VL Deep Learning for Natural Language Processing

16. Contextual Word Embeddings

Prof. Dr. Ralf Krestel

AG Information Profiling and Retrieval

Semester Schedule

Week	1st Date	2nd Date	Hand-in at start of week	Hand-out at end of Week
1	Introduction	Python/Keras/Colab		
2	NLP	Neural Networks Recap		1. Assignment
3	Text Mining	Text Classification		
4	Word Embeddings I	Word Embeddings II		
5	Word Embeddings III	Assignment 1 Discussion	1. Assignment	
6	Named Entity Recognition	Language Models		2. Assignment
7	Recurrent Neural Networks	Holiday		
8	Long Short-Term Memory	Convolutional Neural Networks		
9	Deap Learning in Practice	Assignment 2 Discussion	2. Assignment	
10	Contextual Word Embeddings	Sequence-to-Sequence Models		3. Assignment
11	Transformer Models	Enhanced Language Models		
12	Natural Language Generation	Deep Reinforcement Learning		
13	No lecture	Assignment 3 Discussion	3. Assignment	

Importance of (Pre-Trained) Word Vectors

- In the beginnng: learning word embeddings as part of the final task
  - Not competitive!

- First efforts with pre-rraining
  - Collobert et al. 2011
  - Unsupervised training of WE using 850M tokens (Wikipedia + Reuters)
  - Training lasted 7 weeks

- Breakthrough of DL for NLP
  - The holy grail

Ansatz	POS (ACC)	CHUNK (F1)	NER (F1)	SRL (F1)
SOTA	97.24	94.29	89.31	77.92
Supervised NN	96.37	90.33	81.47	70.99
+ pre-training	97.20	93.63	88.67	74.15
+ hand-crafted	97.29	94.32	89.59	74.72

https://de.wikipedia.org/wiki/Heiliger_Gral

Collobert, Ronan, et al. "Natural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research* 12 (2011): 2493-2537.

Learning Word Representations

- **From Scratch**
  - As part of the target task
  - Word vectors are additional parameters of the overall model
- **Pre-training**
  - Word vectors are learnt independently of a particular task:
    1. From a large external, general coprus
    2. As a pre-processing step on the (domain-specific) training data
- **Fine-Tuning**
  - Initializing of word vectors with pre-trained embeddings and then continuing to learn on (domain-specific) training data
- **Retrofitting**
  - Learnt word vectors are improved/adapted using external (semantic) knowledge, e.g. WordNet similaritities

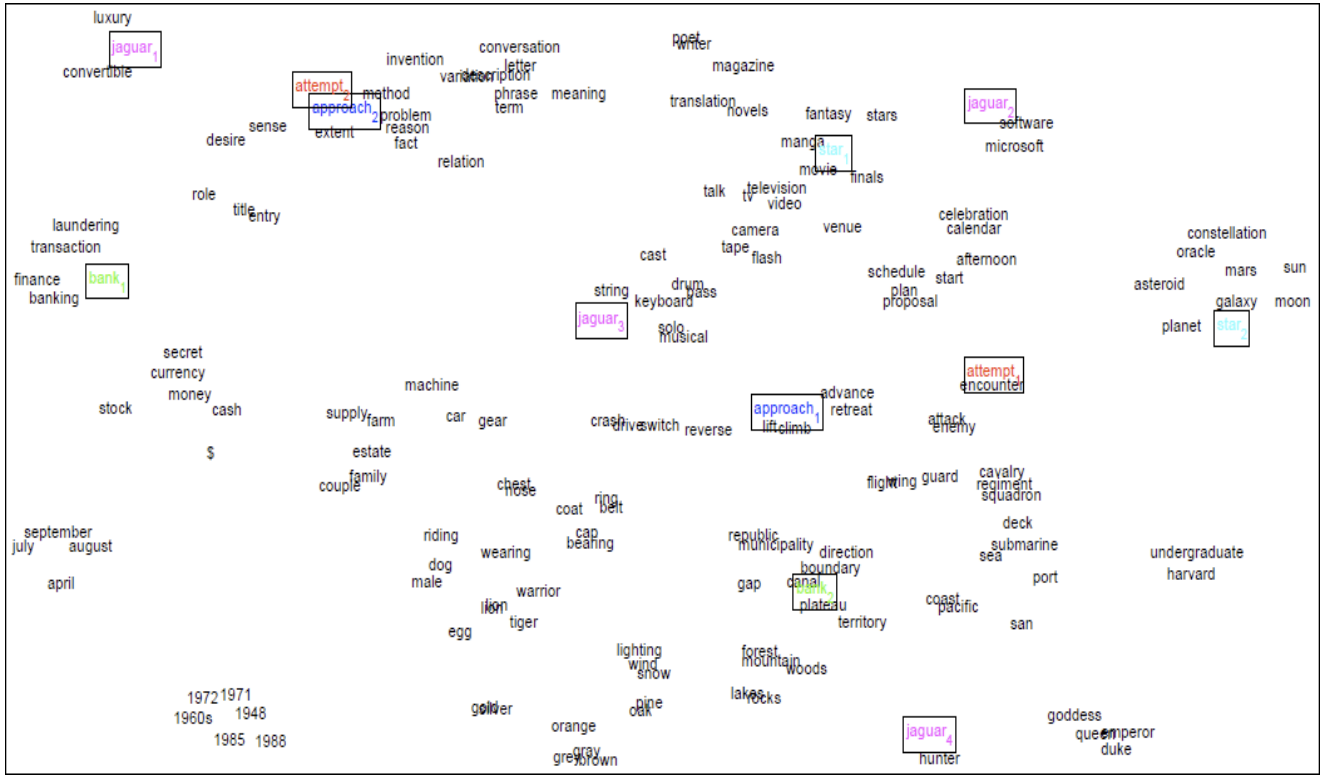
Faruqui, Manaal, et al. “Retrofitting word vectors to semantic lexicons.” In Proc. of NAACL, 2015.

