Multimodal Surprise Adequacy Analysis of Inputs for Natural Language Processing DNN Models

Seah Kim
School of Computing
KAIST
Daejeon, Republic of Korea
bbaa2837@kaist.ac.kr

Shin Yoo
School of Computing
KAIST
Daejeon, Republic of Korea
shin.yoo@kaist.ac.kr

***Abstract*—As Deep Neural Networks (DNNs) are rapidly adopted in various domains, many test adequacy metrics for DNN inputs have been introduced to help evaluating, and validating, trained DNN models. Surprise Adequacy (SA) is one such metric that aims to quantitatively measure how surprising a new input is with respect to the data used to train the given model. While SA has been shown to be effective for computer vision tasks such as image classification or object segmentation, its efficacy for DNN based Natural Language Processing has not been thoroughly studied. This paper evaluates whether it is feasible to apply SA analysis to DNN models trained for NLP tasks. We also show that the input distribution captured in the latent embedding space can be multimodal[1] for some NLP tasks, unlike those observed in computer vision tasks, and investigate if catering for the multimodal property of NLP models can improve SA analysis. An empirical evaluation of extended SA metrics with three NLP tasks and nine DNN models shows that, while unimodal SAs perform sufficiently well for text classification, multimodal SA can outperform unimodal metrics.**

***Index Terms*—Deep Learning, Natural Language Processing, Software Testing**

I. Introduction

Testing Deep Neural Network (DNN) models have received much attention [1]–[4] as DNN models are being incorporated into various software systems [5]. In particular, the successful application of DNN computer vision technology in safety critical domains such as autonomous driving [6] and medical imaging [7] adds urgency to the need to develop effective testing techniques for DNNs.

While the recently proposed test adequacy metrics [1]–[4], as well as data augmentation [8] and model retraining techniques [9], have contributed to the effectiveness of DNN testing, most of the literature concern image recognition tasks, especially image classification, when evaluating proposed techniques. Consider test adequacy metrics for DNNS: almost all existing metrics – Neuron Coverage [1], Strong Neuron Activation Coverage (SNAC) [3], and Surprise Adequacy (SA) [4] – have been evaluated for image classification, using benchmarks such as MNIST [10] and CIFAR10 [11].

Testing of DNN models in other domains – for example, speech recognition or Natural Language Processing (NLP) – has, in contrast, received relatively less attention. In this paper, we attempt to expand the scope of DL testing techniques to DNN models trained for NLP tasks. Since the success of deep learning has been partially attributed to its capability to *learn* latent features that are crucial for the given task [5], we posit that an input domain other than images may exhibit different characteristics from those observed with image recognition DNNs. Our primary goal is to investigate whether it is feasible to apply SA to DNNs trained for NLP tasks, and extend SA metrics specifically for NLP tasks if necessary.

We choose one of the recently proposed test adequacy metrics for DNNs, Surprise Adequacy (SA) [4], to study the characteristics of latent features of NLP tasks. Intuitively, SA is a quantitative measure of how Out-of-Distribution (OOD, i.e., surprising) the Activation Trace (AT)[2] of a given input is with respect to the distribution of inputs seen during training. The more OOD a given input is, the more likely that a DNN will fail to process the input correctly. In the context of NLP, if a text input is highly surprising, the model is more likely to predict imprecisely. The prediction of imprecision is especially important for NLP, as the only available test oracles for NLP tasks are usually human labelling that is very costly. For instance, to evaluate the QA model performance against new pairs of a context paragraph and a query, a human must read both the paragraphs and queries and decide whether the answer is correct, incurring a huge cost. Guided by SA, we expect to prioritise inputs that are likely to induce failures, thereby allowing more efficient use of testing resources [12]. Consequently, we aim to improve testing of AI and NLP based systems, as well as SE tasks that use NLP techniques.

Our study of three different NLP tasks and nine trained models show that the distributions of AT vectors for some NLP tasks are multimodal, unlike those observed from image recognition DNNs. This observation, in turn, allows us to extend existing SA metrics to handle multimodal distributions better, because the degree of ATs being OOD is directly coupled to their distribution. In addition to investigating the

[1]In this paper, we use the term *multimodal* to refer to multimodal distributions, i.e., distributions with more than one mode, in the statistical context. It does not mean multiple data domains.

[2]Activation Trace (AT) [4] is the collection of neuron activation values from selected neurons, typically all neurons in a single chosen layer. ATs capture the behaviour of a DNN w.r.t. given input, similarly to execution traces capturing the behaviour of traditional programs w.r.t. a given input.

feasibility study, we also evaluate whether the multimodal SA for NLP tasks can improve the accuracy of the analysis.

The empirical evaluation includes three different NLP tasks (text classification, sequence labelling, and question answering), nine different DNN models (three per each task), and five different NLP benchmark datasets. We compare both the existing SA metrics and the multi-modal variants we propose against active learning metrics, which can also prioritise inputs that the model is likely to find difficult to handle.

The technical contributions of this paper is as follows:

- We investigate the feasibility of SA analysis for DNNs trained for multiple NLP tasks: text classification, Named Entity Recognition (NER), and Question Answering. Existing DNN testing literature largely focused on image recognition tasks only.
- We show that the distribution of ATs in DNNs trained for some NLP tasks are multimodal. We also show that, when the multimodal nature of NLP models is not properly considered, existing unimodal SA metric can perform sub-optimally, and evaluate two multi-modal variants of two existing metrics: MMLSA (Multi-Modal Likelihood-based SA) and MMDSA (Multi-Modal Mahalanobis distance based SA).
- We conduct a large scale empirical evaluation of our new multimodal SA metrics using nine different DNN models trained for three NLP tasks. The results show that 1) unimodal SAs are sufficient for text classification, 2) multimodal SAs can outperform unimodal metrics for sequence labelling, and 3) Question Answering task, for which no other input prioritisation metric exists, remains challenging for SA analysis.

The rest of this paper is organised as follows. Section II presents the background information about Surprise Adequacy (SA) metrics. Section III-A introduces multi-modal variants of existing SA metrics. Section IV presents experimental setup, introduces the research questions, and describes our baseline metrics from active learning metrics. Section V presents the results of our empirical evaluation. Section VI discusses threats to validity, and Section VII presents related work. Finally, Section VIII concludes.

