

Optimizing Transfer Learning using Embeddings from Language Models

Isaac Ahouma

Overview

- Introduction & Objectives
- Word Embeddings & Language Modeling
- Contextual Word Embeddings
- Knowledge Distillation
- Conclusion



Introduction

- In recent years deep contextualized learning models have been developed to tackle NLP problems
- Models such as ELMo and BERT have achieved state of the art results across most of NLP tasks
- These models consist of millions of parameters and are almost impossible to push into production environment

Objectives

- Use transfer learning to extract knowledge from ELMo or BERT
- Use Extracted knowledge to train better performing models than those currently in production
- Optimize new models for memory and processing speed
- Deploy optimized models into production



Language Modeling

- Fundamental task in natural language processing (NLP)
- Very challenging to build good language models but major improvements have been made using deep learning
- Goal: given the previous $n-1$ tokens in a sequence (the context), predict the probability of each token in the vocabulary of being the next (nth) token
- ELMo is built using word embeddings learned from language models



Word Embeddings

- Numerical (dense, low dimensional) vector representations of tokens
- Robust word embeddings are learned using algorithms trained on very large datasets
- Allow to capture semantic, syntactic and grammar-based relationships between tokens

Problem: Traditional word embeddings give same representation to words independently of the context they appear in



Models

Contextual Word Embeddings



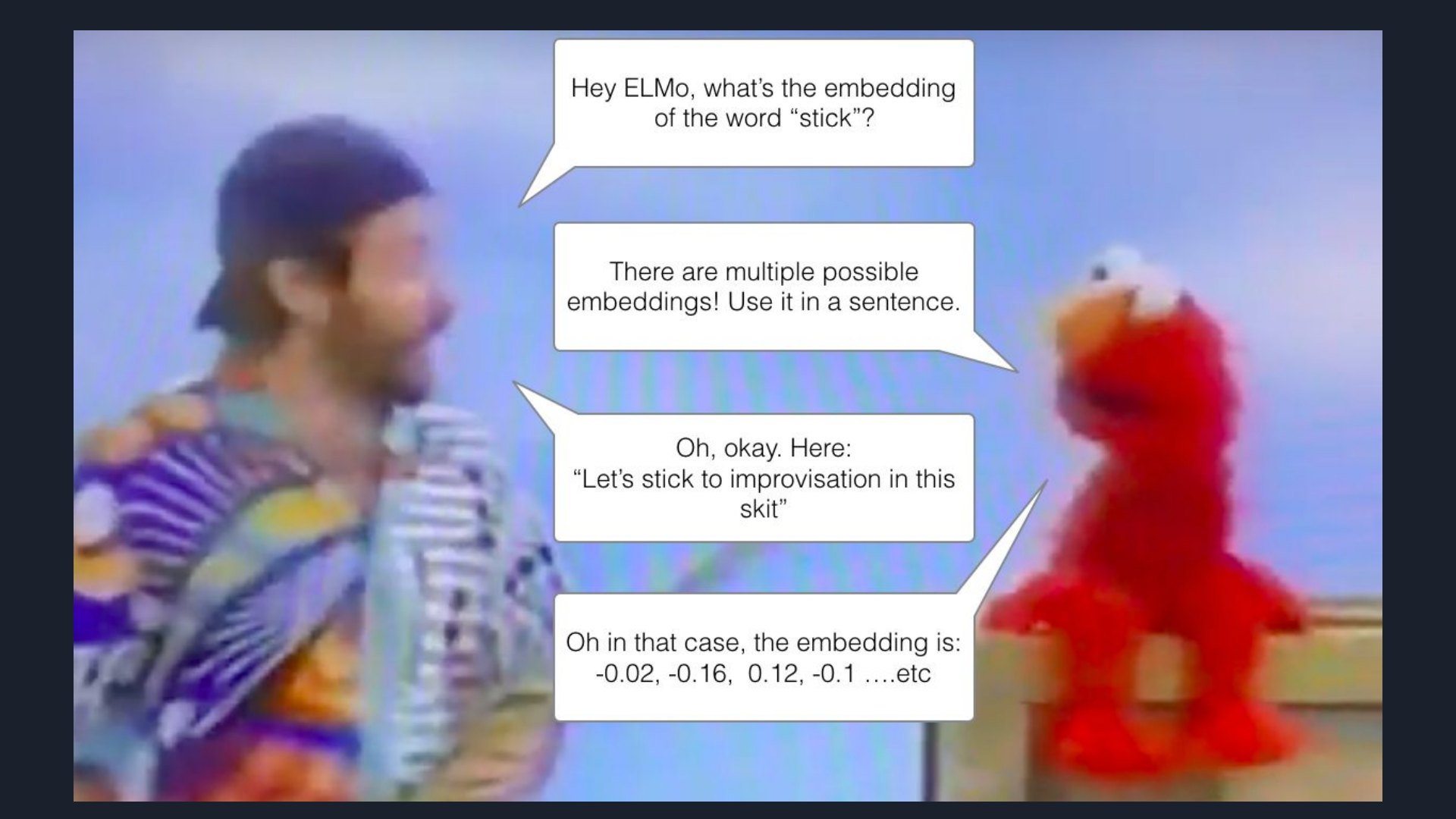
Transfer Learning Using BERT and ELMo

- Use the pre-trained BERT or ELMo available online to create contextualized word embeddings
- Replace word embeddings by contextualized word embeddings in Keanet models
- Train the models as before
- Many possible contextualized word embeddings from BERT or ELMo, so we experimented with each



Embeddings from Language Models aka ELMo

- Words can have different meaning depending on the context
- ELMo introduces the concept of contextualized word embeddings
- Learns different embeddings for words based on the context
- Looks at an entire sentence before assigning each word an embedding

A man with a beard and a dark cap, wearing a colorful patterned shirt, is on the left. Elmo, the red Muppet, is on the right. They are in a room with a blue wall and a window with blinds. Four speech bubbles are overlaid on the image, containing a conversation about word embeddings.

Hey ELMo, what's the embedding of the word "stick"?

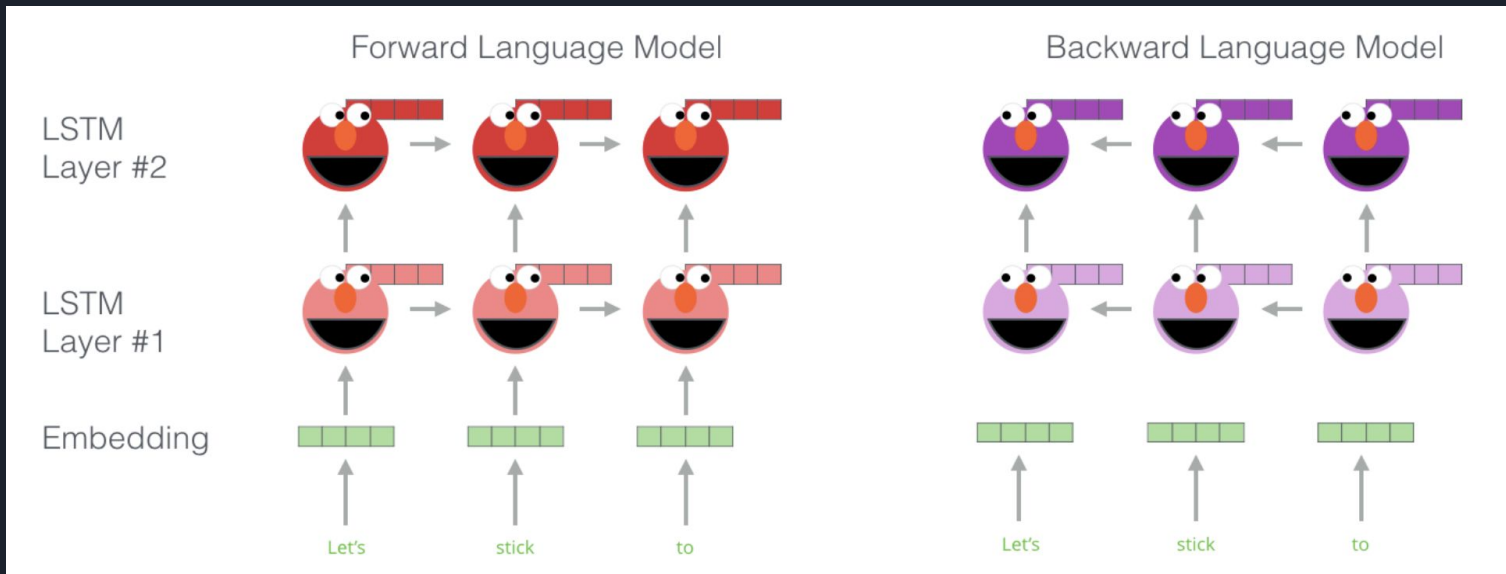
There are multiple possible embeddings! Use it in a sentence.

