Neural Events Extraction from Movie Descriptions

Alex Tozzo, Dejan Jovanović and Mohamed R. Amer
SRI International
201 Washington Rd
Princeton, NJ08540, USA
{dejan.jovanovic, mohamed.amer}@sri.com

Abstract

We present a novel approach for event extraction and abstraction from movie descriptions. Our event frame consists of 'who", "did what" "to whom", "where", and "when". We formulate our problem using a recurrent neural network, enhanced with structural features extracted from syntactic parser, and trained using curriculum learning by progressively increasing the difficulty of the sentences. Our model serves as an intermediate step towards question answering systems, visual storytelling, and story completion tasks. We evaluate our approach on MovieQA dataset.

1 Introduction

Understanding events is important to understanding a narrative. Event complexity varies from one story to another and the ability to extract and abstract them is essential for multiple applications. For question answering systems, a question narrows the scope of events to examine for an answer. For storytelling, events build an image in the reader's mind and constructs a storyline.

For event extraction, we apply Natural Language Processing (NLP) techniques to construct an event frame consisting of: "who", "did what" "to whom", "where", and "when". The more complex questions of "how" and "why" requires significantly more reasoning and beyond this paper's scope. Most syntactic NLP parsers, such as CoreNLP (Manning et al., 2014) and NLTK (Bird et al., 2009), focused on examining characteristics of the words, grammatical structure, word order, and meaning (Chomsky, 1957). On the other hand neural NLP approaches, such as SLING (Ringgaard et al., 2017) relies on large corpora to train such models in addition multiple knowledge databases. These approaches perform event extraction without context (often visual) of a movie or a movie script. When performing event extraction in relation to events in a story, the context can be gleaned from descriptions of the set or characters or prior events in a sequence. Additionally, we intend to develop an event extraction framework for a mixed-initiative, human-computer, system and intend to generate a human-readable event structure for user interaction.

In departure from syntactic approaches, we propose a hybrid, neural and symbolic, approach to address this problem. We benefit from both neural and symbolic formulations to extract events from movie text. Neural networks have been successfully applied to NLP problems, specifically, sequence-to-sequence or (sequence-to-vector) models (Sutskever et al., 2014) applied to machine translation and word-to-vector (Mikolov et al., 2013a). Here, we combine those approaches with supplemental structural information, specifically sentence length. Our approach models local information and global sentence structure.

For our training paradigm, we explored curriculum learning ((Bengio et al., 2009). To the best of our knowledge, we are the first to apply it to event extraction. Curriculum learning proposes a model can learn to perform better on a difficult task if it is first presented with easier training examples. Generally, in prior curriculum learning work, the final model attains a higher performance than if it were trained on the most difficult task from the start. In this work, we base the curriculum on sentence length, reasoning that shorter sentences have a simpler structure. Other difficulty metrics such as average word length, longest word length, and FleschKincaid readability score were not considered in this experiment, but may be considered for future work.

Instead of treating the sentence-to-event problem as a complete black-box putting the burden

Proceedings of the First Workshop on Storytelling, pages 60–66
New Orleans, Louisiana, June 5, 2018.

on the deep learning model, we simplify the problem by adding structure to the output sentence following the event frame structure, where some of the components could be present or absent. Furthermore, some sentences could contain multiple events. Weak labels were extracted from each sentence using the predicate as an anchor. We use structure rather than a bag-of-words because it encodes information about the relationships between words.

Our contributions are three-fold:

- New formulation for event extraction in movie descriptions.
- A curriculum learning framework for difficulty based learning.
- Benchmarking symbolic and neural approaches on MovieQA dataset.

The paper is organized as follows: section 2 reviews prior work; Section 3 formulates our approach; Section 4 specifies the learning framework; Section 5 presents our experiments; Section 6 describes our future work and conclusion.

2 Prior Work

Event extraction is a well established research problem in NLP. Parsers have been developed to extract events and event structures using a variety of methods both supervised and unsupervised.