II. Background: Surprise Adequacy for DNNs

This section contains the description of Surprise Adequacy(SA) [4] used in this paper.

A. Existing SA Metrics

SA is a test adequacy for DNN inputs which that assesses how *surprising* an input is to the model. The basic assumption is that the more familiar an input is to the model (in comparison to the data observed during training), the more likely the model will behave correctly. Kim et al. [4] measure the familiarity of a never-seen-before input by comparing the pattern of neuron activation to the patterns observed during training: instead of observing all neurons in a DNN, they opt to choose a specific layer and take all neuron activation values, which is called Activation Traces (ATs). Once a model is trained, ATs can be obtained for both a new input as well as all inputs in the training data. Subsequently, SA of an input is measured by calculating the similarity between the AT of the given input, and the ATs of the training data. Kim et al. originally introduced Likelihood-based SA (LSA) and Distance-based SA (DSA) [4]. Mahalanobis distance based SA (MDSA) was introduced later by Kim and Yoo [13].

1) Likelihood-based SA: LSA performs Kernel Density Estimation (KDE) over the set of AT vectors obtained from the training data (T). With a Gaussian kernel function K and the AT from a new input, x, LSA computes the density as:

$$\hat{f}(x) = \frac{1}{|T|} \sum_{\alpha_i^T \in T} K(x - \alpha_i^T) \tag{1}$$

LSA is then computed as the negative log of density:

$$LSA(x) = -\log \hat{f}(x) \tag{2}$$

The lower the density of new input is, the more surprising the input is to the model.

2) Distance-based SA: DSA is a measure of a surprise based on the classification boundaries. Given a new input, DSA compares the Euclidean distance between AT of the input and AT of the closest training-set input in the same classification class, to the Euclidean distance between AT of the input and AT of the closest training-set input in another classification class. The ratio of these two distances tells us how close to the boundaries the input is. The closer to the class boundary the input is, the more surprising the input is to the model.

3) Mahalanobis Distance based SA: Mahalanobis Distance measures the distance between a probability distribution and a single data point [14]. Given a new vector x and a set of mean values(μ) and covariance matrix C from training data(T), MDSA is defined as:

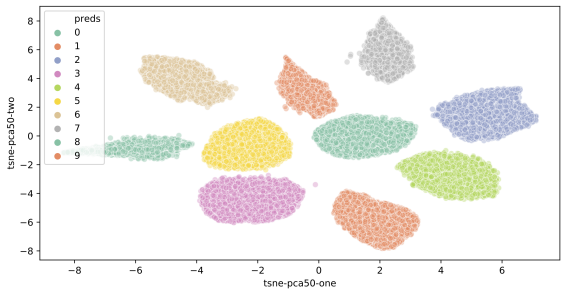
$$MDSA(x) = \sqrt{(\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu})} \tag{3}$$

The farther away an AT vector of a new input is from the training data distribution, the more surprising the input is likely to be to the model. Unlike DSA, MDSA can be applied to non-classification models, as it does not depend on the concept of classification boundaries.

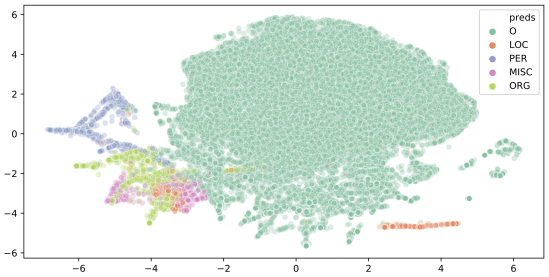
B. Distribution of ATs from NLP Models

The intuition behind SA metrics can be applied to all input and task domains: the AT vectors represent the internal behaviour of the DNN model under consideration, and the more OOD a new input AT is w.r.t. to the training data, the more difficult the model will find it. However, we argue that the detailed method of measuring the OOD-ness can be affected by the shape of the AT vector distribution.

Consider the visualisation of AT vectors from two different DNN models in Figure 1. We first apply PCA to AT vectors of training data and reduce the dimension to 50; subsequently, we use t-SNE [15] to visualise the vectors in 2D. The AT



(a) Visualisation of ATs from ResNet Classifier for CIFAR-10



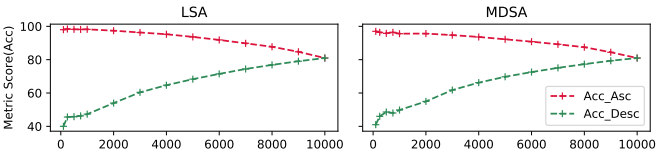
(b) Visualisation of ATs from S-LSTM for CoNLL-2003

Fig. 1: Comparison of ATs from ResNet Image Classifier and TENER Sequence Labelling Model

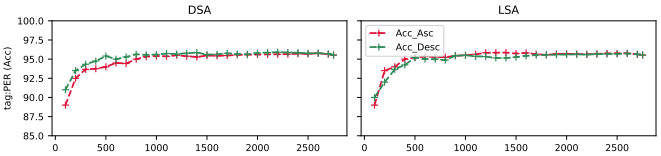
vectors shown in Figure 1a are taken from a ResNet Image Classifier [16] trained with CIFAR-10 dataset [11]. AT vectors shown in Figure 1b are taken from the S-LSTM, a Graph LSTM based sequence labelling model for the NER task [17]. The colour of each dot represents the class label: each vector corresponds to a token (i.e. an entity) in a sequence.

ATs from ResNet form distinct clusters that align with class labels, which is in line with the high training accuracy of the model. More importantly, clusters show clear unimodal distributions. ATs from S-LSTM, on the other hand, show multimodal distribution: we can observe sub-groups within the same label. An AT that is close to the centroid of a sub-group can still be far away from the class centroid.

