



# RNN Transducers for Named Entity Recognition with constraints on alignment for understanding medical conversations

Hagen Soltau, Izhak Shafran, Mingqiu Wang & Laurent El Shafey

Google Brain

soltau, izhak, mingqiuwang, shafey@google.com

## Abstract

Understanding medical conversations requires detecting entities such as Medications, Symptoms, Treatment, Conditions and Diagnosis, which leads to large ontologies with overlapping spans. Moreover, for ease of adoption by the clinicians, the inference also needs to locate the position of the entities in the conversations. Popular solutions to Named Entity Recognition (NER) such as conditional random fields, sequence-to-sequence models, or the question-answering framework are not suitable for this task. We address this problem by proposing a new model for NER task – an RNN transducer (RNN-T), which has hitherto been used only in speech recognition. These models are trained using paired input and output sequences without explicitly specifying the alignment between them, similar to other seq-to-seq models. RNN-T models learn the alignment using a loss function that sums over all alignments. In NER tasks, however, the alignment between words and target labels are available from the human annotations. We propose a fixed alignment RNN-T model that utilizes the given alignment, while preserving the benefits of RNN-Ts such as modeling output dependencies. As a more general case, we also propose a constrained alignment model where users can specify a relaxation of the given input alignment and the model will learn an alignment within the given constraints. In other words, we propose a family of seq-to-seq models which can leverage alignments between input and target sequences when available. Through empirical experiments on a challenging real-world medical NER task with multiple nested ontologies, we demonstrate that our fixed alignment model outperforms the standard RNN-T model, improving F1-score from 0.70 to 0.74.

## 1. Introduction

Extracting relevant information from medical conversations [1, 2, 3] is important for clinical documentation and automating it will reduce the burden on medical providers. Medical conversations contain entities represented by multiple ontologies. Ideally, these ontologies should be modeled jointly, since the entities are related across ontologies. For example, the mention of the symptom *pain* in a conversation increases the likelihood of the mention of pain medications in the same conversation. Modeling these ontologies jointly leads to large label sets and the spans of the entities from different ontology often overlap. For example, in the sentence *so you'll get the metformin*, the term *metformin* might be labeled not only as medication but also as an alleviating factor. Furthermore, the inference needs to locate the span in the conversation for transparency and easier clinical adoption.

These challenges make the task of named entity recognition (NER) [4] considerably more difficult than conventional public data sets such as OntoNotes or ACE and conventional NER models are not well suited for predicting large overlapping ontologies with accurate text spans.

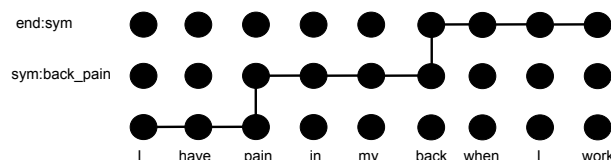


Figure 1: A RNN-T trellis illustrating the alignment of input *pain* in *my back to the label* *sym: back\_pain* where end-markers indicates the end of a span. Horizontal lines denote time steps in the input sequence (with blank output symbols) and vertical lines indicate generating the next label in the target sequence.

In this paper, we propose a novel approach for NER that handles nested and overlapping spans. Specifically, the key component of our NER system is an RNN transducer (RNN-T) [5], a model that was originally proposed for speech recognition. While RNN-T models for speech recognition have to learn the alignment between input and target sequences, we propose a key modification for the NER task by fixing the alignment with the given human annotations. This modification improves training on long sequences with limited supervised data. More generally, we provide a method to relax the alignment between the two extremes of fixed and unconstrained alignment.

In summary, the contributions of our paper are:

- We introduce RNN transducers for NER tasks and show that they are well suited for predicting position and label for nested and overlapping entities with large ontologies.
- We introduce fixed and constrained alignment losses that better utilize human annotations and improve F1 score from 0.70 to 0.74. The fixed alignment loss provides better data efficiency and generalization to long sequences.
- While many public tasks utilize short and well-segmented inputs, we report experiments on medical conversations, a difficult real-world task to highlight the challenges associated with long sequences.

