HLTRI at W-NUT 2020 Shared Task-3: COVID-19 Event Extraction from Twitter Using Multi-Task Hopfield Pooling

Maxwell Weinzierl
Human Language Technology
Research Institute,
University of Texas at Dallas
maw150130@utdallas.edu

Sanda Harabagiu
Human Language Technology
Research Institute,
University of Texas at Dallas
sanda@utdallas.edu

Abstract

Extracting structured knowledge involving self-reported events related to the COVID-19 pandemic from Twitter has the potential to inform surveillance systems that play a critical role in public health. The event extraction challenge presented by the W-NUT 2020 Shared Task 3 focused on the identification of five types of events relevant to the COVID-19 pandemic and their respective set of pre-defined slots encoding demographic, epidemiological, clinical as well as spatial, temporal or subjective knowledge. Our participation in the challenge led to the design of a neural architecture for jointly identifying all Event Slots expressed in a tweet relevant to an event of interest. This architecture uses COVID-Twitter-BERT as the pre-trained language model. In addition, to learn *text span embeddings* for each Event Slot, we relied on a special case of Hopfield Networks, namely Hopfield pooling. The results of the shared task evaluation indicate that our system performs best when it is trained on a larger dataset, while it remains competitive when training on smaller datasets.

1 Introduction

With the outbreak of the COVID-19 pandemic, people turned to social media platforms, such as Twitter, to read and to share timely information about their experiences with testing, treatment and deaths caused by the virus. Extracting information about these types of events has the potential to inform COVID-19 surveillance systems, which play a critical role in the public health mission of agencies at the international, national and local level.

The shared task organized by the 6-th Workshop on Noisy User-generated Text (W-NUT) in 2020 focused on extracting COVID-19 events from Twitter by targeting five types of events of interest, which are illustrated in Table 1, and further detailed in Zong et al. (2020). These five types of self-reported events are typically expressed in Twitter postings of people that indicate when they might be at increased risk of COVID-19 due to a coworker or other close contact testing positive for the virus, or when they have symptoms but were denied access to testing. Moreover, for each Event Type of interest, a set of pre-defined slots were provided, to account for information that may answer important questions involving the events (e.g., *Who tested positive? Where did they recently travel? Who is their employer?*). The complete list of Event Slots associated with each Event Type is illustrated in Table 1.

Event Type	Event Slots
TESTED POSITIVE	*age, close_contact, employer, gender, name, recent_travel, relation, when, where*
TESTED NEGATIVE	*age, close_contact, gender, name, relation, when, where*
CAN NOT TEST	*name, relation, symptoms, when, where*
DEATH	*age, name, relation, when, where*
CURE	*opinion, what_cure, who_cure*

Table 1: The Event Types and their corresponding Event Slots defined in the W-NUT 2020 Shared Task on COVID-19 Event Extraction.

Interestingly, events of type TESTED POSITIVE, TESTED NEGATIVE, CAN NOT TEST and DEATH have slots answering questions about *when* and *where*, which help ground the events temporally and spatially, potentially informing systems that try to capture automatically trends of testing results for the COVID-19 virus. Some slots for events of type TESTED POSITIVE, TESTED NEGATIVE, also encode answers about demographic information, e.g. *age*, *gender* as well as epidemiological information, e.g. *close_contact* or *relation*. Only events of the type TESTED POSITIVE have a slot for *employer*. Some clinical information is available through the

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 530–538
Online, Nov 19, 2020. ©2020 Association for Computational Linguistics

slot *symptoms* of events of type CAN NOT TEST or the slots *what_cure* and *who_cure* of the events of type CURE. For the CURE type of events, a slot capturing the opinion of the tweet reporter was also annotated, providing a means for analyzing the change in opinions throughout time.

The organizers of the W-NUT Shared Task 3 provided participants with a set of $7,500$ annotated tweets, $7,013$ of which we were able to download. The tweets are categorized by the type of event they mention. In each category of tweets, every tweet was annotated with (1) the text spans that can be mapped in any of the slots corresponding to the Event Type, as well as (2) the corresponding Event Slot category. Moreover, in each tweet, text spans that were not mapped to any Event Slot were also provided. The training data we have used contains $3,794$ Event Type mentions and $10,778$ Event Slot instances. This timely and richly-annotated twitter dataset allowed participants to design and train their event extraction systems, expecting to be tested on a different set of tweets, which were labeled with the Event Type, probably a byproduct of tweet retrieval using keywords proven to return relevant posting for the events of interest. In the test set, the tweet text spans are also provided.

