IN5550: Neural Methods in Natural Language Processing
Sub-lecture 11.3
Using contextualized embeddings

Andrey Kutuzov

University of Oslo

31 March 2022



Contents



1. Using contextualized embeddings
   - Practicalities

Using contextualized embeddings

Layers of contextualized embeddings reflect language tiers

For example, ELMo [Peters et al., 2018b]:

1. Representations at the layer of character embeddings (CNN): morphology;
2. Representations at the 1st LSTM layer: syntax;
3. Representations at the 2nd LSTM layer: semantics (including word senses).

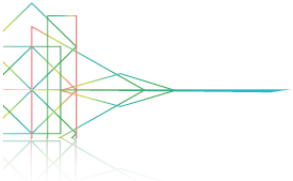
Using contextualized embeddings

Layers of contextualized embeddings reflect language tiers

For example, ELMo [Peters et al., 2018b]:

1. Representations at the layer of character embeddings (CNN): morphology;
2. Representations at the 1st LSTM layer: syntax;
3. Representations at the 2nd LSTM layer: semantics (including word senses).

BERT was shown to be like that as well (more about that later).



Method

ELMo represents a word t_k as a linear combination of corresponding hidden layers (inc. its embedding)

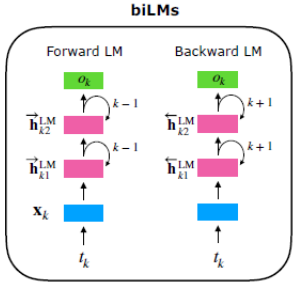
ELMo is a task specific representation. A down-stream task learns weighting parameters

$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}},\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}},\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \right.$$

$$\left. ([\mathbf{x}_k ; \mathbf{x}_k]) \right.$$

Concatenate hidden layers

$$[\overrightarrow{\mathbf{h}}_{kj}^{\text{LM}} ; \overleftarrow{\mathbf{h}}_{kj}^{\text{LM}}]$$



Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*

[Peters et al., 2018a]

Using contextualized embeddings

How one uses contextualized embeddings?

1. As 'feature extractors': pre-trained contextualized representations are fed to the target task (e.g., document classification)

Using contextualized embeddings

How one uses contextualized embeddings?

1. As 'feature extractors': pre-trained contextualized representations are fed to the target task (e.g., document classification)
   - conceptually, the same workflow as with 'static' word embeddings

Using contextualized embeddings

How one uses contextualized embeddings?

1. As 'feature extractors': pre-trained contextualized representations are fed to the target task (e.g., document classification)
   - conceptually, the same workflow as with 'static' word embeddings
2. Fine-tuning: the whole model undergoes additional training on the target task data

Using contextualized embeddings

How one uses contextualized embeddings?

1. As 'feature extractors': pre-trained contextualized representations are fed to the target task (e.g., document classification)
   - conceptually, the same workflow as with 'static' word embeddings
2. Fine-tuning: the whole model undergoes additional training on the target task data
   - Potentially more powerful, but computationally expensive. More on that later.

Using contextualized embeddings

How one uses contextualized embeddings?

1. As 'feature extractors': pre-trained contextualized representations are fed to the target task (e.g., document classification)
   - conceptually, the same workflow as with 'static' word embeddings
2. Fine-tuning: the whole model undergoes additional training on the target task data
   - Potentially more powerful, but computationally expensive. More on that later.

Conditions			Guidelines
Pretrain	Adapt.	Task	
Any	❄	Any	Add many task parameters
Any	🔥	Any	Add minimal task parameters ⚠ Hyper-parameters
Any	Any	Seq. / clas.	❄ and 🔥 have similar performance
ELMo	Any	Sent. pair	use ❄
BERT	Any	Sent. pair	use 🔥

Table 1: This paper's guidelines for using feature extraction (❄) and fine-tuning (🔥) with ELMo and BERT. Seq.: sequence labeling. Clas.: classification. Sent. pair: sentence pair tasks.

