# Reusable Toolkit for Natural Language Processing in an Ambient Intelligence Environment

1st Sirwe Saeedi
*Department of Computer Science*
*Western Michigan University*
Kalamazoo, MI, USA
sirwe.saeedi@wmich.edu

2nd A.C.M. Fong
*Department of Computer Science*
*Western Michigan University*
Kalamazoo, MI, USA
acmfong@gmail.com

3rd Ajay Gupta
*Department of Computer Science*
*Western Michigan University*
Kalamazoo, MI, USA
ajay.gupta@wmich.edu

4th Steve Carr
*Department of Computer Science*
*Western Michigan University*
Kalamazoo, MI, USA
steve.carr@wmich.edu

***Abstract*—Computational natural language processing (NLP) is indispensable in a humanized ambience intelligence environment. NLP facilitates ambient intelligence by making machines understand, and be understood by, humans. This in turn makes machines behave more human-like than they typically are today. Technological advances in machine learning (ML), and especially deep learning (DL), have been a key enabler of NLP research. This paper begins with a survey of recent developments of ML/DL for NLP. It then identifies some of the most promising techniques reported in recent literature. These most promising techniques are then assembled into a reusable toolkit for computational NLP. The adaptable nature of the assembled toolkit allows it to be reused in a broad range of NLP applications. The paper then describes experimental evaluation of our implemented solutions for comparative analysis. Two specific NLP applications form the basis of comparative evaluation. The first involves identifying one of $M$ English sentences that does not make sense. The second, which is harder than the first, involves choosing from among $N$ sentences the one that best explains why a presented sentence is invalid. Human baseline accuracies for these applications are 99.1% and 97.8%, respectively. The observation that these results are somewhat less than perfect demonstrates that even humans can occasionally find these tasks difficult. It further underscores the difficulties involved in some of these computational NLP applications. Experiments conducted on benchmark data show that advanced ML/DL can achieve near-human performance in both computational NLP applications with accuracy scores of 96.1% and 93.7%, respectively.**

***Index Terms*—Computational intelligence, human-like intelligence, natural language processing, machine learning, deep learning, transfer learning, language modeling, transformers, neural networks**

## I. Introduction

Technological improvements in machine learning (ML) [1] have been fueled by algorithmic developments in deep learning (DL) [2] neural networks (NN), increased availability of big data, as well as advances in computer hardware. Ready accessibility of compute-intensive ML/DL methods to affordable cloud services has also contributed to the popularity of ML/DL usage well beyond traditional computer science (CS) applications. All these developments have enabled a range of human-like intelligence applications previously considered very difficult or even impossible. These include applications in areas such as smart cities and transportation, intelligent healthcare, and many others. Indeed, there has been tremendous progress made in various applications of computational intelligence (CI) to natural language processing (NLP), such as conceptual and semantic analysis in social media [3], [4], realistic text generation tools [5], intelligent question and answer (Q&A) systems [6], and text summarization [7].

Members of the Western Michigan University (WMU) Transformative Interdisciplinary Human+AI Research Group (https://fong.cs.wmich.edu/) are committed to the research and development of next-generation human-centric intelligent machines and user-friendly smart applications. It is also an aim of the group to ensure that such enhanced human-centric machine intelligence will positively impact a broad range of disciplines beyond computer science. These include mechanical engineering, civil engineering, statistics, and business analytics. Use cases span commercial, industry, and consumer applications. Machine NLP is one of the key focus areas of the group due to its wide applicability.

This paper presents research intended to address two intertwined problems in NLP. The first problem is sentence validation, which begins with preparing $M$, where $M > 1$, English sentences with exactly one out of $M$ sentences that is not valid. Under this scenario, when a computational machine is presented with the prepared $M$ sentences, it must choose the one that is not valid because it is illogical or violates some natural language rules.

