




## Article

# Intent-Bert and Universal Context Encoders: A Framework for Workload and Sensor Agnostic Human Intention Prediction

Maximillian Panoff , Joshua Acevedo, Honggang Yu, Peter Forcha, Shuo Wang  and Christophe Bobda 

Electrical and Computer Engineering Department, University of Florida, Larsen Hall, 968 Center Drive, Gainesville, FL 32611, USA

\* Correspondence: m.panoff@ufl.edu

**Abstract:** Determining human intention is a challenging task. Many existing techniques seek to address it by combining many forms of data, such as images, point clouds, poses, and others, creating multi-modal models. However, these techniques still often require significant foreknowledge in the form of known potential activities and objects in the environment, as well as specific types of data to collect. To address these limitations, we propose Intent-BERT and Universal Context Encoders, which combine to form workload-agnostic framework that can be used to predict the next activity that a human performs as an Open Vocabulary Problem and the time until that switch, along with the time the current activity ends. Universal Context Encoders utilize the distances between the embeddings of words to extract relationships between Human-Readable English descriptions of both the current task and the origin of various multi-modal inputs to determine how to weigh the values themselves. We examine the effectiveness of this approach by creating a multi-modal model using it and training it on the InHARD dataset. It is able to return a completely accurate description of the next Action performed by a human working alongside a robot in a manufacturing task in ~42% of test cases and has a 95% Top-3 accuracy, all from a single time point, outperforming multi-modal gpt4o by about 50% on a token by token basis.

**Keywords:** HRC; HRI; multi-modal human prediction; open vocabulary; cross-attention

Academic Editor: George F. Fragulis

Received: 25 November 2024

Revised: 18 January 2025

Accepted: 23 January 2025

Published: 2 February 2025

**Citation:** Panoff, M.; Acevedo, J.; Yu, H.; Forcha, P.; Wang, S.; Bobda, C. Intent-Bert and Universal Context Encoders: A Framework for Workload and Sensor Agnostic Human Intention Prediction. *Technologies* **2025**, *13*, 61. <https://doi.org/10.3390/technologies13020061>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As industry, manufacturing, and assisted living increasingly rely on tighter integration of automation, the safety and flexibility of Human-Robot Collaboration (HRC) is increasingly important. This has led to larger demand for mutual cognition in HRC, or proactive robotics as described by Li et al. [1], where robots and humans determine and perform their tasks independently but with foreknowledge of the other's actions. However, robots and humans cannot natively communicate with one another. To address this issue, there have been many proposed solutions to allow humans and robots to better understand the other's current goals and future behaviors, or *intentions*. This communication must happen through the various sensing 'channels' available to humans and robots, as identified by Bonarini, including sight, sound, and touch [2]. Interpreting these signals is still an open task without a universal solution, due to the many diverse tasks and channels that exist

Many of the suggested methods utilize augmented or virtual reality [3,4] to keep the human informed of the robot's planned moments so that humans can 'plan around' the actions of the robot. While such methods do take full advantage of the strengths associated with human workers (their flexibility), as the system does not *proactively* acquire and respond to the human's intentions, but rather only make the robot's available to the

humans, forces the humans to ‘work around’ the robots rather than truly collaborating. Alternatively, there are solutions that enable robots to acquire human intentions, but they often require some form of active input from the human. This can range from gesture control, including simple nods and head movements [5], to complex motions [6], to verbal commands that are parsed and converted to movement constraints [7].

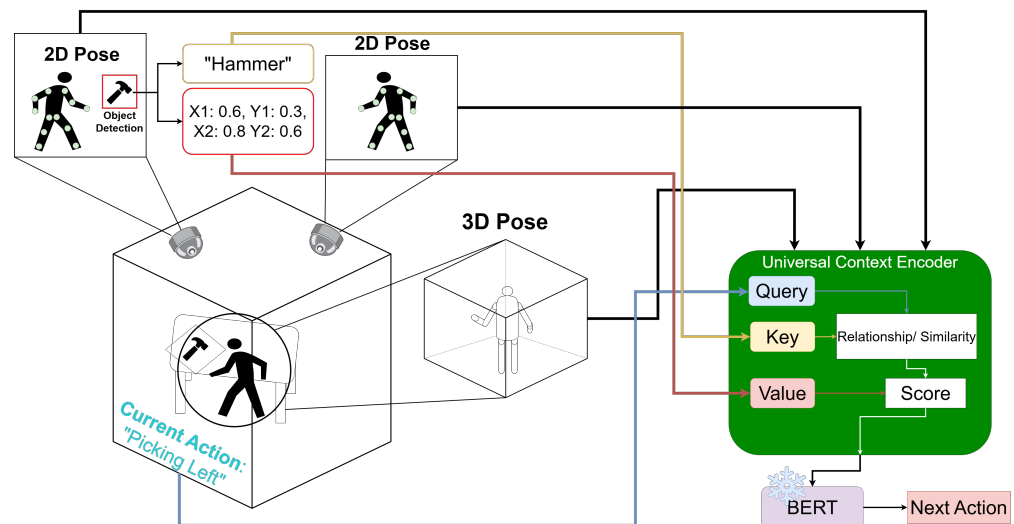
However, these approaches face large hurdles themselves, from requiring computationally expensive digital twins [8], to reliance upon complex sensors such as lidars or depth cameras or even brain-human interfaces [9], which limit their transferability. Additionally, they typically require active human inputs rather than passive contactless sensing. In particular, these active inputs limit the practicality and flexibility of the proposed systems, as human workers would have to alter their workflow to instruct the robot. However, passive acquisition of human intention is a challenging task, with prior methods like Markov chains [10] and LSTM-based models [11] only able to function with complete, *a priori* knowledge of the types of tasks to be completed and adequate training data.

Moreover, as these solutions are workload-specific, they cannot be easily transferred between even highly similar scenarios, such as two manufacturing methods, let alone between completely different ones, such as manufacturing and assisted living. Zheng et al. [12] have attempted to address this by creating a system to encode contextual data to determine the similarity between the current state and nodes in a domain-specific knowledge graph to determine current human actions and identify potential next ones. However, this model still requires premade hand-crafted knowledge graphs and can only be transferred between similar workloads.