(McClosky et al., 2011) uses dependency parsing to extract events from sentences (converted to a dependency tree by a separate classifier) by identifying event anchors in a sentence and graphing relationships to its arguments.

(Chambers and Jurafsky, 2008) and (Chambers and Jurafsky, 2009) develop an unsupervised method to learn narrative relations between events that share a co-reference argument and, later, a sequence of events over multiple sentences.

(Martin et al., 2017) and (Martin et al., 2018) present a neural technique for generating a mid-level event abstraction that retains the semantic information of the original sentence while minimizing event sparsity. They formulate the problem as first, the generation of successive events (event2event in their parlance), then generate natural language from events (event2sentence). The authors use a 4-tuple event representation with subject, verb, object, and a modifier of the sentence including prepositional phrase, indirect object, or causal complement. One key concept is that these events are in generalized WordNet forms and are not easily human-readable. Their event2event network is an encoder-decoder model. The event2sentence model is similar to the event2event model with the exception of using beam search.

(Harrison et al., 2017) introduces a Monte Carlo approach for story generation, a related application of event extraction. Citing RNN's difficulty in maintaining coherence across multiple sentences, they develop a Markov Chain Monte Carlo model that can generate arbitrarily long sentences. In this work, they use the same event representation as (Martin et al., 2018).

Prior work in curriculum learning ((Bengio et al., 2009)) explored shape recognition and language modeling. Specifically for language modeling, they experiment with a model to predict the best word following a context of prior words in a correct English sentence. Their language modeling experiment expanded the vocabulary, increasing the task difficulty as more words were added to the corpus. More recent work ((Graves et al., 2017)) applied curriculum learning to question-answering problems on the bAbI dataset (Weston et al., 2015), designed to probe reasoning capabilities of machine learning systems.

The problem with symbolic approaches is the rigidity of the parsers and only basing the parses on the encoded knowledge. The neural approaches are unbounded and produce a huge variety of generated sentences. However, they are not conditioned on specific text and the results vary, often producing unrealistic sentences. We propose a hybrid of the two approaches to provide structured events conditioned on realistic content.

3 Approach

Our goal is to extract event frames in movie description in the format of "who" "did what" "to whom or what" "when" and "where". By extracting particular components of an event, it becomes easier to instantiate an event as an animation using existing software or present the event object to a human user for them to instantiate on their own terms. Once events are extracted in this format, a sequence of events can be used to animate the script and generate a short movie. In contrast to the purely symbolic approach taken by others, we take a neural approach, applying Recurrent Neural Networks (RNN). The idea is that an RNN will learn to output a structure mirroring the symbolic

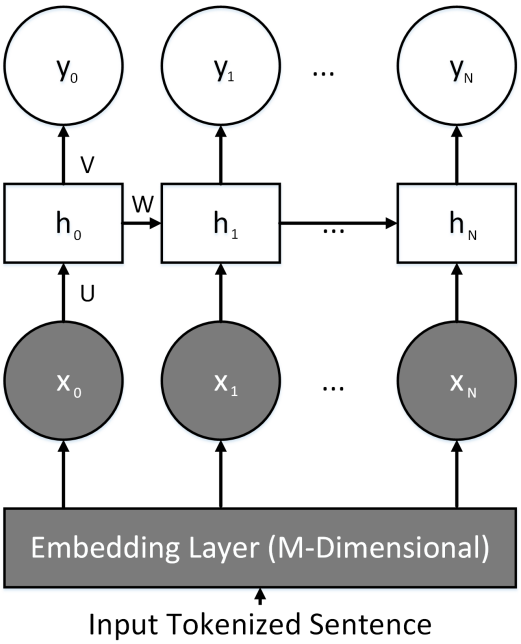

Figure 1: The model takes input word2vec vectors of each word and outputs their labels.