The second problem is reasoning about the validity (or otherwise) of a sentence. For example, suppose that the NLP machine is presented with an English sentence that is known to be invalid. The machine's task then is to choose from $N$, where

The research reported is supported by US National Science Foundation under grant number 2017289

978-1-6654-8768-9/22/$31.00 ©2022 IEEE

$N > 1$, alternative sentences exactly one that best explains why the given sentence is invalid. Obviously, the "best" reason must itself not violate any logical or natural language rules.

Specific contributions of this research include:

- A survey of recent advances in computational intelligence for NLP applications.
- Based on the survey, an effective and reusable learning toolkit is assembled for addressing the two categories of NLP problems outlined above (i.e., sentence validation and validity reasoning).
- A novel way of applying the masked language model (MLM) technique for predicting probabilities of masked tokens for a given set of sentences.
- A method for enhancing the performance of all applied models based on the idea of reformulating the input of sentence validation as a classification task to the input of another downstream stage, i.e., treatment as multiple choice questions.

The rest of this paper is organized as follows. By presenting pertinent research background information with a motivating example, Section II sets the scene and lays the foundation for further discussion. Section III presents a survey of recent computational models for NLP, identifying the most promising ones as the basis for further development. Section IV discusses the key components of an effective and reusable toolkit for NLP applications. These include recurrent neural network (RNN), transfer learning (TL), and transformers. Section V presents experimental details of our developed tools for addressing sentence validation and validity reasoning, leading to near-human level of performance in both applications. Finally, Section VI concludes the paper by summarizing the key advances, as well as limitations, reported in this paper. The concluding section also suggests possible futue research directions, some of which are currently underway.

## II. Background and Motivation

A specific aim of the research is to make computers exhibit human-like ambience intelligence. This is achieved by making computational machines understand and interact with humans in a more humanized and intuitive manner than how they appear now. The well-known Turing test [8] is often considered as a benchmark of inquiry in machine intelligence in general, and NLP research in particular. Also considered an imitation game, the test is designed to determine a computational machine's ability to display human-like intelligence and behavior during a series of interactions with humans.

Apart from the Turing test, there is a less well-known test that uses Winograd schema questions [9] as the basis of inquiry. More specifically, each Winograd schema poses a set of multiple-choice questions in a well-defined structure. As shown in the following examples, solving a Winograd schema question requires effective application of real-world knowledge generally possessed by people.

**Specimen Winograd schemata**
**Example 1**
The horse would not fit in the plastic bag because it was too [big /small]. What was too big (small)?
Answer 1: the horse
Answer 2: the plastic bag

**Example 2** The city police denied the demonstrators a permit because they [feared/suggested] chaos. Who feared (suggested) chaos?
Answer 1: the police
Answer 2: the demonstrators

In each of the two Winograd examples above, if the red word is chosen to complete the sentence, then the pronoun "it" refers to the first answer. Alternatively, choosing the blue word means that the second answer is valid. These examples demonstrate the importance of knowing about the world around us (or a machine in the case of machine NLP). It furthermore demonstrates the importance of knowing how to apply innate general knowledge in new situations and questions not encountered or thought of before. Evidently, Winograd schema questions are designed to be obvious to any typical human proficient in the natural language under consideration (which is English in this case). However, the challenge is to build a computational model to recognize facts obvious to humans. This is often more difficult than anticipated.

## III. Computational Intelligence Models for NLP

This section surveys CI models that have been found effective for NLP problems. The ones that are especially promising are highlighted.

### A. Recurrent Neural Network

A recurrent neural network (RNN) [10] works on sequential data, such as text or strings of characters. RNN is often used for NLP tasks, e.g., language translation, speech synthesis and recognition, Q&A systems, and media captioning. These networks are characterized by their internal memory allowing them to save information from previous inputs to influence the current input. This characteristic enables them to process variable-length sequences of input text data. Different from fully connected neural networks, the current input of a layer in RNN depends on the earlier outputs of the hidden layer.

A disadvantage of standard RNN is that information from a long time ago can be difficult to retrieve because it lacks the ability to remember long term dependencies. This is in turn caused by the problem of vanishing gradient. For example, if the RNN's input is a text paragraph that comprises numerous sentences, then RNN may forget some of the important features from or near the beginning.

