

Improving automatic event understanding through sequential and non-sequential deep learning architectures

Timothy Hao Huang, Jay Zern Ng, Yuki Nishimura,
Jonathan Irvin Santoso, Kevin Christian Wibisono

Supervised by Yuval Marton (Bloomberg),
Asad Sayeed (University of Gothenburg),
Smaranda Muresan (Columbia University)

January 22, 2021

☰ Menu Q Search

Bloomberg

Sign In Subscribe

Deals

Facebook Buys Customer-Service Software Maker Kustomer

- With the rapid increase in the amount of text data available in the world, there has been a growing need to develop tools for *automatic understanding* of events.
- Currently, machines are still not intelligent enough to comprehensively understand events, e.g. to inform financial decisions based on news article headlines.



- Machines can understand events by learning their representations, which are composed by **semantic role-filler representations**.
 - Example: **Uncle Roger**_{AGENT} **makes**_{PREDICATE} **rice**_{PATIENT} with a **rice cooker**_{INSTRUMENT}.



- A desirable event representation is one that reflects *thematic fit*.
 - Given a verb v and an entity x , how well does v fit x in role r ?
 - Example: (eat, apple, PATIENT) is more fitting than (eat, apple, AGENT); (cut, knife, INSTRUMENT) is more fitting than (cut, bowl, INSTRUMENT).

Thematic fit rating data sets

| Verb | Noun | Semantic role | Score |
|---------|--------|---------------|-------|
| advise | doctor | subj | 6.8 |
| advise | doctor | obj | 4.0 |
| confuse | baby | subj | 3.7 |
| confuse | baby | obj | 6.0 |
| eat | lunch | subj | 1.1 |
| eat | lunch | obj | 6.9 |
| kill | lion | subj | 2.7 |
| kill | lion | obj | 4.9 |

Figure: Human scores range from 1 (unlikely) to 7 (most likely)

- Padó et al. (2007): 216 balanced agent/patient ratings.
- McRae (2005): 1,444 unbalanced agent/patient ratings.
- Infeasible to directly optimize thematic fit due to limited data size.
- Instead, train on (word, role) pairs to infer thematic fit.

Hong et al.'s (2018) model

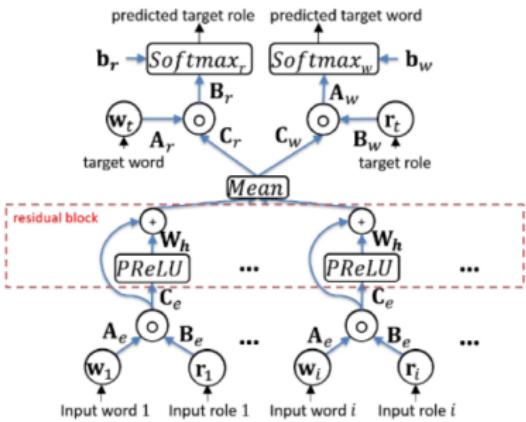
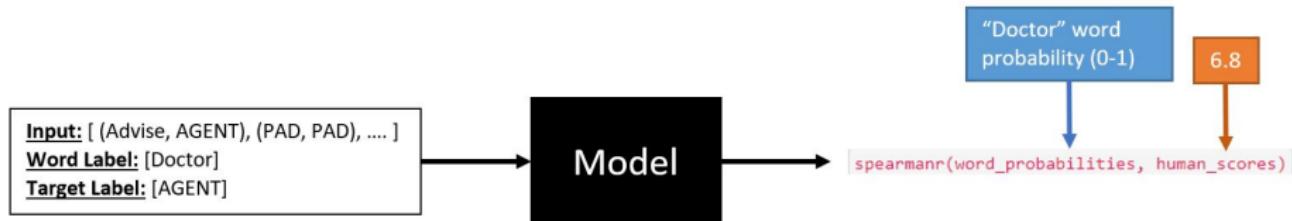


Figure: ResRoFA-MT Architecture (subsequently called Baseline)

