Deriving Entity-Specific Embeddings From Multi-Entity Sequences

Connor Heaton[1], **Prasenjit Mitra**[1,2]
[1]College of Information Sciences and Technology, The Pennsylvania State University, USA
[2]L3S Research Center, Leibniz University Hannover, Germany
{czh5372, pum10}@psu.edu

Abstract

Underpinning much of the recent progress in deep learning is the transformer architecture, which takes as input a sequence of embeddings E and emits an updated sequence of embeddings E'. A special [CLS] embedding is often included in this sequence, serving as a description of the sequence once processed and used as the basis for subsequent sequence-level tasks. The processed [CLS] embedding loses utility, however, when the model is presented with a multi-entity sequence and asked to perform an entity-specific task. When processing a multi-speaker dialogue, for example, the [CLS] embedding describes the entire dialogue, not any individual utterance/speaker. Existing methods toward entity-specific prediction involve redundant computation or post-processing outside of the transformer. We present a novel methodology for deriving entity-specific embeddings from a multi-entity sequence completely within the transformer, with a *loose* definition of *entity* amenable to many problem spaces. To show the generic applicability of our method, we apply it to widely different tasks: emotion recognition in conversation and player performance projection in baseball and show that it can be used to achieve SOTA in both. Code can be found at https://github.com/c-heat16/EntitySpecificEmbeddings.

Keywords: Emotion Recognition, Representation Learning, Sequential Modeling

1. Introduction

Representation learning - learning representation of *things* that are useful in performing tasks related to those *things* (Bengio et al., 2013; Guo et al., 2019; Zhang et al., 2018) - is at the core of much of modern deep learning (DL). Underpinning much of the work in this discipline is the transformer architecture (Vaswani et al., 2017). At a high level, the architecture takes as input a sequence of embeddings *E* and outputs an updated sequence of embeddings *E'* of the same shape. A special [CLS] token is often appended to the sequence, serving as a summary of the input once processed.

While directly useful for many downstream tasks, the processed [CLS] embedding is not as useful for performing entity-specific tasks based on sequences influenced by multiple entities. For example, consider the task of utterance-based emotion and/or sentiment classification in a multi-speaker dialogue in the MELD dataset (Poria et al., 2018), pictured below in Figure 1. The processed [CLS] embedding would describe the conversation as a whole, not an individual utterance.

Accordingly, a number of works have explored how the [CLS] embedding can be directed to accumulate information describing a specific subsection of the sequence being processed, such as a specific utterance in a dialogue (Song et al., 2022a,b). Although useful in directing [CLS] towards a specific utterance, such approaches incur a significant amount of redundant computation as they can only derive an embedding for one utterance at a time. That is, a dialogue containing N utterances must be processed N times to derive an embedding for each utterance therein.

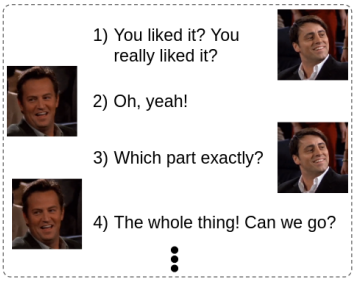

Figure 1: Example from MELD (Poria et al., 2018).

Most often, an language model (LM) processes the sequence of text and outputs an updated embedding of each word/token which is *post-processed* before being used as basis for making the utterance-based classification (Heaton and Schwartz, 2020; Liu et al., 2023a; Shmueli and Ku, 2019; Song et al., 2022b). This *post-processing* can serve two roles: 1) deriving utterance-specific embeddings and/or 2) leveraging the relationships between utterance-specific embeddings, often via a Graph Neural Network (GNN) module. In some cases, utterance-specific embedding are taken as the *average* of their associated contextual embeddings. We speculate this could be improved, as Reimers and Gurevych (2019) have demonstrated a simple average of contextual embeddings provides less utility than a learned function of the same. Furthermore, under such a formulation, the relationships between utterances are not leveraged until the very end of the processing pipeline. We

LREC-COLING 2024, pages 4675–4684
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

question if such relationships should perhaps be leveraged throughout the entire processing pipeline. Since transformers can be viewed as a special type of GNN (Veličković, 2023), such an approach seems plausible.

To this end, we present a general method for deriving entity-specific embeddings for all entities in a multi-entity sequence via the introduction of new special tokens and the manipulation of the attention mask presented to the model. Our entity embeddings can be seen as a special case of the traditional [CLS] embeddings, absorbing signal describing a specific subset of the corresponding input sequence. Furthermore, we enable the model to leverage the relationships between entities at each layer of the LM by further manipulation of the entity-to-entity attention mask, resulting in a fully-connected graph between entity embeddings. Application on the task of emotion recognition in conversation (ERC) yields new SOTA on the EmoryNLP benchmark (Zahiri and Choi, 2017), and ensemble of ours and the previous SOTA pushes the SOTA even higher on the MELD benchmark (Poria et al., 2018). To demonstrate the generic nature of our proposed approach, we apply it to the task of performance projection in Major League Baseball (MLB), surpassing statistical baselines that are traditionally used.