approaches. The input nodes are encoded as a fixed sequence of identical length and the output are labels of the provided structure. Figure 1 illustrates our model. We first encode each word in the input sentence to an *M*-dimensional vector using word2vec (Mikolov et al., 2013b). The embedding output vectors input an *M*-dimensional RNN with LSTM units. We standardized the length of the sentence by padding short sentences and capping the length of the longest sentence to be 25 words. The hidden state of each unit is defined in (1) for $\mathbf{h^{(t)}}$. The output of each unit is $\mathbf{o^{(t)}}$ and is equal to the hidden state. The internal cell state is defined by $\mathbf{C^t}$. Intermediate variables $\mathbf{f^{(t)}}$. $\mathbf{i^{(t)}}$, $\hat{\mathbf{c}}^{(\mathbf{t})}$, and $\mathbf{u^{(t)}}$ facilitate readability and correspond to forget, input, and candidate gates, respectively. The cell state and the hidden state are propagated forwards in time to the next LSTM unit.

$$\begin{aligned}
\mathbf{f^{(t)}} &= \sigma(W_{fh}h^{(t-1)} + W_{fx}x^{(t)} + b_f) \\
\hat{\mathbf{c}}^{(\mathbf{t})} &= \tanh(W_{ch}h^{(t-1)} + W_{cx}x^{(t)} + b_c) \\
\mathbf{i^{(t)}} &= \sigma(W_{ih}h^{(t-1)} + W_{ix}x^{(t)} + b_i) \\
\mathbf{C^{(t)}} &= \mathbf{C^{(t-1)}} \odot \mathbf{f^{(t)}} + \mathbf{i^{(t)}} \odot \hat{\mathbf{c}}^{(\mathbf{t})} \\
\mathbf{o^{(t)}} &= \sigma(W_{oh}h^{(t-1)} + W_{ox}x^{(t)} + b_o) \\
\mathbf{h^{(t)}} &= \tanh(\mathbf{C^{(t)}}) \odot \mathbf{o^{(t)}}
\end{aligned} \tag{1}$$

A significant hurdle in training any of network in this instance is class imbalance. Here, the model is trained using standard back-propagation with a weighted cross-entropy loss function used to avoid over-fitting to the null class.

4 Curriculum Learning

Sentences vary in difficulty due to structure, context, vocabulary, and more. As part of our experiments, we employed curriculum learning to potentially facilitate the learning processes. We compare the curriculum training to standard batch processing.

We divide the training samples into three difficulty groups based on sentence length. We train the model with the easiest set first for 100 epochs before advancing to the medium and hard difficulty training samples, training for 100 epochs each. This results in 300 training epochs total, although the model is only exposed to a third of the dataset for 100 epochs at a time. We compare this to models where the training process exposes the model to the entire corpus for 300 epochs. We use sentence length, assuming that shorter sentences are easier as they contain fewer descriptive words, but other structural and semantic metrics can be used.

5 Experiments

MovieQA Dataset (Tapaswi et al., 2016): We use the descriptive video service (DVS) text from MovieQA. The DVS sentences tend to be simpler and describe the scene explicitly compared to the plot synopsis sentences. We generate an initial training corpus of extracted events using dependency parsers and information extraction annotators from CoreNLP. A total of 36,898 events are generated. Analyzing the corpus of extracted events, we found the longest sentence length contained 62 words. However, by limiting our dataset to sentences with 25 words or less, we retained 97% of the data (35791 sentences). Figure 3 shows the sentence length distribution of the DVS data and the plot synopsis data. We did not experiment with the plot synopsis data, rather we wish to highlight the difference in sentence lengths between the 2 sets of data. The DVS data is heavily skewed towards shorter sentences, most likely due to requiring concise descriptions about what is happening on screen at that time. Plot synopsis sentences tend to be longer as they tend to summarize multiple actions and plot points. Sentences with multiple predicates generated multiple events and this manifested itself as duplicate sentences in our dataset with multiple label sequences. Sentences with multiple events accounted for about 24% of the data. This does lead to complications for train-