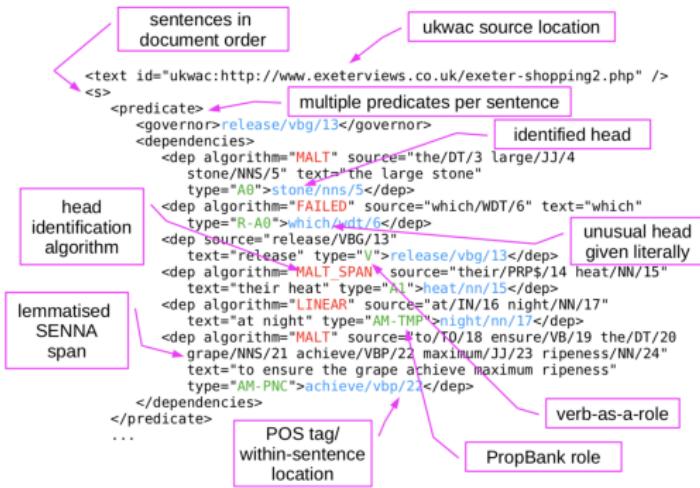
- **Baseline:** the current state-of-the-art thematic fit model.
 - Improved Tilk et al.'s (2016) model by adding:
 - 1 a secondary role prediction task
 - 2 residual blocks to prevent vanishing gradient
 - 3 parametric ReLU (PReLU) layers to introduce positional weightings
 - Naturally interested in (predicted target) role and word accuracies.

Measuring thematic fit



- Thematic fit is measured by the Spearman's correlation between human scores and word predictions.
- Example: (advise, doctor, AGENT, 6.8) from Padó et al. (2007).
 - Use (advise, AGENT) as input, along with padding tokens for other word/roles.
 - From the word prediction, obtain the estimated probability that the target word is *doctor*.

Training data

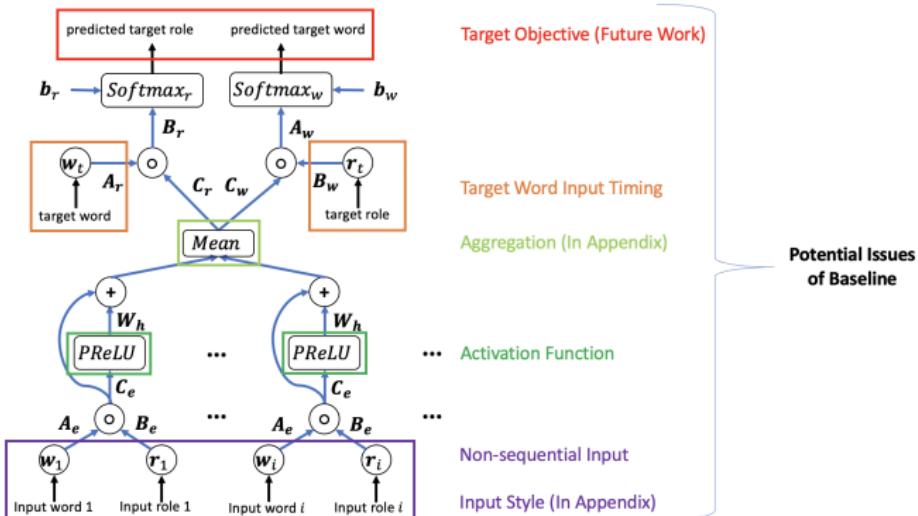


| Data version | Dependency parser | Semantic role labeler |
|--------------------------------------|------------------------------------|----------------------------------|
| v1 (Sayeed et al., 2018) | MaltParser (Nivre et al., 2006) | SENNΑ Collobert et al., 2011) |
| v2 (Marton and Sayeed, submitted) | spaCy (spacy.io) | He-SRL (He et al., 2018) |

RW-Eng corpus (Sayeed et al., 2018)

- A large corpus of automatically labeled semantic frames from:
 - 1 ukWaC (Feraresi et al., 2008)
 - 2 British National Corpus (BNC Consortium, 2007)

Potential issues of Baseline



- Focus on three potential issues of Baseline:
 - target word input timing
 - non-sequential input
 - activation function
- To provide an apple-to-apple comparison with Baseline, modifications are made in a part-by-part basis.

- To deal with the aforementioned issues, three new model variants are created:
 - 1 Target word input timing: **Beginning**
 - 2 Non-sequential input: **SeqAttn**
 - 3 Activation function: **BaselineLeaky**, **BaselineShared**
- Modifications to input word/role pairs aggregation and non-sequential input style can be found in the Appendix.
- Modifications to the target objective, such as adding/removing tasks are not considered this time, and are future directions of work.

