained SSNet with $Model_A$ and $Model_B$ over a different dataset. The transfer accuracy was calculated on the Keras Internet Movie Database (IMDB) dataset (KID) [16]. The KID dataset also contains 25 000 labeled train reviews and 25 000 labeled test reviews. We used the complete 50 000 reviews of KID for calculating the transfer accuracy of SSNet.

We refer the reader to [34] for further details about SLMRD and KID.

3.2 Proposed Architecture

The main idea behind the architecture in Section 2 is to incorporate two separate views on the same corpus. From Figures 2 and 3, it is clear that $Model_A$ and $Model_B$ are critically important components in our architecture. We next consider these two models in detail and discuss their performances.

3.2.1 $Model_A$: BowTie

We choose [34] as $Model_A$ due to its robust and efficient nature in tackling the semantics of the corpora [1, 16]. We introduced some minor tweaks to this model by incorporating an LSTM layer. This resulted in a small increase in the accuracy of the model, e.g., $Model_A$ was trained for 10 epochs and attained 91.2 % as training accuracy and 90 % as validation accuracy. These results were marginally better than the original version [34].

3.2.2 $Model_B$: BLSTM with Attention and Glove Embeddings

For $Model_B$, we used the embedding from [27] and built our model over that. While many other researchers in this area have obtained results by simply using LSTM or BLSTM with [27]; we found that the corpora [1, 16] contain reviews with nuances that are difficult to learn by simply passing the embeddings through LSTM or BLSTM. The models tend to learn the pattern of the inputs rather than the underlying meaning or semantics. This is often the cause of overfitting in a wide range of NLP problems. Further, in the case of sentiment analysis, certain words and their position in the sentence play extremely important role in determining the meaning. It is difficult to incorporate the positional semantics of these words using normal LSTM or BLSTM. The family of attention mechanisms [2, 12, 18] provides a direction to formulate such difficult semantics into the model. We revised the aforementioned attention mechanism to incorporate positional and sentimental semantics for sentences having large number of words.

Let $\overrightarrow{b}$ and $\overleftarrow{b}$ be the forward and the backward components of BLSTM and k be the sequence length, then $h = [\overrightarrow{b}, \overleftarrow{b}]$ where $dim(h) \in \mathbb{R}^{k \times (|\overrightarrow{b}| + |\overleftarrow{b}|)}$. We define the following equations to describe the attention mechanism used in this paper:

$$\begin{aligned} h'^{\leftarrow} &= tanh(h) \\ h'^{\leftarrow} &= softmax(h') \\ C_v &= h \quad h'^{\leftarrow} \\ M &= \sum_k C_v \end{aligned} \tag{1}$$

The third expression in 1 represents the context vector C_v, which we sum up in the fourth expression in 1 over the sequence to remove stochastic behavior, $dim(M) \in \mathbb{R}^{(|\overrightarrow{b}| + |\overleftarrow{b}|)}$. To correctly calculate the *Hadamard product*, it is required to expand the vector space of $h'^{\leftarrow}$ after performing the *softmax* operation. This strategy inside the attention mechanism establishes a probabilistic vector space incorporating the positional and the sentimental semantics into $Model_B$. As a result, with $Model_B$ we were able to achieve 94.20 % as training accuracy and 90.916 % as validation accuracy - see Table 1 below.

It is clear from the table that the component BLSTM with attention outperforms the rest and becomes a suitable candidate for $Model_B$ in our setup.

Our investigation showed that understanding nuances is not very computationally intensive but rather a logically inferential task; hence we used a low vector space, .i.e., 100 of the glove embeddings with sentence length equal to 1200. This resulted in small and robust models. We would also like to notify the readers that the semantic structure of the language as understood by human brain is closely related to word embeddings

Table 1. Computational Results of $Model_B$ with enhancing components.

Component	**Epoch**	**Training**		**Validation**	
		Accuracy	**Loss**	**Accuracy**	**Loss**
LSTM	15	88.03 %	0.2980	85.804 %	0.3882
BLSTM	15	88.59 %	0.2780	86.804 %	0.3282
LSTM (with Attention)	50	90.53 %	0.3060	89.216 %	0.3689
BLSTM (with Attention)	100	94.20 %	0.1449	90.916 %	0.2511

(rather than language modeling). Hence, the authors omitted use of any language modeling techniques in $Model_B$.

