ANLP

Assignment – 2 ELMO

Aditya Raghuvanshi
2021114009

This report provides a comprehensive overview of ELMo (Embeddings from Language Models), a deep contextualized word representation model, developed by researchers at the Allen Institute for Artificial Intelligence (AI2) in 2018. Unlike traditional word embedding models, ELMo excels at capturing the contextual meaning of words within sentences, significantly enhancing performance across various natural language processing tasks. This report delves into the architecture of ELMo, highlighting its bidirectional Long Short-Term Memory (Bi-LSTM) networks, training objectives, and its unique approach to representing words in context. Furthermore, it discusses the pretraining process and the subsequent fine- tuning of ELMo for specific downstream tasks, such as sentiment analysis and natural language inference. Lastly, it emphasizes the role of stacked Bi-LSTMs in capturing the intricacies of language, from syntactic to semantic aspects, and how they contribute to ELMo's effectiveness in various applications.

Introduction: ELMo Overview

ELMo, or Embeddings from Language Models, is a revolutionary word representation model introduced in 2018 by the Allen Institute for Artificial Intelligence (AI2). Its distinguishing feature lies in its ability to capture the contextual meaning of words within a sentence, in contrast to traditional word embedding models.

ELMo Architecture

ELMo's architecture is built upon bidirectional Long Short-Term Memory (BiLSTM) networks. The input to these Bi-LSTMs incorporates non-contextual word embeddings (e.g., word2vec or character convolutional network). The forward and backward language models are trained separately, with the forward model predicting the next word in a sentence, while the backward model predicts the preceding word. Both models leverage contextual information, considering the words that surround the target word.

Pretraining for Language Understanding

Training the Bi-LSTMs with the language modeling objective plays a crucial role in pretraining ELMo's weights. This phase equips ELMo with a profound understanding of language structure and dependencies. Consequently, ELMo becomes adept at solving downstream tasks with less training data and time.

Fine-Tuning for Downstream Tasks

ELMo's utility extends to a wide range of natural language processing tasks, which are collectively referred to as downstream tasks. The model can be fine-tuned for specific tasks, such as sentiment analysis and natural language inference, to adapt its contextual embeddings to the requirements of these tasks.

Stacked Bi-LSTMs for Layered Representation

ELMo employs stacked Bi-LSTMs to represent the complexities of natural language in a layered fashion. Lower layers focus on capturing syntactic aspects, while higher layers delve into more intricate semantic aspects. The use of Bi-LSTMs allows ELMo to consider the bidirectional context of each word based on its surrounding words. At each layer, the forward and backward LSTM representations are concatenated, progressively building a more comprehensive bidirectional context. These representations, combined with non contextual input embeddings, are summed to generate the final contextual word representation in ELMo's architecture.

Conclusion

In conclusion, ELMo is a groundbreaking model in the realm of natural language processing. Its ability to capture context and nuances within sentences through stacked Bi-LSTMs has proven invaluable across various tasks. From pretraining for language understanding to fine-tuning for specific applications, ELMo's versatility makes it a powerful tool for researchers and practitioners alike. This report serves as a foundational understanding of ELMo's architecture and its implications for natural language processing tasks.

Hyperparameter :

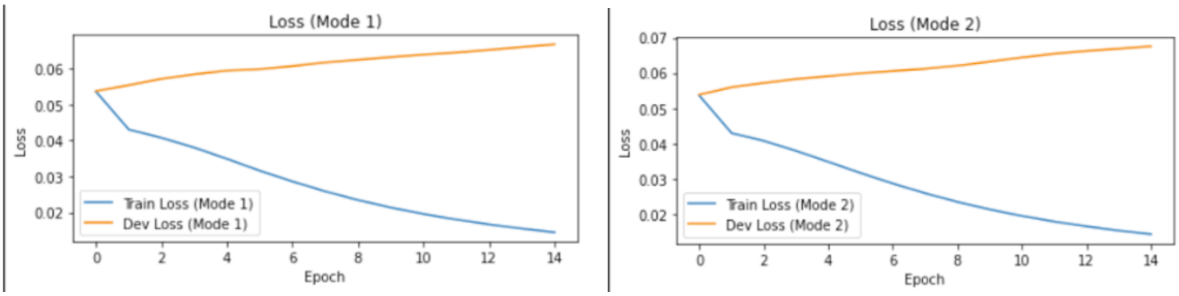
Number of epochs - 10
Criterion – Cross entropy loss
Optimizer – Adam (initial lr → 0.001)

Forward language modeling (Mode 1) results :

Epoch	Train Loss	Dev Loss
1	0.0535	0.0537
2	0.0430	0.0553
3	0.0407	0.0571
4	0.0380	0.0583
5	0.0349	0.0593
6	0.0316	0.0598
7	0.0286	0.0606
8	0.0259	0.0616
9	0.0235	0.0624
10	0.0214	0.0631