## 2. Related Work

Linear-chain conditional random fields (CRF) has proved to be a reliable workhorse for NER. In CRFs, each input token is mapped to an output label and the output dependencies are modeled via a transition matrix. However, the CRFs are not designed to generate multiple targets for a given input token and, hence, are not well-suited for tasks with nested spans and multiple labels associated with them.

A simple approach involves modifying the label set and using standard CRFs. The new label set is created by taking the cross-product of all possible labels that can be tagged on a single input token. This does not scale well due to the explosion in the label space and the resulting data sparsity. The data sparsity is particularly acute in NER tasks since the typical NER corpora are

not very large, resulting in substantial performance degradation (e.g., [1, 6]).

Another approach splits the task as hierarchical inference where spans relevant to all the labels are detected first and then the spans are classified into labels. For example, this hierarchical approach was successfully demonstrated in labeling symptoms in medical conversations [1, 7].

The recent interest in large models has motivated researchers to cast translation, summarization and other NLP problems as question answering (QA) tasks [8]. Framing NER task as QA task is not straightforward and adds more complexity [9]. The query prompts are natural language definitions of the labels, often borrowed from the annotation guidelines. All the entities in the input for a given label are inferred using a two-stage process – detecting start/end positions of the spans using binary classifiers for each input position and then classifying the spans into valid/invalid ones. This approach does not scale well for large label spans.

Another approach is to cast NER as a seq-to-seq task, mapping a sequence of words to a sequence of labels (e.g., [10]). A drawback of this method is that the position of the inferred label with respect to the input sequence is not predicted. This is a consequence of using a soft alignment between encoders and decoders. Without the position information, the model is only solving the NER task partially and is not useful in domains such as medical where explainability is important.

Taken together, these models are not well-suited for NER tasks with nested spans and large label sets, except for seq-to-seq models. Conventional seq-to-seq models also do not solve the NER task, since they do not locate the position of the span in the input.

### 3. RNN Transducers for Named Entities

RNN Transducer (RNN-T) is a sequence transduction model that was designed for speech recognition tasks [5, 11]. Though the model consists of encoder (transcription) and decoder (prediction) networks, it differs from seq-to-seq in how models learn alignments between the input and the output sequences. For RNN-T, this is achieved by relying on a cost function that sums over all possible alignments, computed via forward-backward (or Baum-Welch) algorithm, similar to Hidden Markov Models.

As shown in Figure 1, we use RNN-T to map a sequence of words to a sequence of NER labels. The NER labels include an end marker that allows us to recover the span of words that are associated with the predicted label. Since RNN-T models can emit multiple output labels at any given input position, this allows the model to handle nested and overlapping NER entities as we can associate the same word with multiple NER labels.

A common problem with traditional NER models is that they do not scale well to large label sets. But since RNN-T is a *locally* normalized model, we can apply this model to very large label sets and we demonstrate the effectiveness in our experiments. The ability to scale allows us also to *combine* multiple (related) ontologies into one model. This reduces data sparsity and allows for better modeling of dependencies, e.g. a prediction of a symptom *pain* can influence the prediction of a pain medication.

#### 3.1. Fixed Alignment Loss

As in other seq-to-seq models, RNN-T models are typically trained using paired input and output sequences without explicitly specifying the alignment between them. The model learns

the alignment using a loss function that sums over all alignments, which makes them attractive for ASR where word-level alignments are unavailable.