Word Representations

- So far: **one** representation for **one** word:
  - Word vectors are trained with RNN or CNN
  - Word2vec, GloVe, fastText

- Two Problems:
  1. Words can be ambiguous
     - Always the same representation for a word (**type**) regardless of the context of its mention (**token**) is not enough (e.g., „Bank“, „Jaguar“,…)
       - We need word disambiguation

Vectors Learned from Local and Global Context



Word Representations

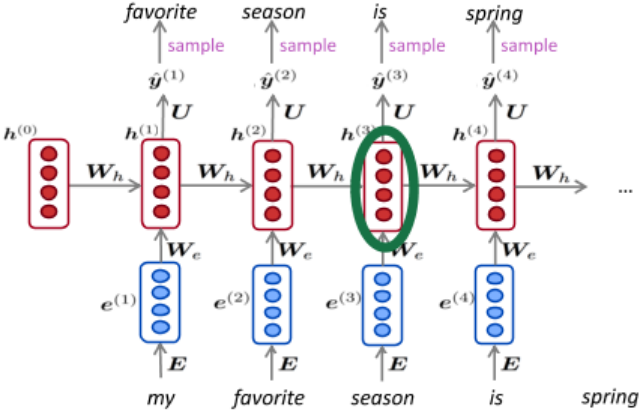
- So far: **one** representation for **one** word:
  - Word vectors are trained with RNN or CNN
  - Word2vec, GloVe, fastText

Bank_1, bank_2 -> different semantics
Arrival, arriving -> different syntactic, similar semantics
Restroom, toilet, shitter -> different connotation, same semantics

- Two Problems:
  1. Words can be ambiguous
     - Always the same representation for a word (**type**) regardless of the context of its mention (**token**)
     - One representation per type is not enough (e.g., „Bank“, „Jaguar“,…)
       - We need word disambiguation
     - But even then, subtle differences in meaning not captured (e.g., FED vs DB)
  2. Words have different aspects
     - Semantics, syntactic behaviour, register/connotation

Did We Already Compute Contextualized WEs?

- In an LSTM language model, we input word vectors (possibly trained on only the training data)
- The LSTM layer learns to predict the next word for each word
- These language models generate context-dependent word representations at each position!



Model	Source	Nearest Neighbor(s)
GloVe	**play**	playing, game, games, played, players, plays, player, Play, football, multiplayer
BiLM	Chico Ruiz made a spectacular **play** on Alusik 's grounder ...	Kieffer , the only junior in the group , was commended for his ability **to hit in the clutch**, as well as his all-round excellent **play**.
BiLM	Olivia De Havilland signed to do a Broadway **play** for Garson ...	... they were **actors** who had been handed fat **roles** in a successful **play** , and had talent enough to fill the roles competently, with nice understatement.

Peters, Matthew, et al. "Deep Contextualized Word Representations." *Proceedings of NAACL*. 2018.

ELMo & ULMFiT

- **Dai and Le (2015)**
  - Semi-supervised approach: train a NLM (instead of word vectors) on an unlabeled corpus; then use the pre-trained NLM for supervised sequence learning
- **Peters et al. (2017)**
  - Idea **TagLM**: We want to capture the meaning of a word in context. To this end, we (additionally) train a task-specific RNN on a small dataset labeled for a particular task, e.g., NER
- **Peters et al. (2018) ELMo**
- **Howard & Ruder (2018)** Universal Language Model Fine-tuning for Text Classification (**ULMFiT**)
  - Same general idea of transferring NLM knowledge (here applied to text classification)

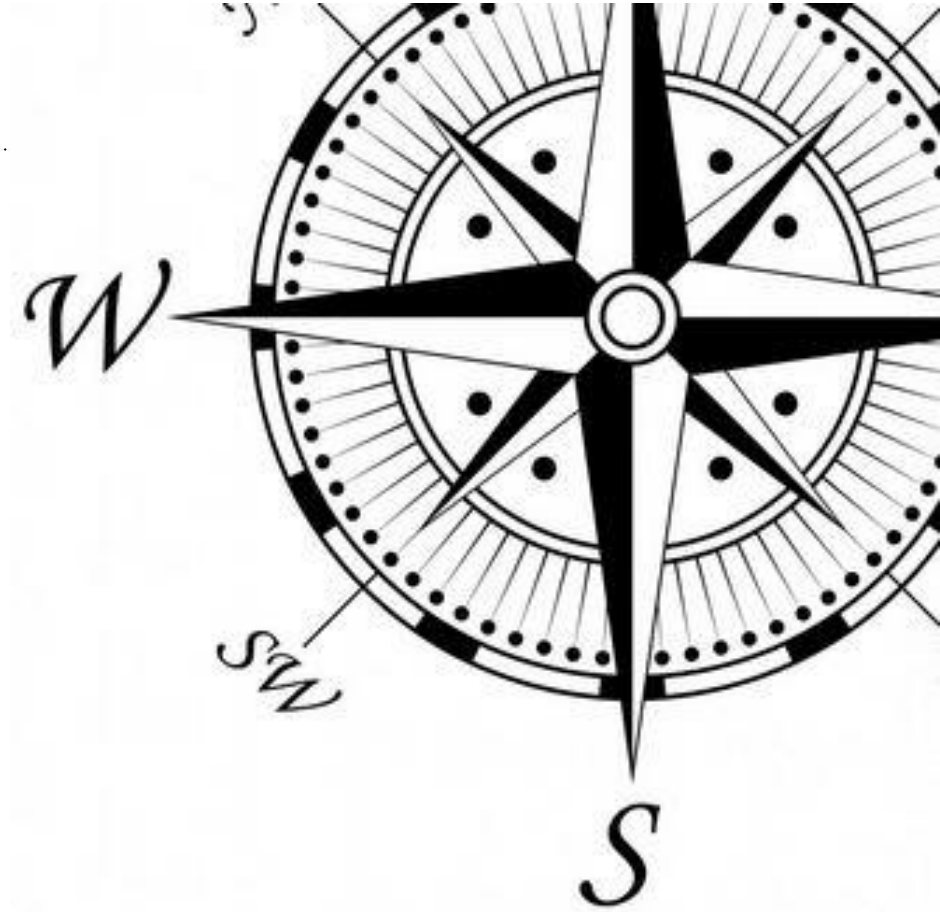
Lerning Goals for this Chapter

- Understand the idea of contextualized word embeddings
- Follow the development of contextualized word embeddings
  - Pre-ELMo
  - ELMo
  - ULMfit