Long Short-Term Memory (LSTM) [11] is a type of RNN specifically designed to overcome the long-term dependency problem associated with standard RNN. LSTM networks can theoretically store information for an arbitrary duration. For NLP applications, average stochastic gradient descent (SGD) weight-dropped LSTM (AWD-LSTM) [12] has been found to

be effective for processing and predicting sequences of words. AWD-LSTM provides a set of regularization and optimization strategies for existing LSTM implementation.

### B. Attention-based Network

The attention mechanism is an important development that is often considered by many to have revolutionized computational NLP. Specific techniques that incorporate the attention mechanism include the transformer [13] architecture, Google's Bidirectional Encoder Representations from Transformers (BERT) [14] , and Generative Pre-trained GPT-3 Model Representation [15]. It is not an exaggeration to assert that attention is the main component of pioneer algorithms in human language understanding. In this research, we investigate the attention mechanism and different state-of-the-art transformers suitable for machine NLP problems. Furthermore, we present a comprehensive view of transfer learning, which is important for explaining the significance of our work.

*1) Transfer Learning (TL):* As Fig.1 shows, TL is a method in which a model, which has been trained on a large dataset for a specific task (primary task), is adapted to a different but related task (secondary task). Typically, the secondary task is the one that the user is interested in, but limited availability of training data prevents the user from training the network for that task from scratch. Alternatively, there are situations in which well trained networks exist and can be adapted to perform more specific tasks. In any case, TL facilitates accumulation of knowledge from related data-rich task(s), and then using that information as a starting point in training the network for the secondary task of interest.

In fact, the use of a pre-trained model is popular among members of the deep learning community. Rather than designing NN layers from scratch to learn useful features, use of a pre-trained model is often considered a good starting point. TL has proved to be a game-changer in numerous computer vision and machine NLP tasks. Examples of pre-trained models that have been trained on very large corpora of text documents include OpenAI GPT series [15], BERT [14] and variants, Embeddings from Language Model (ELMo) [20], Google's word2vec [26], and Stanford Global Vectors for Word Representation (GloVe) [27] models.

In this research, we apply transfer learning to significantly expedite the training process, which also improves the performance of the resulting computational models. We apply pre-trained network to a large benchmark dataset including millions of text data items (books, web pages, paper documents, etc.) and re-purpose the extracted features for our NLP applications. Typically, TL works best if the base network is trained on general features rather than for specific applications.

*2) Self-Attentional Neural Networks and Transformers:* A common trait of transformer-based architectures, such as BERT [14] and variants RoBERTa [16], ALBERT [17], and DistilBERT [18], is that they all have some form of built-in self-attention mechanism. Before the advent of transformers and self-attention mechanism, RNN-based networks struggled to capture the context in many NLP applications. The root

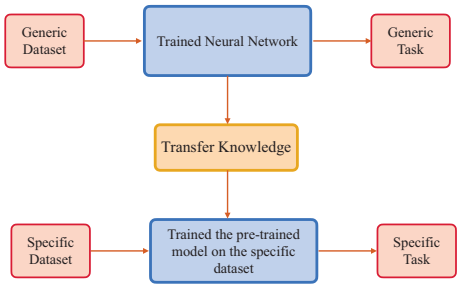


Fig. 1. Transfer learning.

cause is that these models are sensitive to the length of sentences. For long input sequence length more updates increase the chances of losing earlier inputs and updates.

The attention mechanism was proposed in [19] to deal with this loss problem originally intended for machine translation. The attention mechanism allows the model to look at the entire context and identify relationships between words that are far apart. Attention-based models give particular "attention" to some hidden states in RNN-based models, while decoding each word during the translation.

