Adapting BERT to different tasks and languages.

Alexey Sorokin[1,2]

[2]Moscow Institute of Science and Technology, [1]Moscow State University

Huawei Machine Learning Conference,
Nizhny Novgorod, December, 18th, 2019

Introduction

- Most NLP tasks require labeled data.
- For some tasks the dataset can be quite large for reasonable performance:
  - Machine translation.
  - Grammar error correction.
  - Dialogue generation.

Introduction

- Most NLP tasks require labeled data.
- For some tasks the dataset can be quite large for reasonable performance:
  - Machine translation.
  - Grammar error correction.
  - Dialogue generation.
- Often annotation is time- and knowledge-consuming:
  - Morphological tagging and syntactic parsing.
  - Semantic role labeling.
- For some tasks data generation helps.
- But in general, NLP tasks often suffer from the lack of data.

Introduction: word embeddings

- For most tasks the model requires general knowledge about the language, which can be extracted from other tasks.
- This tasks must not require annotation.
- An ideal task — language modeling — the prediction of missing words:

This book is extremely ? to read.

Introduction: word embeddings

- For most tasks the model requires general knowledge about the language, which can be extracted from other tasks.
- This tasks must not require annotation.
- An ideal task — language modeling — the prediction of missing words:

This book is extremely ? to read.

- Classical approach: word2vec (Mikolov, 2013) — predicts a missed word using the words around it.
- Similar predictions $\rightarrow$ similar word vectors.

Introduction: word embeddings

- For most tasks the model requires general knowledge about the language, which can be extracted from other tasks.
- This tasks must not require annotation.
- An ideal task — language modeling — the prediction of missing words:

This book is extremely ? to read.

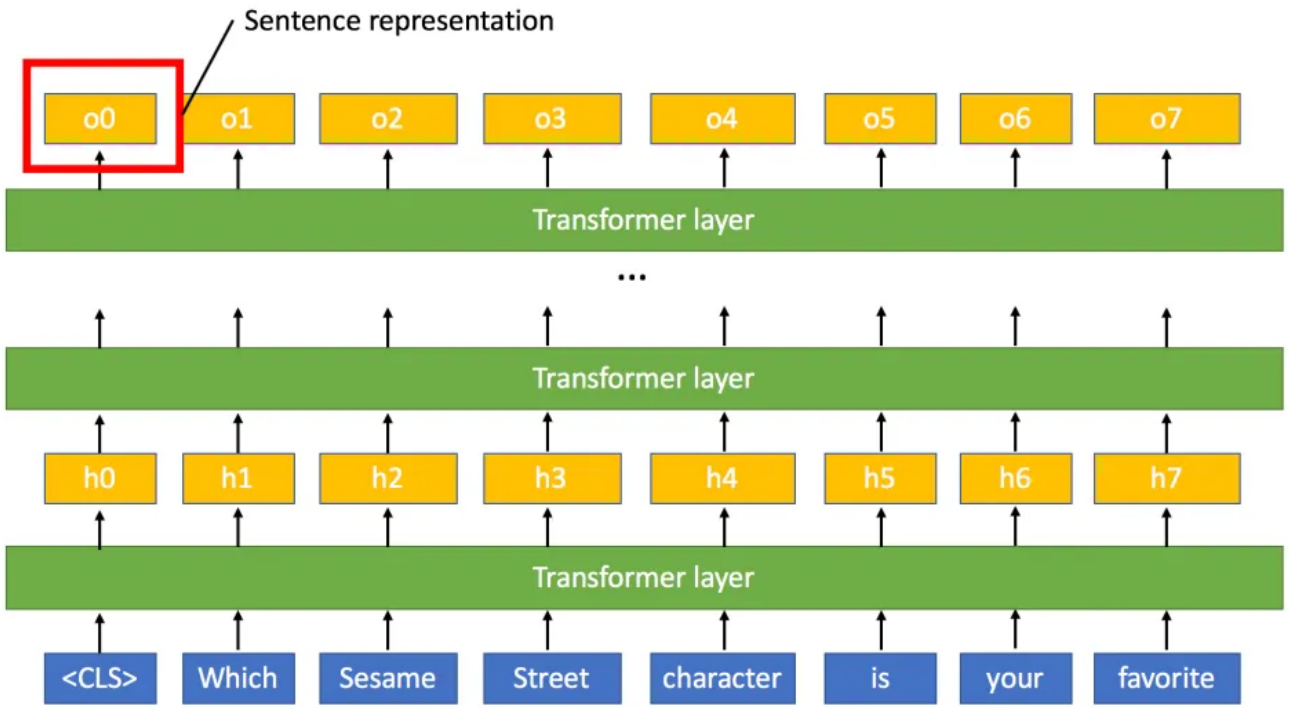
- Classical approach: word2vec (Mikolov, 2013) — predicts a missed word using the words around it.
- Similar predictions $\rightarrow$ similar word vectors.
- Drawbacks of word2vec:
  - No word order information.
  - No subword information (fixed in FastText, Bojanovski et al., 2016).