- Relevant chapters:
  - S13 (2019): https://www.youtube.com/watch?v=S-CspeZ8FHc

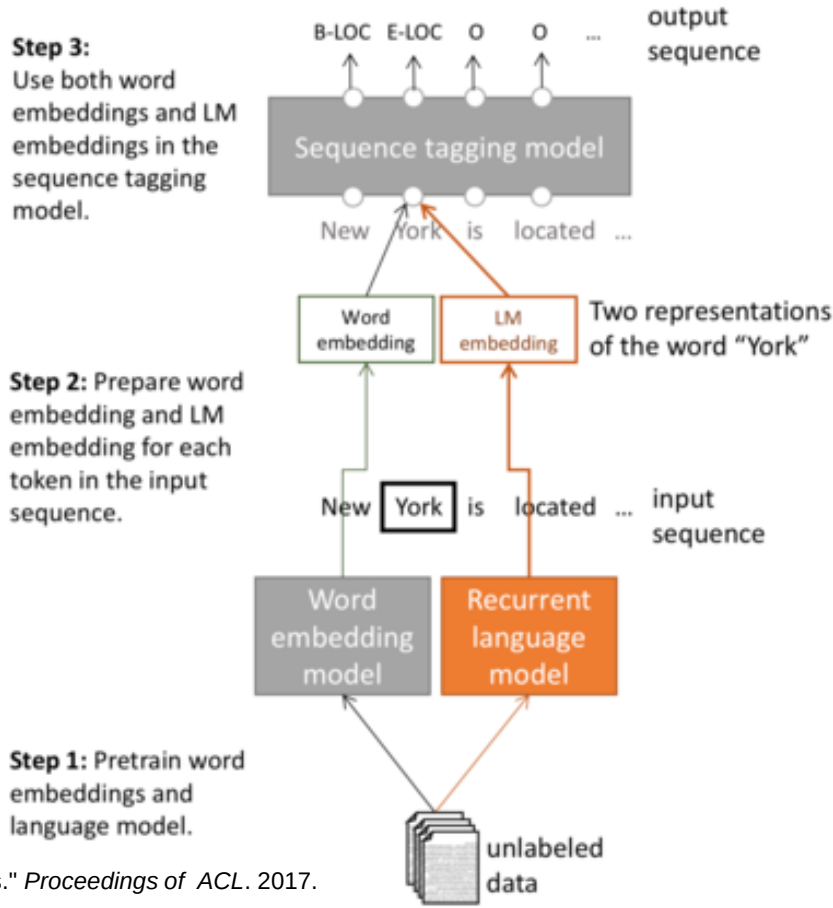
Topics Today

1. **TagLM**
2. ELMo
3. ULMFiT



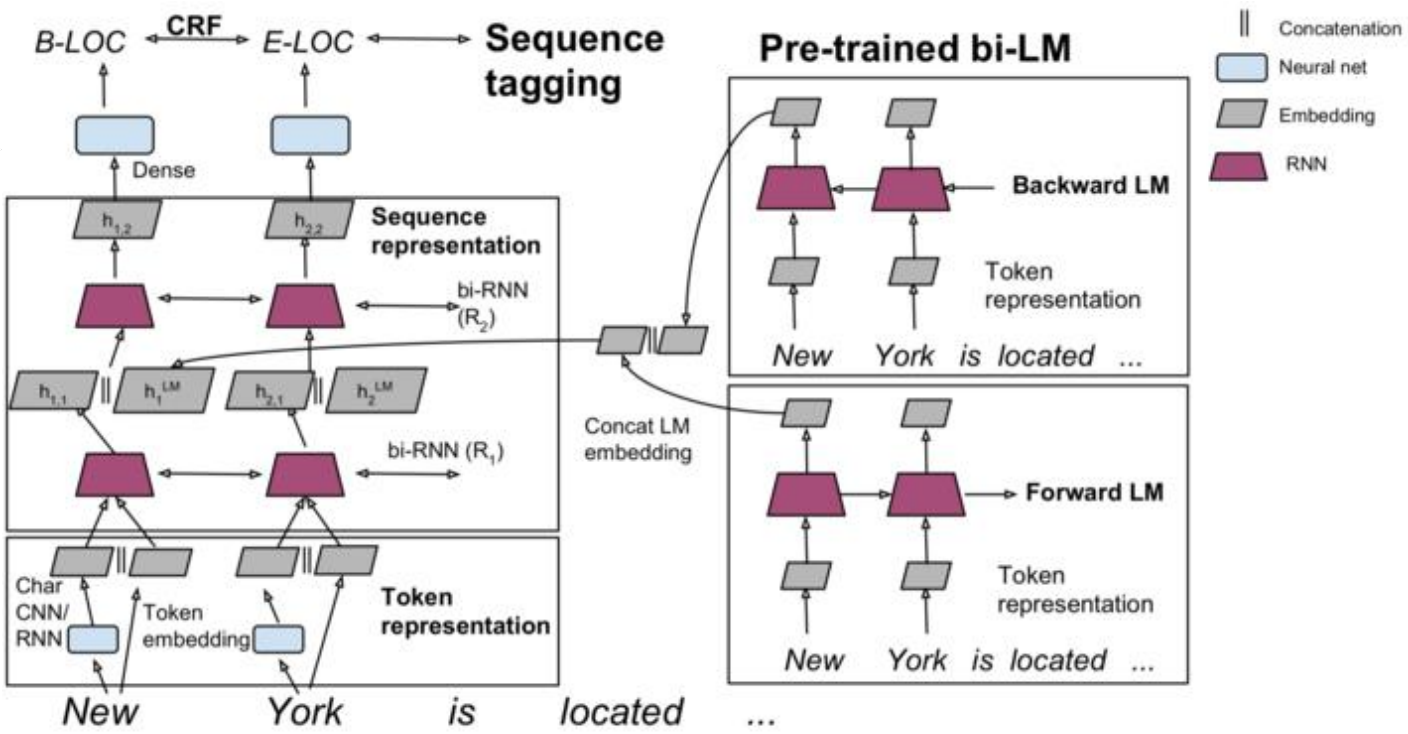
TagLM I

- Semi-supervised approach:
  - Step 1: unsupervised
  - Step 3: supervised, task-specific (here NER)



Peters, Matthew, et al. "Semi-supervised sequence tagging with bidirectional language models." *Proceedings of ACL*. 2017.