Our contributions are as follows:

- Present a novel method for deriving entity-specific embeddings from a multi-entity sequence and leveraging the relationship between said entities, completely within a transformer model
- Empirically show that our solution improves performance in both pre-training and fine-tuning, establishing a new SOTA on the MELD and EmoryNLP ERC datasets
- Demonstrate the generic nature of the proposed methodology via application on a completely different task: player performance prediction in the MLB, outperforming statistical baselines that have been previously used for this task.

2. Related Work

Here we describe previous approaches towards making entity-specific predictions from multi-entity sequences and a brief introduction on player performance projection in professional baseball.

2.1. Analyzing Multi-Entity Sequences

Dialogues are a well-known example of a sequence of text involving two or more entities (Ni et al., 2023). The meaning of "entity of interest" (hereafter we just use "entity" as a shorthand) can change based on the task at hand. In the MELD benchmark (Poria et al., 2018) (Figure 1), for example, the *entity* is assumed to be the utterance - the task is to make a classification with respect to a particular utterance. In tasks such as speaker attribute classification (*i.e.* inferring a speaker's age/gender), one or more speakers can effectuate one or more utterances, and the task is to infer some characteristic of the speaker given their utterances (Tigunova et al., 2019; Welch et al., 2019). The *entity* is assumed to be each speaker in such case.

Many models designed for such applications are only able to make predictions for an individual entity at once, even though they analyze multiple entities in the course of doing so. Song et al. (2022a,b), for example present a RoBERTa model (Liu et al., 2019) model with a dialogue formatted as

$$X_i = [< s >, s_{t-k}, u_{t-k}, ..., s_t, u_t, < /s >, Q]$$

where s_i and u_i denote the *speaker* and spoken *utterance* at timestep $t = i$, while Q contains the text "for u_t, s_t feels [MASK]." Q is included to help the model "hone-in" on the task at hand, denoting the speaker and utterance pair about which questions will be asked.

The *InstructERC* model applies a similar technique to the Llama2 model (Lei et al., 2023) and currently sits atop the leaderboard for the emotion recognition benchmarks MELD[1] (Poria et al., 2018) and EmoryNLP[2] (Zahiri and Choi, 2017). In addition to *prompting* the model to generate an emotional label for a particular utterance within the dialogue, they *augment* each record by *retrieving* example utterances with similar semantic content and their corresponding labels.

Although current SOTA on the two aforementioned benchmarks, *InstructERC* is not without downsides, primarily redundant computation. When processing a dialogue containing N utterances, the model is *required* to process the dialogue N times to make N predictions, incurring a significant amount of redundant computation. While this formulation may be desired in some applications, a more computationally efficient approach would have the model make emotion predictions for all utterances it is given at once.

Methods that derive entity-specific embeddings typically combine a language model "backbone" and a supplemental entity processing module (Heaton and Schwartz, 2020; Huang et al., 2019; Shmueli and Ku, 2019; Song et al., 2022b). Liu et al.

[1] https://paperswithcode.com/sota/emotion-recognition-in-conversation-on-meld

[2] https://paperswithcode.com/sota/emotion-recognition-in-conversation-on-4

(2023b), for example, utilize RoBERTa (Liu et al., 2019) to process a multi-speaker dialogue before extracting embeddings for each utterance/speaker, which are further processed by RNNs. Similarly, Lee and Choi (2021) use a graph neural network (GNN) module in conjunction with a recurrent neural network (RNN) module to process embeddings emitted by RoBERTa, which correspond to each utterance . While these methods provide utility, a first-step in many of the post-processing phases is the *averaging* of context embeddings corresponding to a particular utterance/speaker before being post-processed by a supplemental module. We posit that this could be improved, as Reimers and Gurevych (2019) have demonstrated that a simple average of processed contextual embeddings provides less utility than a well-learned function of the same. Furthermore, post-processing is done without regard for the original context, which may be sub-optimal.

2.1.1. Attention Mask Manipulation

The RoBERTa-based HiDialog is the only model we are aware of that constructs entity-specific embeddings from a multi-entity sequence completely *within* the transformer backbone *without* the use of averaging (Liu et al., 2023a). They introduce special *speaker* and *turn* embeddings added to the model's vocabulary. By appending the new embeddings to the input sequence and carefully manipulating the attention mask, the authors aim to capture *speaker* and *turn* specific information that can be used towards downstream prediction. Instead of biasing the entire model by manipulating the input sequence, the model's attention mask is manipulated to bias a particular sub-sequence of the input. Once processed by RoBERTa, the [CLS], *speaker*, and *turn* embeddings are extracted from the output sequence and processed by a "heterogeneous graph module" to model the intra-utterance relations before making utterance-level predictions. The attention mask is presented in Figure 2.