1. Modifying target word input timing

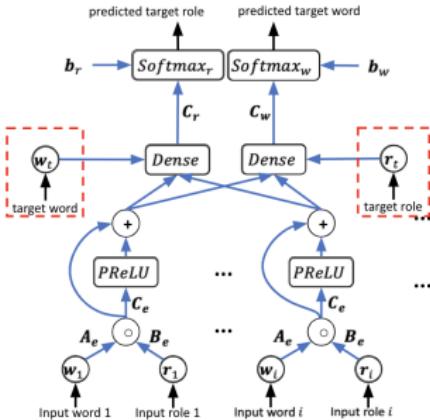


Figure: Beginning architecture

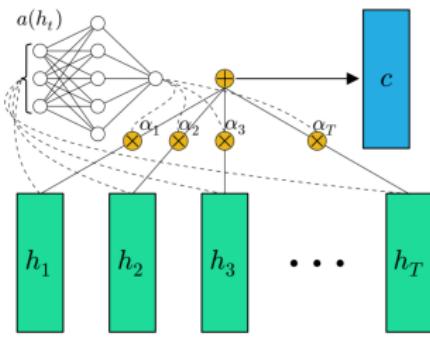
- **Beginning:** introduces the target word + role in dense layer.
 - Reduces tensor factorization from 2 to 1, but introduces two task-specific dense layers.
 - Potentially improves performance as event representation will have information about the target.

2. Modifying non-sequential input

- All previous models did not take word-role ordering into account.
- Experiments with sequence-based models to find out whether sequential information might help improve the quality of event representations.
 - This induces the need for modified input preprocessing and evaluation scripts which ensure correct ordering of the event participants.
- Based on experiment outcomes, models based on attention mechanism (particularly self-attention) perform better than those based on CNN, RNN, LSTM or bidirectional LSTM.

The attention mechanism

- The attention mechanism assigns weights to hidden states.
- Concretely, given a sequence of hidden states (h_1, h_2, \dots, h_T) and encoder contextual information c_t for $1 \leq t \leq T$, the attention score is computed as $\alpha = \text{softmax}(e_1, e_2, \dots, e_T)$, where $e_t = f(c_t, h_t)$ for $1 \leq t \leq T$.
- The attention score α is applied to the hidden states, producing a weighted representation of hidden states $\sum_{t=1}^T \alpha_t h_t$.
- In the absence of contextual information, it is usually assumed that $c_t = h_t$. This is called *self-attention*.



Applying the attention mechanism

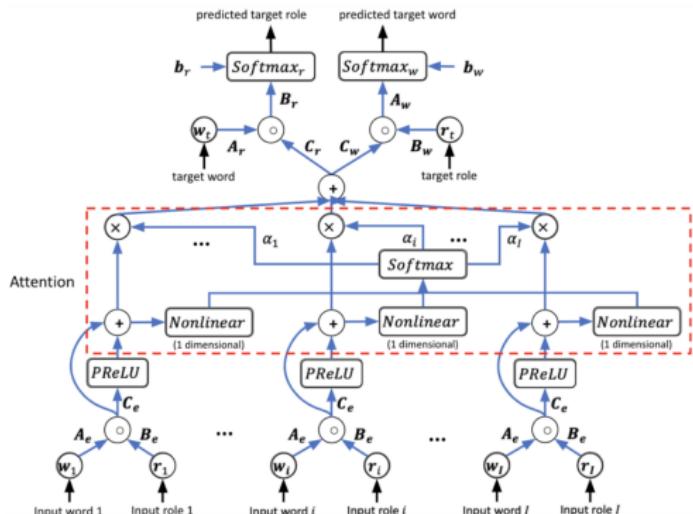
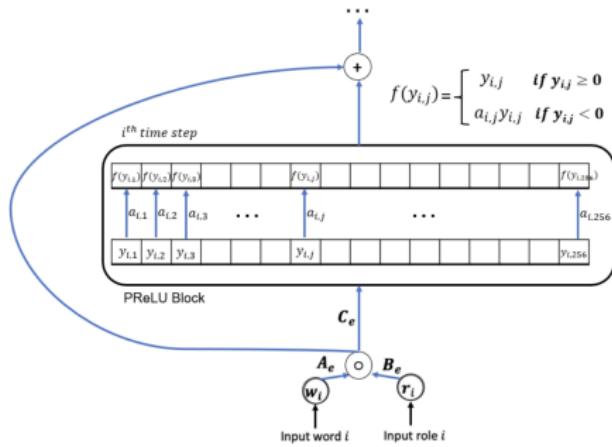


Figure: SeqAttn architecture

- One of the most common self-attention scoring mechanisms is Bahdanau's (2015) additive mechanism, which is utilized here.
 - $e_t = f(h_t) = \tanh(Wh_t + b)$ for $1 \leq t \leq T$.
 - Here, W and b are learnable parameters.