TagLM II



$$\mathbf{h}_{k,1} = [\overrightarrow{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

Peters, Matthew, et al. "Semi-supervised sequence tagging with bidirectional language models." *Proceedings of ACL*. 2017.

CoNLL 2003 NER (en news testb)

Name	Description	Year	F1
TagLM	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma & Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Ratinov & Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford	MEMM softmax markov model	2003	86.07

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

Peters et al. (2017): TagLM–“Pre-ELMo”

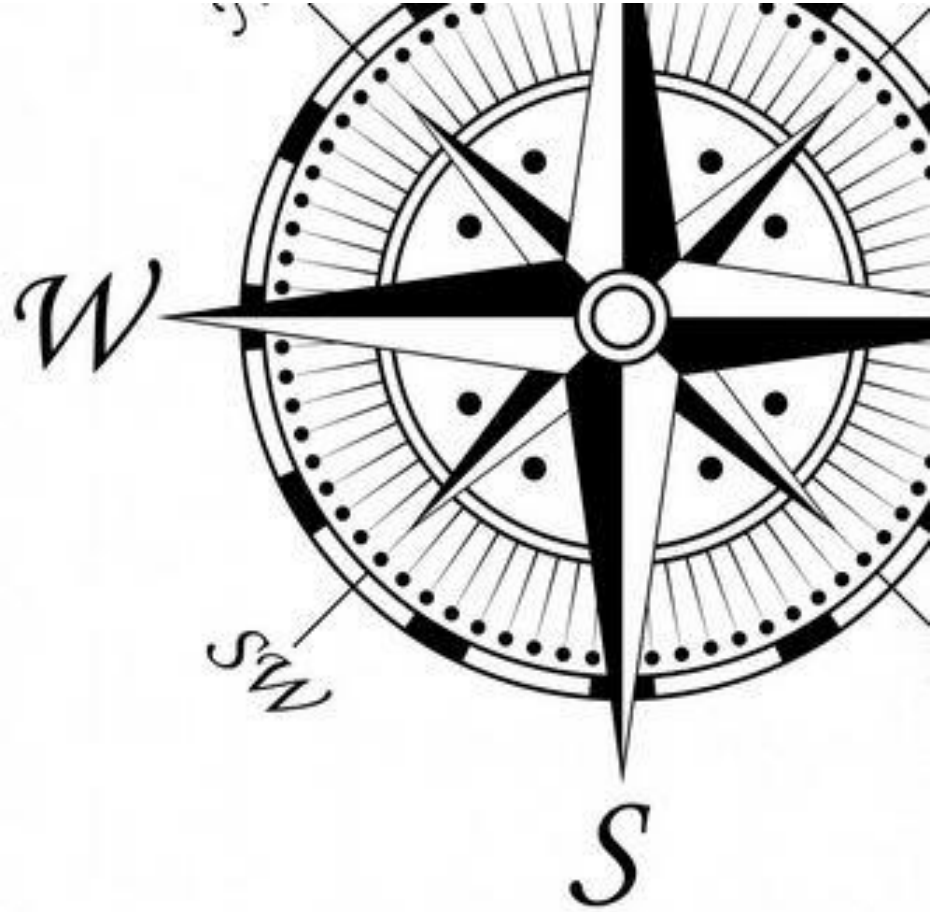
- Language model trained on 800 million training words of “Billion word benchmark”

- Language model observations
  - An LM trained on supervised data does not help
  - Having a bidirectional LM helps over only “forward”, by about 0.2
  - Having a huge LM design (perplexity=30) helps over a smaller model (perplexity=48) by about 0.3

- Task-specific BiLSTM observations
  - Using just the LM embeddings to predict isn’t great: 88.17 F1
  - Well below just using an BiLSTM tagger on labeled data

Topics Today

1. TagLM
2. **ELMo**
3. ULMFiT



ELMo: Embeddings from Language Models I

- Deep contextualized word representations
  - NAACL 2018
  - https://arxiv.org/abs/1802.05365

- Initial breakout version of **word token vectors** or **contextual word vectors**
  - Learn word token vectors using long contexts not context windows (here, whole sentence, could be longer)
  - Learn a deep Bi-NLM and use all its layers in prediction

https://pixy.org/783745/

ELMo: Embeddings from Language Models II

ELMo: Embeddings from Language Models III

- Train a bidirectional LM
- Aim at **performant** but not overly large LM:
  - Use 2 biLSTM layers
  - Use character CNN to build initial word representation (only)
    - 2048 char n-gram filters and 2 highway layers, 512 dim projection
  - Use 4096 dim hidden/cell LSTM states with 512 dim projections to next input
  - Use a residual connection
  - Tie parameters of token input and output (softmax) and tie these between forward and backward LMs

ELMo: Embeddings from Language Models IV

- ELMo learns task-specific combination of biLM layer representations
  - This is an innovation that improves on just using the top layer of the LSTM stack

$$
\begin{aligned}
R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\
&= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\},
\end{aligned}
$$

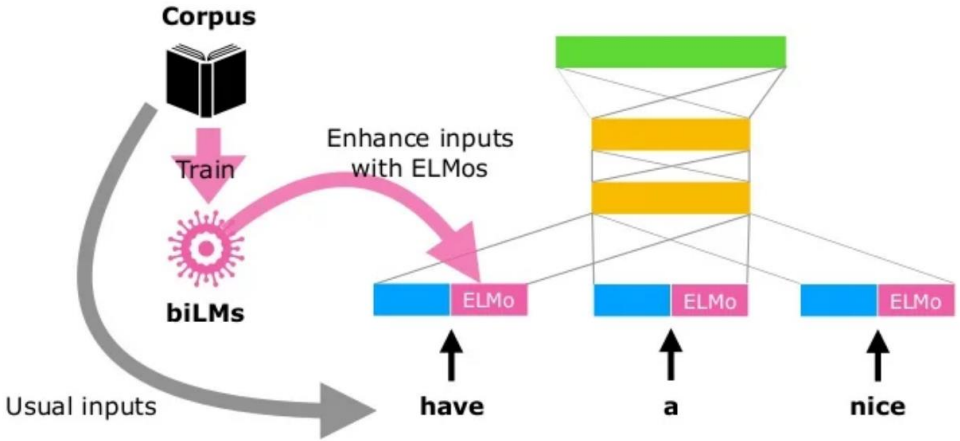
$$
\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}
$$