We instead propose a framework to recover human intention without the limitations of being tied to specific activity classes, data modality and formats, or *a priori* knowledge graphs by formulating this challenge as an Open Vocabulary problem [13]. This method works through ontology or the relationships between the *concept* of items or objects. Specifically, by embedding objects and poses in a shared latent space originating from a Natural Language Processing (NLP) model, our proposed Universal Context Encoders can better extract the relationships between various observed system attributes and components such as objects, joints, and activities with related or similar names in Human Readable English (HRE). This additional contextual information provides meaningful insight into the next activities humans will take while not being limited to small, hand-crafted workloads. Instead, The proposed Intent-BERT framework extracts important contextual information and uses it to predict the next action to be performed by a human as shown in Figure 1. Specifically, Figure 1 demonstrates how data from multiple modalities and sources are embedded into a single shared latent space through a Universal Context Encoder (UCE), which uses cross attention between the HRE description of the data source (e.g., “hammer”) and the HRE description of the current Task/Action (e.g., “Picking Left”) to embed the raw data (e.g., 2D pose/object detection bounding boxes).

Our contributions can be summarized as:

1. A framework, Intent-BERT, using Universal Context Encoders that supports encoding a diverse range of data modalities and formats into a single latent space through word embedding similarity
2. The use of Intent-BERT to passively recover a description of the next activity a human performs in English from a single time point in a scene
3. Additionally, using Intent-BERT to predict the time until the current activity a human is performing will end and the time until the next activity begins
4. A demonstration and evaluation of this approach on the InHARD Dataset [14].



**Figure 1.** A high level overview of Intent-BERT demonstrating how textual descriptions of data and their sources are used to weight the data through Universal Context Encoders.

The rest of this paper is laid out as follows: In Section 2, we discuss a few related works. Next, in Section 3, we introduce our methodology before presenting our results in Section 4. Finally, we conclude our paper in Section 5.

## 2. Background

The proposed technique builds upon several techniques that combine multiple types of data to better understand the current scenario and, thus, the *context* of the system. Once the context is recovered, it should provide better insights into the intentions of any human operators. By understanding human intentions, the system can support future human actions without the humans directly communicating with the system, which meets the definition of proactive robotics [1].

Additionally, it is important to our discussion to introduce our terminology of *Actions*, *Tasks*, and *Workloads*, which may vary slightly from prior literature, including the InHARD dataset [14]. An Action is a particular movement or set of movements that exist independent of a particular object, i.e., pick up, set down. A Task pairs an Action with an object, i.e., pick up the cup. A Workload is a set of tasks to be completed, i.e., the steps to follow to create a cake. Therefore, an ‘action’ in InHARD is a *Task* in this paper, while a ‘meta-action’ is an *Action*. Several related and highly similar terms are also used throughout this work, including Human-Robot Collaboration, Human-Robot Interaction, and Human-Robot Collaboration, all of which are defined in Table 1.

**Table 1.** A brief comparison between the areas relevant to this work. \* The main focus of this work.

Area	Description
Human-Robot Collaboration	Humans and Robots working jointly (but potentially spatially or temporally separated) towards the same goals [1]
Human Robot Interaction	Humans and Robots directly interacting to exchange information or complete a task
Human-Robot Communication	Humans and Robots exchanging information
Human Intention Prediction *	Predicting the next long term action to be done by a human.

### 2.1. Human Robot Collaboration

As mentioned in Li et al. [1], the ability of humans and robots to work together is essential to further progress in automation, as well as existing methods. This includes traditional areas of automation such as assembly and heavy industry, but also rehabilitative care [15], and many other areas besides. However, humans and robots are able to collaborate to a limited degree without any communication, with existing assembly line work being the chief example of such. However, this lack of communication leads to potential safety issues and efficiency losses due to poor or unstable controllers [15].

### 2.2. Human-Robot Communication

There are a multitude of proposed methodologies that allow humans and robots to communicate without contact, which meet the criteria of symbiotic robotics from Li et al. [1]. The medium of communication can take many forms in these approaches, but most current approaches focus on contactless readings, which do not require humans to touch any sensors. Vision techniques in this space have been highly successful [16], especially when expanded to include, gaze [5], gesture [6,17], and pose detection [11]. We outline a few recent examples and their intuitive reasoning below. An early example of this is given by Heinzmann et al. [5], where intersections between a human's eye line and objects on a table are detected. The human then can provide specific commands by moving their head in different ways, such as nodding or shaking their head. Gesture Control is another common approach explored by Strazdas et al. [6]. In particular, they propose a method that allows humans to 'mime' the action they wish the robot to perform to intuitively communicate their intentions [6].

### 2.3. Multi-Modal Techniques in Robotics

Many of these human-robot communication forms combine the results of individual communication methods into 'multi-modal' techniques. Wang et al. provides a fairly complete example of this, which combines gesture, haptics, voice, and brainwave processing [9], where many techniques focus only on two or three methods [18–20]. The individual results of the communication and perceptual methods are combined following statistical, rules-based, or deep learning fusion techniques, resulting in a more complete estimation of the context within a scene.

Multi-modal techniques are also common for tasks outside HRC, both within the robotic domain [21] and in terms of general computer vision [22]. These multi-modal techniques can distill to obtain the relevant information or 'knowledge' contained in each input type and convert them into a shared domain. This is often done to process commands given through language using Natural Language Processing (NLP) to create outputs in other domains, such as vision. Latte [7] demonstrates how verbally expressed human intentions can be translated into robot velocities.

### 2.4. Human Intention Prediction

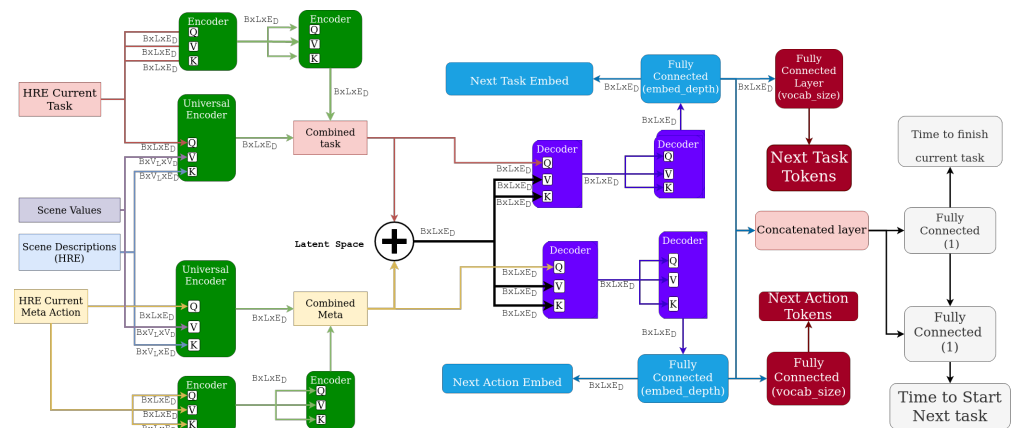
Many works prior to the recent focus on context-awareness focused on predicting human motion from past behavior. This took many forms, including Kalman filters [23] or other pure spatio-temporal analysis [24]. Alternatively, Ragaglia et al. predict all spaces humans may physically be able to occupy within a certain time range under some velocity and limb orientation constraints [25]. However, these have the obvious weakness that in many Workloads, future Tasks may involve different parts and tools than prior ones, which limits the adaptability of these techniques.