3. Modifying activation function



| Model | PReLU parameters | Number of trainable parameters |
|----------------|---------------------------|--------------------------------|
| Baseline | $a_{i,j} \in \mathbb{R}$ | $I \times J$ |
| BaselineLeaky | $a_{i,j} = 0.3$ | 0 |
| BaselineShared | $a_{i,j} = a_{k,j} = a_j$ | J |

- $I = 6$ represents the number of input pairs.
- $J = 256$ represents the embedding dimension.
- Using a pre-trained model, BaselineLeaky and BaselineShared will produce non-varying event representations regardless of input order.

Model results on 10% data

| Model (10% v1 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|---------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 94.40% | 9.20% | 45.1 | 31.9 |
| Beginning | Adam 0.01 | 94.40% | 9.20% | 18.3 | 16.8 |
| SeqAttn | Adam 0.001 | 87.80% | 9.50% | 52.3 | 39.5 |
| BaselineLeaky | Adam 0.01 | 88.80% | 9.50% | 34 | 38.1 |
| BaselineShared | Adam 0.01 | 88.70% | 9.40% | 36.3 | 37.3 |

| Model (10% v2 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|---------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 97.20% | 15.10% | 48 | 41.3 |
| Beginning | Adam 0.01 | 97.30% | 15.10% | 18.8 | 18.5 |
| SeqAttn | Adam 0.001 | 93.50% | 15.80% | 54.1 | 38.6 |
| BaselineLeaky | Adam 0.01 | N/A | | | |
| BaselineShared | Adam 0.01 | | | | |

- Beginning does not show any good potential on thematic fit.
- SeqAttn performs well on all metrics except role accuracy.
- Modifying the PReLU layer boosts McRae but reduces Padó scores.

Variants of SeqAttn

- From the results, we decide to invest more time into sequential attention models.
- We experiment with the attention mechanisms summarized below.

| Mechanism | Scoring function | Learnable parameter | Model name |
|---------------------------------------|------------------------------|---------------------|----------------------|
| Location-based (Luong, 2015) | $a_t = W e_t + b$ | W, b | SeqAttnLocation |
| General (Luong, 2015) | $a_t = e_t^T W e_t$ | W | SeqAttnGeneral |
| Dot product (Luong, 2015) | $a_t = e_t^T e_t$ | None | SeqAttnDotProd |
| Scaled dot product (Vaswani, 2017) | $a_t = e_t^T e_t / \sqrt{n}$ | None | SeqAttnScaledDotProd |

- Also, we introduce a new task-specific attention mechanism.

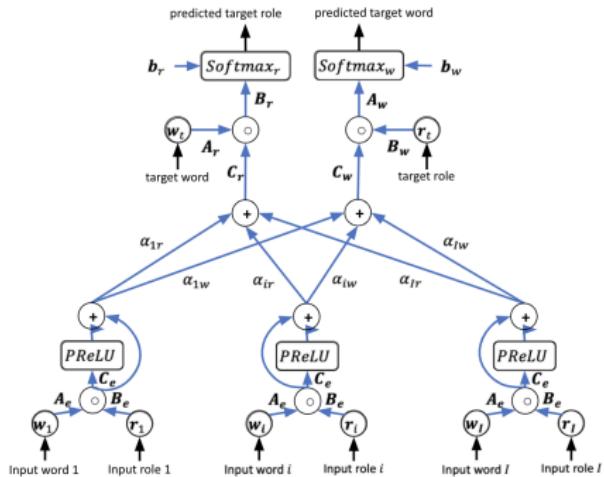


Figure: SeqTargetAttn architecture

- **SeqTargetAttn**: task-specific attention mechanism (one for word prediction and another for role prediction).
 - Inspired by Liu et al. (2009), who posits that task-specific attention mechanism in a multi-task model setting may increase model generalization and performance.