The authors of [13] claimed that "self-attention" is all encoding needs and no interfering with RNN variations. Self-attention is the core idea behind transformer, such that each token in the sequence attends to every other token in the same sequence. Consequently, the relationships between words in the sequence can be captured. Transformer is fundamentally a sequence-to-sequence, length independent model that consists of encoders and decoders. The encoding and decoding blocks are a stack of encoders and decoders. For example, the paper [13] stacks six encoders and decoders. These numbers are fundamentally the model's hyperparameters. The encoder takes the input sequence and flows them through a self-attention layer, followed by a feed-forward NN. The output vector is fed into the decoder (which includes an attention layer, too), turning it into an output sequence. The output sequence can be in another language as in the case of machine translation. The output sequence can also be symbols, a copy of the input, etc. The self-attention layer in the encoder helps the model to pay attention to other words in the sequence while encoding a specific word. The attention layer in the decoder focuses on relevant parts of the encoder's output.

To illustrate how attention makes a difference, consider a scenario in which a person is reading some text. Normally, the reader cannot remember the text word by word and simply focuses on the important keywords of the text which represents the key content. The attention mechanism works similarly to our brain in that it takes a sequence of words at the same time and decides which ones are important by attributing different weights to inputs. Fig.2 [13] illustrates the architecture of a

single transformer encoder-decoder pair.

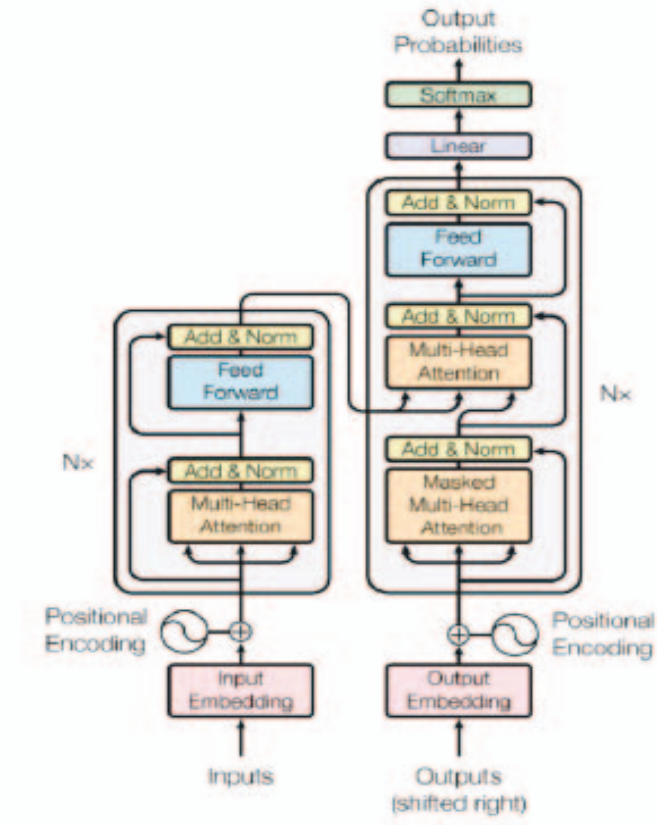


Fig. 2. Single encoder-decoder transformer [13] .

In Fig.2, both encoder (left) and decoder (right) are an order of modules stacked on top of one another ($Nx$ is the number of encoders/decoders). These modules are mostly Multi-Head Attention and Feed Forward layers. Word embeddings of the input and output (n-dimensional vectors) are passed to the first encoder and decoder, respectively. The positional encoding remembers the order of words in the sequence that are fed into the model. Each encoder/decoder pair propagates its output to the next encoder/decoder. The last encoder's output is passed to all decoders.

*3) Bidirectional Encoder Representations from Transformers (BERT):* One of the improved results of transformer is presented by BERT [14]. It achieved state-of-the-art results in a range of machine NLP applications. These include the Q&A task on Stanford Question Answering Dataset (SQuAD), machine translation, understanding human linguistics, Natural Language Inference (NLI), and time series prediction.