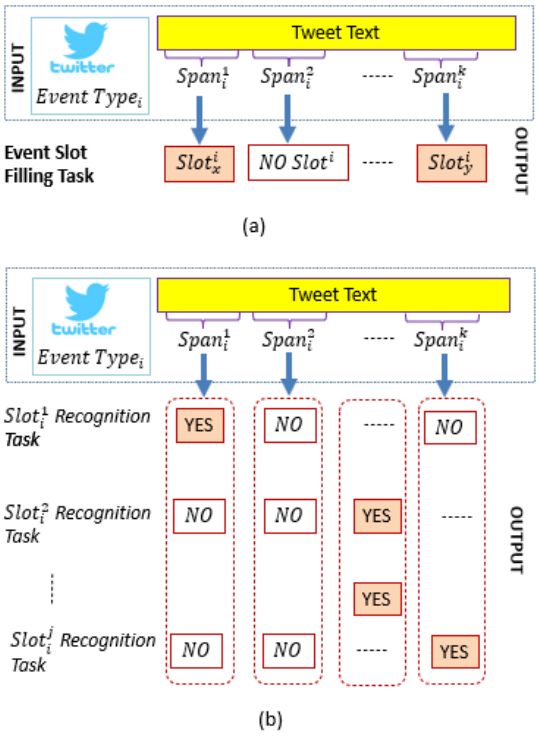

Figure 1: (a) The Task of Event Slot Filling; (b) Multi-Task Binary Classification of Tweet Text Spans.

The challenge of event extraction from Twitter texts was defined as an Event Slot filling task, as illustrated in Figure 1(a). Each tweet text is associated with one of the five *Event Type*$_i$ listed in Table 1 and its defined Event Slots $ES_i = (ES_1^i, ES_2^i, , , ES_{n(i)}^i)$. In addition, a set of text spans from the tweet, $(Span_i^1, Span_i^2, ...Span_i^k)$ are provided. Filling the slots of the mention(s) of an event of *Event Type*$_i$ is made possible by assigning to each tweet text span either to one of the Event Slots ES_j^i for the *Event Type*$_i$ or to none of these Event Slots. To be noted that this is a many-to-many assignment, as (1) the same $Span_i^1$ may be assigned to more than one Event Slot from ES_i and (2) more than one text span can be assigned to the same Event Slot from ES_i. This is because, the notion of *event mention* is avoided, thus when multiple mentions of the same Event Type are expressed in the same tweet, some of the Event Slots that are filled for one of the mention, whereas other are filled for different mentions.

Casting the Event Slot filling as a binary classification problem, as suggested in Zong et al. (2020), is illustrated in Figure 1(b), where we show how a separate recognition task is considered for each of the Event Slots from ES_i. In this case, for each tweet text span, the recognition task of Event Slot ES_j^i is identifying whether any of the provided text spans can fill that slot or not. In this way, the Event Slot filling task is cast as a Multi-Task binary classification problem. As before, the same text span may fill different Event Slots, pertaining to different event mentions.

Because event extraction in this challenge is based on a mapping operation from tweet text spans to Event Slots, we contemplated methods of representing these spans that could take into account more than deep contextual information. We hypothesized that the representation of the tweet text spans should be specific to each of the Event Slot recognition tasks depicted in Figure 1(b). While contextual span representations based on the widely used BERT model (Devlin et al., 2019) have been successful in many applications, e.g. end-to-end relation extraction (Eberts and Ulges, 2020) or coreference resolution (Joshi et al., 2020), we believe that representing tweet text spans could be improved when considering modern Hopfield Networks in which the update rule is the attention mechanism used in the transformer and BERT, an idea recently advocated in Ramsauer et al. (2020). These ideas led the design of our Multi-Task, Event-specific Ex-