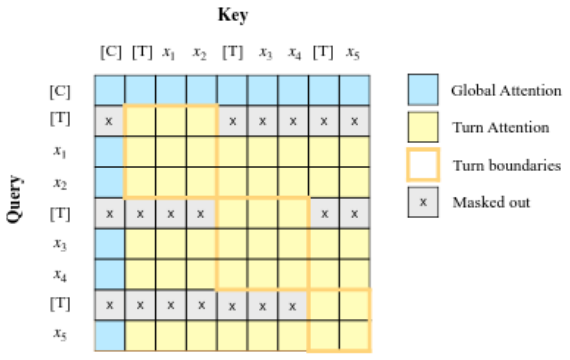

Figure 2: Visualization of the HiDialog attention matrix as proposed by Liu et al. (2023a).

As we can see in Figure 2, the *turn* embeddings are given *symmetric* attention between themselves and the contextual embeddings in that turn, and the [CLS] embedding is allowed to tend to the *turn* embeddings, but not vice-versa. While this approach has shown to be efficacious, sitting at #3 on the MELD leaderboard, we can improve upon it. For example, if multiple entity embeddings can be derived in one pass of a transformer, can the relationships between the utterances *also* be modeled *within* the transformer? Transformers can be viewed as GNNs (Veličković, 2023), so this seems plausible.

2.2. Player Performance Projection

The vast majority of methods underpinning sports analytics, particularly player performance projection in professional baseball, are reminiscent of the "bag-of-words" (BoW) approach from the early days of computational linguistics (Harris, 1954). An "expert" identifies *important* in-game events and players are evaluated based on how often these *important* events occur while they are in the game. While such methods provide some utility, they incur the same limitations as the BoW approach, namely (for this application) the lack of word-sense (*event-sense*) disambiguation. We posit more insights can be gained by presenting a machine learning model with a description of the game as a sequence of pitches, allowing it to construct rich, contextual embeddings for each event. See the work of (Costa et al., 2019) for an in-depth discussion of the field.

3. Deriving Entity-Specific Embeddings

In this section, we present our general method for deriving entity-specific embeddings from multi-entity sequences. The proposed approach can be leveraged to train a model from scratch or applied to an already pre-trained model.

First, an [ENTITY] token is added to the model's vocabulary. Records are then constructed as they would be normally - *i.e.* vanilla construction of the input sequence (token IDs), attention mask, padding mask, *etc*. Entity-related construction begins by identifying the entities $E = \{e_1, ..., e_N\}$ for which embeddings should be derived. We assume no strict definition of *entity*, using simply to denote sub-sections of the context which are of interest.

Once identified, a boolean mask denoting portions of the input sequence with which each entity interact $T = \{t_{e_1}, t_{e_2}, ..., t_{e_N}\}$, where $t_{e_i} \in [0, 1]^{1 \times SeqLen}$, is constructed. This is often as simple as identifying the speaker or utterance ID corresponding to a particular timestep. Then, $|E|$ entity embeddings are appended to the input sequence, and T is used to update the attention mask such

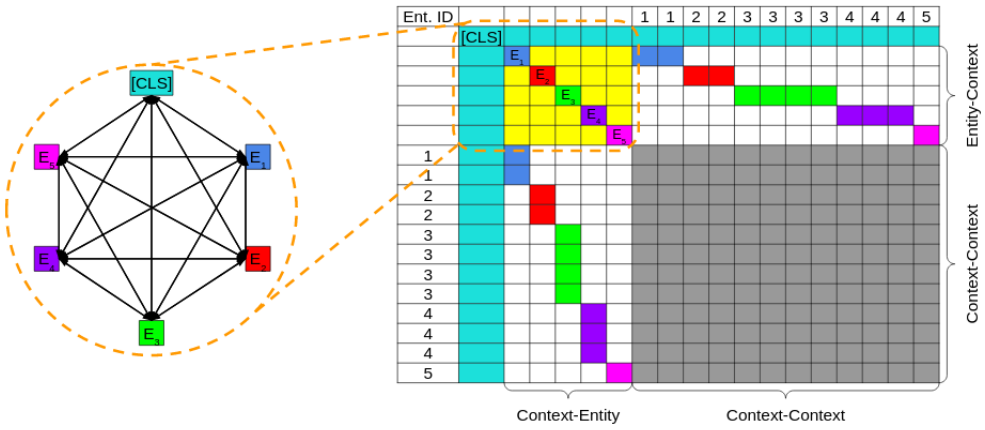

Figure 3: Example manifestation of our approach. White/transparent cells denote a closure of the attention mask. "Ent. ID" denotes the entity to which contextual embeddings correspond, included for illustration purposes only. Optional entity-to-entity attention is denoted in yellow - when not employed, the attention denoted in yellow is closed (white). Enabling entity-to-entity attention results in a fully connected graph, visualized in the left portion of the image.

that the embedding for e_i can only tend to indices with which e_i participates or interacts. All entity embeddings are allowed to tend to themselves. Each entity begins the same, but accumulates different information based on the portion of the sequence to which it can tend. The [ENTITY] tokens act as a special case of the [CLS] token - the latter describes the entire sequence, the former a particular subsequence thereof.