- L=#layers f the BiLM
- Each token gets 2*2+1 representations

- γ^{task} scales overall usefulness of ELMo to task;
- s_j^{task} are j softmax-normalized mixture model weights

ELMo: Use with a Task

- First run biLM to get representations for each word
- Then let (whatever) end-task model use them
  - Freeze weights of ELMo for purposes of supervised model
  - Concatenate ELMo weights into task-specific model



ELMo: Use with a Task

- Concatenate ELMo weights into task-specific model
  - Details depend on task
    - Concatenating into intermediate layer as for TagLM is typical
    - Can provide ELMo representations again when producing outputs, as in a question answering system

- Where to insert ELMo?

Task	Input Only	Input & Output	Output Only
SQuAD	85.1	**85.6**	84.8
SNLI	88.9	**89.5**	88.7
SRL	**84.7**	84.3	80.9

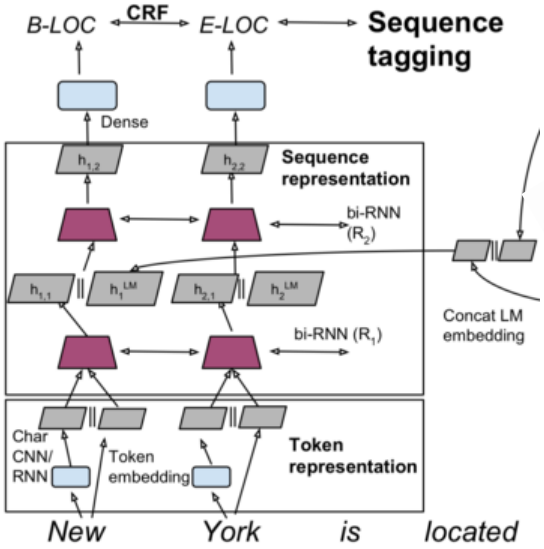
- Which layers of the biLM to use and how to weight them?

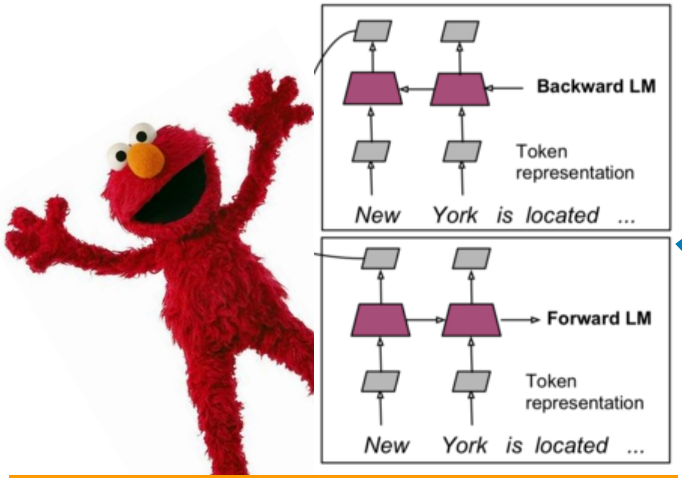
Reg. Parameter

Task	Baseline	Last Only	All layers λ=1	All layers λ=0.001
SQuAD	80.8	84.7	85.0	**85.2**
SNLI	88.1	89.1	89.3	**89.5**
SRL	81.6	84.1	84.6	**84.8**

ELMo as Part of an NER-Taggers

- Breakout version of **deep contextual word vectors**





Use learned, task-weighted average of 2 hidden layers

ELMo representation:
A deep bidirectional neural LM

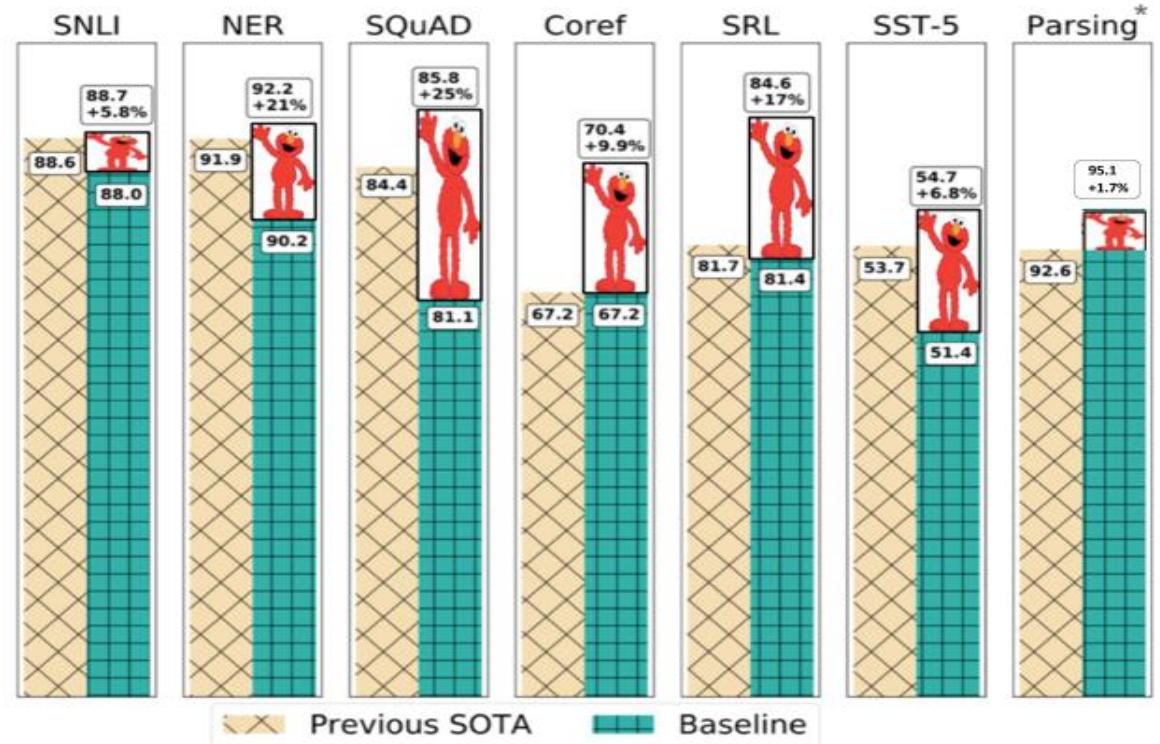
$$\mathbf{h}_{k,1} = [\overrightarrow{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