Thus, many solutions begin attempting to *understanding* human intentions to predict motion. To solve this, Liu et al. use hidden Markov chains to predict future contexts from

previous and current ones for particular manufacturing Workloads [10]. This approach relies upon probabilities of the last few actions to predict future ones, which works well for repetitive or highly deterministic tasks but less well for complex graph-like Workloads. To address this, Zhang et al. proposed a method to encode context vectors into a latent space and match a vector describing the current context against a pre-made domain-specific graph [26]. This allows for greater flexibility than hidden Markov chains, including support for transferring solutions across similar but different Workloads. Li et al. similarly create a method that constructs a graph representing spatial relationships within a scene [27].

### 3. Methodology

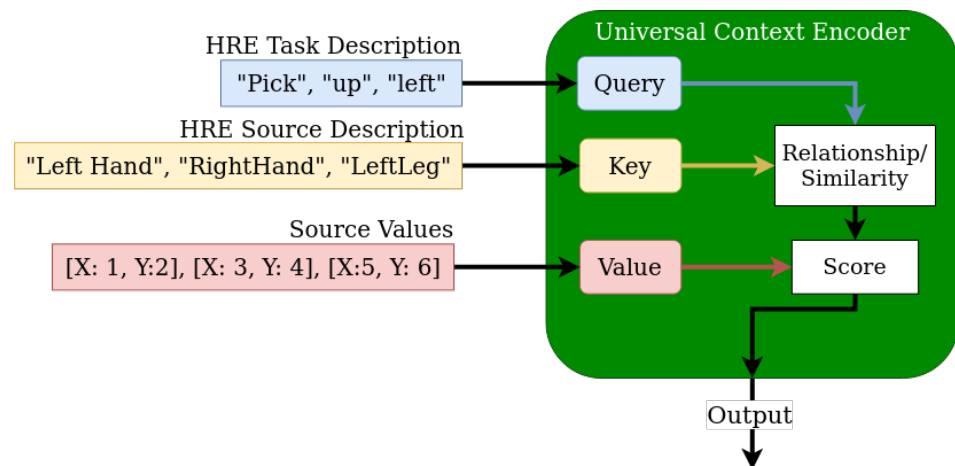
One of the main shortcomings of current Human-Intention predictions is their reliance upon hand-crafted rules, specific data types, or Workload-specific graphs. To address this, we propose a method that formulates the prediction of the next action to be performed as an Open Vocabulary problem [13] which can be seen in Figure 2. Specifically, by labeling the sources, modalities, and channels [2] of all observed system attributes in Human Readable English (HRE), Natural Language Processing (NLP) can be used to weight and merge all such data into a single latent space describing the context of the system, with the weights determined by cross attention between the data source or modality to the work being done. This latent space is then mined to predict future contexts, without any domain-specific foreknowledge, increasing its transferability.



**Figure 2.** A complete overview of the proposed architecture. The Image and Joint Encoders are introduced in Section 3.2, while ‘Encoders’ are transformer encoders from [28]. B is batch size, L is sequence length, and  $E_D$  is Embed dimension.

#### 3.1. Intuitive Rationale

Multi-modal techniques have strongly benefited from including transformers and attention, which more easily allows diverse data types to be embedded into a shared latent space. We use BERT [29] from Keras NLP [30] to tokenize and embed Human-Readable English (HRE) Action and Task descriptions along with HRE joint and object (i.e., *source* or *origin*) names detected within the scene at any given time. These are used as Queries and Keys for a cross-attention network, where the Values can be any arbitrary data source. Figure 3 provides an example of 2D joint positions being embedded through a Universal Context Encoder (UCE). The UCE uses cross attention between the current Task or Action and the HRE description of the data to weight the relevancy and importance of the data when embedding. As a result a UCE is capable of embedding data from a diverse range of modalities into a single shared latent space by exploiting the relationships between descriptions of each data source and the current task.



**Figure 3.** The Proposed Universal Context Encoder. It uses similarity in token embeddings to extract relationships between source values and the current task.

This is due to the meaning of words captured by the NLP embedding (done by BERT in our case), placing more similar words and phrases more closely to each other in the embedded space than unrelated words. This proximity in the embedded space allows the UCE to exploit the relationships between these HRE descriptions to weight the importance of the Values (i.e., raw data) to each dimension in the latent space without placing any requirements on the types, origins, or formats of that incoming raw data. In short, this enables a UCE to flexibly weigh the importance of collected data through its relationship to the current task. e.g., ‘LeftHand’ is more correlated to ‘Pick up Left’ than ‘RightLeg’ is, and if ‘RightHand’ has higher velocity during this time, that its more likely that the next Action is ‘Put Down Right’. This also works for bounding boxes or segmentation results as opposed to velocity measurements, where the HRE class name of the object is provided as the key or any other types of data that have HRE descriptions.