The key technical innovation behind BERT is its application of bidirectional training of transformer to language modeling. While a traditional language model reads text from left to right and predicts a token conditioned on the previous tokens, BERT is bidirectionally trained. This means BERT can predict masked tokens conditioned on the rest of the tokens in the sentence. Although BERT is a Transformer-based language model, it does not need a decoder. So, only the encoder part is necessary for BERT.

## IV. An Effective and Reusable NLP Toolkit

### A. Toolkit Overview

Innate human knowledge helps us to differentiate between simple false and true statements or answer questions that people sometimes encounter, such as "can a horse fly to Jupiter?" quickly. However, using computational intelligence to mimic this kind of human response has proven difficult for machines [21]. Recent advances in ML emphasize the importance of NLP as a critical aspect of CI. In much of the first fifty-year history of CI research, progress was at times slow [22] in NLP-related problems. However, when transfer learning [23], and then transformers were introduced to the NLP research community [13], significant breakthroughs have occurred at an accelerated pace [24]. These advances form the basis of an effective toolkit for addressing multiple related issues in computational NLP.

An effective CI NLP toolkit begins with language modeling (LM). Fundamentally, LM is about assigning a probability distribution over sequences of words or tokens. LM is followed by transfer learning (TL) to reuse a pre-trained model on different data distribution and feature space. This serves as the starting point of any particular machine NLP application. Using TL significantly improves the learning process in terms of both time and computational effort through the transfer of knowledge from a related application that has already been well learned to the new one under consideration [25]. Specifically, TL involving transformers, such as BERT [14], have been found effective for a range of machine NLP applications, e.g., multilingual Q&A systems [28], tweet act classifier (speech act for Twitter) [29], and text-based emotion recognition [30]. Other potentially useful NLP tools include (AWD-LSTM) [12], Transformer T5 [31], Transformer XL [32], and Generative Pre-trained GPT [15]. A comparative study [31] found that GPT-2 outperformed Transformer XL in terms of model stability and human-like performance in some NLP applications. However, further study is needed to establish the efficacy of these techniques in a broad sense across NLP tasks. This remains an open research question.

### B. BERT-MLM

Although BERT [14]and its variants show promise for NLP, further development has been undertaken for optimized performance. The original BERT is a language model used to predict masked tokens and next sentence, which has a range of possible applications. Consequently, we need to modify BERT specifically for the machine NLP problems under consideration. Many machine NLP problems can be formulated as classification and Q&A problems. Therefore, we can begin by adding a classification layer on top of the transformer output. After that, we can feed embedded vectors of question and answers into the model separated by

special tokens. Notable variants of BERT that have shown promise include RoBERTa [16] and ALBERT [17]. These are useful for both NLP sentence validation and validity reasoning. RoBERTa is trained longer on more long length sequences of data; comparable ALBERT has fewer parameters.

A specific extension of BERT for the machine NLP problems under consideration involves a training strategy known as Masked Language Model (MLM). MLM involves masking out some of the words in a presented input, which is then followed by conditioning each word bidirectionally to predict the masked words.

Particularly, a random sample of tokens in the input sequence is selected to be replaced with the special token '[MASK]'. Under this arrangement, the objective is a cross-entropy loss on predicting the masked tokens [14]. BERT uniformly selects 15% of the input tokens for possible replacement. Of the selected tokens, 80% are replaced with '[MASK]', 10% are replaced by some randomly selected token drawn from the same vocabulary, and the remaining 10% of the tokens are left unaltered. The following is an algorithm that we have developed to enhance the performance of MLM for use in the present context.

1) Insert two special tokens into each sentence. One of the two special tokens is inserted at the beginning of the sentence while the other is inserted at the end.
2) Replace each token from left to right with the special '[MASK]' token one token at a time. This will result in multiple sentences.
3) Feed these resultant sentences to MLM for predicting the probabilities of the original masked tokens.
4) Normalize the predicted probabilities using softmax activation function in the output layer.
5) Multiply the predicted probabilities of masked tokens for each pair of sentences. This means prediction is made for identifying the sentence with the highest probability of occurrence.