Model results on 10% data

| Model (10% v1 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|----------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 94.40% | 9.20% | 45.1 | 31.9 |
| SeqAttn | Adam 0.001 | 87.80% | 9.50% | 52.3 | 39.5 |
| SeqAttnDotProd | Adam 0.001 | 88.10% | 9.30% | 50.6 | 39.5 |
| SeqAttnScaledDotProd | Adam 0.001 | 88% | 9.30% | 50.5 | 38.5 |
| SeqAttnGeneral | Adam 0.001 | 87.80% | 9.30% | 52.4 | 40.5 |
| SeqAttnLocation | Adam 0.001 | 87.50% | 9.30% | 49.2 | 38 |
| SeqTargetAttn | Adam 0.001 | 87.60% | 9.40% | 53.1 | 37.5 |

| Model (10% v2 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|----------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 97.20% | 15.10% | 48 | 41.3 |
| SeqAttn | Adam 0.001 | 93.50% | 15.80% | 54.1 | 38.6 |
| SeqAttnDotProd | Adam 0.001 | 93.60% | 15.70% | 56 | 37.9 |
| SeqAttnScaledDotProd | Adam 0.001 | 94% | 15.70% | 55.9 | 38.8 |
| SeqAttnGeneral | Adam 0.001 | 93.60% | 15.70% | 54.3 | 39.1 |
| SeqAttnLocation | Adam 0.001 | 93.10% | 15.70% | 56.9 | 33.2 |
| SeqTargetAttn | Adam 0.001 | 93.40% | 15.70% | 56.9 | 38.9 |

- Attention models generally achieve higher thematic fit correlations (except for McRae score for v2 data); SeqTargetAttn obtains the highest Padó score for v1 and v2.
- Using a smaller learning rate (i.e. 0.001) leads to better performance of sequential models, but not for Baseline.

Further modifications

| Model (10% v1 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|----------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 94.40% | 9.20% | 45.1 | 31.9 |
| BaselineShared | Adam 0.01 | 88.70% | 9.40% | 36.3 | 37.3 |
| Beginning | Adam 0.01 | 94.40% | 9.20% | 18.3 | 16.8 |
| BeginningShared | Adam 0.01 | 94.50% | 9.10% | 4.2 | 12.7 |
| SeqAttn | Adam 0.001 | 87.80% | 9.50% | 52.3 | 39.5 |
| SeqAttnShared | Adam 0.001 | 87.40% | 9.00% | 49.8 | 38.6 |
| SeqAttnGeneral | Adam 0.001 | 87.80% | 9.30% | 52.4 | 40.5 |
| SeqAttnGeneralShared | Adam 0.001 | 87.40% | 9.00% | 51.5 | 39.4 |
| SeqTargetAttn | Adam 0.001 | 87.60% | 9.40% | 53.1 | 37.5 |
| SeqTargetAttnShared | Adam 0.001 | 87.00% | 9.00% | 46.1 | 37.9 |

- Modification of activation function and ...
 - target word input timing (i.e. BeginningShared).
 - non-sequential input (e.g. SeqAttnShared).
- Shared PReLU layers generally decrease accuracies and thematic fit scores.

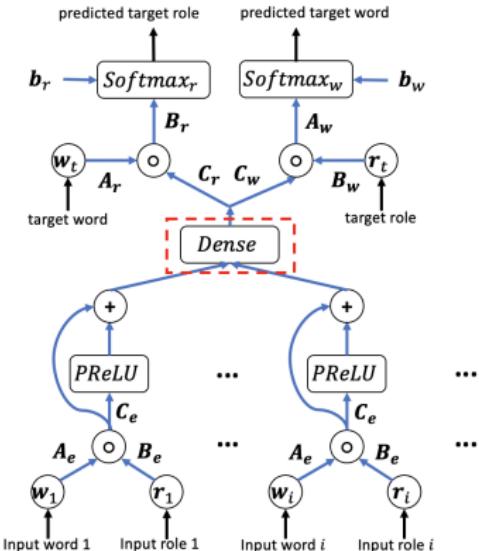
- Modifying target word input timing does not improve baseline model performance.
- Attention mechanisms generally improve thematic fit correlations.
- PReLU layers generally give better results than Shared PReLU or Leaky ReLU layers.
- Code and model training instructions available: <https://github.com/15huangtimothy/bloomberg-event-embedding-team2>.