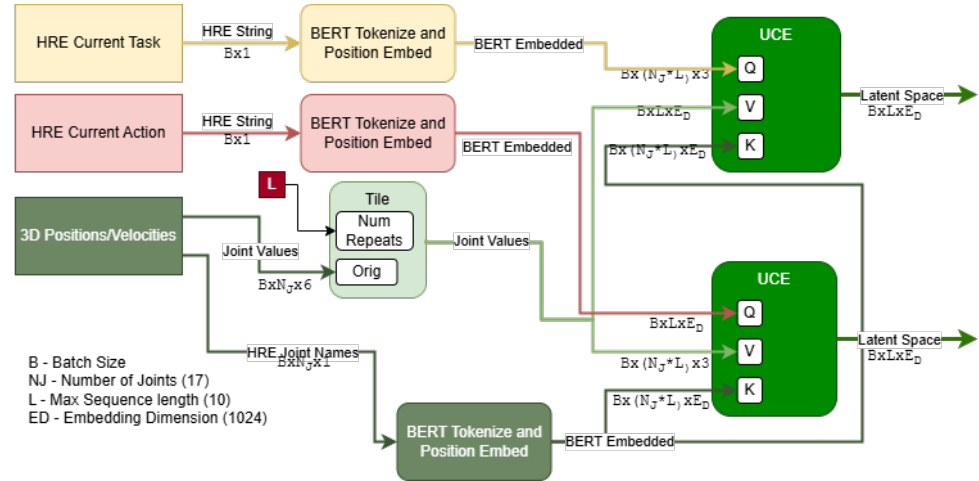
### 3.2. Proposed Framework

To accomplish this, we propose a framework that natively supports multi-modal analysis, combining inputs from various contextual sources. As shown in Figure 2 for the InHARD dataset, we prepare an example architecture using this framework. Firstly, the textual inputs (i.e., the HRE Actions/Meta-Actions and Tasks from Intent-BERT) are tokenized and embedded by BERT. The Task and Action tokens directly go through two layers of self attention. They are also passed to the model’s UCE as Queries (e.g., “Picking Left”, “Consult Sheets”), while the various modalities of other data such as 3D poses have any HRE descriptions (e.g., “Right Elbow”, “Left Hip”) extracted and used as Keys for their corresponding data, as shown in Figure 4. This allows the network to learn relationships between the origin of each data point and the Task and Action being performed. The Values passed to the Encoders are the data values themselves and are thus modified by the relationships extracted between the datum’s origin and the current Task/ Action. This same process is repeated for 2D poses, with the pose values and HRE descriptions extracted from the recorded video using Yolo v8 [31] before being passed into the same UCE instance as the 3D data as shown in Figure 5 Finally, the outputs from the UCEs are summed and normalized to create a single latent space capturing the entirety of the scene.

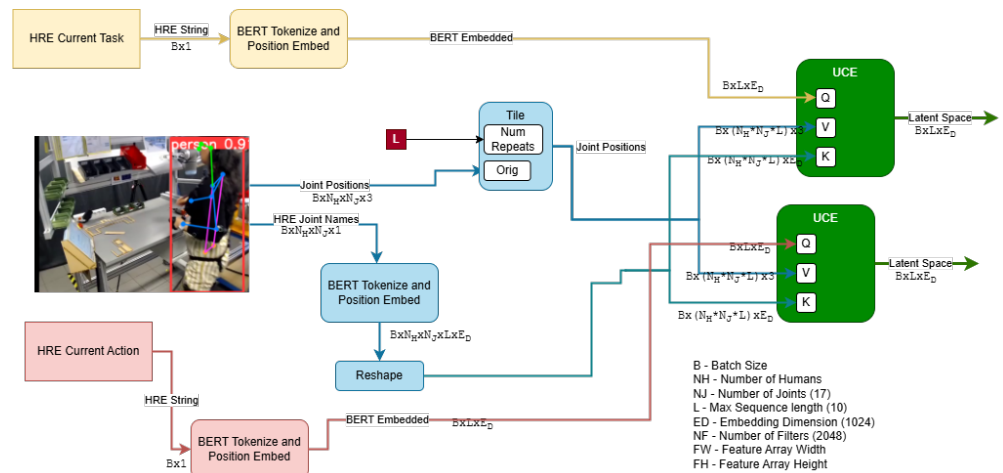
However, Universal Context Encoders only support human-describable content. Extracted image features, which do not have HRE data, are handled as follows: a  $1 \times 1$  convolutional layer is used to represent the same depth as the BERT embedding. Positional Embedding, similar to that used in Visual Transformers [32] is then used to encode embedded features with additional spatial meaning (to help identify the location of tools and



equipment). This is then squeezed into three dimensions, preserving the batch size and embedding depth before being passed to an Encoder as both the Value and Key. Additionally, the HRE embeddings of the 2D Joints are concatenated with the joint data itself before going through a fully connected layer to return to the original embedding depth. This is done to add spatial information to the Keys to better support the meaning of joints located at different places in the image.



**Figure 4.** Structure of the proposed Universal Context Encoder being used to encode 3D joint data. Note that the Query (Task/Action) and Key (Tokenized Joint Name) vectors share the same last dimension as they were both created by BERT, and cross attention between the two is used to weigh the raw 3D position and velocities.



**Figure 5.** Structure of the proposed Universal Context Encoder being used to encode 2D joint data. Note that 2D pose information is extracted via a separate network, with the HRE joint names tokenized into the space used by BERT. These are then used as Keys to the raw 2D data of each joint's pose (which in turn act as Values). Cross attention occurs between the Keys and the Action/Task description.

This combined latent space is then decoded by two Transformer Decoders in the style of Vaswani et al. [28], one to recover the next Task and one to recover the next Action. These Decoders are completely self-attended, using the combined space for both the encoder and decoder sequences. Moreover, the raw embedded spaces are also returned to calculate losses using the Embed Loss as described in Section 3.3 and Equation (3). Finally, the combined latent space is flattened and passed through a fully connected layer with ReLU activation to predict the time until the current activity stops. The result of this is concatenated with the flattened space and passed through a second Fully Connected layer, also with ReLU activation, to predict the time until the next activity starts.

### 3.3. Losses

To assist our model in recovering the Action and Task descriptions, the model outputs both the recovered HRE descriptions, as well as the raw embeddings. The raw embeddings are used to provide additional feedback to the model to better place predictions in the latent space, in addition to more standard objective functions. In this section, we will introduce all the losses used for each output and the rationale behind the selection of each.

$$\text{ClippedPredictor} = \text{clip}(\hat{y}, \text{Pred}_{\min}, \text{Pred}_{\max}) \quad (1)$$

We begin with the Action/Task outputs. The HRE Token outputs are trained using Categorical Cross-Entropy [33], jointly optimized along with our novel Embed Loss from Equation (3) calculated from the raw embedded output. Embed Loss seeks to align the output embeddings directly with embeddings of the target HRE strings as processed by BERT to maintain the coherency of the embedded process. Embed Loss merges our novel LatentGapLoss (LGL, Equation (2)), a metric based on Mean Average Percentage Error (MAPE) [34], with standard Cosine Similarity [35]. For these,  $y$  denotes ground truth values and  $\hat{y}$  denotes predictions, and  $|x|$  is the absolute value of  $x$ .  $\text{pred}_{\min}$  and  $\text{Pred}_{\max}$  are user-defined limits for the clipping operation.