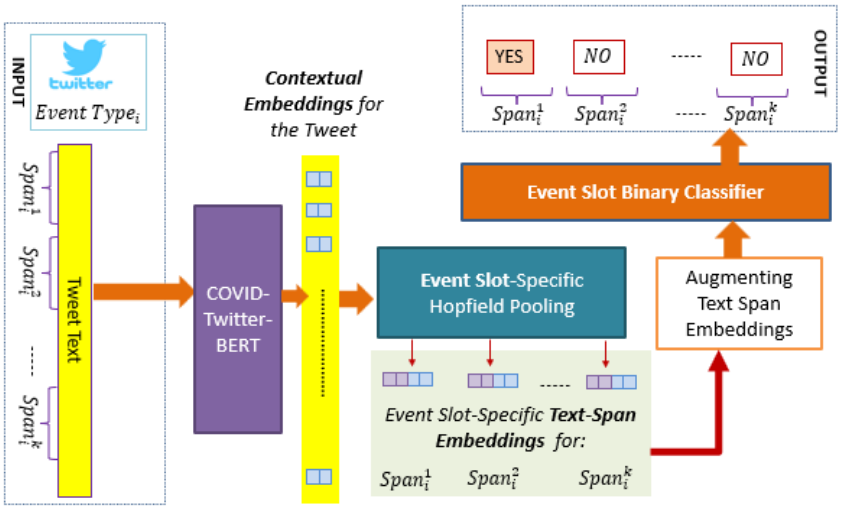

Figure 2: The Multi-Task Event-specific Extraction system using BERT and Hopfield Pooling (MT-EsE.BHP).

traction system using BERT and Hopfield Pooling (MT-EsE.BHP).

2 Related Work

Zong et al. (2020) introduced the collection of tweets used for training in this shared-task along with the annotation schema and provided two baselines: a logistic regression model and a multi-task BERT model. Their multi-task BERT model utilized the contextual embedding for the token <E> as their span embedding, while we expand on this span embedding representation with width embeddings and a slot-specific Hopfield pooling embedding. We also make use of COVID-Twitter-BERT, as opposed to BERT-base used by Zong et al. (2020).

Mackey et al. (2020) utilized an unsupervised approach to cluster tweets in which users discuss experiences associated with possible COVID-19 symptoms. They used the biterm topic model (BTM) (Yan et al., 2013) to identify tweet topic clusters, and then manually annotated tweets which fell within relevant topic clusters. In contrast, we made use of the annotated tweets which discuss relevant COVID-19 events, as provided by the W-NUT 2020 Shared Task-3, designing a supervised approach capable to extract events and their slots, not only symptom discussions.

Zhang et al. (2020) manually identified twitter users which appeared depressed and retrieved historical tweets from these users. They attempted to identify whether a twitter user may be depressed based on their public tweet history using the language model XLNet (Yang et al., 2019) and they perform user-level classification using a Support Vector Machine (SVM). They apply this system to detect and monitor trends of depression during the COVID-19 pandemic. We differ from this system by utilizing a domain-specific language model with COVID-Twitter-BERT is trained to extract several types of events and their corresponding slots from tweets related to COVID-19.

3 The Approach

3.1 The MT-EsE.BHP

The overall architecture of the MT-EsE.BHP is illustrated in Figure 2. Given a tweet in which events of type *Event Type*$_i$ are discussed, along with the text spans that can be potentially mapped into any of the events slots from ES_i, the pre-trained domain-specific language model COVID-Twitter-BERT (Müller et al., 2020) is used to produce for each word-piece token in the tweet a contextual embedding. COVID-Twitter-BERT was pre-trained on a large corpus of Tweets related to COVID-19, and it also contains learned embeddings for URLs (<url>) and @ mentioned Twitter users (@<user>) which are used in place of URLs and usernames.

The contextual embeddings are used to learn embedding representations for each text span from the tweet that can be potentially mapped to any of the Event Slots of the events of *Event Type*$_i$ discussed in the tweet. In the MT-EsE.BHP a separate *text-span embedding* is learned for each Event Slot from ES_i. A special case of a Hopfield Network, namely Hopfield Pooling is used for learning each