3.2.3 Combining $Model_A$ and $Model_B$

Now our task is to combine $Model_A$ and $Model_B$ taking into account that they represent very different views on the semantics in the same text. Recall from the observation in Section 2 that the only meaningful way to combine the two models is through the probability assessment each of them produces for a given review and in turn the entire corpus. While the ensemble techniques [25] have been known for good performance in computer vision related tasks [17, 26, 28, 30], the same is not true for natural language processing based problems. This analogy can also be backed by the fact that models using different encodings have different latent space and hence merging such latent spaces may not produce an optimal solution due to the varying rate of convergence of individual models. But the major issue is the projection of one model's latent space onto another. Due to different encodings, such projections may produce inconsistent coalesced models.

To combine $Model_A$ and $Model_B$, we introduce two parameters *threshold* θ, and *bias*. The idea of threshold as a parameter is to keep a check on the robustness of the base model ($Model_A$ in our case). The concept of a base model here is just a matter of an assumption which model we perceive as more robust and accurate. This could be based on the performance of the model on the training dataset as well as other considerations. For example, validation accuracy and validation loss may be used as parameters to judge the robustness of the models. Another criteria could be the prediction confidence of the respective model which can be inferred by evaluating the prediction probabilities of the models on judiciously chosen datasets. In our experiments, we found $Model_A$ to be outperforming $Model_B$ in these criteria.

Generally speaking, in cases where the comparison between two candidate models is inconclusive, the choice of the base and auxiliary models may be approached analogously to how humans interpret ambiguous voice: do they trust more the deciphering of the words or the evaluation of the facial expression of the speaker to decide what they mean? Some people may choose to weigh the words heavier than the voice in a given circumstance, others may opt the other way around. However, it is always important to be aware of the limitations the models may have in the context of the potential application.

Once the confidence of prediction of the base model falls below θ, we use the input of the secondary model ($Model_B$) based on *bias*. For a given sample $i \in [1, n]$, where n is the number of prediction samples; let $P_A^i = Pr(\mathbf{y}^i|\mathbf{x}^i, \mathbf{w}_A)$ be the probability predicted by $Model_A$ for sample i and let $P_B^i = Pr(\mathbf{y}^i|\mathbf{x}^i, \mathbf{w}_B)$ be the probability predicted by $Model_B$ for the same sample i. Then we can combine the predicted probabilities as follows:

$$P_{SS}^i = \max_{\substack{0<\theta<1\\1\leq i\leq n}} \begin{cases} \max \begin{cases} P_A^i \\ \overline{P_A^i} \end{cases} , & \text{if } (P_A^i > \theta) \vee (\overline{P_A^i} > \theta) \\ \max\limits_{0<bias<1} \begin{cases} bias * P_A^i + (1 - bias) * P_B^i \\ bias * \overline{P_A^i} + (1 - bias) * \overline{P_B^i} \end{cases} , & otherwise \end{cases} \tag{2}$$

Formula 2 finds the maximum probability P_{SS}^i of the combined models based on the parameters θ and $bias$. Note that at first glance it seems one may try out the entire set of possible values of θ and $bias$ to ensure a global maxima. Such an exhaustive search could start from a low value of θ and $bias$ and gradually increase storing all the probabilities. Once done, the maximum probability can be recorded. Let $\theta \in \{\theta_1, \theta_2, ..., \theta_m\}$ and $bias \in \{bias_1, bias_2, ..., bias_t\}$; where $(\theta_1, bias_1) > 0$ and $(\theta_m, bias_t) < 1$. Then the computational time complexity for such an exhaustive search would be bounded by $O(mt)$, while the space complexity is $O(m)$. Upon deeper insight, it can be observed that formula 2 has a large overlapping substructure due to the calculation of probabilities on the basis of monotonically increasing θ and hence a large amount of calculations can be memoized [21, 22], resulting in a much lower computational time complexity. We next explain how this memoization technique can be used in evaluating 2 for a given prediction sample $i \in \{1, n\}$.