Oh, okay. Here:
"Let's stick to improvisation in this skit"

Oh in that case, the embedding is:
-0.02, -0.16, 0.12, -0.1etc

More on ELMo

- ELMo is a bidirectional language model
- Built using a network of Bidirectional LSTMs
- Instead of predicting the next word like in a traditional language model, it also predicts the previous word





Issues with ELMo

- ELMo theoretically achieve superb results on most NLP tasks, but is limited by its size and speed at prediction time
- ELMo is more than 25-50 times slower on the benchmarks than the model currently in production and requires at least four times more memory
- Training and optimizing ELMo on AWS caused a steep increase in costs
- All of these issues make it impossible to deploy ELMo in production



Bidirectional Encoder Representations from Transformers aka BERT

- Very large NLP model developed by Google
- Trained on massive datasets in multiple languages
- Builds on top of ELMo and other novel models (Transformer, ULM-Fit, ...)
- Broke several records on NLP tasks (sentence classification, question answering,...)



More Details on the Architecture of BERT

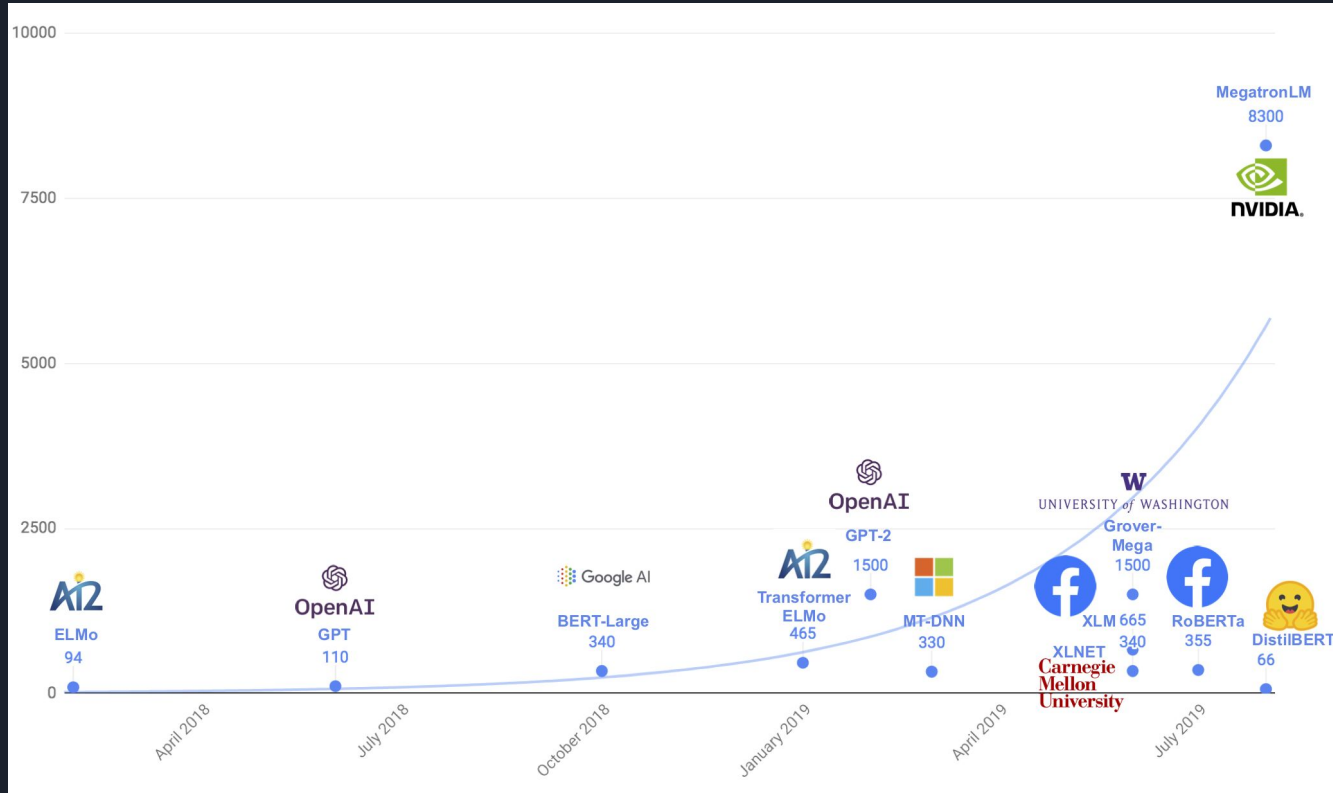
- Deep and complex architecture consisting of 12 or 24 encoding layers
- Implements a bidirectional masked language model
- Each encoding layer has very large feed forward networks and attention heads
- Encoding layers each learn a different representation for the original input
- Last layer of the model is a feed forward neural network that can be used for other NLP task