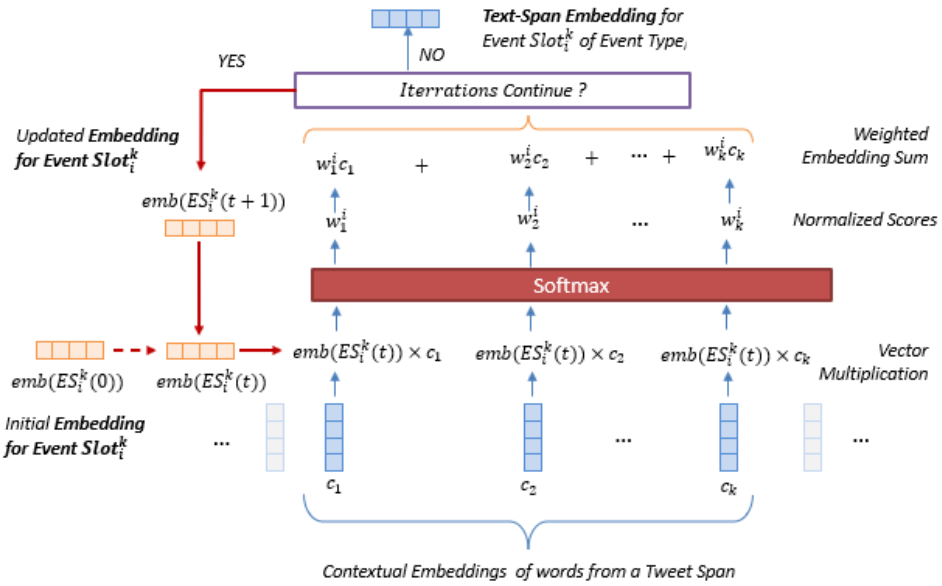

Figure 3: Hopfield Pooling for an Event Slot ES_i^k of any *Event Type*$_i$.

Event Slot-specific text-span embedding. Details of how Hopfield Pooling works are provided in Section 3.2. As Figure 2 shows, because the text-span embeddings have the same dimension, regardless of the width of the text span, the text-span embeddings were augmented by concatenating an embedding corresponding to the width information for each text span. The augmented text-span embeddings are used by each Event Slot Binary classifier, which decides the mapping of each text span from the tweet to any of the Event Slots from ES_i. The binary classifier is implemented as a single-layer Softmax classifier.

3.2 Hopfield Pooling

Continuous Hopfield Networks have recently been shown to be equivalent to iterative attention (Ramsauer et al., 2020), where current attention systems are equivalent to a Hopfield Network with no update steps. Hopfield pooling is a special case of a Hopfield Network, where the query for the network is a single vector. Hopfield pooling provides a way to summarize k embeddings into a single fixed-length embedding in an iterative way. Additionally, the single query vector can be a learned embedding, and this embedding can be different for different tasks in a multi-task system, which is useful when the semantics for each task are different. For each Event Slot ES_i^j from the set of Event Slots ES_i defined for *Event Type*$_i$ (with $i = 1, ..., 5$ corresponding to the Event Types and their slots listed in Table 1), Hopfield pooling enables the MT-EsE.BHP to learn text-span embeddings. For example, when learning the text-span embedding for the span "*my sister*" while considering the Event Slot *gender*, the pooling will likely entirely focus on the contextual embedding for "*sister*", while when considering the *relation* Event Slot, the pooling will likely focus equally on both contextual embeddings for the words *"my"* and *"sister*, as they are both relevant to identify that the author of the tweet has a relationship with someone mentioned in that tweet text span.

Figure 3 illustrates the process of performing Hopfield pooling for an Event Slot ES_i^k of any *Event Type*$_i$, resulting in the text-span embedding for a tweet span. This process is based on the learning of an *embedding* for the Event Slot ES_i^k of *Event Type*$_i$, denoted as $emb(ES_i^k)$. As shown in Figure 3, $emb(ES_i^k)$ is randomly initialized. At each iteration, a vector multiplication between the contextual embeddings of the words from the tweet text span and the current Event Slot embedding $emb(ES_i^k(t))$ takes place. The vector product is producing unnormalized scores, which we normalize into $w_1^i, w_2^i, \ldots, w_k^i$ with the Softmax operation. These scores are then multiplied by their respective contextualized embeddings $c_1, c_2, \ldots, c_k$ and point-wise added together to produce a new single embedding. If we have not yet performed the required number of Hopfield update iterations, then the new embedding becomes $emb(ES_i^k(t+1))$ and