We suspect that the multimodality can actually affect the accuracy of SA analysis. Figure 2 compares the result of SA correlation analysis (similar to those by Kim et al. [4]) for both CIFAR-10 images classified by ResNet and the PER tags of CoNLL-2003 classified by S-LSTM. In all plots, we gradually expand a set of unseen test inputs by adding them in the ascending (red) and descending (green) order of their SA values, and plot the accuracy of the model prediction. By definition of SA, we expect the red line to start at a accuracy and gradually come down, as we start with less surprising inputs and gradually add more surprising ones. Similarly, we expect the green line to start at the low accuracy and gradually go up, as we start with more surprising inputs and gradually add less surprising ones. For ResNet, the results show that SA behaves as expected. However, for S-LSTM,



(a) SA Correlation Analysis for ResNet/CIFAR-10



(b) SA Correlation Analysis for S-LSTM/CoNLL-2003

Fig. 2: Comparison of SA Correlation Analysis

the degree of surprise does not correlate well with the model performance. This result motivates our approach with multimodal SA analysis. We suspect that the inaccurate measure of distance and familiarity is affecting the analysis, and posit that more accurate measurements based on multimodal distribution can improve the analysis.

Since we cannot fully explain how the latent features are learnt by DNNs [18], it is difficult to fully explain where the multimodality in NLP models comes from. However, we can cautiously propose some relevant characteristics of NLP tasks: unlike images, words in natural language are inherently more discrete, with each token conveying different meanings. In addition, semantic meanings of words are also highly contextual: not only the word itself, but also its relative position in the sequence (i.e., sentence) can affect its meaning. Combined, these factors suggest that the distribution of AT vectors from NLP models may be more complicated than those from image recognition models.

III. Multimodal SA and NLP Models

This section first introduces two new variants of SA: Multimodal Likelihood based SA(MLSA) and Multimodal Mahalanobis distance based SA (MMDSA). Subsequently, we describe how Activation Traces (ATs) are extracted from each of the NLP models we study.

A. Multimodal Surprise Adequacy

The visualisation of ATs shown in Section II-B suggests that ATs extracted from NLP tasks can be multimodal, i.e., even ATs that belong to what is thought to be similar inputs can show more than one mode. Ignoring the multimodal nature of ATs can adversely affect SA analysis. For example, the existing MDSA assumes unimodal distributions of ATs: an AT that is actually very close to one of the modes can be assigned relatively higher distance from the unimodal centroid.

The key intuition behind our multimodal variants of SA metrics is that we identify different modes and measure the distance (or similarity) of a new test input to only one of the modes. Instead of using Kernel Density Estimation, we can use Gaussian Mixture Model (GMM) to represent the multimodal

distribution of ATs from the training data. Similarly, instead of using the global mean and covariance for Mahalanobis distance, we can cluster the multimodal ATs from training data, and only compute the Mahalanobis distance between the input AT and the mean and covariance from the closest cluster of training data ATs. The following subsections formally define these multimodal variants of SA metrics.

1) Multimodal Likelihood-based SA (MLSA): Gaussian Mixture Model (GMM) is essentially a clustering algorithm in which each cluster is modelled as a Gaussian distribution. We can train a GMM using the AT vectors of the training data, and represent the membership of each AT vector to the clusters using a mixture of Gaussian densities. Given a number of cluster, k, as well as the mean, μ, covariance, C, and weight, w, for each Gaussian distribution N, GMM produces the probabilistic density function, f, as follows:

$$f(x) = \sum_{j=1}^{K} w_j N(x|\mu_j, C_j) \tag{4}$$

Similarly to LSA, we define MLSA for a input x to be the negative of the log of density:

$$MLSA(x) = -\log(f(x)) \tag{5}$$

The KDE used by the original LSA can be considered as a specific case of GMM where all kernels share the same parameters. In contrast, GMM allows each Gaussian distribution to be different from others: we expect GMM based MLSA to capture the multimodal distribution of training data ATs more accurately.

2) Multimodal Mahalanobis Distance-based SA (MMDSA): The original MDSA is a surprise adequacy which utilises the Mahalanobis distance between the given input AT and the unimodal distribution of the training data ATs. The multimodal version of MDSA, which is called MMDSA, first applies k-means clustering [19] to the training data ATs. Subsequently, given an input AT, MMDSA is defined as the Mahalanobis distance between the input AT and the distribution of ATs in the closest cluster, using the mean values and the covariance matrix obtained from that cluster.

Deciding the optimal number of clusters is a fundamental challenge in clustering [20], especially for the k-means clustering that takes the number of clusters to generate as an input. We adopt the widely used Silhouette index [21] to determine k without depending on any external information or analysis. *Silhouette index* evaluates the quality of clustering results by measuring how a data point is similar to its assigned cluster compared to other clusters. More formally, the Silhouette index for an input x assigned to cluster C_i is defined as:

$$S(x) = \frac{B(x) - A(x)}{\max(A(x), B(x))} \tag{6}$$

where $A(x)$ is the mean of intra-cluster distance between x and other members of C_i, and $B(x)$ is the mean of inter-cluster distance between x and its nearest neighbour cluster.

$$A(x) = \frac{1}{|C_i| - 1} \sum_{x \in C_i, x \neq y} d(x, y) \tag{7}$$

$$B(x) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{y \in C_k} d(x, y) \tag{8}$$

We compute the mean silhouette score over all ATs while varying the clusters' number and determine the number of clusters k with the highest score.

B. Task-specific AT Extraction for NLP Models

This paper studies three NLP tasks: Text Classification (CLS), Named Entity Recognition (NER) and Question Answering (QA). Text classification simply aims to classify given natural language text using the pre-determined set of labels. For example, sentiment analysis aims to label given sentences as having positive or negative sentiments. Named Entity Recognition aims to classify each token in a given natural language sentence as named entities, such as names, locations, and quantities. Finally, given a context paragraph and a question sentence, QA aims to choose the part of the given context paragraph that contains the answer.

A major difference between image recognition tasks and NLP tasks is the variability of input size in NLP. Unlike images that can be rescaled to a fixed size, the length of natural language data is inherently variable. As the number of words differs between separate inputs, it is impossible to aggregate ATs from different sequences into a single fixed dimension. In turn, this makes the straightforward computation of KDE or Mahalanobis distance difficult.