CoNLL 2003 NER (en news testb)

Name	Description	Year	F1
			
	https://www.zdf.de/nachrichten/panorama/sesamstrassenmonster-talkshow-elmo-100.html		
ELMo	BiLSTM-CRF + ELMo	2018	92.22
TagLM	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma & Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Ratinov & Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford	MEMM softmax markov model	2003	86.07

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

ELMo Results

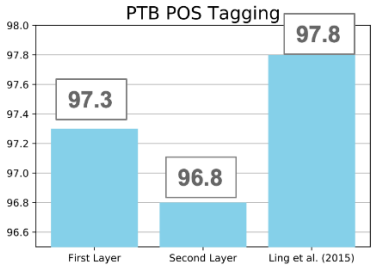


Baseline + ELMo

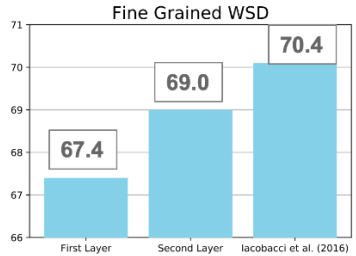
- SQuAD: question answering
- SNLI: textual entailment
- SRL: semantic role labeling
- Coref: coreference resolution
- NER: named entity recognition
- SST-5: sentiment analysis

ELMo: Weighting of Layers

- The two biLSTM NLM layers have differentiated uses/meanings
  - Lower layer is better for lower-level syntax, etc.
    - Part-of-speech tagging, syntactic dependencies, NER
  - Higher layer is better for higher-level semantics
    - Sentiment, Semantic role labeling, question answering, SNLI



First Layer > Second Layer



Second Layer > First Layer

- This seems interesting, but it'd seem more interesting to see how it pans out with more than two layers of network

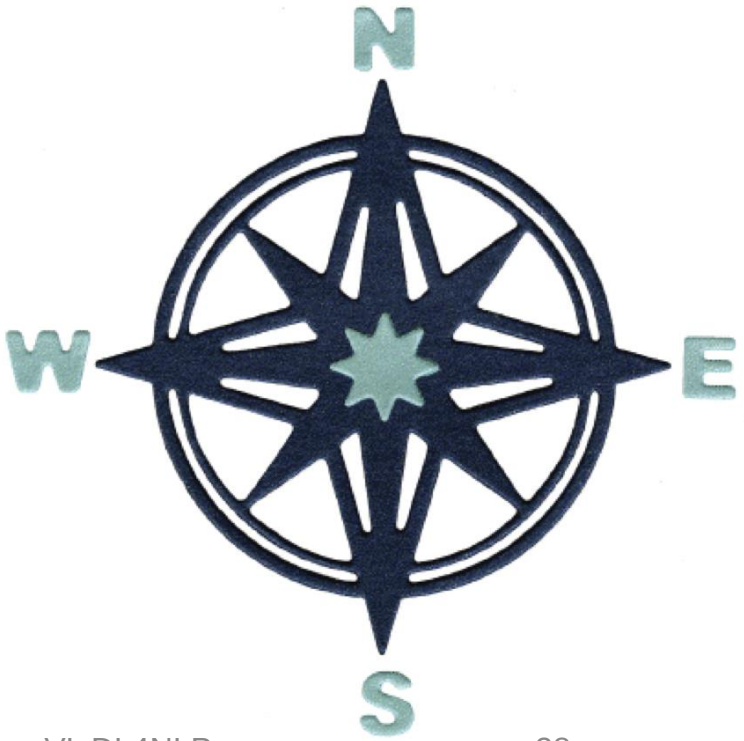
Exercise

- What are the advantages and disadvantages of contextualized vs. static word embeddings?

Start 5 4 3 2 1 End

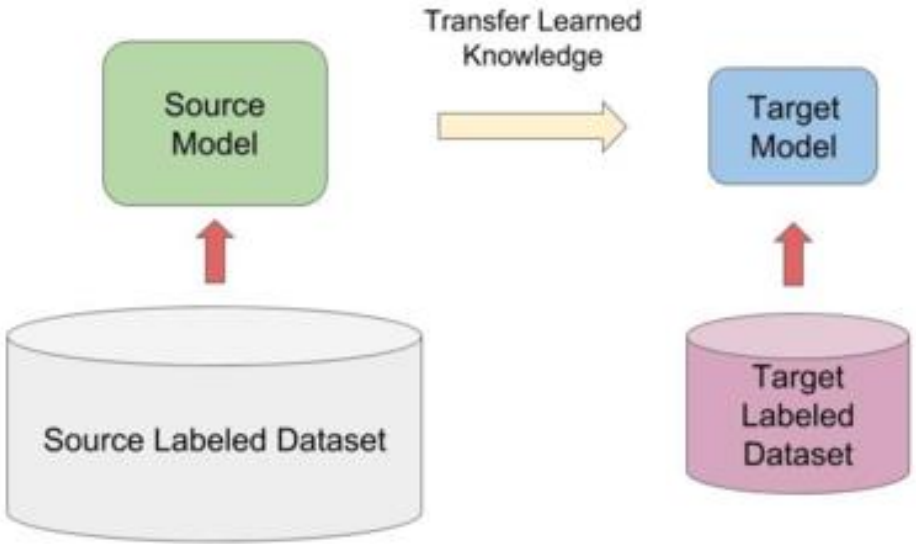
Überblick

- TagLM
- ELMo
- **ULMFiT**



Universal Language Model Fine-Tuning (ULMFiT)