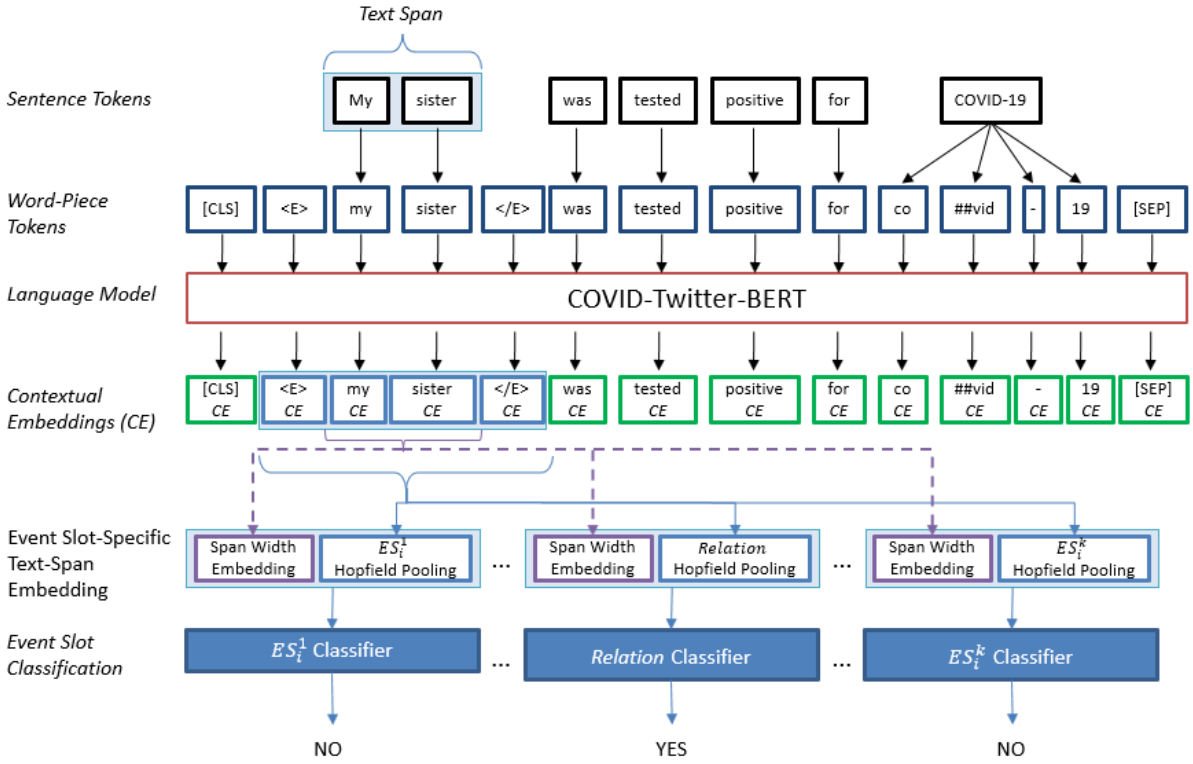

Figure 4: Example of Event Extraction for the *relation* Event Slot of the TESTED POSITIVE Event Type.

we repeat the whole process again. Otherwise, the update iterations are finished, which results in the final Event Slot-specific Hopfield span embedding.

3.3 A Detailed Example

Figure 4 provides a detailed description of how the MT-EsE.BHP operates on a sentence from a tweet categorized under the TESTED POSITIVE Event Type, when mapping the text span "*My sister*" into the *relation* Event Slot. First, the text span is surrounded by special tokens (<E> ... </E>). These tokens provide the language model contextual information as to where the text span is located in the sentence. Next, the sentence is word-piece tokenized (Devlin et al., 2019) and provided as an input to the COVID-Twitter-BERT language model. The language model generates a contextualized embedding for each word-piece token, which provides a representation of the word-piece token with respect to the whole sentence. Next, we perform Event Slot-specific Hopfield pooling for every Event Slot associated with the TESTED POSITIVE Event Type to produce the embedding of the text span, in the same manner as detailed in Section 3.2 and illustrated in Figure 3. As we have discussed Section 3.2, Hopfield pooling also learns an embedding for each Event Slot of TESTED POSITIVE. In Figure 4 the embedding of the Event Slot *relation* is illustrated, as well as embeddings learned for other Event Slots, e.g. ES_i^1 or ES_i^k. These embeddings are concatenating together a learned *width embedding* which represents the width of the text span that may be potentially mapped into the Event Slots. The width embedding provides information to each Event Slot classifier as to the width of the text span, since this information is lost when performing Hopfield pooling. Finally, a binary Event Slot Classifier for every Event Slot of TESTED POSITIVE decides whether the text span can be mapped in its corresponding Event Slot. Figure 4 shows the illustrated text span is mapped in the *relation* Event Slot.