Issues with BERT

- BERT is even bigger and slower than ELMo
- BERT has its own tokenizer which is more complex than Keanet's tokenizer
- All of these issues make it impossible to deploy these models in production

Bidirectional Encoder Representations from Transformers



Knowledge Distillation





Knowledge Distillation Setup

- Transfer learning technique in machine learning
- Consists of training a very large model (the teacher) and transfer its learning to a smaller model (the student)
- Student model is trained to reproduce the behavior of its teacher
- Student uses pseudo labelled data
- Goal: Predict faster and consume less memory than teacher
- Find the right balance between capacity and performance

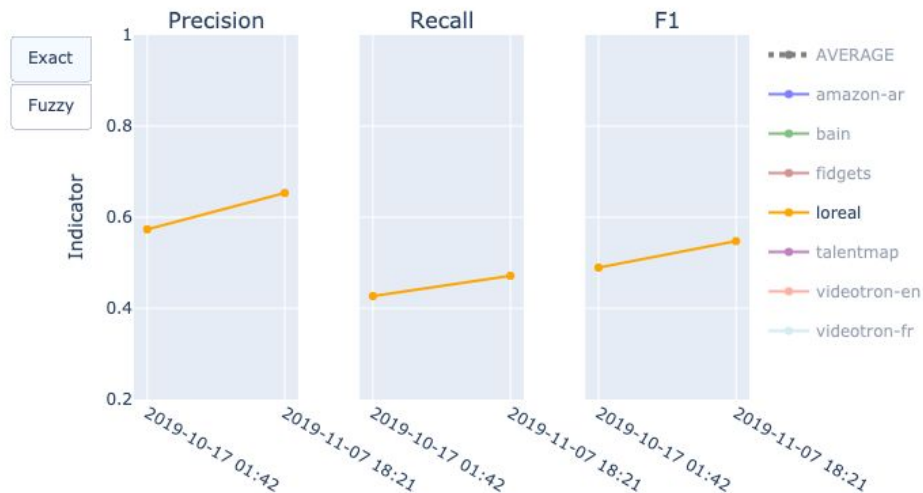


Experiment on L'Oréal Dataset

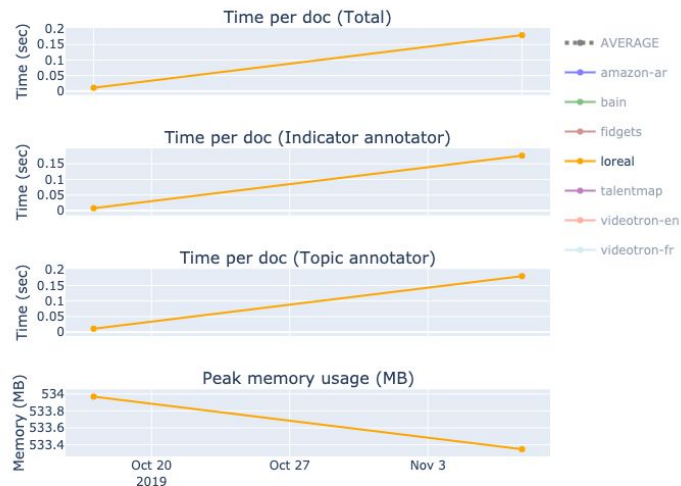
- Annotated train set, test set and non-annotated train set available
- Train and evaluate ELMo on L'Oréal annotated subsets
- Use ELMo teacher to assign pseudo labels to non annotated subset
- Train student model on the pseudo labelled dataset and evaluate on test set