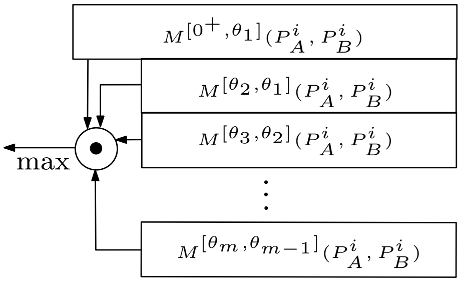
$M^{[0^+,\theta_1]}(P_A^i, P_B^i)$
$M^{[\theta_2,\theta_1]}(P_A^i, P_B^i)$
$M^{[\theta_3,\theta_2]}(P_A^i, P_B^i)$
$M^{[\theta_m,\theta_{m-1}]}(P_A^i, P_B^i)$
max

Figure 4. The image shows how memoization can be used to save computation. Instead of calculating all the previous thresholds, only relevant ones are calculated.

Let $\theta \in \{\theta_1, \theta_2, ..., \theta_m\}$ such that $\{\theta_1 < \theta_2 < ... < \theta_m\}$. Let $M^{[\theta_i,\theta_{i+1}]}(P_A^i, P_B^i))$ be the subroutine that calculates $\{bias * P_A^i + (1 - bias) * P_B^i\}$ and $\{bias * \overline{P_A^i} + (1 - bias) * \overline{P_B^i}\}$ where $0 < bias < 1 \wedge 1 \leq i \leq n$. Then we begin $\forall P_A^i \in Model_A \wedge P_A^i < \theta_1$ and evaluate $M^{[0^+,\theta_1]}(P_A^i, P_B^i)$ and store it in memory. 0^+ is the probability predicted by $Model_A$ and less than θ_1. Once θ_1 is calculated, we do not need to calculate the values already calculated and stored for θ_1. Subsequently, we only calculate for values that fall in between θ_i and θ_{i+1}. This results in a drastic decline of computational time complexity. This approach is similar to the dynamic programming paradigm, but does not possess an optimal substructure and hence the space complexity cannot be reduced.

3.3 Performance

In Figure 5, we show the combined performance of our proposed architecture for varying choice of epochs of $Model_B$. It can be seen from Figure 5 that the performance of combined system is best when $Model_B$ from epoch 60 is combined. We attained maximum accuracy of 94.93 % for $\theta \in \{0.86, 0.88, 0.90, 0.92, 0.94, 0.96, 0.98\}$. Figure 6 shows the receiver operating characteristic (ROC) curve for the best performing model for 7 threshold values together with ROC of $Model_A$ and $Model_B$. From figure 6, it is clear that area under curve (AUC) for the 7 threshold are similar with threshold at 0.98 attaining the maximum AUC.

From this we conclude that:

- When $Model_B$ is robust, it forces the base model $Model_A$ to perform better. This means that θ is pushed to a much higher value

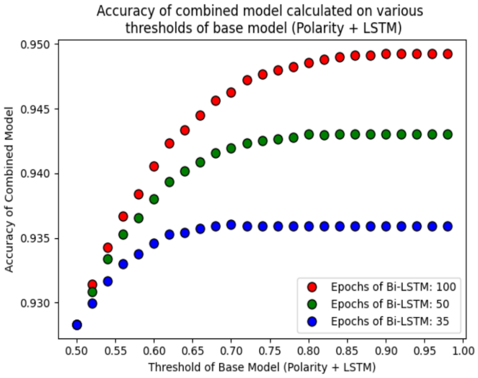

Figure 5. Accuracy of BLSTM with attention at various epochs

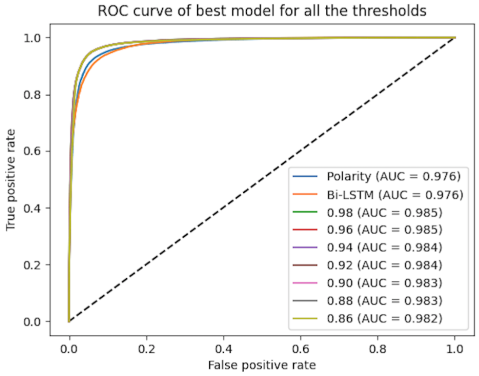

Figure 6. ROC of the combined setup relative to $Model_A$ and $Model_B$

- Whenever $Model_B$ has some part to play in the final outcome, $Model_B$ contributes with a high numerical value (approximately 42 % of its original numerical value)

These conclusions are directly related to our discussions in the introduction section regarding the role of SS in human cognition. The summary of computational results from our proposed system are shown below.

Table 2. Computational results of the SS classifier from Figure 3

Model	**Accuracy**	**Inference Time**
SSNet (Combined $Model_A$ and $Model_B$)	94.93 %	28.3 μs

These results confirm that the neural network system inspired by biological entities and concepts is robust and computationally fast while acting similarly to human decision making.