4 Results

We trained five separate models, for each of the five Event Types, and they all share the same training schedule and settings: We use an initial learning rate of $2e^{-5}$, warmed up to from 0 over the first epoch and then decayed to 0 linearly for the 9 remaining epochs, with the ADAM optimizer (Kingma and Ba, 2015). We train with gradient accumulation, where we accumulate the gradients for a batch size of 32 with 4 batches of size 8. We initialize the span width embeddings with a nor-

Overall	Precision	Recall	F_1
Best	75.32	71.18	65.98
Median	68.14	56.30	62.76
Worst	53.77	42.77	51.14
MT-EsE.BHP	75.32	56.79	64.76
TESTED POSITIVE	**Precision**	**Recall**	**F_1**
Best	85.69	62.67	69.73
Median	78.32	57.54	67.67
Worst	44.32	44.71	44.52
MT-EsE.BHP	82.98	60.13	69.73
TESTED NEGATIVE	**Precision**	**Recall**	**F_1**
Best	71.07	71.94	70.30
Median	66.14	64.30	63.98
Worst	53.91	40.83	50.56
MT-EsE.BHP	71.07	61.87	66.15
CAN NOT TEST	**Precision**	**Recall**	**F_1**
Best	68.63	72.40	65.23
Median	64.13	52.11	55.79
Worst	46.46	43.51	45.19
MT-EsE.BHP	68.63	56.82	62.17
DEATH	**Precision**	**Recall**	**F_1**
Best	72.40	78.55	69.42
Median	57.42	64.36	61.16
Worst	49.17	52.15	52.12
MT-EsE.BHP	61.64	62.05	61.84
CURE	**Precision**	**Recall**	**F_1**
Best	84.05	78.43	62.05
Median	76.96	46.61	59.76
Worst	49.61	34.82	45.11
MT-EsE.BHP	83.52	45.30	58.74

Table 2: Overall results and results from each Event Type for the MT-EsE.BHP in the official shared-task evaluation.

mal distribution, an embedding size of 25, and 100 embeddings for widths $0-99$, where spans above width 99 are represented by the width 99 embedding. We apply a dropout rate of 0.10 to the output of the Slot-specific Hopfield span embedding, and we found best development performance when the number of update steps for Hopfield pooling was 2 with only 1 attention head. We split the provided collection of tweets into train (60%), dev (15%), and test (25%) sets for model development and assessment. We select threshold values for each binary slot classifier based on maximum F_1-score[1] performance on the dev set. We selected this training schedule and these hyper-parameters based on initial experiments of our system on the TESTED POSITIVE collection of tweets. We discuss the impact of this decision further in Section 5.

[1]F_1-score is defined as $F_1 = 2 \times Precision \times Recall/(Precision + Recall)$

Performance on the shared-task was determined based on micro F_1 score for the mapping of text spans into Event Slots, computed for each Event Type and overall, on an unseen evaluation set of 2500 tweets with 500 tweets per Event Type. Participating teams were provided the evaluation tweets 5 days in advance of the run submission deadline, and only one submission was allowed per team. Our team name and single submission were titled "*HLTRI*" and utilized the MT-EsE.BHP described in this paper. Our system produced an overall F_1 score of 64.76% for mapping Event Slots across all Event Types, which accounts for the 2nd-best overall results in the official shared-task evaluation. Our system obtained the best results for mapping tweets text spans in the Event Slots of the TESTED POSITIVE Event Type, and 3rd-best for the TESTED NEGATIVE Event Type, 2nd-best for the CAN NOT TEST Event Type, 3rd-best for the DEATH Event Type, and 4th for CURE Event Type. Overall the precision, recall, and F_1 scores for the mapping of text spans into Event Slots in the official shared-task evaluation are provided in Table 2. In the Table, we also show the best, median, and worst precision, recall, and F_1 scores across all team submissions. We believe that our system performed best on the mapping into Event Slots for the TESTED POSITIVE Event Type, because in the training data, we were provided with the most annotated tweets for this Event Type. We also performed well on mapping into the Event Slots of the CAN NOT TEST Event Type, out-performing the median F_1 score by 6.38%. Our system was competitive on all but the CURE Event Type, scoring above the median F_1 scores and coming close to the best F_1 scores on all other Event Types.

Detailed results are provided in Table 3 for each mapping of a text span into an Event Slot when using the official shared-task evaluation script. As seen, with the exclusion of the CURE Event Type, we largely score well on mapping into Event Slots for which more training examples were provided, and even score well for mapping into Event Slots for which much less training data was available, due to our multi-task learning framework setting for learning how to map jointly text spans into Event Slots. We explain our poor performance for mapping into the Event Slots pertaining to the CURE Event Type because our system obtained poor recall across all three Event Slots, but particularly in the *who_cure* Event Slot.