To overcome the input dimension variability and to minimise the information loss, we need to design customised AT extraction method for each NLP task at hand. Simply clipping the internal product of a NLP model into a fixed size based on a single universal heuristic is not ideal, as NLP models are expected to label either each word, or an entire sentence, depending on the task. Extracting a sequence-level AT vector by taking the average of AT vectors of all included tokens can cause loss of information about token-level differences.

Our aim is to minimise the information loss in ATs, while capturing fixed dimension ATs for the given task. Text Classification (CLS) is the most straightforward case, as any text classification model should aggregate all latent features into a single classification result for the entire sentence: inputs are of variable length, but text classification models inherently remove the variability during the process of classification. Consequently, we can extract ATs in a similar way to image recognition DNNs. NER, on the other hand, aims to identify the spans in the sequence that belong to different entities, and also assign the labels to each entity. Consequently, we look at NER task as a collection of smaller classification tasks. While inputs for NER are given as sentences, we analyse test adequacy at entity level.

Question Answering (QA) models have the most complicated internal structure: models are expected to understand the given context and the question, and to find the location of

the corresponding answer embedded in the context. Therefore, we argue that the internal representation for the *span* of the answer, passed through multiple neural layers, conveys the essence of QA model behaviour. However, the length of answer spans is also variable. Instead of taking ATs of all the tokens in the answer span, we simply concatenate ATs of the beginning and the end of the answer span.

IV. Experimental Setup

This section introduces our research questions and describes the experimental setup of our empirical evaluation. We describe the baseline metrics from active learning literature, and introduce the NLP task benchmarks and the DNN model architectures we use.

A. Research Questions

We present the following research questions and try to answer them with the results of our empirical evaluation.

1) RQ1. Effectiveness of Unimodal SA Metrics: How effective are the existing unimodal SA metrics for DNNs trained for NLP tasks? With RQ1, we investigate whether SA works as a test adequacy criterion for DNNs trained for NLP tasks. Following Kim et al. [4], we compare the results of the correlation analysis between SA and AL metric values and model accuracy.

2) RQ2. Effectiveness of Multimodal SA: Are multimodal SA metrics more accurate for DNNs trained for NLP tasks? With RQ2, we evaluate the newly proposed multimodal SA metrics, MLSA and MMDSA. We perform the same correlation analysis with multimodal SA metrics and compare their accuracy to unimodal metrics.

B. Active Learning Methods

We use four metrics from Active Learning (AL) literature as the baseline to compare SA metrics to. In Active Learning, the learner can actively request labelling a specific new input, to alleviate the high cost of labelling large sets of data [22]. While Active Learning is motivated by the need to lower the cost of learning, and SA as test adequacy is motivated by the need to identify inputs that are most likely to reveal unexpected behaviour of the given model, both provide a technique to prioritise unseen inputs in the order of their *difficulty* from the perspective of the model. Consequently, the prioritisation metrics used by AL can be used as a baseline to compare SA metrics to. Compared to AL metrics, other existing test adequacy metrics for DNNs (e.g., Neuron Coverage [1]) are designed for sets of inputs, and cannot be directly used as a measure of how surprising the model will find a single input. This is why we compare SA metrics, both unimodal and multimodal, to the AL metrics.

Entropy is the most wide-spread method for measuring the uncertainty [23]. Given an input x, a set of training data, D, and a set of class labels, C, we can compute the entropy of the classification result, y, as follows:

$$H(y \mid x, D) = -\sum_{c \in C} P(y = c \mid x, D) \cdot \log P(y = c \mid x, D) \tag{9}$$

To convert entropy as a measure of OOD-ness, we take the negative of entropy.

Following two uncertainty acquisition methods are proposed by recent approaches to combine Bayesian deep learning into active learning methods [24]–[26]. AL with Bayesian deep learning has achieved considerable reduction in the amount for labelling in training deep neural networks, by performing random Monte Carlo dropout to the model.

Bayesian Active Learning by Disagreement (BALD) represents the mutual information between data points and the model weights [27]. The mutual bond between the model predictions for a given data and the model parameters can tell us the degree of confidence the model has about the output. BALD subtracts the expectation of the entropies over the posterior of model parameters from the entropy over model predictions. Given a set of dropout models, M, model weights w_m for $m \in M$, a set of class labels, C, and training data D, we can compute:

$$\begin{aligned} &I(y; w \mid x, D) = H(y \mid x, D) \\ &-\frac{1}{|M|} \sum_{m \in M} \sum_{c \in C} -P\left(y = c \mid x, w_m\right) \log P\left(y = c \mid x, w_m\right) \end{aligned} \tag{10}$$

Variation Ratio aims to measure the level of model confidence through the use of random dropouts [24], [28]. After independently applying $|M|$ dropout masks, let f_m denote the number of dropout predictions that agrees with the model prediction without any dropout. We measure the proportion of predictions that disagree with the model prediction as follows; the larger the value is, the more uncertain the model is.

$$v(y) = 1 - \frac{f_m}{|M|} \tag{11}$$

Following [24], [29], we independently draw $|M| = 100$ dropout masks to estimate BALD and Variation Ratio.

Maximum Normalized Log-Probability (MNLP) is an uncertainty-based, cost-aware input prioritisation strategy for NER [29]. MNLP aggregates the probability assigned by the model during the decoding step, and divides it with the length of sequence to reflect the higher labelling efforts required by longer sequences. Given an input x, and probabilities for n class labels, MNLP is defined as[12]:

$$M(x) = -\max_{y_1, \ldots, y_n} \frac{1}{n} \sum_{i=1}^{n} \log P(y_i \mid y_1, \ldots, y_{n-1}, \{x_{ij}\}) \tag{12}$$

[12]Notation x_{ij} denotes the one-hot encoding of jth character in the ith word of input sequence x [29].