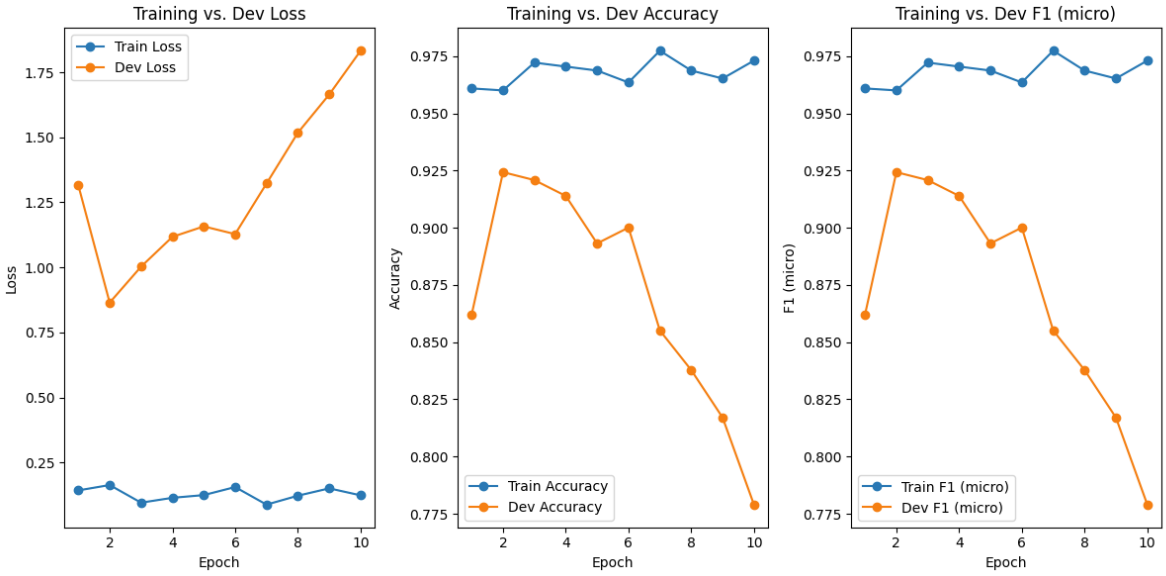
Backward language modeling (Mode 2) results :

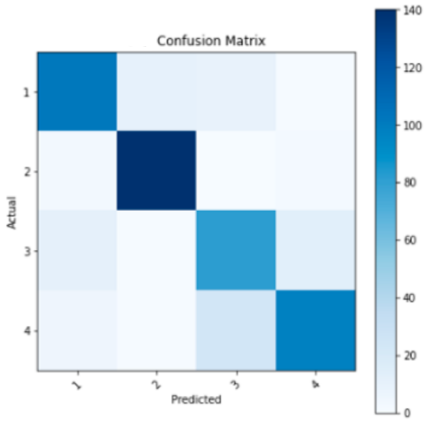
Epoch	Train Loss	Dev Loss
1	0.0536	0.0538
2	0.0429	0.0559
3	0.0407	0.0572
4	0.0379	0.0583
5	0.0348	0.0590
6	0.0316	0.0599
7	0.0286	0.0605
8	0.0259	0.0612
9	0.0234	0.0620
10	0.0213	0.0632



Downstream Tasks STS ELMo Results

Epoch	Dev Loss	dev Accuracy	dev F1 (micro)	train Loss	train Accuracy	F1 (micro)
1	1.3155572	0.86204152	0.86204152	0.14254993	0.9609375	0.9609375
2	0.86439419	0.92432526	0.92432526	0.16315392	0.96006944	0.96006944
3	1.00258624	0.92086505	0.92086505	0.09536567	0.97222222	0.97222222
4	1.11729398	0.91394464	0.91394464	0.11401182	0.97048611	0.97048611
5	1.15744456	0.89318339	0.89318339	0.12448922	0.96875	0.96875
6	1.12780949	0.90010381	0.90010381	0.15539427	0.96354167	0.96354167
7	1.32226998	0.85512111	0.85512111	0.08787075	0.97743056	0.97743056
8	1.51756157	0.83782007	0.83782007	0.12210879	0.96875	0.96875
9	1.66459856	0.81705882	0.81705882	0.15035977	0.96527778	0.96527778
10	1.83291838	0.77899654	0.77899654	0.12354733	0.97309028	0.97309028

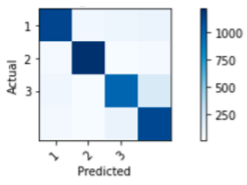




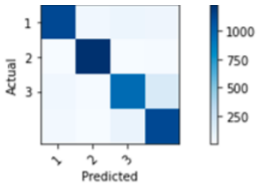
BONUS

We present the weight vectors used in the ELMo model for combining embeddings. These weight vectors play a significant role in shaping the characteristics of the resulting ELMo embeddings. We observe that each weight vector consists of three values, which control the contribution of different layers of the pre-trained language model to the final ELMo representation.

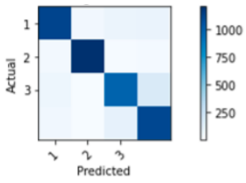
[0.33, 0.33, 0.33]



[0, 0.5, 0.5]



[0.5, 0.25, 0.25]



[0.4, 0.4, 0.2]