During training, the loss of a regular RNN-T model is computed by marginalizing over all valid alignments between input  $x$  and output  $y$  sequences and can be computed efficiently by forward-backward algorithm on accelerators as matrix products [12, 13].

$$L_{\text{regular}} = P(y | x) = \sum_A P(A | x) \quad (1)$$

In NER tasks, however, the alignment between words and target labels are available from the human annotations. There is no need for the model to learn the alignment and we can simplify the task by modifying the loss to a *fixed-alignment loss*.

$$L_{\text{fixed}} = P(y | x) = P(A | x) \quad (2)$$

We explicitly encode the alignment information in the input and compute the loss with respect to the *given* alignment only. This is in particular useful when training data is sparse which is often the case for NER tasks. As we demonstrate in experiments, this improves learning when training with long sequences. Being able to model long sequences in turn allows us to increase the contextual information available to the model and improve accuracy on the NER tasks.

#### 3.2. Constrained Alignment Loss

Filling the gap between the two extremes of the original *unconstrained alignment* loss and the new *fixed-alignment* loss, we propose a *constrained alignment* loss. The key idea is to allow a user-defined relaxation of the given training alignment. This could be very useful when human annotations are noisy and model can learn perturbations of the given alignment with better likelihood.

We accomplish this by manipulating the label ( $y$ ) and *blank* ( $b$ ) matrices of RNN-T models which are employed in the forward-backward algorithm to compute the sum over alignments [12, 13]. Specifically, they correspond to labels ( $\log P(y_u | x_t)$ ) and blanks ( $\log P(b_u | x_t)$ ) defined over input and output time steps. Intuitively, the  $(t, u)$  entries in the two matrices represent the associated with the horizontal and vertical transitions in the the alignment trellis in Figure 1. The *fixed-alignment* loss can be viewed as an instance where all the entries of the matrices are set to negative infinity except for those corresponding to the given alignment.

The *constrained alignment* loss is parameterized by user-specified relaxations in  $(t, u)$  dimensions of the alignment, from which we create a 2D unit convolution filter ( $r$ ).

$$L_{\text{constr.}} = P(y | x) = \sum_{A \in C(\text{delta})} P(A | x) \quad (3)$$

The given training alignment is represented as two Boolean matrices (masks) corresponding to the alignment in the  $y$  and  $b$  matrices. The masks are then expanded (relaxed) to a set of valid alignments  $C(\text{delta})$  using a 2D convolution with a unit rectangular filter, whose dimensions are  $1 + 2 * \text{delta}(t)$  and  $1 + 2 * \text{delta}(u)$ . Thus, the degree of relaxation from fixed alignment is controlled by  $\text{delta}(t)$  and  $\text{delta}(u)$  and we use the same  $\text{delta}$  for both dimensions in our experiments. The expanded Boolean masks are then intersected with the  $y$  and  $b$  matrices and all the entries not masked are set to negative infinity.

Once the alignment constraints are incorporated into the  $y$  and  $b$  matrices, we compute the forward-backward algorithm to obtain the *constrained alignment loss*.

## 4. Experimental Setup

We report experimental results on a real-world NER task on medical conversations [14] that contains overlapping and nested entities with multiple ontologies and large label sets. The corpus contains about 100K unlabelled conversations between medical providers and patients. Of these about 6k conversations were labeled by professional scribes and correspond to 5 ontologies – medications, symptoms, conditions, diagnoses and treatments.

As evaluation metric we use the common *F1 score* that measures the correctness of the predicted labels and the location of input text over which the labels are predicted. Note, that is in contrast to some seq-to-seq approaches, where the F1 score is relaxed to not account for position errors. In our case, we use the regular *F1 score* where the predicted label and text span has to be correct.

Ontology	training examples
Medications	100,000
Symptoms	64,000
Conditions	42,000
Diagnoses	38,000
Treatments	6,000

Table 1: *Corpus Ontologies*

### 4.1. Pre-Processing

The average sequence length of our conversations is about 1600 words, with a long tail of very long conversations. While it is quite common to break conversations by sentence end markers, this approach does not work well in our use case for the following reasons. First, the model is intended to be used with audio as input modality and the current ASR models do not predict sentence end markers sufficiently well [15]. Second, our data is conversational in nature, and many sentences are very short (e.g. answering questions, yes/no, etc). A single sentence will often not provide sufficient context to predict medical entities and choosing a fixed window of sentences might break the context at the wrong boundaries.