TABLE I: List of models used in study

Task	Evaluation Metric	Dataset	Description	Model	Performance
Text Classification	Accuracy	IMDB	Sentiment classification on film reviews with binary labelling (positive/negative) on 50K multi-sentences	L-mixed[3]	94.55
				Transformer[4]	83.76
		AG-News	200K news articles with 4 topic labels	L-mixed	94.21
		SST-5	215,154 phrases from film reviews with fine-grained (five-way) sentiment labels	Transformer	44.6
Named Entity Recognition	F1	CoNLL-03	22,137 English sentences labelled with NER tags	LSTM-CRF[5]	86.77 (97.35)
				S-LSTM[6]	90.74 (98.09)
				TENER[7]	91.43 (98.15)
Question Answering	Exact Match (F1)	SQuAD 1.1	QA dataset with 100K crowd-sourced questions for Wikipedia articles. Answer to the question is guaranteed to be inside of the context	BiDAF[8]	77.18 (67.31)
				DocQA[9]	80.78(71.59)
				QANet[10]	79.83(70.65)
				FusionNet[11]	81.94(72.94)

Note that MNLP score is computed for the entire input sequence x: we map the sequence level score to each tagged token to map MNLP to tag level analysis.

C. Datasets and Models

Table I shows the list of datasets and models we conducted our empirical evaluation with.

1) Text Classification: Three datasets are used for evaluating SA for the text classification task: IMDB [30] and SST-5 [31] for sentiment analysis of film reviews and AG-News [32] for topic classification of news articles. L-mixed [33] is a single-layer BiLSTM classifier which shows good performance on IMDB and AGNews dataset. Transformer is a self-attention based model that has shown state-of-the-art performance in many NLP tasks [34]: we exploit the encoder of Transformer to evaluate IMDB and SST-5 dataset.

2) Sequence Labelling: CoNLL-2003 [35] is a widely studied dataset for NER. It contains labelled sequences with four different named entity types: person, location, organization, and miscellaneous. To evaluate SA for the NER task, we use three models: LSTM-CRF [36], S-LSTM [17], TENER [37]. Each model uses different architectures to encode the character and word-level information (LSTM-CRF uses Bi-LSTM, S-LSTM uses Graph LSTM, and TENER uses Transformer), but all use Conditional Random Field (CRF) layer to model the tagging decision.

3) Question Answering: To evaluate SA metrics for QA models, we use version 1.1 of the Stanford Question Answering Dataset (SQuAD) [38], which is a widely used to study Question Answering. A context paragraph in SQuAD corresponds to one paragraph of a Wikipedia article; answers for each question are guaranteed to be a segment of the context paragraph. We evaluate four different QA models with end-to-end architectures: the BiDAF model [39] by Seo et al., two subsequent developments of BiDAF that are DocQA [40] and QANet [41], and finally the FusionNet [42]. The BiDAF model as well as its variants uses a bi-directional attention flow layer, which provides a single representation that reflects both the context and query words. This architecture, in turn, allows easy extraction of AT vectors. FusionNet introduces a new concept called "history of words" and exploits different levels of fusion information between context and question to capture the complete content.

[4]https://github.com/DevSinghSachan/ssl_text_classification
[5]https://github.com/huggingface/transformers
[6]https://github.com/Hironsan/anago/
[7]https://github.com/leuchine/S-LSTM
[8]https://github.com/fastnlp/TENER
[9]https://github.com/allenai/bi-att-flow
[10]https://github.com/allenai/document-qa
[11]https://github.com/NLPLearn/QANet
[12]https://github.com/felixgwu/FastFusionNet

D. Model Training & Environment

For BiDAF, DocQA, and FusionNet models for the QA task, as well as the L-mixed model for the text classification of the IMDB dataset, we use pre-trained model weights provided with the original publications. For all other studied models, we could not obtain pre-trained weights, and instead trained our own models using the official implementations made publicly available with the original publications. All models have been trained using a machine equipped with an Intel Core i7-8700K CPU, 32GB of RAM, and an NVidia RTX2080 GPU.

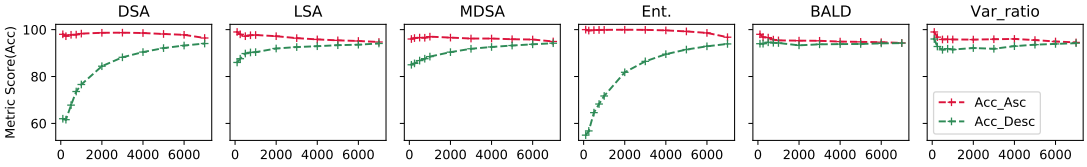
E. Evaluation Metrics

For each NLP task, we adopt a widely used, standard evaluation metric. For text classification task, we simply use accuracy (i.e., the percentage of correct predictions out of total inputs). For NER, we adopt the standard F1 score, which is the harmonic mean of the precision and the recall metric computed against the ground truth labels. For Question Answering, typically Exact Match (EM) and F1 are used. Exact Match (EM) which is the percentage of questions for which the given model produces the exactly correct answers.

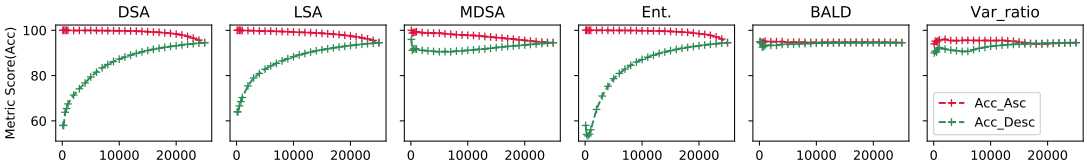
V. Results

A. RQ1. Effectiveness of Unimodal SA Metrics

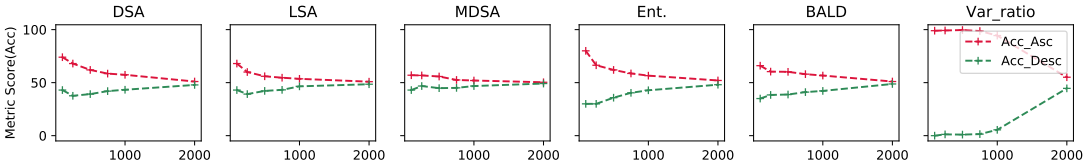
Figure 3 shows the result of correlation analysis for the text classification task. We show the results from L-mixed



(a) Lmixed(AGNews)



(b) Lmixed(IMDB)



(c) Transformer(SST-5)

Fig. 3: Correlation between various SA/AL metrics and Text Classification Model Accuracy for Test Inputs: red lines show accuracy for a set of inputs that gradually incorporates new test inputs in the ascending order of their SA (expected go from high to low accuracy, as the inputs change from easy to hard), whereas green lines show a set of inputs expanding in the descending order of SA (expected to go from low to high, as the inputs change from hard to easy).

models applied to AGNews and IMDB datasets, as well as the Transformer applied to SST-5 due to the limited space; Transformer applied to IMDB dataset shows a similar trend.