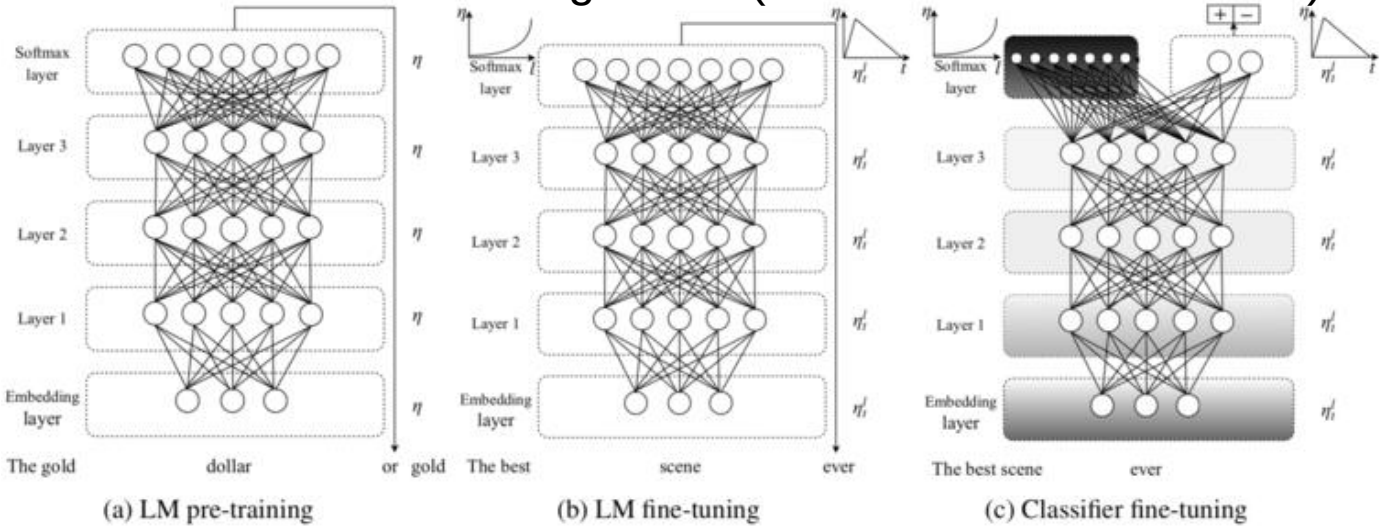
- Howard and Ruder (2018) Universal Language Model Fine-tuning for Text Classification
  - https://arxiv.org/pdf/1801.06146.pdf
  - Same general idea of transferring NLM knowledge
  - Here applied to text classification



ULMFiT II

a) Train LM on big general domain corpus (use biLM)

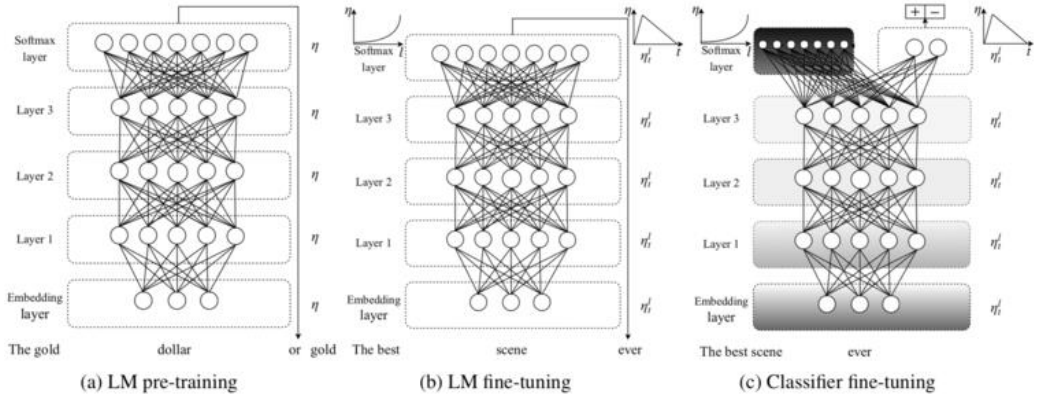
- Two-fold fine-tuning
  b) Tune LM on target task data
  c) Fine-tune as classifier on target task (LM or Softmax are frozen)



(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

ULMFiT Focus

- Use reasonable-size “1 GPU” language model
- A lot of care in LM fine-tuning
- Different per-layer learning rates
- Slanted triangular learning rate (STLR) schedule
- Gradual layer unfreezing and STLR when learning classifier
- Classify using concatenation $[h_T, maxpool(\boldsymbol{h}), meanpool(\boldsymbol{h})]$

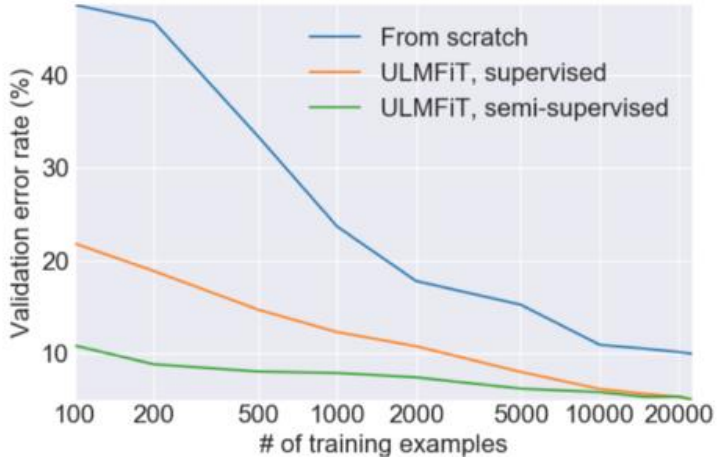


ULMFiT Performance

- Text classifier error rates

	Model	Test		Model	Test
IMDb	CoVe (McCann et al., 2017)	8.2	TREC-6	CoVe (McCann et al., 2017)	4.2
	oh-LSTM (Johnson and Zhang, 2016)	5.9		TBCNN (Mou et al., 2015)	4.0
	Virtual (Miyato et al., 2016)	5.9		LSTM-CNN (Zhou et al., 2016)	3.9
	ULMFiT (ours)	**4.6**		ULMFiT (ours)	**3.6**

- Transfer learning works!