One important feature to note is the ability of two or more entity embeddings to attend to the same set of contextual embeddings. In the HiDialog approach (Liu et al., 2023a), for example, **one and only one** turn (*entity*) embedding can tend to a particular timestep. This formulation makes sense for their domain of application, as only one speaker can effectuate an utterance at any given time. However, it limits the number of potential applications for such approaches. In the audio domain, for example, one may want to derive an embedding for each instrument in a symphony, similar to (Shi et al., 2022). In such cases, multiple instruments play at once, a scenario HiDialog can not handle. Additionally, in the domain of sport, it is common for two or more players to mutually influence the events at a particular timestep, e.g., a pitcher, batter, and one or more outfielders may be involved in one play.

3.1. Entity-to-Entity Attention

While the formulation above will be sufficient in many applications, there exist applications in which leveraging the relationship between entities in the sequence is desired, such as emotion recognition in conversation (Liu et al., 2023a). For example, if all of the utterances in the surrounding context of utterance (entity) u_i express the emotion *joy*, while occasionally possible, it is perhaps unlikely that u_i would express an emotion such as *fear*. Thus, a method for leveraging the relationship between entities in the sequence is desired. While existing work has shown this can be done *outside* of the transformer architecture, we explore if this can be done *within* the transformer architecture.

Specifically, we propose a method for allowing a transformer model to leverage the "global", bidirectional relationship between the entities. First, we simply *open* the attention mask between all entities E present in the sequence, resulting in a fully-connected graph between entity embeddings, visualized in the left side of Figure 3. Next, we introduce new additive *position embeddings* and add them to the E embeddings present in the sequence. These *position embeddings* are distinct from the position embeddings already learned by the model. During training, we randomly mask a subset of the entity-to-entity (E2E) attention, removing the ability of the entities to tend to a portion of surrounding entities. A visual representation of our approach when entity-to-entity attention is enabled is presented in Figure 3, with E2E attention depicted in yellow.

4. Experiments

We demonstrate the efficacy of our method and its generic nature by experimenting with two widely different domains: NLP and sports analytics. NLP experiments are performed "on top" of pre-trained models, while we trained models from scratch for our baseball application, achieving SOTA and thus demonstrating the versatility of our approach.

4.1. Emotion Recognition In Conversation

We begin with the pre-trained RoBERTa (Liu et al., 2019) when applying our method to this task. Instead of learning a special entity embedding from scratch, we instantiate the entity embedding as the pre-trained [CLS] embedding. The token serves a similar role - deriving summary information from the context to which it has access - and is thus a good starting point for the entity embedding.

Special care must also be taken when implementing the entity-to-entity attention. While each entity could be given the *same* positional embedding, such approach would not instill a temporal order on the entity embeddings themselves. To this end, we create a new set of learnable position embeddings, for use **only** with entity embeddings. These embeddings are instantiated from the model's vanilla positional embeddings, but optimized separately.

We leverage *utterance IDs* present in each dataset to equip RoBERTa with the tooling to derive embeddings for each utterance in the dialogue. The model is first subjected to extended-pre-training via dynamic masked language modeling (MLM) (Liu et al., 2019) to learn how to leverage the new entity embeddings before being fine-tuned for the task of ERC. In fine-tuning, the model optimizes a combination of cross-entropy and prototype-cosine similarity loss (Song et al., 2022a), weighted 9:1. The prototype for each label is computed by randomly sampling 64 of the 256 most recent records for each label, taking the average.

4.1.1. Attention Analysis

In addition to evaluating the models on the tasks they were trained to perform, we also explore the behaviour of the attention heads within the models to better understand how the newly introduced entity embeddings are utilized. To this end, we record and dissect the attention probabilities constructed by the model during processing.

First, we explore the extent to which utterance (entity) embeddings and contextual embeddings tend to each other during processing. In previous approaches which use a GNN to post-process utterance embeddings emitted by a LM backbone, inherent is the assumption that the inter-utterance relationships need not be leveraged until the final stages of processing. We are curious if our model manifests with the same behavior or if it leverages the entity-entity relationships earlier in the pipeline.

We also explore if and how the emotion expressed in each utterance influences attention behavior. Specifically, we explore the propensity for utterance embeddings to tend to other utterance embeddings according to their emotion. The most populous emotion classes will inherently draw more attention from utterances in the corresponding context, regardless of emotion, so we account for the co-occurrence between emotions in our analysis.