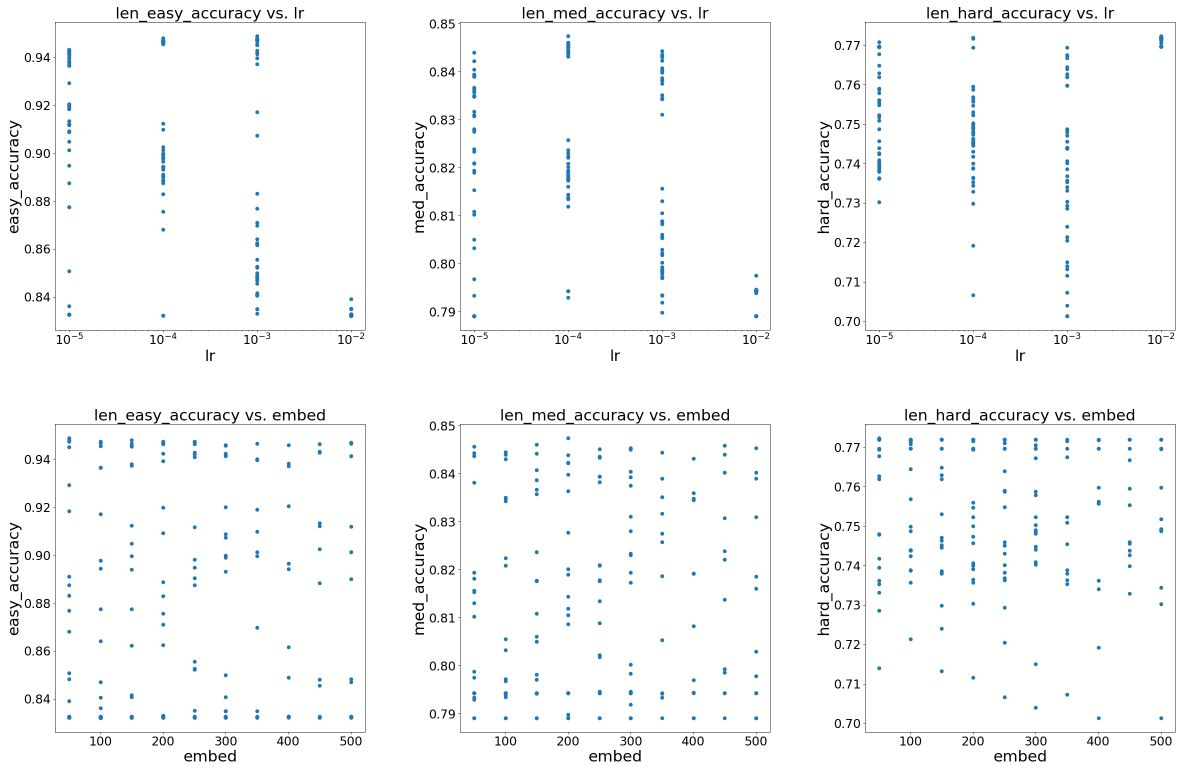

Figure 2: (Top, left to right) Easy, Medium, Hard Accuracy vs Learning Rate. (Bottom, left to right) Easy, Medium, Hard Accuracy vs Embedding Dimension.

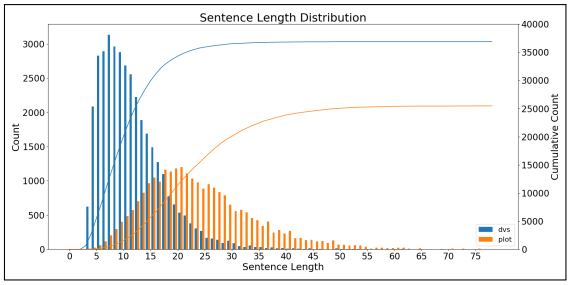

Figure 3: Sentence Length Distribution from DVS (blue) and Plot Synopsis (orange) Data and Cumulative Distribution (solid lines and right axis)

ing as we did not distinguish between events.