- Examine possible reasons as to why sequential models generally have low role accuracies.
- Explore different target objectives and consider adding or removing tasks.
- Collaborate with Team 1 on the impact of non-random word/role embeddings on model performance.

Thanks!

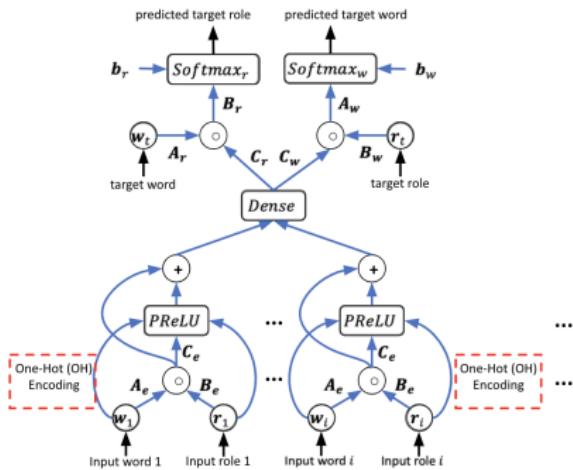


Appendix. Dense



- Dense only differs from Baseline by substituting the mean aggregation layer with a dense layer.
 - The rationale behind this change is to allow the neural network to automatically find a functional form which best summarizes the word-role representations.

Appendix. WideDeep



- WideDeep only differs from Baseline by including the one-hot encoding of each input word and role in the input layer.
- This architecture is inspired by the ubiquitous wide-and-deep learning architecture (Cheng et al., 2016), which has been shown to perform well on recommendation tasks.

Model results on 1% v1 data

| Model (1% v1 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|----------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 93.50% | 7.40% | 27.6 | 18.8 |
| Dense | Adam 0.01 | 93.40% | 7.20% | 2.1 | 7.1 |
| Beginning | Adam 0.01 | 93.90% | 7.60% | 17.9 | 17.4 |
| WideDeep | Adam 0.01 | 99.90% | 8.90% | 10.7 | 9.7 |
| SeqAttn | Adam 0.001 | 86.20% | 7.50% | 43.1 | 27.1 |
| SeqAttnDotProd | Adam 0.001 | 85.10% | 7.30% | 41.3 | 29.3 |
| SeqAttnScaledDotProd | Adam 0.001 | 86.00% | 7.40% | 40 | 24.9 |
| SeqAttnGeneral | Adam 0.001 | 78.40% | 7.50% | 45.3 | 31.3 |
| SeqAttnLocation | Adam 0.001 | 85.90% | 7.40% | 41 | 30.1 |
| SeqTargetAttn | Adam 0.001 | 85.50% | 7.00% | 40.7 | 24.8 |
| BaselineLeaky | Adam 0.01 | 86.10% | 7.50% | 47.1 | 30.3 |
| BaselineShared | Adam 0.01 | 86.40% | 7.40% | 45.8 | 33.9 |

Model results on 1% v2 data

| Model (1% v2 data) | Optimizer + Learning Rate | Val Role Accuracy | Val Word Accuracy | Pado-All | McRae-All |
|----------------------|---------------------------|-------------------|-------------------|----------|-----------|
| Baseline | Adam 0.01 | 96.40% | 13.20% | 34 | 21.1 |
| Dense | Adam 0.01 | 96.00% | 10.90% | 9.5 | 3.7 |
| Beginning | Adam 0.01 | 96.70% | 13.30% | 34.2 | 18.2 |
| WideDeep | Adam 0.01 | 99.90% | 13.10% | 10.2 | 22.2 |
| SeqAttn | Adam 0.001 | 92.00% | 13.20% | 42.3 | 29.3 |
| SeqAttnDotProd | Adam 0.001 | 82.90% | 12.30% | 42.9 | 30.7 |
| SeqAttnScaledDotProd | Adam 0.001 | 92.10% | 13.00% | 42.9 | 28.9 |
| SeqAttnGeneral | Adam 0.001 | 82.70% | 12.40% | 51.3 | 30 |
| SeqAttnLocation | Adam 0.001 | 92.10% | 13.60% | 47.1 | 29.3 |
| SeqTargetAttn | Adam 0.001 | 92.10% | 12.50% | 50.2 | 28.3 |
| BaselineLeaky | Adam 0.01 | N/A | | | |
| BaselineShared | Adam 0.01 | | | | |