TESTED POS.	Precision	Recall	F_1	#
age	66.67	40.00	50.00	5
close_contact	71.43	32.79	44.94	61
employer	75.93	33.88	46.86	121
gender	91.23	51.49	65.82	101
name	86.93	81.60	84.18	375
recent_travel	62.50	18.52	28.57	27
relation	50.00	55.00	52.38	20
when	60.00	40.91	48.65	22
where	84.03	56.82	67.80	176
TESTED NEG.	**Precision**	**Recall**	F_1	#
age	100.00	55.56	71.43	9
close_contact	20.00	14.81	17.02	27
gender	75.53	60.17	66.98	118
name	76.01	75.18	75.60	274
relation	75.00	51.92	61.36	52
when	40.00	29.63	34.04	27
where	60.53	46.94	52.87	49
CAN NOT TEST	**Precision**	**Recall**	F_1	#
relation	90.62	46.77	61.70	62
symptoms	71.79	60.87	65.88	46
name	65.38	66.67	66.02	153
when	66.67	11.76	20.00	17
where	56.00	46.67	50.91	30
DEATH	**Precision**	**Recall**	F_1	#
age	83.33	90.91	86.96	33
name	57.93	60.43	59.15	139
relation	100.00	30.30	46.51	33
when	57.14	72.73	64.00	33
where	55.56	61.54	58.39	65
CURE	**Precision**	**Recall**	F_1	#
opinion	83.70	50.66	63.11	152
what_cure	84.85	53.44	65.57	262
who_cure	81.05	32.77	46.67	235

Table 3: Detailed results produced by the MT-EsE.BHP.

5 Discussion

The event organizers released the annotated evaluation tweets at the same time they released evaluation results, therefore we were able to analyze Event Slot-specific performance and perform a qualitative error analysis on specific tweets and Event Slots which were incorrectly classified.

We found that our system performed best on mapping text spans into Event Slots for the largest Event Type: TESTED POSITIVE, and within that Event Type also performed better on average on Event Slots which had more examples in the training data. While this is partially due to the fact that the TESTED POSITIVE Event Type has the largest collection of annotated tweets, approximately double that of all other Event Types, this is also likely due to the fact that most of the training hyper-parameters, listed in Section 4, were selected based on initial experimental performance on the training TESTED POSITIVE collection. This hyper-parameter selection decision likely biased our overall architecture towards improved performance on mapping of a text span into an Event Slot for the TESTED POSITIVE Event Type at the cost of reducing performance in other Event Types. This hypothesis is supported by our shared-task evaluation ranking, where we came in 1st in TESTED POSITIVE but placed lower in all other Event Types.

Table 3 provides detailed performance for each Event Slot along with the number of examples in the evaluation dataset. We see that our multi-task learning framework maintains performance for some Event Slots that were provided with a small set of examples, such as *age*, while other for Event Slots, provided also with a small set of examples, such as *when*, there is clearly room for improvement. We believe that sharing a learned language model informed by COVID-Twitter-BERT contributes to this baseline level of performance, while the slot-specific span representation using Hopfield pooling improves performance upwards when performing mapping into Event Slots that were provided with a larger set if examples. Hopfield pooling provides the system a mechanism by which to learn how to best merge the shared contextual representation produced by COVID-Twitter-BERT into a slot-specific representation useful for slot-specific classification. We also see that mapping text spans into Event Slots for the CURE Event Type had high precision, but very poor recall even with a relatively large number of provided examples, leading to our poor F_1 scores for this Event Type. More specifically, we see that the recall of the *who_cure* slot is extremely poor, therefore we investigate this further in our error analysis.

Table 4 lists tweets in which our system was incorrect. Example 1 demonstrates the failure of the MT-EsE.BHP to identify that "*Vice President Mike Pence*" is an *employer* who's staff member tested positive for coronavirus. This is likely due to the fact that, in the collection of tweets, it is atypical to list a single person as an employer as opposed to the organization they employ the employee through. More tweets which follow this pattern would likely be necessary to learn this use of *employer*. A typical mistake made by the MT-EsE.BHP on the *close_contact* slot is visible in Example 2, where the system fails to recognize the implication that "*the mother*" was not a close