The difficulty classes computed by sentence length and are as follows: easy sentences are 8 words or less with an average sentence length of 6.5 words, medium sentences are 9-12 words with an average length of 10.4 words, and hard sentences are 13 words or longer with an average length of 16.8 words. Each difficulty class is contains one third of the original data set. For training with and without a curriculum, we used an 80-20 train-test split. For curriculum learning, each difficulty was trained on 80% of each of the respective difficulty sets and tested on the remaining 20%. The extracted events provide weak labels generated by the CoreNLP algorithm approach.

Pre-processing: We tokenized the text assigning an integer to each word after removing capitalization and apostrophes. Sentences are vectorized using this index. The output format assigns integers between 1-5 to parts of the sentence based on which elements of the sentence are part of the subject (1), predicate (2), object (3), location (4), or time (5) phrases. Articles, prepositions, conjunctions, adjectives, and adverbs were often assigned the null class (0) although some may be included parts of event phrases. Sentences are left-padded with zeros make all sentence vectors the same length.

Implementation Details: The trained model is a basic LSTM model. We employ two different training approaches. In the first approach, we ignore the sentence length and use random batches of training data. We train for 300 epochs. Second, we use a training curriculum based on sentence length, starting training with shorter sentences and progressing to longer sentences. The sentences are divided into easy, medium, and hard difficulty sets with each set containing roughly one-third of the

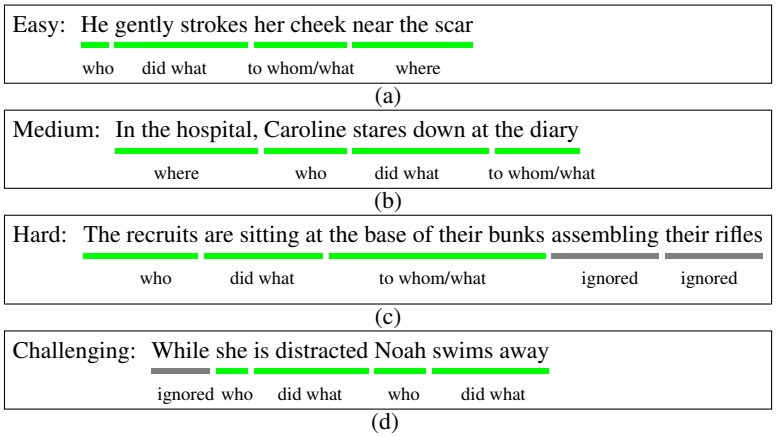

Figure 4: Successful Cases

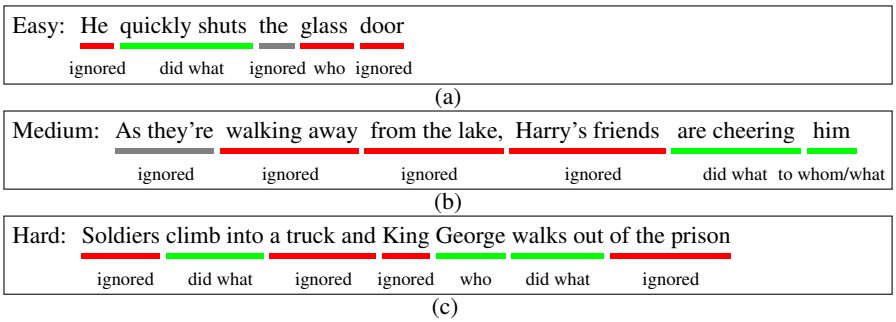

Figure 5: Failure Cases

total data set. We hold out a set of data from each difficulty level for validation. We train the model for 100 epochs on each level of difficulty to match the total of 300 epochs of the random training curriculum. The final approach is to start with shorter sentence lengths and train with longer sentences towards the end.

We examine four parameters: the learning rate, the embedding dimension, the hidden dimension, and number of training epochs. A grid search was employed to examine the effects of these parameters on the validation accuracy. We varied learning rate in powers of 10 from $1e-5$ to $1e-2$. The embedding dimension was varied from 50-500 in increments of 50. The hidden dimension was varied from 48-512 in increments of 16. Lastly, we varied the number of training epochs from 300-2000 in increments of 200. Below we include a sample of figures from this search where we fixed the number of epochs (300) and the hidden layer dimension (512) while adjusting the learning rate and the embedding dimension. For these parameters, we found an embedding dimension of 350 performs best on the easy and medium difficulty levels. Additional training is in progress.