4.1.2. Datasets

We identify three multi-speaker dialogue datasets for use in this study, MELD (Poria et al., 2018), EmoryNLP (Zahiri and Choi, 2017), and IEMOCAP (Busso et al., 2008). MELD and EmoryNLP have prescribed train/test splits which are used when appropriate for pre-training and fine-tuning, but the same can't be said for IEMOCAP. For that reason, IEMOCAP is only used for pre-training.

MELD (Poria et al., 2018) and EmoryNLP (Zahiri and Choi, 2017) are both derived from the Friends sitcom and serve as ERC benchmarks for the community. MELD includes 1,433 dialogues containing 13,708 utterances annotated as one of seven emotions: *anger*, *disgust*, *fear*, *joy*, *neutral*, *sadness*, and *surprise*. EmoryNLP contains 897 *scenes* (dialogues) and a total of 12,606 utterances annotated as one of seven emotions: *sad*, *mad*, *scared*, *powerful*, *peaceful*, *joyful*, and *neutral*. Models applied to EmoryNLP are trained on *both* ERC corpora, with *scared* mapped to *fear* and *mad* to *anger*.

4.2. Sports Analytics

To demonstrate the generic nature of the proposed method, we also apply it to the task of player performance projection in the MLB (professional baseball). At a high level, the game of baseball can be viewed as a sequence of pitches with two players influencing the result of each pitch - *i.e.* a multi-entity sequence. We describe the application in this domain at a high level, but direct the interested reader to related work by Heaton and Mitra (2022).

Application begins by pre-training a transformer model from scratch. We equip the model with the tooling to derive *entity embeddings* for the 5 pitchers and 45 batters who appear most frequently in each sequence. The model is pre-trained with an infilling task similar to BERT's masked language modeling (MLM) (Devlin et al., 2018) which we term masked *gamestate* modeling (MGM). When presented with a sequence of events, roughly 15% of the timesteps are *masked* and the model is tasked with discerning *what happened* at said timesteps. The model predicts *what happened* in terms of two categorical values - the *event* and the *gamestate delta*. The *event* describes what happened in terms fans of the game typically use - *single*, *home run*, *flyout*, *etc.* - while the *gamestate delta* describes the structural changes in the game - base occupancy, ball-strike count, run-count, and out-count.

Once pre-trained, the model is fine-tuned to make predictions about future player performance given the sequence describing said player's last 15

games. During fine-tuning the model can be trained to predict various aspects of a player or team's performance. We fine-tune the models to predict the number of strikeouts, walks, and hits that will be recorded by the starting pitcher and starting batters *against the opposing starting pitcher*. These targets were selected as they are general indicators of in-game player performance and are commonly discussed by fans/analysts of the game[3].

4.2.1. Dataset

We construct our dataset using the same methodology as Heaton et al. (Heaton and Mitra, 2022), updated to included data through the 2021 MLB season. Our resulting dataset contains 4.6 million pitches, 1,884 unique pitchers, 2,229 unique batters and 15,743 games. This data is converted to a pitch-by-pitch sequence describing the 1) gamestate when each pitch was thrown, 2) the pitcher and batter involved, and 3) change in gamestate resulting from the pitch via real-valued statistics and learned embeddings. Data is presented to the model the same as in (Heaton and Mitra, 2022) and we direct the curious reader to that paper as it is beyond the scope of this work.

5. Results

5.1. NLP

5.1.1. Extended Pre-training

The results of performing extended pre-training as described above is presented in Table 1. RoBERTa is subjected to 15 epochs of extented pre-training with an Adam optimizer, batch size of 32, $l2$ weight of 0.01, a learning rate of $1e-5$ with a cosine schedule and 150 iterations of warmup. We perform experiments with and without our entity embedding method, with and without fine-tuning, to determine its efficacy.

Entity Embeddings	Fine-tune	Ppl
		2.79
	✓	2.32
✓		2.58
✓	✓	**2.26**

Table 1: MLM perplexity scores.

The presented evaluation metric for MLM predictions is perplexity adapted for the bidirectional case, computed as $PPL(P) = \frac{1}{N}\sum_{i=1}^{N} 2^{H(p_i)}$, where p_i is the probability distribution for MLM prediction i and $H(p_i)$ is the cross-entropy for prediction distribution p_i. Perplexity has a range of $[1, \infty)$, with lower values indicative of a stronger model.

We see that even *without* fine-tuning the model, adding entity embeddings to the input sequence and corresponding manipulation of attention mask lowers uncertainty score by 7.5%. Finetuning the model results in a decrease in uncertainty score of almost 19%. Comparing the fine-tuned models reveals that the model equipped with entity embeddings achieves a 2.5% lower uncertainty score. This suggests careful manipulation of the input sequence and attention mask is able to improve the performance of a pre-trained model without any updates, a frontier perhaps related to the growing field of *prompt engineering* (White et al., 2023).