$$\text{LGL}(y, \hat{y}) = \frac{|y - \hat{y}|(|y_{\text{true}}| + |\text{ClippedPredictor}| + 2\epsilon)}{|y| * |\text{ClippedPredictor}| + \epsilon} \quad (2)$$

MAPE is known to have several limitations as an objective metric, but unlike logarithmic-based losses, it supports negative values smaller than  $-1$ , which often occur in embeddings. MAPE is used over Mean Square Error (MSE) [36] as the values of each axis in the embedded space are not normalized, which means that a small error on one axis can carry far more impact than a larger error along another. Thus we optimize for the relative position along each axis with MAPE, rather than the absolute position with MSE. LGL differs from standard MAPE by including a denominator factor based on  $\hat{y}$ , the predictions, to penalize all predictions smaller than a user-defined  $\epsilon$ , to mitigate the ‘all 0’ predictions that can occur with MAPE. This prediction factor is clipped (Equation (1)) to prevent rewarding meaningless large predictions. Cosine Similarity is added into the Embed Loss function to optimize for the absolute distances away from the origin of the latent space [37], to better align the absolute positioning of embedded predictions. This Cosine Similarity (CS) is multiplied by  $-1$  to convert it to be a loss (smaller score better), then summed with a 1 to ensure the output is in the range of  $[0, 2]$  rather than  $[-1, 1]$ . Thus between LGL and CS, both the relative and absolute positions of the predictions in the embedded space are optimized.

$$\text{EmbedLoss}(y, \hat{y}) = \text{LGL}(y, \hat{y}) \times [(-\text{CS}(y, \hat{y}) + 1) + 1] \quad (3)$$

Comparatively, the error for the time outputs are comparatively simple. Mean Square Logarithmic Error is used, as the time values predicted cannot be under 0 and the relative accuracy is more important than the absolute. In other words, we want to ensure smaller error as the true time to finish the current task or start the next approaches and accept larger errors as the true times are more distant.

## 4. Results

We validate our approach by training the proposed architecture on the InHARD dataset [14]. It is important to note that our intention with this case study is to demonstrate how Universal Context Encoders can be used to combine various multi-modal data in an intuitive, easily transferable fashion, rather than evaluate the specific Universal Context



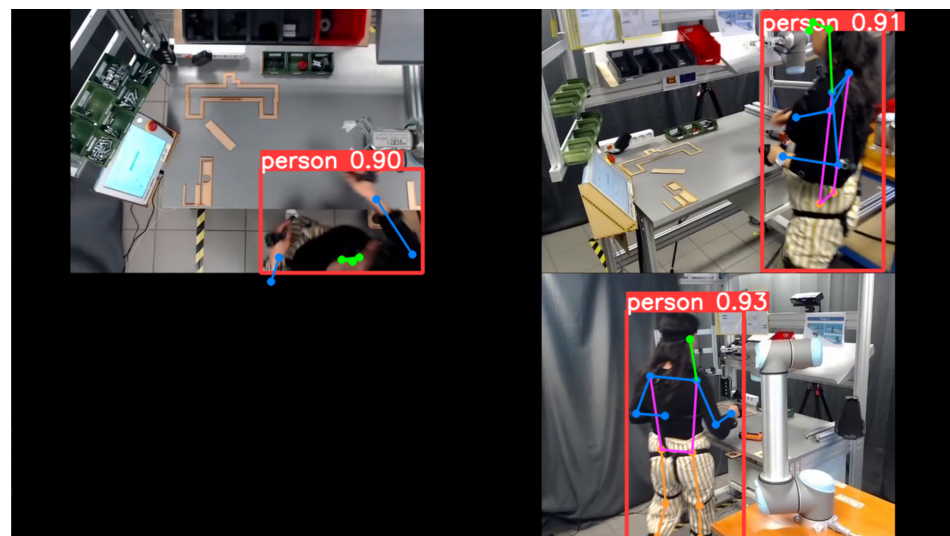
Encoders and data modalities used. InHARD includes multiple cameras placed around a workstation while human workers collaborate with a robotic arm to complete assembly tasks. Each human is wearing 3D location sensors across their body, which recover the position and orientation of their joints. The current Action ('meta-action' in the dataset) and Task ('action') of the worker at each time point are recorded. InHARD includes 14 Actions (e.g., 'Take component', 'Picking left') and 75 Tasks (e.g., 'Place LARD on Profile P360-1', 'Put down Cover CAPO').

#### 4.1. Experimental Setup

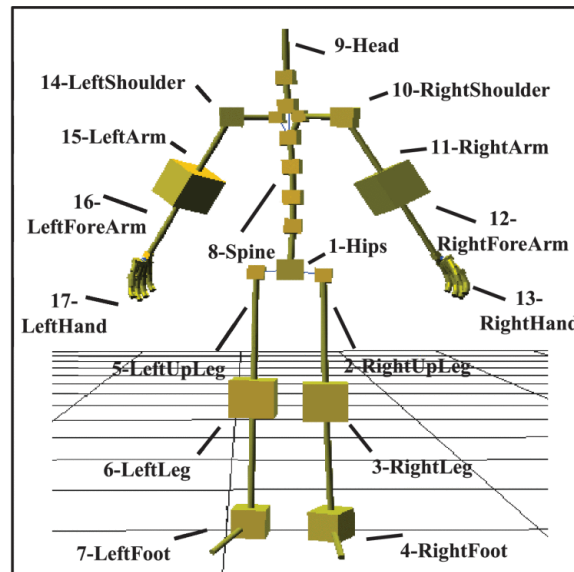
All training was completed on a computer with 32 GB of DDR4 RAM, an 11700k CPU at 3.60 GHz, and an Nvidia 2080 Ti, running Tensorflow 2.13 and Pytorch 2.0.1.

##### 4.1.1. Data Cleaning and Preparation

2D normalized human poses were gathered using Yolov8 [31] with a required human confidence level of 0.7 from the videos captured of human operators within the InHARD dataset [14] as seen in Figure 6. As the 3D human poses were sampled at a rate exceeding the frame rate within InHARD, the mean of all samples for each joint was computed during each frame and used as a single value. These were then scaled by the position of the head, to track the relative pose of the human. The 3D pose velocities were calculated by computing the difference from the prior pose, with an initial velocity value of 0. Keras\_NLP's [30] *bert\_large\_en\_uncased* BERT Backbone is used to process and embed the Human-Readable English (HRE) descriptions of each task, with examples of these descriptions for the 3D joints in Figure 7.



