Deep contextualized word representations

NLP Reading Assignment

Vaibhav Nagrale
112001046

ABSTRACT

This paper introduces a new type of word representation, deep contextualized word representation. This representation models both semantics and context. Here we use a bidirectional language model (biLM). This model can be easily used across various NLP tasks.

INTRODUCTION

In models, learning high quality representation can be challenging. They must model both semantics and context. Here we use a type of word representation contextualized word representation, in contrast to traditional word vector representations that use the entire context when a word should appear in order to form its representation. We use vectors from bidirectional LSTM trained with Language Model on large corpus, thus it is called Embeddings from Language Models (ELMo). We combine all internal states of ELMo to get good word representation.

MOTIVATION

Language understanding requires context.

So traditional word vectors use context to form the representation however since they have an assumption of learning a single vector per word they have to compress all the context into the same vector and when they are used for tasks in other places these vectors are used in isolation.

In this paper they proposed Embeddings from Language Models(ELMo).Instead of forming single representations for each word we have infinite vectors to represent the word. Now representation is dependent on context.

They evaluated this approach on wide NLP tasks like NER dataset, Question dataset, etc.

METHODOLOGY

I. **ELMo**
   We want to compute a contextual vector, given a context in our case sentence and a word in that sentence, we want to compute a contextual representation of what that word is. And property that this representation would change for different contexts. Compute contextual vector:

$$C_k = f(w_k|w_1, \ldots, w_n) \in R^N$$

$$\text{f(play | Elmo and Cookie Monster play a game.)}$$

$$\neq$$

$$\text{f(play | The Broadway play premiered yesterday.)}$$

Key Ideas:

1. Neural LMs embed the left context of a word Neural LMs already learn how to encode one sided context
2. We can introduce a bidirectional LM to embed left and right context We can get context for both left and right of the word.

They are computed on top of two-layer biLMs with character convolutions, as a linear function of the internal network states. A biLM combines both a forward and backward LM.

II. Working

To illustrate this, a example:
The Broadway play premiered yesterday

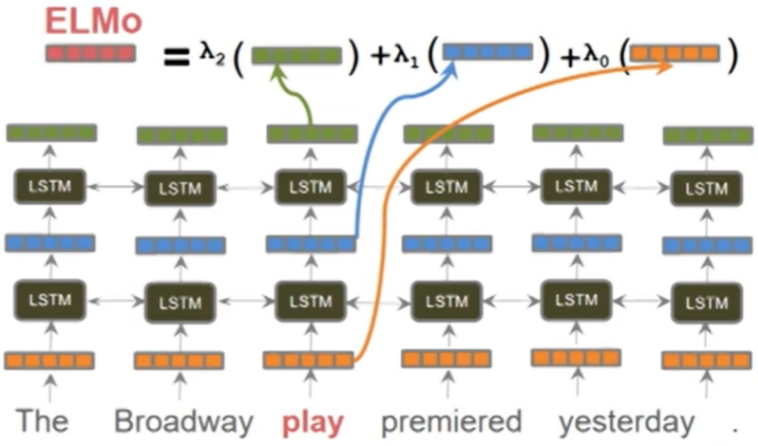
We would like to compute contextual representations for play in this context.
We do this by a four way language model

Here we first form a context insensitive representation, at bottom type representation in our case it is purely character based. We use large set of character n-gram convolutional filters to do this and then you pass these through very large LSTM, trained on large dataset of large tokens so that we can effectively learn this high capacity model and then we can use hidden state from the top layer to predict the next word using softmax function.

The four directional model would go on the left of the word play to get the context to the right. We use a backward language model, this works the same as the forward model except run over the sentence in reverse and predict the previous word. In this work they combined them into bidirectional language models, which work in both directions jointly maintaining separate parameters for each LSTM.

So for word “play” in our example,
We could take the top state of our model. It works well, however we can do better if we use all the layers and average points over all the layers. In our case we got the highest performance by doing learned weighted average points.

$$ELMo = \lambda_2() + \lambda_1() + \lambda_0()$$



RESULT

I. Properties

1. Unsupervised: It is completely unsupervised you can use any annotated data.
2. Contextual: Now representation depends on the entire context and it'll change for different contexts.
3. Deep: We are now computing representations and doing learned averages over multiple layers of representation and also shown a little that language models effectively learn different types of contextual information at all different layers.
4. Character based: They are purely character based. So this allows us to compute representations for tokens that we haven't seen at training time.
5. Extremely versatile: Now we have a new type of word representation so we can use wherever we want (word2vec, glove, etc.).

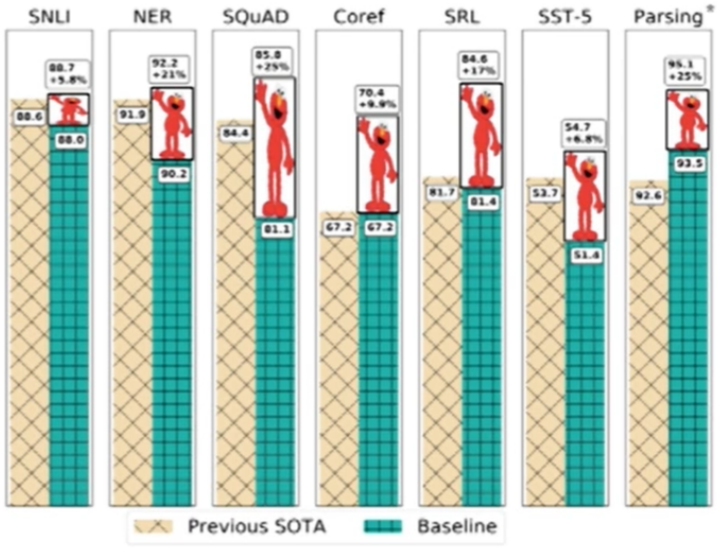
II. Analysis

Key Ideas:

1. LM objective forces the network to implicitly learn how syntax and semantics vary across context.
2. Deep LM allows downstream models to preferentially re-use representations as needed.

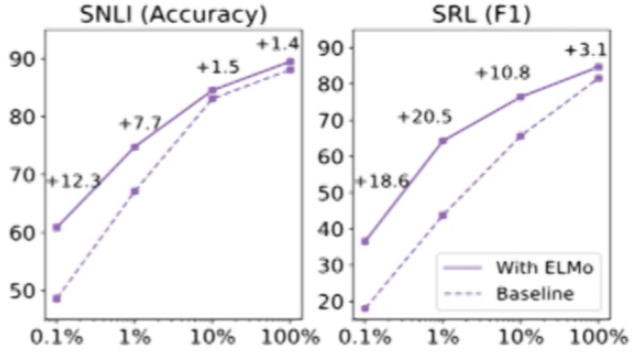
III. Results

Evaluation of models on different NLP tasks.



* Kitaev and Klein, ACL 2018 (see also Joshi et al., ACL 2018)

Evaluated models on a wide range of NLP tasks like Named Entity Recognition dataset, Question Answering dataset, Coref, etc. to compare. A standard state-of-the-art model was taken which was good at that time (yellow). We just added these pretrained ELMo representations to our model (green) and in all cases we got good improvement in results.



ELMo enhanced models are remarkably sample efficient

Two dataset SNLI and SRL, We see that with ELMo we get significant improvement.

CONCLUSION

We see how ELMo improves our accuracy.
Language models learn very useful contextual representations that are transferable to a lot of NLP tasks.