Model	MELD	EmoryNLP	# Records Processed
EmotionIC	66.40	40.01	D
HiDialog	66.96	N/A	D
SPCL-CL-ERC	67.25	40.94	$D \times U$
InstructERC	69.15	41.39†	$D \times U$
EE (Ours)	66.53	40.69 (41.39†)	D
EE+E2E (Ours)	66.91	41.98 (42.43†)	D
EE+E2E (Ours) & SPCL-CL-ERC	**70.26**	**47.61** (**51.85**†)	$(D \times U)$ $+D$

Table 2: Fine-tuning results (w-F1). D/U denote the number of dialogues/utterances in each dataset. †COSMIC test split (Ghosal et al., 2020).

5.1.2. Emotion Recognition

In Table 2 we present the results of fine-tuning RoBERTa on MELD and EmoryNLP after extended pre-training. Models are trained using an Adam optimizer, batch size of 32, $l2$ weight of 0.01, and learning rate of $2e-5$. For MELD, models were trained for 10 epochs and 120 warmup iterations, while for EmoryNLP models were trained for 12 epochs with 160 warmup iterations. Scores on EmoryNLP reflect a model trained on both datasets.

We also explore how our model can be ensembled with the SPCL-CL-ERC model, retraining the model using the codebase provided by the authors (Song et al., 2022a). Our reproduction achieved a w-F1 score of only 65.97/39.73 on MELD/EmoryNLP; the authors report scores of 67.25 and 40.94. A checkpoint for InstructERC is not available, and we were unable to train a model to convergence using provided code. Upon inspection, it became apparent InstructERC was applied to the COSMIC (Ghosal et al., 2020) test split for EmoryNLP, which only contains ∼75% of the records. We also apply our model to this subset, denoting it with † in Table 2.

[3]https://www.mlb.com/glossary/standard-stats

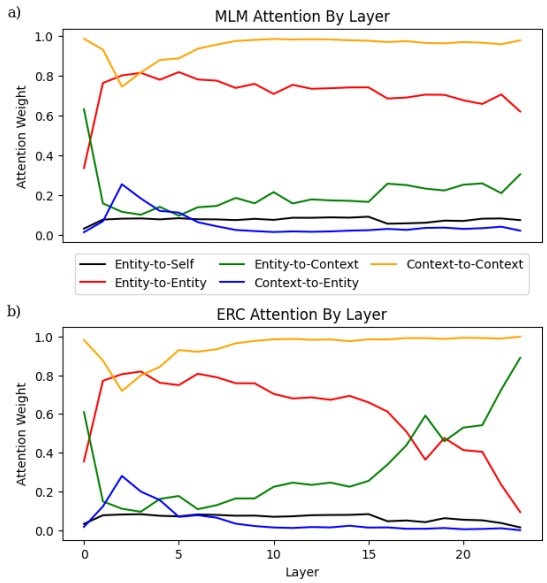

Figure 4: Attention behavior in RoBERTa-Large after a) pre-training (MLM) and b) fine-tuning (ERC).

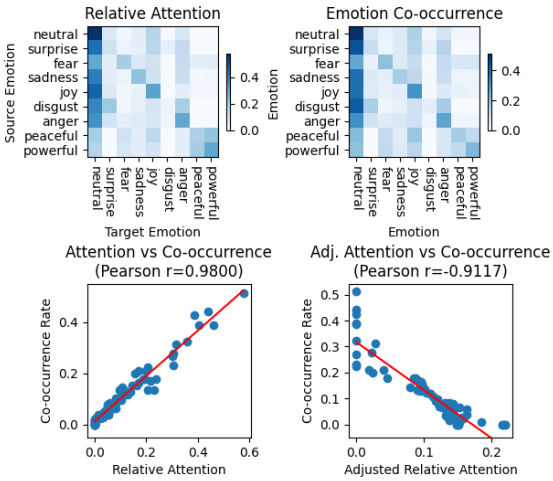

Figure 5: Analysis of how the emotion of an utterances influences attention behavior.

Model checkpoints/codebases are not provided for each model in Table 2, so we cannot evaluate the run-time of each model. As a proxy for general run-time we present the number of records processed by each model to make predictions for all utterances in each dataset, highlighting the redundant compute in many approaches.

Comparing the efficacy of our approach *with* and *without* entity-to-entity attention demonstrates the ability of RoBERTa to leverage relationships between entities. Enabling entity-to-entity attention improves performance on MELD by 0.6% and on EmoryNLP by 3.1% compared to the base entity embedding model. The score of 66.91 weighted-F1 is on-par with the performance of HiDialog, which leverages a "heterogeneous graph module" **outside** of RoBERTa. Our approach establishes new SOTA on EmoryNLP, surpassing *InstructERC* which is based on the significantly larger Llama2 (7B vs 350M parms) and processes more records by a factor of U. When our predictions are ensembled with those of SPCL-CL-ERC a new SOTA is established at 70.26/47.61 on MELD/EmoryNLP.