**Figure 6.** An example of the 2D poses being extracted from the InHard Dataset recordings. Each dot represents a joint labeled by Yolov8 [31] with the joint's HRE description used as the Key passed into the UCE and the joint's x and y coordinates within the (quarter) frame used as the corresponding Value.



**Figure 7.** Example of each 3D joint and its corresponding HRE description. Originally Figure 5 from [14]. Obtained under License number 5957321463576 from IEEE, Copyright 2020 Mejdi Dallel.

#### 4.1.2. General Hyper-Parameters

We use the suggested test-train split from InHARD, and created the model reserving the first three sessions as validation while exploring architectures and hyperparameters. The results in this paper are from a model trained on the entirety of the training set and tested on the test set using the found hyperparameters. Namely, Adam [38] was used as an optimizer with a clip norm of 3, a learning rate of  $1 \times 10^{-4}$ , and an epsilon of 0.01, with a batch size of 1. All operations were performed with 32 bit floating point precision.

#### 4.1.3. UCE/Intent-BERT Hyper-Parameters

ResNet 50 trained on ImageNet is used to perform feature extraction, taking an input of  $(224 \times 224)$  across an image of  $(720 \times 1280)$  pixels.  $E_D$  is set to 1024,  $Pred_{min}$  and  $Pred_{max}$  are set to  $-5$  and  $5$  respectively, and  $\epsilon$  is set to 0.001. Additionally, we note that the proposed Intent-BERT architecture supports using multiple time points when creating the latent space by summing the inputs from each, but for this evaluation, we predict from only one time point.

#### 4.2. Next Task/Action Prediction

There are two outputs for each Task and Action, the embedded space and the reconstructed tokens, which can be thought of as HRE words in this context. It is important to note that the output tokens in Intent-BERT support the entire BERT uncased vocabulary of 30,552 classes (i.e., many possible tokens ( $\sim 30,000$ ) exist that are not part of the InHARD dataset, but may exist in other workloads). Intent-BERT outputs a series of these tokens, not specific activity classes, unlike all prior work we are aware of. This allows for easy modification and potential truly workload-agnostic models to be formed. We evaluate the success of our trained model on these through several metrics: Perplexity [39], Per-Token Accuracy (PTA), Top-3 Per-Token Accuracy, and Phrase Accuracy, which are compared in Figure 8. Cross-Entropy (CE) is calculated on a per-token basis across each sequence. Perplexity is a standard metric in NLP [39], which is simply the exponent of the CE. Perplexity represents the model's confusion between its choices but excludes all masked (empty) tokens. PTA is the overall accuracy across all non-mask, non-reserved (e.g., beginning of sequence, end of sequence) tokens. Top 3 PTA is the percentage of time that the correct token is in the top 3 guesses of the model at that position (again excluding non-mask, non-reserved). Finally,

Phrase Accuracy is the model’s accuracy in reconstructing the entire HRE sentence of the next Task/Action. The results of each for both Task and Action prediction can be found in Table 2. Although Task prediction only has 14% Phrase Accuracy, it successfully predicts the correct token for each position in the sequence within the top 3 candidates 80% of the time. Meanwhile, Action recovery is completely correct in 41% of test instances, and has a Top-3 PTA of 95%.

	Token 1	Token 2	Token 3	Token 4	Token 5
Ground Truth	Start	51	50	End	Mask
Prediction	Start	51	End	Mask	Mask
Match	✓	✓	✗	✗	✓
Included in Per-Token	✗	✓	✓	✗	✗

**Figure 8.** Comparison of PTA, Token Predictions, and Phrase Accuracy. Each token must match and be included in the per-token metric to count for it, and Phrase Accuracy requires all tokens that are included in the per-token metric to match.

**Table 2.** Token metrics for the proposed architecture, evaluated on the InHARD test set. CE denotes Cross-Entropy and PTA Per-Token Accuracy.

Activity	Perplexity	PTA	Top-3 PTA	Phrase Accuracy
Task	3.3594	60.08%	80.14%	14.63%
Action	3.3593	60.08%	95.78%	41.69%

We also evaluate the similarity of the predictions’ embedded space to the true phrase’s embedded space in Table 3. We use several metrics here, including LGL (Equation (2)), Mean Square Error (MSE), Cosine Similarity (CS) and Mean Average Percentage Error (MAPE). Cosine Similarity is a measure between  $-1$  to  $1$ , with higher being better, while all other lower values for the other metrics indicate superior performance. In particular, the MAPE is still fairly high, which indicates that there is room for improvement in aligning the predicted embedded spaces with the true ones.

**Table 3.** Embedded space metrics, evaluated on the InHARD test set. MSE denotes Mean Square Error, CS cosine similarity, and LGL Latent Gap Loss.

Activity	LGL	MSE	CS	MAPE
Task	6.924	0.6490	0.4377	927.71%
Action	4.867	0.2990	0.4728	598.1%

#### 4.3. Activity Timing Prediction

Finally, we examine the results of our model in predicting the timing of each activity change in Table 4. Intent-BERT is able to recover the time to activity switch within about 2 s of the switch, and the time that the human operator stops their current task to about 1 s. It is important to note that there are non-negligible gaps between activities in the InHARD dataset, which is why a single prediction for the switch is not used. e.g., The user may stop their activity in 3 s but wait 7 s before the next starts.

**Table 4.** Errors in predicting the time the current activity finishes and the next one begins.

Prediction Type	MSLE	MAE	MAPE
Time to finish task	0.2765	1.263s	432.6%
Time to next task	0.4094	2.186s	702.3%

#### 4.4. Timing Evaluation

On the stated test machine (without any LLM tokenization/embedding or cleaning or preprocessing) Intent-BERT takes around 260 ms to predict the next action. Yolo8 preprocessing and cleaning adds around 22 ms for a total of 282 ms without tokenization or embedding. The current implementation of this tokenization/embedding in this work is highly inefficient due to it being performed well prior to inference/training as a preprocessing step, but NVIDIA has released a BERT model through TensorRT capable of 76 ms inference on CPU only [40]. Together, this comes to 358 ms for a single prediction on a capable but older system without any quantization. With just over 40M parameters in the entire Intent-BERT Model and 2M parameters in the UCE, this solution may be feasible for pseudo-real-time (~1 s) inference on higher-end edge devices.