Instead, we feed entire conversations to our model at inference time. This avoids breaking context at fixed intervals. However, given the long tail of long conversations and limited memory on accelerator devices, during training time we have to break conversations. As we do not have good break points, we randomly cut segments from conversations. The segment lengths of the training examples are between 280 to 320 words. This randomized cutting is performed when each example is loaded during training. As the training progresses, the model essentially sees different views of the data; this can also be seen as a form of data augmentation. During the test time, the full conversations are fed as input sequence.

### 4.2. Pre-Training

The text encoder is a Transformer-XL [16] architecture with 15 layers and has in total 473m parameters. The pre-training corpus consists of 100k unlabeled medical conversations with 177m words. We apply the same pre-processing for pre-training and training and randomly cut segments of 280 to 320 words.

For pre-training of the encoder, we mask input tokens similar to the approach in BERT [17], but the model needs to learn not

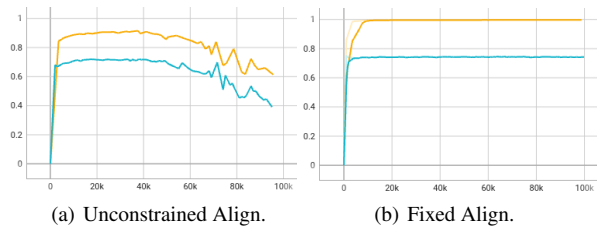


Figure 2: *F1 score for train (orange) and test (blue) data, measured over 100k training steps. Training with unconstrained alignment leads to unstable NER performance, even though Log Likelihood converges well in training.*

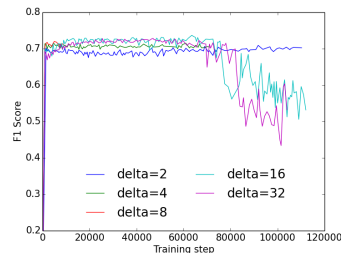


Figure 3: *Constrained alignment loss: when the constraints are relaxed, the performance ranges from that of fixed alignment loss to unconstrained alignment loss.*

only to recover the masked inputs, but also the entire sequence including the unmasked tokens. This is done to encourage the encoder to learn words as well as context representations. The masking operation is done at word level before tokenization, so that entire words are masked out. This means also the input and target sequences are not of the same length, and we use a RNN-T model for pre-training.

## 5. Experiments

### 5.1. Fixed alignment loss

The advantages of fixed alignment loss can be seen in Figure 2 comparing side-by-side fixed and unconstrained alignment models, both of which used the same inference setup with regular RNN-T beam search. The overall F1 scores improve from 0.702 (Precision=0.705, Recall=0.66) with unconstrained alignment loss to 0.743 (Precision=0.789, Recall=0.702) with fixed alignment loss.

Figure 2 shows also that performance of using an unconstrained loss is also worse for *training* (orange) data, i.e. it is not an issue of over-training. It is rather that the model does not learn the named entity recognition task. It is also not an optimization issue, the negative log-Likelihood loss on the training data is minimized and approaching 0 for both unconstrained and fixed alignment loss.

While the log-Likelihood of the training data is getting optimized, the model has difficulty learning the task and the learning is unstable as evidenced by the F1 scores on the training data. We have observed that the posterior distribution becomes sharper and sharper during training, minimizing the negative likelihood, while not learning to predict the class labels. With unconstrained alignments, the model does not learn to predict entities well while still minimizing the loss function.

For studying the impact of relaxing the constraints on the

given training alignment, we trained models with different alignment tolerances in  $(t, u)$  axes. We specified equal tolerance on both axes with values of 2, 4, 8, 16 and 32. As evident from the Figure 3, when the permitted alignments are close to the given training alignment, the model performs similar to fixed alignment loss. When the alignments are allowed to stray from the given training alignment, the performance resembles that of unconstrained alignment. This demonstrates how constraints in the alignment loss can be effectively used to steer the behavior of the learned model.