Algorithm 1. Enhanced MLM

As an example to illustrate how Steps 1 and 2 in Algorithm 1 work, consider the following.

1) Insert special tokens into the sentence at the beginning and end: ['[CLS]', 'She', 'eats', 'water', '[SEP]']

2) Replace each token from left to right with '[MASK]':
['[MASK]', 'She', 'eats', 'water', '[SEP]'],
['[CLS]', '[MASK]', 'eats', 'water', '[SEP]'],
['[CLS]', 'She', '[MASK]', 'water', '[SEP]'],
['[CLS]', 'She', 'eats', '[MASK]', '[SEP]'],
['[CLS]', 'She', 'eats', 'water', '[MASK]']

### C. Problem Reformulation

Our idea of improving the performance of the reusable toolkit is to reformulate the input of sentence validation as a binary classification problem to the input of another downstream task, i.e., multiple choice questions. The difference between these two models is the amount of attention paid to the sentences. In the self-attention layer, the encoder looks at other words in the input sentence as it encodes a specific word. For classification models using BERT, RoBERTa, and Albert, we concatenate the two sentences. Then, the self-attention layer attends to each position in the input sequence, including both sentences. Fig. 3 shows a RoBERTa classifier for sentence validation.

Reformulating RoBERTa for multiple choice questions, we feed each sentence to the network individually. Consequently, the attention layer attends to the sequence of words for each individual sentence for gathering useful information. The purpose of this is that it can lead to better encoding for each word. Fig.4 shows a RoBERTa question answering schematic for sentence validity reasoning.

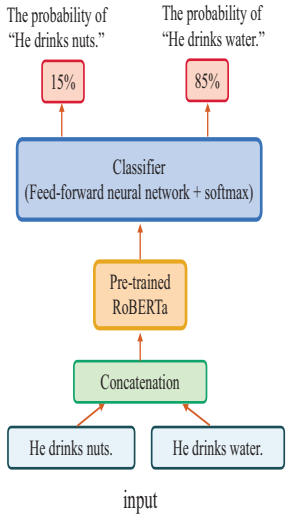


Fig. 3. RoBERTa Classifier Model.

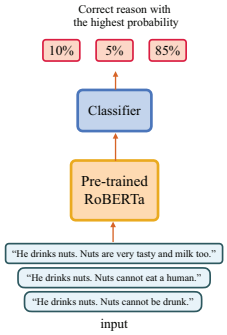


Fig. 4. RoBERTa Question Answering Model.

## V. Experiments

Experiments have been conducted to quantify the performance of a range of tools for both sentence validation and validity reasoning. We used a large benchmark dataset [33] prepared by [34]. The dataset comprises 10,000 and 2,021 human-labeled pairs of sentences ($M$=2) for training and validation models, respectively. After releasing the dev set, we combined the two datasets for training and used the dev set to test our models.

Table I summarizes the results for sentence validation. With accuracy scores of 95% and 96.1%, RoBerta performed the best. The result was especially high when the task was reformulated as multiple choice. However, BERT-MLM did not perform as well as expected. Further analysis revealed that it was not suited to the nature of this particular dataset, which typically consists of short sentences of only a few words. Intuitively, the nature of BERT-MLM is such that it should work much better on longer sentences than the short ones in the benchmark dataset. Further research is necessary to validate this intuition once datasets with suitably longer sentences become available.

TABLE I
Sentence validation results

| Model | Accuracy (%) |
|---|---|
| BERT-MLM | 74.3 |
| BERT classification | 88 |
| Albert classification | 92 |
| RoBerta classification | 95 |
| RoBerta multiple choice | 96.1 |

As the most promising approach, RoBerta was then further developed for sentence validity reasoning. It achieved an accuracy of 93.7% for reasoning with the number of candidate sentences $N$ = 3. These scores achievable by machines represent near-human levels of performance at 99.1% and 97.8% for sentence validation and validity reasoning, respectively. The fact that the human scores are less than perfect indicates that some humans find these tasks occasionally difficult. This observation underscores the challenge in getting machines to do likewise.