#### 4.5. Comparison of UCEs Against GPTs

The performance of human intention prediction via UCE vs standard state-of-the-art LLMs such as GPT 4o-mini (version gpt-4o-mini-2024-07-18) was evaluated to further highlight the strengths of UCE based embedding. Similarly to Section 4.2, the ‘Test’ split of the In-Hard dataset was used for this evaluation. GPT 4o was accessed through the OpenAI API version using an Assistant created with a temperature of 0.4 and the following instructions Quote 4.5:

The user will give you data about a person’s body parts and positioning, this data is given from a 3d perspective where you receive each body parts position in a x, y, z, format and their rotations in an x, y, z format. You also get each body parts position in a 2d perspective from multiple camera angles in an x,y format and you get a confidence level. I then need you to tell the user what person observed is doing and predict what they are about to do. You are to match each action to a specific meta-action and action found in the file you were given. Take your time, there is absolutely no rush I want the answer to be correct not quick.

PTA and EmbedLoss metrics were used to compute the difference in accuracy between GPT and UCE based methods at predicting the next Action and Task a human will perform, with the results in Table 5. PTA represents the word for word match between the evaluated model and ground truth, and EmbedLoss represents the difference in meaning between the two predictions. *Intent-BERT* and UCEs outperformed the evaluated *GPT* model at every metric. While *GPT* had around 11–12% accuracy in the PTA category, *Intent-BERT* had an accuracy of 60%. *GPT* had a greater difference of around 4–5 tokens for both Task and Action EmbedLoss analysis when compared to *Intent-BERT*.

**Table 5.** EmbedLoss (lower is better), PTA (higher is better), PSA (Phrase Accuracy) of Intent-BERT and GPT 4o-mini on the test set of In-HARD. The best performance in each category is bolded.

Model	Task EmbedLoss	Task PTA	Task PA	Action EmbedLoss	Action PTA	Action PA
GPT	7.56	12.1%	0.19 %	7.73	10.9%	0.19%
Intent-BERT	<b>3.47</b>	<b>60.1%</b>	<b>14.63%</b>	<b>2.78</b>	<b>60.1%</b>	<b>41.69%</b>

## 5. Conclusions

In conclusion, we present Intent-BERT and Universal Context Encoders as frameworks for recovering human intentions from passive contextual data. This framework uses BERT's embedding of words to determine the relationships between Human Readable English descriptions of the current action and the sources of data through our proposed Universal Context Encoders. We use this framework to create an architecture and train it on the In-HARD dataset, where it is able to correctly describe the next Action performed by a human ~42% of the time. As this architecture predicts HRE descriptions of activities, our next step will be to train it on datasets from across multiple workloads and domains, in order to create a single flexible model to predict human intention in a variety of circumstances. In particular, diverse and complex scenarios focusing on medical treatment or rehabilitation would be an excellent test for UCEs/Intent-BERT. To further evaluate how UCEs adapt to other modalities, additional types of data can be used in addition to or instead of the pose data collected by In-Hard, such as modalities such as heart rate or other biometric information for rehabilitative/medical contexts. This evaluation should be performed both with and without additional retraining to further evaluate the generalization capabilities of UCEs. Tests against Assembly 101 [41] and EPIC-Kitchens [42] datasets may also prove insightful for generalization across tasks, though not across modalities.

**Author Contributions:** Conceptualization, M.P. and H.Y.; methodology, M.P. and P.F.; software, M.P.; validation, M.P. and J.A.; formal analysis, M.P.; investigation, M.P. and J.A.; resources, S.W. and C.B.; data curation, M.P. and J.A.; writing—original draft preparation, M.P. and J.A.; writing—review and editing, M.P. and C.B.; visualization, M.P. and J.A.; supervision, C.B.; project administration, M.P.; funding acquisition, S.W. and C.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This Study was funded in part by the National Science Foundation Grant numbers 2007210 and 2106610 and the OpenAI Researcher Access Program grant number 0000003029.

**Data Availability Statement:** The data and code used in this study are available at <https://zenodo.org/records/4003541> and <https://github.com/smartsystemslab-uf/Intent-BERT.git>, accessed on 17 January 2025.

**Conflicts of Interest:** The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. Maximillian Panoff started working at JLG Industries after the completion of work on this study.

## References

1. Li, S.; Wang, R.; Zheng, P.; Wang, L. Towards Proactive Human-Robot Collaboration: A Foreseeable Cognitive Manufacturing Paradigm. *J. Manuf. Syst.* **2021**, *60*, 547–552. [CrossRef]
2. Bonarini, A. Communication in human-robot interaction. *Curr. Robot. Rep.* **2020**, *1*, 279–285. [CrossRef] [PubMed]
3. Matsas, E.; Vosniakos, G.C.; Batras, D. Prototyping proactive and adaptive techniques for human-robot collaboration in manufacturing using virtual reality. *Robot. Comput.-Integr. Manuf.* **2018**, *50*, 168–180. [CrossRef]
4. Zhou, Z.; Li, R.; Xu, W.; Yao, B.; Ji, Z. Context-aware assistance guidance via augmented reality for industrial human-robot collaboration. In Proceedings of the 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 16–19 December 2022; pp. 1516–1521. [CrossRef]
5. Heinzmann, J.; Zelinsky, A. Quantitative Safety Guarantees for Physical Human-Robot Interaction. *Int. J. Robot. Res.* **2003**, *22*, 479–504. [CrossRef]
6. Strazdas, D.; Hintz, J.; Felßberg, A.M.; Al-Hamadi, A. Robots and Wizards: An Investigation Into Natural Human–Robot Interaction. *IEEE Access* **2020**, *8*, 207635–207642. [CrossRef]
7. Buckner, A.; Figueredo, L.; Haddadin, S.; Kapoor, A.; Ma, S.; Vemprala, S.; Bonatti, R. LATTE: LAnguage Trajectory TransformEr. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 7287–7294. [CrossRef]