As described in Section II-B, we expect the accuracy shown by red lines in the correlation analysis to monotonically decrease as inputs given to the model change gradually from less surprising (i.e., easier to handle) to more surprising (i.e., harder to handle); we expect the opposite trend for the green lines. For all unimodal SAs, the observed trends satisfy our expectations. Inputs with higher SA values are harder to classify correctly. Entropy metric also performs well for all models and datasets. However, BALD and Variation Ratio fail to prioritise inputs that make the model misbehave: there are little differences between accuracies of the red and the green.

The notable exception is the SST-5 dataset, for which SA metrics and Entropy perform worse than BALD and Variation Ratio. The low test set accuracy around 50% suggests that the model is either insufficiently trained, or over-fitted to the training data, when using the configuration included in the publicly available implementation. In either of such cases, the distribution of the training data AT vectors cannot be reliably used as a reference to measure the OOD-ness of a new input.

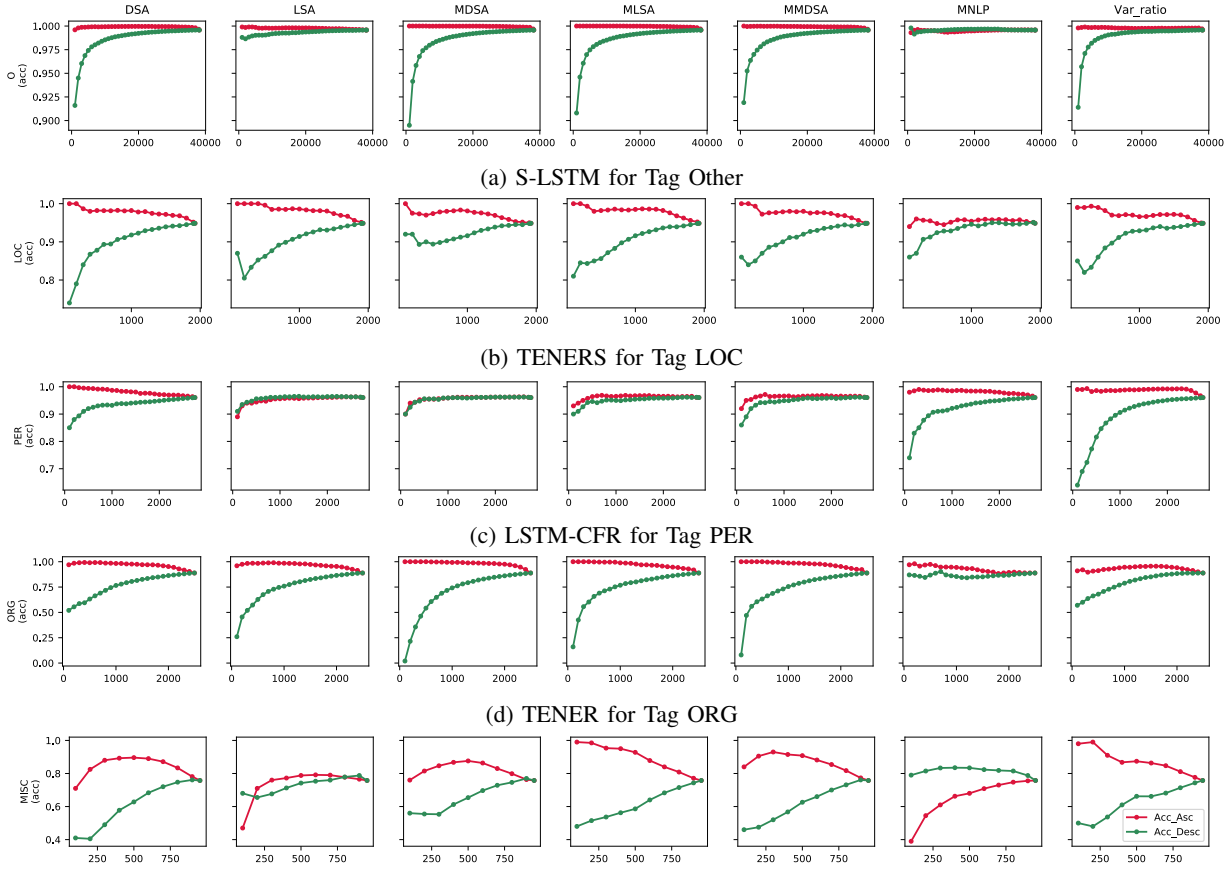
While SA metrics perform well for text classification models, the results are suboptimal for other two tasks. See the plots of unimodal SA metrics in Figure 4, which contains the correlation analysis for tag level NER accuracy from different models. Unimodal SA metrics show suboptimal behaviours, such as failing to distinguish easy and difficult inputs (LSA in Figure 4a, MDSA in Figure 4d), and inverted red and green lines (LSA in Figure 4d and 4e). Interestingly, the AL metric MNLP also shows suboptimal behaviour (Figure 4a and 4e). We suspect that assigning sequence level metric value to individual tokens adds too much noise to MNLP analysis. In comparison, DSA and Variation Ratio perform reliably well, but note that both are relatively expensive methods.

Answer to RQ1: For text classification task, which we expect to be both relatively easy, and of a unimodal nature due to the single unified classification layer, existing unimodal SA metrics not only work well but also can outperform Active Learning metrics. However, for more complicated NER task with multimodal neuron activation, unimodal SA metrics can show suboptimal behaviour.