## VI. Conclusion

This paper has presented an overview of some of the recent advances in learning techniques that are applicable to computational natural language processing (NLP). Following a thorough review of the relevant literature, we identified that most promising approaches and assembled a reusable machine NLP toolkit from among those promising techniques. We further demonstrated an application of some of those tools toward solving two machine NLP problems: sentence validation and validity reasoning. Experimental results using benchmark data confirmed the usefulness of the tools by achieving near-human performance in solving the two specific machine NLP problems.

The reusable toolkit is not without limitations. In its present form, limitations of the reusable machine NLP toolkit include the following. First, the toolkit has not been tested on beyond the two categories of machine NLP problems described in this paper, namely validating sentences and reasoning about the validity of a given sentence. Second, and while staying with the two categories of NLP problems, the available choices are currently somewhat restricted. Third, the innovative BERT-MLM technique has not been applied to very long sentences due to a current lack of suitable dataset. There is an expectation that BERT-MLM will demonstrate its effectiveness when applied to sentences that are significantly longer than those in the benchmark dataset used in the experiments.

Future research directions include evaluating the reusable machine NLP toolkit for more variety of test data. For example, an interesting research direction is to test the BERT-MLM technique on texts that are made up of long sentences. Another direction is to expand on parameters like $M$ and $N$ by using data augmentation, resampling, or similar techniques. For example, instead of setting $M$ = 2 in sentence validation, we can allow the machine to choose one logical sentence from among several options, where $M-1$ sentences are illogical or invalid. This could be formulated as a multiple choice problem and/or classification. Alternatively, the choice can be one invalid sentence out of several plausible sentences. Likewise, the number of alternative explanations can be increased to a number more than $N$ = 3 to test the toolkit for generalizability.

Other possible future work will involve further development of novel machine learning / deep learning models specifically optimized for computational NLP applications, as well as factorization and streamlining of repetitive steps in a complete machine NLP pipeline or workflow. Finally, another possible future research direction is to adapt and apply the tools and findings reported in this paper to languages other than English.

## References


[1] S. Shekkizhar and A. Ortega, "Revisiting local neighborhood methods in machine learning," 2021 IEEE Data Science and Learning Workshop (DSLW), 2021, doi: 10.1109/DSLW51110.2021.9523409.

[2] N. Shlezinger, J. Whang, Y. C. Eldar and A. G. Dimakis, "Model-based deep learning: key approaches and design guidelines," 2021 IEEE Data Science and Learning Workshop (DSLW), 2021, doi: 10.1109/DSLW51110.2021.9523403.

[3] A.C.M. Fong, Conceptual analysis for timely social media-informed personalized recommendations, 33rd IEEE International Conference on Consumer Electronics (ICCE 2015), Las Vegas, NV, USA, Jan 2015.

[4] A.C.M. Fong and J. Tang, "Evolving social networks in consumer electronics," IEEE Consumer Electronics Magazine, Vol. 2/4, Oct 2013.

[5] Q. Zhang, B. Guo, H. Wang, Y. Liang, S. Hao and Z. Yu, "AI-powered text generation for harmonious human-machine interaction: current state and future directions," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 859-864, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00176.

[6] Y. Sun, "Joint learning of question answering and question generation," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 5, pp. 971-982, 1 May 2020, doi: 10.1109/TKDE.2019.2897773.

[7] K. Chettah and A. Draa, "A discrete differential evolution algorithm for extractive text summarization," 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2021, pp. 1-6, doi: 10.1109/INISTA52262.2021.9548632.