8. Choi, S.H.; Park, K.B.; Roh, D.H.; Lee, J.Y.; Mohammed, M.; Ghasemi, Y.; Jeong, H. An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102258. [\[CrossRef\]](#)
9. Wang, L.; Gao, R.; Váncza, J.; Krüger, J.; Wang, X.; Makris, S.; Chrysosouris, G. Symbiotic human-robot collaborative assembly. *CIRP Ann.* **2019**, *68*, 701–726. [\[CrossRef\]](#)
10. Liu, H.; Wang, L. Human motion prediction for human-robot collaboration. *J. Manuf. Syst.* **2017**, *44*, 287–294. [\[CrossRef\]](#)
11. Orsag, L.; Stipancic, T.; Koren, L. Towards a Safe Human–Robot Collaboration Using Information on Human Worker Activity. *Sensors* **2023**, *23*, 1283. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Zheng, P.; Li, S.; Xia, L.; Wang, L.; Nassehi, A. A visual reasoning-based approach for mutual-cognitive human-robot collaboration. *CIRP Ann.* **2022**, *71*, 377–380. [\[CrossRef\]](#)
13. Wu, J.; Li, X.; Xu, S.; Yuan, H.; Ding, H.; Yang, Y.; Li, X.; Zhang, J.; Tong, Y.; Jiang, X.; et al. Towards open vocabulary learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5092–5113. [\[CrossRef\]](#)
14. Dallel, M.; Havard, V.; Baudry, D.; Savatier, X. InHARD—Industrial Human Action Recognition Dataset in the Context of Industrial Collaborative Robotics. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy, 7–9 September 2020; pp. 1–6.
15. Sharkawy, A.N.; Koustoumpardis, P.N. Human–robot interaction: A review and analysis on variable admittance control, safety, and perspectives. *Machines* **2022**, *10*, 591. [\[CrossRef\]](#)
16. Wang, P.; Liu, H.; Wang, L.; Gao, R.X. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. *CIRP Ann.* **2018**, *67*, 17–20. [\[CrossRef\]](#)
17. Mendes, N.; Safeea, M.; Neto, P. Flexible programming and orchestration of collaborative robotic manufacturing systems. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 913–918. [\[CrossRef\]](#)
18. Park, K.B.; Choi, S.H.; Lee, J.Y.; Ghasemi, Y.; Mohammed, M.; Jeong, H. Hands-Free Human–Robot Interaction Using Multimodal Gestures and Deep Learning in Wearable Mixed Reality. *IEEE Access* **2021**, *9*, 55448–55464. [\[CrossRef\]](#)
19. Liu, H.; Fang, T.; Zhou, T.; Wang, L. Towards Robust Human-Robot Collaborative Manufacturing: Multimodal Fusion. *IEEE Access* **2018**, *6*, 74762–74771. [\[CrossRef\]](#)
20. Papanastasiou, S.; Kousi, N.; Karagiannis, P.; Gkournelos, C.; Papavasileiou, A.; Dimoulas, K.; Baris, K.; Koukas, S.; Michalos, G.; Makris, S. Towards seamless human robot collaboration: Integrating multimodal interaction. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 1–17. [\[CrossRef\]](#)
21. Prakash, A.; Chitta, K.; Geiger, A. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7077–7087.
22. Huang, Z.; Zeng, Z.; Liu, B.; Fu, D.; Fu, J. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv* **2020**, arXiv:2004.00849.
23. Zhang, P.; Jin, P.; Du, G.; Liu, X. Ensuring safety in human-robot coexisting environment based on two-level protection. *Ind. Robot* **2016**, *43*, 264–273. [\[CrossRef\]](#)
24. Unhelkar, V.V.; Lasota, P.A.; Tyroller, Q.; Buhai, R.D.; Marceau, L.; Deml, B.; Shah, J.A. Human-Aware Robotic Assistant for Collaborative Assembly: Integrating Human Motion Prediction With Planning in Time. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2394–2401. [\[CrossRef\]](#)
25. Ragaglia, M.; Zanchettin, A.M.; Rocco, P. Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. *Mechatronics* **2018**, *55*, 267–281. [\[CrossRef\]](#)
26. Zhang, Y.; Ding, K.; Hui, J.; Lv, J.; Zhou, X.; Zheng, P. Human-object integrated assembly intention recognition for context-aware human-robot collaborative assembly. *Adv. Eng. Inform.* **2022**, *54*, 101792. [\[CrossRef\]](#)
27. Li, S.; Zheng, P.; Wang, Z.; Fan, J.; Wang, L. Dynamic Scene Graph for Mutual-Cognition Generation in Proactive Human-Robot Collaboration. *Procedia Cirp* **2022**, *107*, 943–948. [\[CrossRef\]](#)
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
29. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
30. Watson, M.; Chollet, F.; Sreepathihalli, D.; Saadat, S.; Sampath, R.; Rasskin, G.; Zhu, S.; Singh, V.; Wood, L.; Tan, Z.; et al. KerasNLP. 2022. Available online: <https://github.com/keras-team/keras-nlp> (accessed on 23 October 2024).
31. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. Version 8.0.178. 2023. Available online: [https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE\\_Reference\\_Guide.pdf](https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf) (accessed on 17 January 2025).
32. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.



33. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. [[CrossRef](#)]
34. McKenzie, J. Mean absolute percentage error and bias in economic forecasting. *Econ. Lett.* **2011**, *113*, 259–262. [[CrossRef](#)]
35. Rahutomo, F.; Kitasuka, T.; Aritsugi, M. Semantic cosine similarity. In Proceedings of the 7th International Student Conference on Advanced Science and Technology ICAST. University of Seoul South Korea, Seoul, Republic of Korea, 29–30 October 2012; Volume 4, p. 1.
36. Tsokos, C.P.; Welch, R. Bayes discrimination with mean square error loss. *Pattern Recognit.* **1978**, *10*, 113–123. [[CrossRef](#)]
37. Li, B.; Han, L. Distance weighted cosine similarity measure for text classification. In Proceedings of the Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, 20–23 October 2013; Proceedings 14; Springer: Berlin/Heidelberg, Germany, 2013; pp. 611–618.
38. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–2.
39. Chen, S.F.; Beeferman, D.; Rosenfeld, R. *Evaluation Metrics for Language Models*; Carnegie Mellon University: Pittsburgh, PA, USA, 1998. [[CrossRef](#)]
40. NVIDIA. TensorRT—BERT. 2019. Available online: <https://developer.nvidia.com/blog/real-time-nlp-with-bert-using-tensorrt-updated/> (accessed on 5 January 2020).
41. Sener, F.; Chatterjee, D.; Shelepov, D.; He, K.; Singhanian, D.; Wang, R.; Yao, A. Assembly101: A Large-Scale Multi-View Video Dataset for Understanding Procedural Activities. In Proceedings of the the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 21096–21106.
42. Damen, D.; Doughty, H.; Farinella, G.M.; Furnari, A.; Ma, J.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *Int. J. Comput. Vis.* **2021**, *130*, 33–35. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Reproduced with permission of copyright owner. Further reproduction  
prohibited without permission.