Event Type	Tweet with Text Span	MT-EsE.BHP	Annotation
TESTED POS.	"Staffer for [*Vice President Mike Pence*] tests positive for coronavirus <url>"	×	*employer*
TESTED NEG.	"Reports coming out that the infant child of [*the mother*] who died of Covid-19 tested negative for Covid-19. The Sars-COV-2 is mysterious."	*close_contact*	×
CAN NOT TEST	"[@*LouisianaGov*] @LADeptHealth Still can't get tested though."	×	*where*
DEATH	"@FALLOFTHECABAL [*I*] personally know 3 1ST respknders and 2 nurses who have died from complications of COVID-19. i have no horse in this race so to speak."	×	*relation*
CURE	"At @WhiteHouse briefing today, a so-called reporter said that [*Biden*] recommends flying our flag at half-staff. Well,, That sure must be the cure for #COVID19. My suggestion: Wait till AFTER the PANDEMIC is OVER, then, in honor of those who died, fly the mast at half-staff."	*who_cure*	×
CURE	"@DonaldJTrumpJr [*You*] advocate drinking bleach to cure covid-19. We are all aware of how much you understand disease."	×	*who_cure*

Table 4: Tweets and text spans where the MT-EsE.BHP incorrectly classifies an Event Slot.

contact, since she had COVID-19 and died from it, but the "*infant child*" was a close contact due to the implied closeness of a mother to her infant child. Example 3 shows the difficulty of noticing that an author tweeting "@*LouisianaGov*" due to their inability to get tested likely lives in the location of Louisiana. Typos are prevalent in user-generated text, and Example 4 demonstrates a typo with "*1st respknders*" which likely causes the language model to miss the contextual clues that the author of the tweet has some *relation* with first responders and nurses who have contracted and died from complications due to COVID-19. Example 4 and 5 demonstrate one of the primary reasons we believe our system performs poorly on the CURE Event Type, and specifically the *who_cure* Event Slot. We see sarcasm demonstrated in Example 4, where the author of the tweet is sarcastically stating that Biden's recommendation of flying a flag at half-staff is an actual potential cure for COVID-19. Tay et al. (2018) discuss the "sophisticated speech act" of sarcasm in the context of social communities such as Twitter and Reddit, and they note that sarcasm can severely disrupt opinion mining systems. Sarcasm can be very difficult to identify (Joshi et al., 2017), and many of the tweets discussing COVID-19 cures contain sarcastic content which can be difficult to distinguish. Our system mistakenly identified Biden as *who_cure*, while the annotator picked up on the sarcasm and did not annotate this instance. Example 5 demonstrates a debatable instance of sarcasm on the other side, where the annotation states that "*You*", being Donald Trump Jr., advocates for drinking bleach as a cure for COVID-19. Our system does not identify that Donald Trump Jr. advocates for the cure of drinking bleach, but the annotator agrees that, in this instance, the author of the tweet is legitimately making the claim that Donald Trump Jr. advocates for this cure.

6 Conclusion

In this paper we described the Multi-Task Event-specific Extraction system using BERT and Hopfield Pooling (MT-EsE.BHP) that was developed for the W-NUT 2020 Shared Task 3. Our system learned how to take advantage of contextual embeddings such that embeddings for text spans can be learned, while also learning embeddings for each Event Slot of each Event Type. This was made possible by using Hopfield Pooling. The text span embeddings informed binary classifiers (one for each Event Slot) that decided whether tweet text spans can be mapped into an Event Slot or not. Separate such binary classifiers were trained for each Event Type. The results that we obtained are promising. These results could be further used to learn how to associate Event Slots for each mention of an Event Type in tweets, instead of producing only a bag of filled Event Slots. This would be a requirement for knowledge extraction from COVID-relevant tweets useful for Public Health applications.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *European Conference on Artificial Intelligence*.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5).

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tim Mackey, Vidya Purushothaman, Jiawei Li, Neal Shah, Matthew Nali, Cortni Bardier, Bryan Liang, Mingxiang Cai, and Raphael Cuomo. 2020. Machine learning to detect self-reporting of symptoms, testing access, and recovery associated with covid-19 on twitter: Retrospective big data infoveillance study. *JMIR Public Health Surveill*, 6(2):e19509.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *https://arxiv.org/abs/2005.07503*.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2020. Hopfield networks is all you need. *https://arxiv.org/abs/2006.02567*.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia. Association for Computational Linguistics.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. pages 1445–1456.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Yipeng Zhang, Hanjia Lyu, Yubao Liu, Xiyang Zhang, Yu Wang, and Jiebo Luo. 2020. Monitoring depression trend on twitter during the covid-19 pandemic. *https://arxiv.org/abs/2007.00228*.

Shi Zong, Ashutosh Baheti, Wei Xu, and Alan Ritter. 2020. Extracting covid-19 events from twitter.