We used ADAM (Kingma and Ba, 2014) for gradient optimization. Our loss function was a weighted categorical cross entropy function using the class distribution from the training data as class weights. Accuracy is calculated by the frequency with which the predicted class matches the labels. The accuracy is then the total count of actual matches by the total potential matches.

5.1 Qualitative Results

In this work, we used the symbolic algorithm as a weak label for the neural network approach. The symbolic approach appears to work well for shorter sentences with simple sentence structure. However, as the sentences become longer and additional descriptive phrases, the dependency parsing becomes more complex. We present 3 examples (1 from each difficulty level) of a simple sentence the symbolic algorithm does well.

Figure 4 shows examples where our approach was successful. Figure 4(a) illustrates an easy difficulty sentence. The parser correctly identifies the *he* as the subject, *gently strokes* as the verb phrase, *her cheek* as the object, and *near the scar* as the location. Figure 4(b) illustrates a medium difficulty sentence. The parser identifies *Caroline* as the subject again, the verb phrase *stares down at*, the object *diary* and the location *in the hos-*

pital. Figure 4(b) illustrates a medium difficulty sentence. Figure 4(c) illustrates a hard difficulty sentence. The parser identifies *recruits* as the subject. The verb phrase identified here is *are sitting at* and the object is *base of their bunks*. Another verb phrase that could be identified is *assembling* with the object *their rifles*. These examples also show how the model ignored articles in the sentences. Finally, Figure 4(d) illustrates a challenging sentence. One pleasantly surprising example of the model learning multiple events in a single short sentence. The model correctly identifies 2 subjects (*she, Noah*) and 2 verb phrases (*is distracted, swims away*).

Figure 4 shows examples where our approach failed. Figure 5(a) illustrates an easy difficulty sentence. The model incorrectly identifies *glass* as a predicate while correctly identifying *shuts*, suggesting the model does not anchor events around a predicate phrase. Figure 5(b) illustrates an medium difficulty sentence. The model identifies the verb phrase *are cheering* and the object *him*, but fails to recognize the subject *Harry's friends*. This is odd as it would suggest the model doesn't recognize possessive apostrophes as part of a noun phrase, but may be confusing it with a contraction. However, other situations show the model does not recognize the contraction either. Figure 5(c) illustrates a hard difficulty sentence. The model fails to identify *soldiers* as a subject of any predicate, yet correctly identifies *climb into* and *walks out of* as predicate phrases. It does, however, identify *George* as a subject, but not his descriptor of *King*.

As sentences get longer, the model begins to breakdown. This may be due to the weak labels provided by the symbolic algorithm. It may also be due to the non-linear relationship between subject, predicate, and object in the sentence. The neural model also fails when the location is a generic place such as a *cafe* or *garage*. One improvement could be to incorporate the WordNet meaning of each word in the sentence.

5.2 Quantitative Results

We show results from curriculum learning using the sentence length as a basis for the curriculum. The accuracy is shown in Figure 6 for both the non-curriculum learning and the length-based curriculum. We first train with easy data, wit the easy validation data closely tracking it. After 200 epochs, we begin training with medium difficulty data. At this point, the easy validation data changes only a little, while medium and hard difficulty validation data continue to increase slightly. At 400 epochs, we begin training with the most difficult data: data containing the longest set of sentences our dataset. Introducing the hard training data affects the easy and medium validation accuracy. The hard difficulty validation accuracy continues to increase while easy and medium drop. Due to the semi-supervised method of labeling the data using symbolic methods, we believe longer sentences tend to be noisier and less accurately labeled compared to shorter sentences. This introduces noisy labels for the network, confusing it on previously learned examples leading to degraded performance. Descriptive phrases containing nouns can complicate the network and hinder identification of subject or object.