4 Conclusions and next steps

We successfully followed our intuition inspired by the biological underpinning of the human brain for understanding sarcasm to construct a neural network architecture for sentiment analysis and demonstrated excellent performance on benchmark datasets. Next, we plan to explore possibilities for enhancing the security of the computation of this network through secure multiparty computation protocols to facilitate adoption in sensitive application domains where high security and privacy is required. In addition, we are going to look for effective parallelization techniques to accelerate the computation of the training and prediction phases on multi-GPU platforms.

Acknowledgments

We thank the NIST Information Technology Laboratory (ITL) for the research funding and support.

References

[1] Andrew Maas. Large movie review dataset. http://ai.stanford.edu/~amaas/data/sentiment/, 2011.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (Prague)*, pages 858–867, June 2007.

[4] E. Cambria. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107, 2016.

[5] D. T. D. Carlo, N. Benedetto, H. Duffau, F. Cagnazzo, A. Weiss, M. Castagna, M. Cosottini, and P. Perrini. Microsurgical anatomy of the sagittal stratum. *Acta Neurochirurgica*, 161:2319–2327, 2019. https://doi.org/10.1007/s00701-019-04019-8.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] N. Chomsky. Center for brains minds + machines, research meeting: Language and evolution. https://youtu.be/kFR0LW002ig, May 2017.

[8] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[9] A. Conneau, D. Kiela, H. Schwenk, L. Barraul, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Emprical Methods in Natural Language Processing (Copenhagen, Denmark, September 7-11))*, Association of Computational Linguistics, pages 670–680, 2017, See also update at https://arxiv.org/abs/1705.02364v5.

[10] L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603, 2013.

[11] Google Brain Team. Open source library for ml models. https://www.tensorflow.org/, 2020.

[12] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[13] M. Hoang and O. A. Bihorac. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), (Turku, Finland, 30 September - 2 October))*, Linköping University Electronic Press, pages 187–196, 2019.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[16] Keras Documentation. Imdb movie reviews sentiment classification. `https://keras.io/datasets/`, 2018.

[17] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 21(1):31–40, 2017.

[18] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[19] D. M. Magerman. Learning grammatical structure using statistical decision-trees. In L. Miclet and C. de la Higuera, editors, *Grammatical Interference: Learning Syntax from Sentences*, pages 1–21, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[20] L. Màrquez and H. Rodríguez. Part-of-speech tagging using decision trees. In C. Nédellec and C. Rouveirol, editors, *Machine Learning: ECML-98*, pages 25–36, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[21] J. Mayfield, T. Finin, and M. Hall. Using automatic memoization as a software engineering tool in real-world ai systems. In *Proceedings the 11th Conference on Artificial Intelligence for Applications*, pages 87–93. IEEE, 1995.

[22] D. Michie. “memo” functions and machine learning. *Nature*, 218(5136):19–22, 1968.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013, see also `https://arxiv.org/abs/1310.4546`.

[24] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep learning based text classification: A comprehensive review, 2020.

[25] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *J. Artif. Int. Res.*, 11(1):169–198, July 1999.

[26] R. Paul, L. Hall, D. Goldgof, M. Schabath, and R. Gillies. Predicting nodule malignancy using a cnn ensemble approach. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[28] F. Perez, S. Avila, and E. Valle. Solo or ensemble? choosing a cnn architecture for melanoma classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[29] P. M. Pexman. How do we understand sarcasm? *Frontiers for Young Minds*, 6:56, November 2018. `https://doi.org/10.3389/frym.2018.00056`.

[30] B. Savelli, A. Bria, M. Molinara, C. Marrocco, and F. Tortorella. A multi-context cnn ensemble for small lesion detection. *Artificial Intelligence in Medicine*, 103:101749, 2020.

[31] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[32] C. Sun, L. Huang, and X. Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[33] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950.

[34] A. Vassilev. Bowtie - a deep learning feedforward neural network for sentiment analysis. In G. Nicosia, P. Pardalos, R. Umeton, G. Giuffrida, and V. Sciacca, editors, *Machine Learning, Optimization, and Data Science*, pages 360–371, Cham, 2019. Springer International Publishing.

[35] A. R. Voelker, I. Kajić, and C. Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[36] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.