B. Effectiveness of Multimodal SA

Let us turn to multimodal SA metrics for NER and QA models. For NER models, consider the plots for MLSA and MMDSA in Figure 4. In many cases, the multimodal versions show improved behaviour compared to their unimodal counterparts. For example, MLSA can reliably distinguish OOD inputs for the Other tag with S-LSTM model (see LSA and MLSA in Figure 4a), or untwist crossed accuracy lines (see LSA and MLSA in Figure 4e).

The results from QA models are shown in Figure 5. Since no AL metric is available for QA models, we simply compare unimodal and multimodal SA metrics here. Also note that DSA is not applicable here as QA cannot be formulated as a classification problem. In general, we can observe that BiDAF and its variants – DocQA and QANet – are especially



(a) S-LSTM for Tag Other

(b) TENERS for Tag LOC

(c) LSTM-CFR for Tag PER

(d) TENER for Tag ORG

(e) S-LSTM for Tag MISC

Fig. 4: Correlation between various SA/AL metrics and Tag Level Test Accuracy of Various NER Models

challenging to SA analysis. However, there are notably suboptimal behaviour from unimodal SA metrics for these models. For example, with the DocQA model, inputs with high LSA values can actually show higher EM score than inputs with low LSA values. With BiDAF model, MDSA fails to produce any difference in EM or F1 between high and low SA inputs. In both cases, while going multimodal does not solve all the issues, they ease the severity of issues (MLSA for DocQA, or MMDSA for BiDAF).

Answer to RQ2: Multimodal SA metrics can produce better results for sequence labelling models trained for NER tagging. QA task is challenging for both uni- and multimodal SA metrics, although going multimodal does bring some benefits.

VI. Threats to Validity

Threats to internal validity concerns any factors that could have interfered with the our experimentation and measurements. We only use the publicly available original implementations of the studied NLP models, and reuse their pre-trained weights, whenever possible. Threats to external validity concern any factors that may limit the generalisation of our claim. The relative merits of studied metrics are only valid within the scope of our experimentation. Other models with different DNN architectures, or other models trained with different datasets, may respond differently to the studied metrics. Our results also depend on the specific choice of AT extraction points: ATs captured from different locations may show different behaviour. The randomness in clustering algorithms can also affect our results. To mitigate the effect of non-determinism, we repeat MLSA and MMDSA computation five times and perform the accuracy correlation analysis using the average of five runs for each input. Threats to construct validity concern any potential misuse or misinterpretation of measured metrics. To mitigate this, we only use standard evaluation metrics that are widely used in the literature for each studied NLP task.

VII. Related Work

The very first work on DNN test adequacy is DeepXplore [1], which introduced Neuron Coverage (NC). Given a threshold for neuron activation value and a set of test inputs, NC is the percentage of neurons that have activated above threshold by at least one input during testing. Higher NC means that more neurons have been activated, pointing to more diverse model invotations. Subsequently, DeepGauge [3] proposed a number of additional test adequacy metrics that aim to improve upon NC. The k-Multisection Neuron Coverage (kMNC) replaces the activation threshold with buckets of

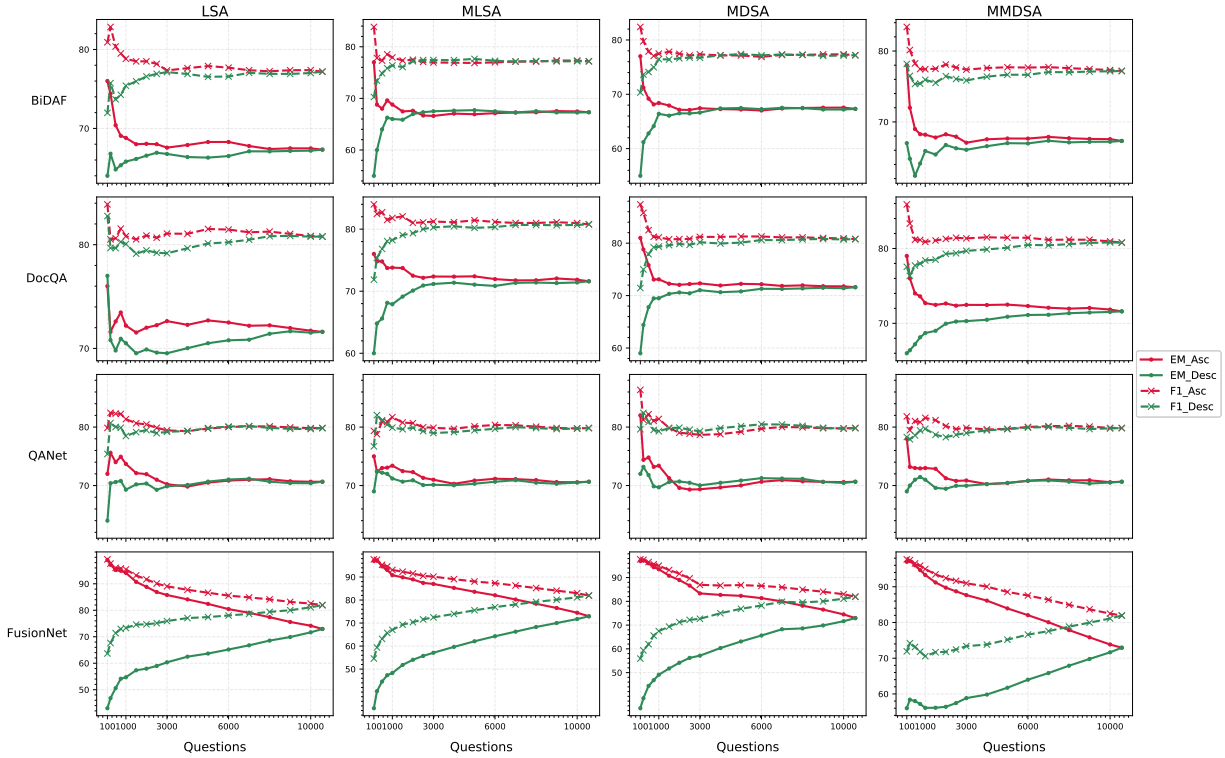


Fig. 5: Correlation between various SA metrics and QA Model Accuracy for Test Inputs

activation values, and computes the percentage of buckets that contain at least one activation during testing. While kMNC allows us to capture neuron activation at a much finer granularity, it remains vulnerable for the analysis of a single input, as most inputs are likely to fill one out of k buckets per neuron, resulting in overall kMNC value of roughly $\frac{1}{k}$. Strong Neural Activation Coverage (SNAC), on the other hand, computes the percentage of neurons are activated beyond the range observed during training. SNAC approximately captures the number of out-of-distribution episodes at the neuron level. However, its criterion is Boolean (out of bound or not) and corser than Suprise Adequacy.