5.2. Attention Analysis

To better understand how the model utilizes the newly introduced entity embeddings, we visualize the attention patterns exhibited by the model. First, we explore the propensity for embeddings of each type - *entity* (uttterance) embeddings and traditional *contextual* embeddings - to tend to different portions of the sequence during both stages of training (Figure 4). As we can see, the attention of context tokens (*i.e.* Context-to-Entity and Context-to-Context) is largely the same after both phases of training. The *context* tokens devote more attention towards entity embeddings in the early layers (1-6) of the model, but then largely tend to each other in subsequent layers. The attention behaviour of the entity embeddings change significantly for ERC compared to MLM. Trained for MLM, entity embeddings primarily tend to each other after the first layer, although they gradually tend to more of the context with each layer of processing. This behavior changes in fine-tuning, particularly in the upper layers of the model, where entity embeddings tend to the context more than other entity embeddings. This learned behavior is in stark contrast with the way the relations between entities (utterances) have previously been leveraged. Previous approaches (§2.1) would leverage contextual information first, and *then* the relationship between entities. Our results suggest that the relationship between entities (utterances) should be leveraged earlier in the processing pipeline.

Furthermore, we explore if the emotion expressed in each utterance influences the behavior of attention in the model, presenting the results in Figure 5. As we can see, the average attention that an utterance expressing one emotion prescribes to an utterance expressing some other emotion is highly correlated with the co-occurrence of each emotion pair. However, when the co-occurrence of each emotion pair is considered - subtracting co-occurrence frequency from from the average attention probability - we see that utterances expressing emotions that seldom appear in the same context garner a relatively larger amount of attention. Intuitively, this makes sense - uncommon artifacts are useful in discerning the emotional landscape.

Config	Strikeouts (K)				Walks (BB)				Hits (H)					Average	
	Pitcher		Batter		Pitcher		Batter		Pitcher		Batter				
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	Strk	MSE	R^2
Stat	5.43	0.20	**0.43**	0.05	1.73	0.04	0.19	**0.02**	**4.48**	0.07	**0.43**	0.03	10	2.11	0.07
Embd Avg	5.46	0.20	0.44	0.04	1.49	0.03	**0.17**	0.01	4.71	0.06	**0.43**	0.03	5	2.12	0.06
Entity Embd	**5.34**	**0.22**	**0.43**	**0.06**	**1.48**	**0.05**	**0.17**	**0.02**	4.57	**0.08**	**0.43**	**0.04**	**11**	**2.07**	**0.08**

Table 3: Fine-tuning results. (Mean Squared Error (MSE) and the coefficient of determination (R^2)). *Embd Avg* denotes a transformer-baseline in which embeddings corresponding to each entity are extracted and averaged. STAT denotes a statistics-based baseline comparable to industry-standard models.

5.3. Sports Analytics

5.3.1. Pre-training

We pre-train our models to understand sequences describing *all* types of player records - starting pitchers, relief pitchers, and starting batters. However, because relief pitchers can enter the game in the middle of the inning, such records would not make for a fair analysis, akin to trying to generate the correct continuation without access to the context. Thus, we present our trained models' MGM performance on sequences of starting players *only*.

Config	Ppl	F1
Vanilla	1.43	0.55
Entity Embeddings	**1.28**	**0.63**

Table 4: MGM performance after pre-training. Metrics are perplexity and weighted F1.

Specifically, an 8 layer, pre-norm transformer model (Xiong et al., 2020) with an internal dimension of 768 and 8 attention heads was trained using an Adam optimizer with a learning rate of `5e-4`, *L2* weight of `1e-4`, and a batch size of 30. When performing MGM, *event*/*gamestate delta* predictions were given a weight of 0.1/0.9, respectively, in computing the overall MGM loss. Models were pre-trained on 5M records, with repeats. Pre-training results are presented in Table 4. Metrics were obtained by having each model perform MGM on the same set of 25k records for the 2021 season.

Pre-training results presented in Table 4 show that our entity-embedding approach improves overall modeling performance. Compared to baseline, our methodology yields a decrease in perplexity of 10.49% and increase in F1 score of 14.55%. These findings are in agreement with those from the realm of language, in which RoBERTa saw a 2.50% decrease in perplexity. We speculate our method provides our method yields more improvement in the sports domain because a weaker baseline model, resulting from a smaller training corpus (4.6M vs 100M unique tokens). Lower perplexity magnitude is likely due to the smaller vocabulary size ($\sim$100 vs $\sim$50k tokens), and baseball games having a more stringent structure than language. These findings perhaps suggest that our method can provide benefits in low-resource domains.

5.3.2. Fine-tuning

We then took the pre-trained models and fine-tuned them to make predictions about future player performance. During fine-tuning, models were trained on data from the 2015-2020 seasons and evaluated on data from the 2021 season. Simple, linear prediction heads were added on top of our pre-trained models and fine-tuned using an Adam optimizer with a learning rate of `2e-4` and *L2* of `1e-6` for up to eight epochs. Predictions were evaluated directly, but also used to identify how many days in a row each model could successfully identify a batter to reach base (Strk), similar to MLB's "Beat the Streak" competition[4].