## 5.2. Semi-Supervised Learning

Named Entity recognition models are often trained with limited amounts of labeled data since manually labeling spans of input text is labor intensive. In our case, only 5k of the 100k conversations are labeled. In such settings, semi-supervised learning often provides additional gains when the initial model can produce reasonably accurate labels. For evaluating gains from semi-supervised learning, we generate pseudo-labels on the remaining 95k unlabeled conversations and fold them back into the training data.

Table 2 summarizes the results from adding semi-supervised training data. The additional data benefits models with unconstrained alignment loss more than the fixed alignment loss. If enough data is available, models can learn alignments without specific human annotations, potentially saving annotation costs. When the data is limited and the model has difficulty learning the alignment all by itself, utilizing the given alignment through fixed-alignment loss is very beneficial. The fixed-alignment loss is so effective that the additional data only brings marginal benefits and the benefits are observed largely for ontologies that have limited training examples.

Loss	5k labeled	+ 100k unlabeled
Unconstrained	0.702	0.747
Fixed	0.743	0.762

Table 2: Adding unsupervised training data gives substantial gains for the unconstrained alignment case, where the added training data helps the model to learn aligning words to labels.

## 5.3. Training on Short vs Long Sequences

The difficulty of learning alignments increases with the length of sequences. This is true for modern sequence transduction models (RNN-T, seq2seq), as well as traditional HMM models. In our experimental setting, the raw data are unsegmented conversations and we randomly cut training examples from these conversations during training infeed. To alleviate the training issues with Forward/Backward training (as shown in Figure 2), we could reduce the segment lengths during model training.

However, during inference the test data is unsegmented and long. While training on short sequences helps with learning alignments, the models are unable to generalize to long sequences. We demonstrate this by training two models, with 40-60 and 280-320 word segments respectively, both with fixed-alignment loss. Their performance, as shown in Figure 4, clearly demonstrates poor generalization of the model trained with shorter segments.

To verify that this is not a training issue and rather a mismatch between train and test conditions, we generated a segmented version of the test data by applying the same random segment cutting procedure that we apply in training. Each conversation gets segmented multiple times and we average the results of the F1 scores. While this is clearly not a suitable

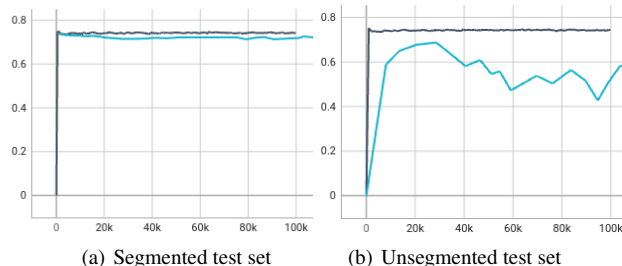


Figure 4: F1 score over 100k training steps with fixed alignment model trained on short (50 words, light blue) vs long segments (300 words, dark blue). While the model trained on short segments performs well on segmented test data, it fails to generalize to a more realistic setting with unsegmented tests data (right plot, light blue curve). In contrast, the model trained on longer sequences generalizes well.

approach when deploying models, it helps us to verify that is indeed a generalization issue where models do not learn to generalize to longer sequences. The results on the segmented test data can be seen in the left plot in Figure 4. Both models (train on short and long sequences) perform well on segmented data, while only the model trained on long sequences performs well on unsegmented conversations.

## 6. Discussion

We demonstrated that RNN-T works well for nested and overlapping named entity recognition and showed that the approach scales for large ontologies. The model is simple and elegant. Compared to Hierarchical CRF or reading comprehension models, only one model and one loss function is needed, making optimization much easier.

We introduced a fixed alignment loss that better utilizes human annotations and improves F1-score from 0.70 to 0.74. We generalized this notion to a constrained alignment loss where users can vary the degree of reliance on the position information from the training data. This is useful in scenarios where the training data is less carefully labeled or ontologies are ambiguous.