Surprise Adequacy (SA), proposed by Kim et al. [4], focuses more on the distribution of neuron activations (see Section II for details of SA). Unlike existing test adequacy criteria, SA can measure the similarity between a *single* new input and the data seen by the model during training. This, in turn, allows the prioritisation of inputs in the likelihood of model misbehaviour. Chen et al. [43] have compared SA metrics to other test adequacy criteria using a range of image classification benchmarks (MNIST [10], CIFAR-10 and CIFAR-100 [11], and ImageNet [44]) and found it to be better than structural coverage metrics such as NC or kMNC. However, they only considered image classifiers. On testing of NLP models, Ribeiro et al. have recently proposed CheckList, a set of NLP specific methodology for test input generation [45]. We note that CheckList is a guideline for manual test data generation, while the approach presentd in this paper is an automated technique of evaluating adequacy of test inputs.

VIII. Conclusion and Future Work

We present a feasibility study for applying Surprise Adequacy (SA) metric to Natural Language Processing (NLP) tasks. Like many other test adequacy metrics, SA has been mostly evaluated using image recognition models. This paper shows that it can be successfully applied to NLP tasks to prioritise inputs that will reveal incorrect behaviour of a model. We also report that the distribution of Activation Traces of some NLP models are multimodal, and investigate whether multimodal variants of SA metrics can improve the accuracy of the analysis. We show that, for NER tagging task, the new metrics can accuracy prioritise tokens that will be mislabelled by the model. However, more complicated tasks such as Question Answering remains as a challenge for SA analysis. Future work includes expanding the domain even further to other NLP tasks, as well as developing improved AT extraction and analysis techniques that can handle complicated NLP tasks more effectively.

Acknowledgement

This work was supported by the Engineering Research Center Program through the National Research Foundation of Korea funded by the Korean Government (MSIT) (NRF-2018R1A5A1059921) and Institute for Information & communications Technology Promotion grant funded by the Korean government (MSIT) (No.1711073912).

REFERENCES

[1] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17. ACM, 2017, pp. 1–18.

[2] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 303–314.

[3] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang, "Deepgauge: Multi-granularity testing criteria for deep learning systems," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ser. ASE 2018. ACM, 2018, pp. 120–131.

[4] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," in *Proceedings of the 41th International Conference on Software Engineering*, ser. ICSE 2019. IEEE Press, 2019, pp. 1039–1049.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[6] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[7] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60 – 88, 2017.

[8] X. Gao, R. K. Saha, M. R. Prasad, and A. Roychoudhury, "Fuzz testing based data augmentation to improve robustness of deep neural networks," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1147–1158.

[9] H. Zhang and W. K. Chan, "Apricot: A weight-adaptation approach to fixing deep learning models," in *IEEE/ACM International Conference on Automated Software Engineering*, 2019, pp. 376–387.

[10] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *Available: http://yann. lecun. com/exdb/mnist*, 2010.

[11] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[12] J. Kim, J. Ju, R. Feldt, and S. Yoo, "Reducing dnn labelling cost using surprise adequacy: An industrial case study for autonomous driving," in *Proceedings of ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE Industry Track)*, ser. ESEC/FSE 2020, 2020, pp. 1466–1476.

[13] S. Kim and S. Yoo, "Evaluating surprise adequacy for question answering," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ser. DeepTest, 2020, pp. 197–202.

[14] P. C. Mahalanobis, "Reprint of: Mahalanobis, p.c. (1936) "on the generalised distance in statistics"," *Sankhya A*, vol. 80, no. 1, pp. 1–7, 2018.

[15] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[17] Y. Zhang, Q. Liu, and L. Song, "Sentence-state lstm for text representation," in *ACL*, 2018.

[18] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3359786

[19] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[20] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley, 1990.

[21] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, Nov. 1987.

[22] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep. TR1648, 2009.

[23] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[24] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1183–1192.

[25] W. H. Beluch, T. Genewein, A. Nrnberger, and J. M. Khler, "The power of ensembles for active learning in image classification," in *Proceedings of the Computer Vision and Pattern Recognition*, June 2018.

[26] A. Kirsch, J. van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019, pp. 7026–7037.

[27] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[28] L. C. Freeman, *Elementary applied statistics: for students in behavioral science*. John Wiley & Sons, 1965.

[29] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep active learning for named entity recognition," in *International Conference on Learning Representations*, 2018.

[30] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150.

[31] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of Empirical Methods in Natural Language Processing*, Oct. 2013, pp. 1631–1642.

[32] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 649–657.

[33] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting lstm networks for semi-supervised text classification via mixed objective function," in *AAAI*, 2019.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[35] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147.

[36] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Jun. 2016, pp. 260–270.

[37] H. Yan, B. Deng, X. Li, and X. Qiu, "TENER: adapting transformer encoder for named entity recognition," *CoRR*, vol. abs/1911.04474, 2019.

[38] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *CoRR*, vol. abs/1606.05250, 2016.

[39] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *CoRR*, vol. abs/1611.01603, 2016.

[40] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 845–855.

[41] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," 2018.

[42] H.-Y. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," *ArXiv*, vol. abs/1711.07341, 2018.

[43] J. Chen, M. Yan, Z. Wang, Y. Kang, and Z. Wu, "Deep neural network test coverage: How far are we?" 2020.

[44] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[45] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 4902–4912. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.442