As a baseline, we also explore how a model trained without our entity embedding method performs. To this end we select the subset of event embeddings in which each player participated, taking the average as that player's embedding. "Stat" denotes a statistics-based baseline comparable with industry-standard approaches to this task (Bailey, 2017). Statistics describing each player in the 15 games leading up to game G are calculated, filtered via a mutual-information-based feature selection process in the python package `SKLearn`(Pedregosa et al., 2011), and used to train an XGBoost model (Chen and Guestrin, 2016).

In analyzing the results in Table 3 we see that the model equipped with our entity embeddings outperforms both baseline methods. The statistical baseline yields a lower MSE when predicting pitcher hits, but upon inspection, it is because the statistical model emits predictions that are much closer to the population mean than our approach. Specifically, the statistical model has a range of 4.58 in terms pitcher hit predictions, while our approach has a range of 6.62, leaving ours preferred. Improvement of our approach over the embedding average baseline also reinforces the notion that a simple average of contextual embeddings provides less utility than a well-learned function of the same.

[4]https://www.mlb.com/apps/beat-the-streak

6. Conclusion

We proposed and evaluated a simple, effective, and versatile approach for deriving entity-specific embeddings from a multi-entity sequence. Application of our approach yields significant improvements during pre-training and fine-tuning in the disparate domains of NLP and sports analytics. Alone, our approach improves over the previous SOTA for ERC on the EmoryNLP dataset by 1.43%. Ensembled with the SPCL-CL-ERC model, we advance SOTA by 15.03% on EmoryNLP and 1.61% on MELD. Applied to player performance projection, we outperform previous approaches by 1.90% and 14.29% in MSE and R^2, respectively. In doing so, we explore the attention patterns within each layer of our model, finding interesting behavior with respect to model layer and utterance emotion.

7. Bibliographical References

Sarah Reid Bailey. 2017. Forecasting batting averages in mlb.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Gabriel B Costa, Michael R Huber, and John T Saccoman. 2019. *Understanding sabermetrics: An introduction to the science of baseball statistics*. McFarland.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. Cosmic: Commonsense knowledge for emotion identification in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2470–2481.

Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Connor Heaton and Prasenjit Mitra. 2022. Using machine learning to describe how players impact the game in the mlb. In *The 16th Annual MIT Sloan Sports Analytics Conference*.

Connor T Heaton and David M Schwartz. 2020. Language models as emotional classifiers for textual conversation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2918–2926.

Yen-Hao Huang, Ssu-Rui Lee, Mau-Yun Ma, Yi-Hsin Chen, Ya-Wen Yu, and Yi-Shin Chen. 2019. Emotionx-idea: Emotion bert–an affectional model for conversation. *arXiv preprint arXiv:1908.06264*.

Bongseok Lee and Yong Suk Choi. 2021. Graph based network with contextualized representations of turns in dialogue. *arXiv preprint arXiv:2109.04008*.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Xiao Liu, Jian Zhang, Heng Zhang, Fuzhao Xue, and Yang You. 2023a. Hierarchical dialogue understanding with special tokens and turn-level attention. *arXiv preprint arXiv:2305.00262*.

Yingjian Liu, Jiang Li, Xiaoping Wang, and Zhigang Zeng. 2023b. Emotionic: Emotional inertia and contagion-driven dependency modelling for emotion recognition in conversation. *arXiv e-prints*, pages arXiv–2303.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2023. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, 56(4):3055–3155.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Xuan Shi, Erica Cooper, and Junichi Yamagishi. 2022. Use of speaker recognition approaches for learning and evaluating embedding representations of musical instrument sounds. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:367–377.

Boaz Shmueli and Lun-Wei Ku. 2019. Socialnlp emotionx 2019 challenge overview: Predicting emotions in spoken dialogues and chats. *arXiv preprint arXiv:1909.07734*.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022a. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713*.

Xiaohui Song, Liangjun Zang, Rong Zhang, Songlin Hu, and Longtao Huang. 2022b. Emotionflow: Capture the dialogue level emotion transitions. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8542–8546. IEEE.

Anna Tigunova, Andrew Yates, Paramita Mirza, and Gerhard Weikum. 2019. Listening between the lines: Learning personal attributes from conversations. In *The World Wide Web Conference*, pages 1818–1828.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković. 2023. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538.

Charles Welch, Verónica Pérez-Rosas, Jonathan K Kummerfeld, and Rada Mihalcea. 2019. Look who's talking: Inferring speaker attributes from personal longitudinal dialog. *arXiv preprint arXiv:1904.11610*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2018. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28.

8. Language Resource References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Sayyed M Zahiri and Jinho D Choi. 2017. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. *arXiv preprint arXiv:1708.04299*.