6 Conclusions and Future Work

This work presents an initial study of neural event extraction. We intend to study a bidirectional LSTMs and encoder-decoder models in future work. We anticipate bidirectional models and encoder-decoder models will enable the network to capture longer-term dependencies between object and predicate. We also plan to extend the data set to additional sources with human annotations for more accurate ground truth labels.

An additional direction for future work is to incorporate graphs as a mechanism to enforce structure. In addition, we can extract events from visual information and use it to guide the events extracted from textual information. Using graphs generated from both visual and textual information will result in a more complete, and less noisy event representation.

This experiment is a component for our work towards developing a mixed-initiative system for visual storytelling. Here, we take preliminary steps towards extracting events from movie descriptions with the intention of then instantiating events in an animation module. The simplified event structure facilitates the mixed system where either the computer or human can suggest events or render the event in a particular style or genre. Our vision for the system is to have a human and a computer take turns suggesting new events in a story or suggest a story arc and generate pertinent, relevant events to justify the conclusion.

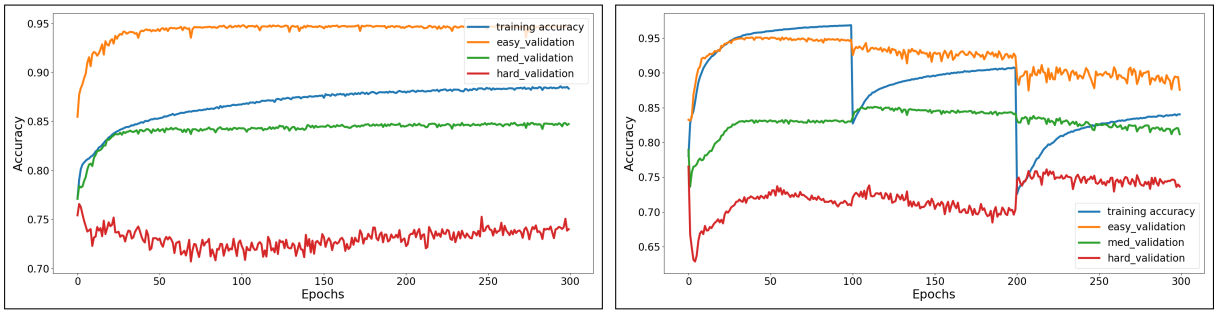

Figure 6: (Left) No curriculum learning with all difficulties mixed into training data. Validated against all difficulty levels at each epoch. (Right) Curriculum learning based on sentence length. Each difficulty trained for 100 epochs in easy, medium, hard order. Validated against all difficulty levels at each epoch.

Acknowledgments

This work is funded by DARPA W911NF-15-C-0246. The views, opinions, and/or conclusions contained in this paper are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied of the DARPA or the DoD. Special thanks to Karine Megerdoomian for the helpful discussions.

References

Yoshua Bengio, Jrme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. *ICML*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. OReilly Media Inc.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. *ACL*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. *IJCNLP*.

N. Chomsky. 1957. Syntactic structures.

Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. *ICML*.

Brent Harrison, Christopher Purdy, and Mark O. Riedl. 2017. Toward automated story generation with markov chain monte carlo methods and deep neural networks. In *Workshop on Intelligent Narrative Technologie*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL-Systems*.

Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, Shruti Singh, Brent Harrison, Pradyumna Tambwekar Murtaza Dhuliawala, Animesh Mehta, Richa Arora, Nathan Dass, Chris Purdy, and Mark O. Riedl. 2017. Improvisational storytelling agents. In *NIPS-Workshops*.

Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, Shruti Singh, Brent Harrison, and Mark O. Riedl. 2018. Event representations for automated story generation with deep neural nets. In *AAAI*.

David McClosky, Mihai Surdeneau, and Christopher D. Manning. 2011. Event extraction as dependency parsing. *Proceedings of BioNLP Shared Task 2011 Workshop*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Michael Ringgaard, Rahul Gupta, and Fernando C. N. Pereira. 2017. SLING: A framework for frame semantic parsing. *CoRR*, abs/1710.07032.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.