We highlight the importance of reporting experiments on long sequences and demonstrate the sensitivity of the typical RNN-T models to long sequences. Fixed alignment loss makes the learning process much easier and allows the model to generalize to long sequences with higher accuracy.

We demonstrated that unsupervised training improves the performance from 0.74 to 0.76, even when the model utilizes a large pre-trained encoder. We also showed that the use of unlabeled data reduces the gap with the unconstrained alignment.

While conventional seq-to-seq models do not completely solve NER tasks, a variant with hard attention was proposed in [6] that alleviates the issues and it turns out that their work essentially converts a seq-to-seq model to RNN-T to a very large degree. In their work, the targets are not only the labels, but also include a special *ew* (end-of-word) token for every input token. The model uses *ew* tokens to advance the decoder and predict the label for the input word under consideration, thus uses a hard-alignment between encoders and decoders. With these two modifications, their model is essentially equivalent to RNN-T, where the *ew* tokens serves the same role as *blank*.

Lastly, it is worth noting that the RNN-T architecture can be employed for both ASR and NER, reducing the engineering overhead of deploying a complete spoken NLU system.

## 7. References

- [1] N. Du, K. Chen, A. Kannan, L. Tran, Y. Chen, and I. Shafran, "Extracting symptoms and their status from clinical conversations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 915–925. [Online]. Available: <https://aclanthology.org/P19-1087>
- [2] N. V. Meripo and S. Konam, "Extracting appointment spans from medical conversations," in *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*. Online: Association for Computational Linguistics, Jun. 2021.
- [3] B. Hao, Y. Liu, and J. H. Hao, "Enhanced medical dialogue diagnosis with intra-inter window attention encoder," in *2021 International Conference on Intelligent Computing, Automation and Applications (ICAA)*, 2021.
- [4] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999. [Online]. Available: <http://nlp.stanford.edu/fsnlp/>
- [5] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, 2012.
- [6] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5326–5331. [Online]. Available: <https://aclanthology.org/P19-1527>
- [7] N. Du, M. Wang, L. Tran, G. Lee, and I. Shafran, "Learning to infer entities, properties and their relations from clinical conversations," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4979–4990. [Online]. Available: <https://www.aclweb.org/anthology/D19-1503>
- [8] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," 2018.
- [9] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5849–5859. [Online]. Available: <https://www.aclweb.org/anthology/2020.acl-main.519>
- [10] A. Kannan, K. Chen, D. Jaunzeikare, and A. Rajkomar, "Semi-supervised learning for information extraction from dialogue," in *Proceedings of Interspeech*, 2018, pp. 2077–2081.
- [11] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6645–6649.
- [12] K. C. Sim, A. Narayanan, T. Bagby, T. N. Sainath, and M. Bacchiani, "Improving the efficiency of forward-backward algorithm using batched computation in tensorflow," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2017, pp. 258–264.
- [13] T. Bagby and K. Rao, "Efficient implementation of recurrent neural network transducer in tensorflow," in *Proceedings of the 2018 IEEE Spoken Language Technology Workshop*. IEEE, 2018.
- [14] I. Shafran, N. Du, L. Tran, A. Perry, L. Keyes, M. Knichel, A. Domin, L. Huang, Y.-h. Chen, G. Li, M. Wang, L. El Shafey, H. Soltau, and J. S. Paul, "The medical scribe: Corpus development and model performance analyses," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 2036–2044. [Online]. Available: <https://www.aclweb.org/anthology/2020.lrec-1.250>
- [15] H. Soltau, M. Wang, I. Shafran, and L. E. Shafey, "Understanding Medical Conversations: Rich Transcription, Confidence Scores & Information Extraction," in *Proceedings of Interspeech*, 2021, pp. 4418–4422.
- [16] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. [Online]. Available: <https://aclanthology.org/P19-1285>
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423>