Results Achieved with ELMo

Indicator accuracy overall (Exact)



Model performance

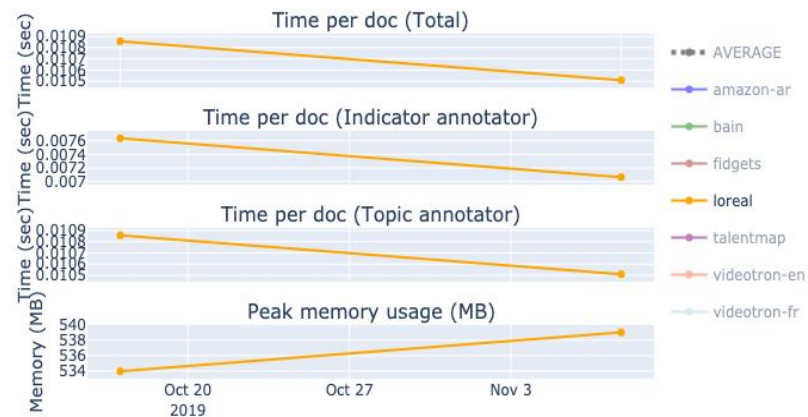


Student Model Results

Indicator accuracy overall (Exact)



Model performance



Summary of results

Model	F1	Precision	Recall	Inference Time per document (Seconds)	Peak Memory Usage (MB)
Baseline	0.4894	0.5734	0.4269	0.0076	533.97
ELMo	0.5476	0.6532	0.4714	0.18	533.35
Student Model	0.5503	0.6584	0.4727	0.0071	539.03



Experiment on Entire Data

- Training set[1]: Datasets from multiple companies (Bain, Videotron, L'Oréal, ...)
- Test set[1]: Keatext's test set + L'Oréal test set
- 2 teacher models
 - Teacher[1]: ELMo model trained on Training set[1] and evaluated on Test set[1]
 - Teacher[2]: ELMo model presented in previous section
- Use trained ELMo model to assign pseudo labels to non-annotated L'Oréal dataset
- 2 student models:
 - Student[1]: Trained using Training set[1] + pseudo labelled data from Teacher[1]
 - Student[2]: Trained using Training set[1] + pseudo labelled data from Teacher[2]
 - Both models are evaluated on Test set[1]
- Train student models with different size (3000, 5000, 10000) subsets of pseudo labelled dataset to decide how much pseudo labelled data should be used
- We used 2 teachers to understand if student benefits more from a teacher specialized on a single dataset or a generalist teacher.

Summary of results

Model	F1	Precision	Recall	Inference Time per document (Seconds)	Peak Memory Usage (MB)
Baseline	0.6939	0.7251	0.6692	0.0095	532.8343
ELMo (Teacher[1])	0.6815	0.7059	0.6631	0.1172	535.8856
Student Model[1]	0.7033	0.7507	0.6661	0.0158	532.8856
Student Model[2]	0.7095	0.7491	0.6789	0.0108	532.9593

Conclusions: Limitations and Future Work

Limitations

- ELMo and BERT
 - Source imbalance (More data from some companies than others)
 - Not a lot of labelled data is available
 - Tradeoff between performance, memory usage and processing time
- Knowledge Distillation
 - Need unlabelled data from source other than L'Oréal

Future Work

- Knowledge Distillation setup where student learns to predict teacher's probabilities or score instead of classes
- Knowledge Distillation with BERT teacher
- Knowledge Distillation where student learns both from BERT and ELMo teachers

QUESTIONS?