Let’s Scale it Up!

It would cost **~$4,600,000** to train GPT-3 using he cheapest GPU cloud provider

ULMfit	GPT	BERT	GPT-2	GPT-3
Jan 2018	June 2018	Oct 2018	Feb 2019	Juni 2020
Training: 1 GPU day	Training 240 GPU days	Training 256 TPU days ~320–560 GPU days	Training ~2048 TPU v3 days	Training 355 years on a Tesla V100 GPU

fast.ai OpenAI Google AI OpenAI OpenAI

https://lambdalabs.com/blog/demystifying-gpt-3/

GPT-3 Example Applications

- Text generation
- Question answering
- Chatbot
- Information
- Text classif
- Translation
- Virtual assi
- …

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: What party did he belong to?
A: He belonged to the Republican Party.

Why it is important to be grate

The following is a list of companies and the categories they fall into and their most recent market cap

Facebook: 'Social media, Technology' '$683.95B'

LinkedIn: 'Social media, Technology, Enterprise, Careers' '$14.2B'

Uber: 'Transportation, Technology, Marketplace' '$56.3B'

Unilever: 'Conglomerate, Consumer Goods' '$144.2B'

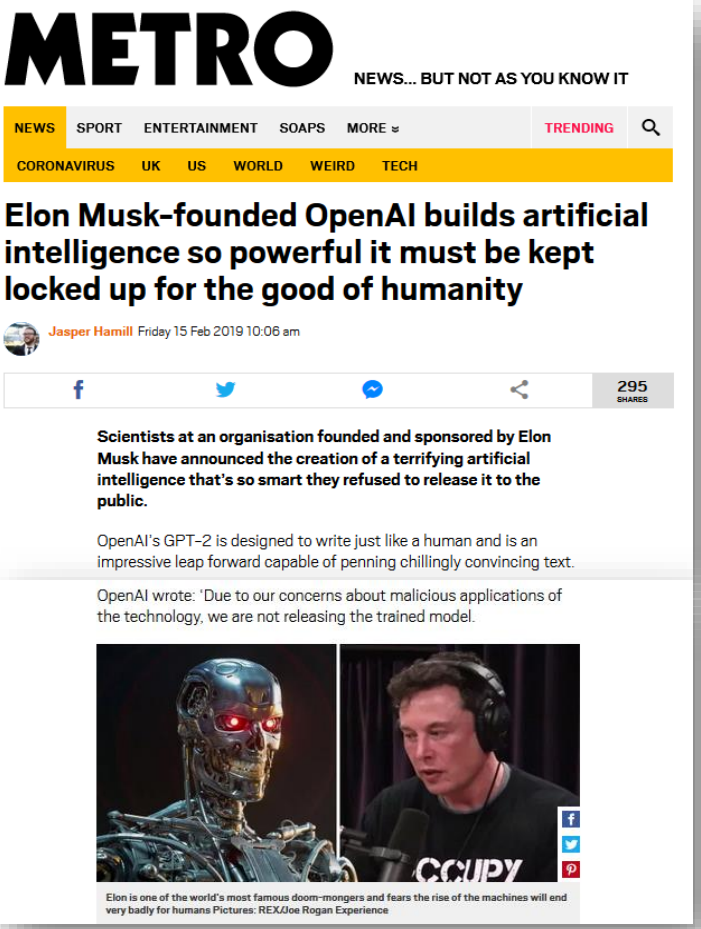
McDonald's: 'Fast Food, Restaurants' '$98.6B'

Google: 'Technology, Internet' '$542.8B'

Apple: 'Technology, Consumer Electronics' '$724.8B'

https://towardsdatascience.com/gpt3-the-dream-machine-in-real-world-c99592d4842f

GPT-3 in the News



https://metro.co.uk/2019/02/15/elon-musks-openai-builds-artificial-intelligence-powerful-must-kept-locked-good-humanity-8634379/

CoNLL 2003 NER (en news testb)

Name	Description	Year	F1
BERT Large	Transformer bidi LM + fine tune	2018	92.8
BERT Base	Transformer bidi LM + fine tune	2018	92.4
ELMo	BiLSTM-CRF + ELMo	2018	92.22
TagLM	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma & Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Ratinov & Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford	MEMM softmax markov model	2003	86.07

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

C

DeBERTa (Microsoft)	Optimized BERT + fine tune	2021	96.1
RoBERTa (Facebook)	Optimized BERT + fine tune	2021	95.7

Name	Description	Year	F1
LUKE	Entity-aware self attention	2020	94.3
ACE	Automated concat. of embeddings + doc context	2020	94.14
Strakova et al.	LSTM-CRF+ELMo+BERT+Flair	2019	93.38
Flair (Zalando)	Character-level language model	2018	93.09
BERT Large	Transformer bidi LM + fine tune	2018	92.8
BERT Base	Transformer bidi LM + fine tune	2018	92.4
ELMo	BiLSTM-CRF + ELMo	2018	92.22
TagLM	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma & Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Ratinov & Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford	MEMM softmax markov model	2003	86.07

https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003

Lerning Goals for this Chapter

- Understand the idea of contextualized word embeddings
- Follow the development of contextualized word embeddings
  - Pre-ELMo
  - ELMo
  - ULMfit

- Relevant chapters:
  - S13 (2019): https://www.youtube.com/watch?v=S-CspeZ8FHc

Literature

- Collobert, Ronan, et al. "Natural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research* 12 (2011): 2493-2537.
- Faruqui, Manaal, et al. “Retrofitting word vectors to semantic lexicons.” In Proc. of NAACL, 2015.
- Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning.“ In Proc. of NIPS, 2015.
- Peters, Matthew, et al. "Semi-supervised sequence tagging with bidirectional language models." In *Proc. of ACL,* 2017.
- Peters, Matthew, et al. "Deep Contextualized Word Representations." In *Proc. of NAACL*, 2018.
- Howard, Jeremy, and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification." In *Proc. of ACL,* 2018.