Recommendations from [Peters et al., 2019]

Using contextualized embeddings

Pre-trained models

- ELMo models for various languages at the NLPL vector repository:
  - `http://vectors.nlpl.eu/repository/`
- Transformer models are also available via *HuggingFace* library:
  - `https://huggingface.co/models`

Using contextualized embeddings

Pre-trained models

- ELMo models for various languages at the NLPL vector repository:
  - `http://vectors.nlpl.eu/repository/`
- Transformer models are also available via *HuggingFace* library:
  - `https://huggingface.co/models`

Contextualized models for Norwegian

- NorELMo_{30} (NLPL id 217) and NorELMo_{100} (NLPL id 218)
- NorBERT-1 (NLPL id 216) `https://huggingface.co/ltgoslo/norbert`
  - Trained on a corpus of 2 billion word tokens:
  - Norsk Aviskorpus (NAK); Bokmål and Nynorsk Wikipedias.
  - Described in [Kutuzov et al., 2021]

Using contextualized embeddings

Pre-trained models

- ELMo models for various languages at the NLPL vector repository:
  - `http://vectors.nlpl.eu/repository/`
- Transformer models are also available via *HuggingFace* library:
  - `https://huggingface.co/models`

Contextualized models for Norwegian

- NorELMo_{30} (NLPL id 217) and NorELMo_{100} (NLPL id 218)
- NorBERT-1 (NLPL id 216) `https://huggingface.co/ltgoslo/norbert`
  - Trained on a corpus of 2 billion word tokens:
  - Norsk Aviskorpus (NAK); Bokmål and Nynorsk Wikipedias.
  - Described in [Kutuzov et al., 2021]
- NorBERT-2 (NLPL id 221) `https://huggingface.co/ltgoslo/norbert2`
  - Trained on a corpus of 15 billion word tokens:
  - C4 web-crawled data for Norwegian [Xue et al., 2021]; Norwegian Colossal Corpus (NCC) [Kummervold et al., 2021].

Using contextualized embeddings



LTG Oslo
@ltgoslo

Today LTG Oslo releases large-scale language models for Norwegian
This includes NorBERT, the 1st fully functional Norwegian BERT, trained on Wikipedia and news (Norsk Aviskorpus). Also available via #HuggingFace as ltgoslo/norbert

Details and download:
norlm.nlpl.eu

2:08 pm · 13 Jan 2021 · Twitter Web App

LTG Oslo
@ltgoslo

We released a new BERT model for Norwegian: NorBERT 2, trained on a corpus an order of magnitude larger than the one used for NorBERT 1. It features a larger vocabulary and whole word masking. NorBERT 2 excels in binary sentiment analysis

See more at wiki.nlpl.eu/Vectors/norlm/...

Model/task	mBERT	NorBERT
mål (accuracy) code	98.0	98.5
norsk (accuracy) code	97.9	98.0
alysis (Mean Targeted F1 across 5 runs) code	34.9	**35.0**
F1 score) code	67.7	77.1
lokmål (F1 score) code	78.8	85.5
lynorsk (F1 score) code	81.7	82.8

4:29 PM · Feb 11, 2022 · Twitter Web App

Using contextualized embeddings

NorBERT models were trained from scratch on Norwegian data

- Important: a custom WordPiece vocabulary (case-preserving, accented characters)
- *'Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft.'*

Using contextualized embeddings

NorBERT models were trained from scratch on Norwegian data

- Important: a custom WordPiece vocabulary (case-preserving, accented characters)
- *'Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft.'*
  - Google's Multilingual BERT: 'Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft .'

Using contextualized embeddings

NorBERT models were trained from scratch on Norwegian data

- Important: a custom WordPiece vocabulary (case-preserving, accented characters)
- *'Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft.'*
  - Google's Multilingual BERT: 'Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft .'
  - NorBERT-1: 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterl ##engt ##et løft .'
  - NorBERT-2: 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterleng ##tet løft .'

Using contextualized embeddings

NorBERT models were trained from scratch on Norwegian data

- Important: a custom WordPiece vocabulary (case-preserving, accented characters)
- *'Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft.'*
  - Google's Multilingual BERT: 'Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft .'
  - NorBERT-1: 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterl ##engt ##et løft .'
  - NorBERT-2: 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterleng ##tet løft .'
- Trained on Saga and Fox HPC clusters for about a month each.
- 16 NVIDIA P100 GPUs for NorBERT-1, 4 NVIDIA A100 GPUs for NorBERT-2

Using contextualized embeddings

BERT or ELMo?