[8] A. M. Turing. “I.—computing machinery and intelligence” In: Mind LIX.236 (Oct. 1950), pp. 433–460. issn: 0026-4423. doi: 10.1093/mind/LIX.236.433.
[9] E. Davis, “Winograd schemas and machine translation,” arXiv preprint arXiv:1608.01884 (2016).
[10] J.J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.” Proceedings of the national academy of sciences 79.8 (1982): 2554-2558.
[11] S. Hochreiter and J. Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.
[12] S. Merity, N.S. Keskar, and R. Socher. “Regularizing and optimizing LSTM language models.” arXiv preprint arXiv:1708.02182 (2017).
[13] A. Vaswani et al., “Attention is all you need.” Advances in neural information processing systems. 2017.
[14] J. Devlin et al., “Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).
[15] T.B. Brown et al. “Language models are few-shot learners.” arXiv preprint arXiv:2005.14165 (2020).
[16] Y. Liu et al. “Roberta: A robustly optimized bert pretraining approach.” arXiv preprint arXiv:1907.11692 (2019)
[17] Z. Lan et al. “Albert: A lite bert for self-supervised learning of language representations.“ arXiv preprint arXiv:1909.11942 (2019).
[18] V. Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” arXiv preprint arXiv:1910.01108 (2019).
[19] D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473 (2014).
[20] C. Wang, S. Liang, Y. Zhang, X. Li, and T. Gao. 2019a. Does it make sense? and why? a pilot study for sense making and explanation. arXiv preprint arXiv:1906.00363.
[21] E. Davis. . Logical formalizations of commonsense reasoning: A survey. Journal of Artificial Intelligence Research, 59:651–723, May, 2017.
[22] E. Davis and L. Morgenstern. Introduction: Progress in formal commonsense reasoning. Artificial Intelligence, 153(1-2):1–12, March 2004. Logical Formalizations and Commonsense Reasoning; Conference date: 01-05-2001 Through 01-05-2001.
[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In NIPS2014.
[24] S. J. Pan and Q. Yang. 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359.
[25] E. S. Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, and Antonio Jose Serrano Lopez. 2009. Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.
[26] O. Mikolov, K. Chen, G. S. Corrado, and J. A. Dean. 2015. Computing numeric representations of words in a high-dimensional space, May 19. US Patent 9,037,464.
[27] J. Pennington, R. Socher, and C. D. Manning. “Glove: Global vectors for word representation,” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532-1543, 2014.
[28] N. T. M. Trang and M. Shcherbakov, “Vietnamese question answering system from multilingual BERT models to monolingual BERT model,” 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), 2020, pp. 201-206, doi: 10.1109/SMART50582.2020.9337155.
[29] T. Saha, S. Ramesh Jayashree, S. Saha and P. Bhattacharyya, ”BERT-Caps: a transformer-based capsule network for tweet act classification,” IEEE Transactions on Computational Social Systems, vol. 7, no. 5, pp. 1168-1179, Oct. 2020, doi: 10.1109/TCSS.2020.3014128.
[30] A. F. Adoma, N. -M. Henry and W. Chen, “Comparative analyses of Bert, Roberta, Distilbert, and Xlnet for text-based emotion recognition,” 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020, pp. 117-121
[31] I. Ganguli, R. S. Bhowmick, S. Biswas and J. Sil, “Empirical auto-evaluation of Python code for performance analysis of transformer network using T5 architecture,“ 2021 8th International Conference on Smart Computing and Communications (ICSCC), 2021, pp. 75-79, doi: 10.1109/ICSCC51209.2021.9528123.
[32] A. Das and R. M. Verma, “Can machines tell stories? a comparative study of deep neural language models and metrics,“ IEEE Access, vol. 8, pp. 181258-181292, 2020, doi: 10.1109/ACCESS.2020.3023421.
[33] Benchmark commonsense reasoning dataset. Available at https://github.com/wangcunxiang/SemEval2020-Task4-Commonsense-Validation-and-Explanation/tree/master/ALL%20data
[34] C. Wang, S. Liang, Y. Jin, Y. Wang, X. Zhu, and Y. Zhang, “SemEval-2020 Task 4: Commonsense Validation and Explanation”, Proc. 14th Workshop on Semantic Evaluation, pp. 307–321, Barcelona, Spain, December 2020.