Both are good. For many tasks, BERT outperforms ELMo, but is much heavier computationally.

- Let's apply both to the NoREC dataset of sentence-level binary sentiment classification for Norwegian [Øvrelid et al., 2020].
- Both BERT and ELMo are pre-trained on exactly the same corpus.

Using contextualized embeddings

BERT or ELMo?

Both are good. For many tasks, BERT outperforms ELMo, but is much heavier computationally.

- Let's apply both to the NoREC dataset of sentence-level binary sentiment classification for Norwegian [Øvrelid et al., 2020].
- Both BERT and ELMo are pre-trained on exactly the same corpus.

Model	**NorBERT-1**	**NorELMo$_{30}$**
F1 score	**77.1**	75.0
Number of parameters (weights)	110 mln	**57 mln**
Model size	450 Mbytes	**220 Mbytes**
Time to adapt	35 sec	**5 sec**

Using contextualized embeddings

Code

- Python library for using pre-trained ELMo: `https://pypi.org/project/simple-elmo/`

Using contextualized embeddings

Code

- Python library for using pre-trained ELMo: `https://pypi.org/project/simple-elmo/`
- Python library for using pre-trained BERT: `https://pypi.org/project/transformers/`

Using contextualized embeddings

Code

- Python library for using pre-trained ELMo: `https://pypi.org/project/simple-elmo/`
- Python library for using pre-trained BERT: `https://pypi.org/project/transformers/`
- Code for training ELMo: `https://github.com/ltgoslo/simple_elmo_training`

Using contextualized embeddings

Code

- Python library for using pre-trained ELMo: `https://pypi.org/project/simple-elmo/`
- Python library for using pre-trained BERT: `https://pypi.org/project/transformers/`
- Code for training ELMo: `https://github.com/ltgoslo/simple_elmo_training`
- Original BERT training code: `https://github.com/google-research/bert`
  - Takes about 24 hours to train one ELMo epoch on 1 billion words using two NVIDIA P100 GPUs.
  - For BERT, the same training data requires about 3-4 weeks with the same hardware.

Using contextualized embeddings

Code

- Python library for using pre-trained ELMo: `https://pypi.org/project/simple-elmo/`
- Python library for using pre-trained BERT: `https://pypi.org/project/transformers/`
- Code for training ELMo: `https://github.com/ltgoslo/simple_elmo_training`
- Original BERT training code: `https://github.com/google-research/bert`
  - Takes about 24 hours to train one ELMo epoch on 1 billion words using two NVIDIA P100 GPUs.
  - For BERT, the same training data requires about 3-4 weeks with the same hardware.
- Hmm, they said Transformers are better paralellized, and thus are faster than RNNs, right...?
- Things do not always end up as expected :-)

References I

Kummervold, P. E., De la Rosa, J., Wetjen, F., and Brygfjeld, S. A. (2021).
Operationalizing a national digital library: The case for a Norwegian transformer model.
In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Kutuzov, A., Barnes, J., Velldal, E., Øvrelid, L., and Oepen, S. (2021).
Large-scale contextualised language modelling for Norwegian.
In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 30–40, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Øvrelid, L., Mæhlum, P., Barnes, J., and Velldal, E. (2020).
A fine-grained sentiment dataset for Norwegian.
In Proceedings of the 12th Language Resources and Evaluation Conference, pages 5025–5033, Marseille, France. European Language Resources Association.

References II

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018a).
Deep contextualized word representations.
In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Zettlemoyer, L., and Yih, W.-t. (2018b).
Dissecting contextual word embeddings: Architecture and representation.
In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

References III


Peters, M. E., Ruder, S., and Smith, N. A. (2019).
To tune or not to tune? adapting pretrained representations to diverse tasks.
In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 7–14, Florence, Italy. Association for Computational Linguistics.


Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021).
mT5: A massively multilingual pre-trained text-to-text transformer.
In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.