Pretraining and fine-tuning

- Many modern approaches include two phases: pretraining and fine-tuning.
- Pretraining: train a network on language modeling.
- Fine-tuning:
  - Change the output layer of the network to solve the current task.
  - Train the output layer (possibly with fine-tuning the remaining network).
- Main examples:
  - ELMo (Peters et al., 2018): training on language modeling (in both directions), fine-tuning on the task in question.
  - GPT (Radford et al., 2019): training on next word prediction, fine-tuning on supervised tasks (natural language inference, sentence similarity, etc.)

BERT

- BERT (Devlin, 2018) — another method for bidirectional language model pretraining.
- The network is based on Transformer-like (Vaswani, 2017) self-attention.



BERT training details

- Words in BERT are represented as sequence of subtokens.
- Subtokens dictionary is obtained uisng Byte-Pair encoding
- BERT is trained on 2 tasks:
  - Missing subtoken restoration.
  - Next sentence classification (does a pair of sentences contain two consecutive sentences).
- (Devlin, 2019) released two BERT models:
  - English (dictionary size 30K)
  - Multilingual (dictionary size 119K) – trained on the concatenation of 103 Wikipedias.

Multilinguality

- BERT knows something about all 103 langauges it was trained on.
- The quality of this knowledge differs from language to language (Cloze task results from [Rönnqvist et al., 2019]):

	Mono	Multi
English	**45.92**	33.94
German	**43.93**	28.10
Swedish		22.30
Finnish		14.56
Danish		25.07
Norwegian (Bokmål)		25.21
Norwegian (Nynorsk)		22.28

- We should fine-tune BERT to a particular language.

BERT tokenization

- Multilingual BERT tokenization is trained on 103 Wikipedias.
- Some cyrillic subtokens do not correspong to Russian words:
  - *року*,
  - *годині*,
  - *була*,
  - *##лар*,
- Word tokenization with these tokens violates the structure of the words.

Mikhail Arkhipov, Yury Kuratov
Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language
Dialogue 2019

BERT adaptation

- Two stages of BERT adaptation:
  - vocabulary recalculation.
  - subtoken embeddings fine-tuning
- Vocabulary is recalculated on language-specific data.

BERT adaptation

- Two stages of BERT adaptation:
  - vocabulary recalculation.
  - subtoken embeddings fine-tuning
- Vocabulary is recalculated on language-specific data.
- Token embeddings are initialized by Multilingual BERT representations (average of its subtokens).
- Other layers – also multilingual initialization.
- Further training on language-specific corpus.

Results on Russian

model	F-1	Accuracy
Neural networks [11]	79.82	76.65
Classifier + linguistic features [11]	81.10	77.39
Machine Translation + Semantic similarity [6]	78.51	81.41
BERT multilingual	85.48 ± 0.19	81.66 ± 0.38
RuBERT	87.73 ± 0.26	84.99 ± 0.35

Table 1: ParaPhraser. We compare BERT based models with models in non-standard run setting, when all resources were allowed.

model	F-1 (dev)	EM (dev)
R-Net from DeepPavlov [2]	80.04	60.62
BERT multilingual	83.39 ± 0.08	64.35 ± 0.39
RuBERT	84.60 ± 0.11	66.30 ± 0.24

Table 3: Results on question answering with SDSJ Task B. Models performance was evaluated on development set (public leaderboard subset).

BERT for several languages

- Fine-tuning BERT for one languages, we loose its multilinguality.
- What if we need a model for several related languages?
- You can fine-tune on them simultaneously!

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, Alexey Sorokin, Tuning Multilingual Transformers for Named Entity Recognition on Slavic languages, BSNLP 2019

Model	Span F_1	RPM	REM	SM
Bi-LSTM-CRF (Lample et al., 2016)	75.8	73.9	72.1	72.3
Multilingual BERT [5]	79.6	77.8	76.1	77.2
Multilingual BERT-CRF	81.4	80.9	79.2	79.6
Slavic BERT	83.5	83.8	82.0	82.2
Slavic BERT-CRF	87.9	85.7 (90.9)	84.3 (86.4)	84.1 (85.7)

Table 1: Metrics for BSNLP on validation set (Asia Bibi documents). Metrics on the test set are in the brackets.

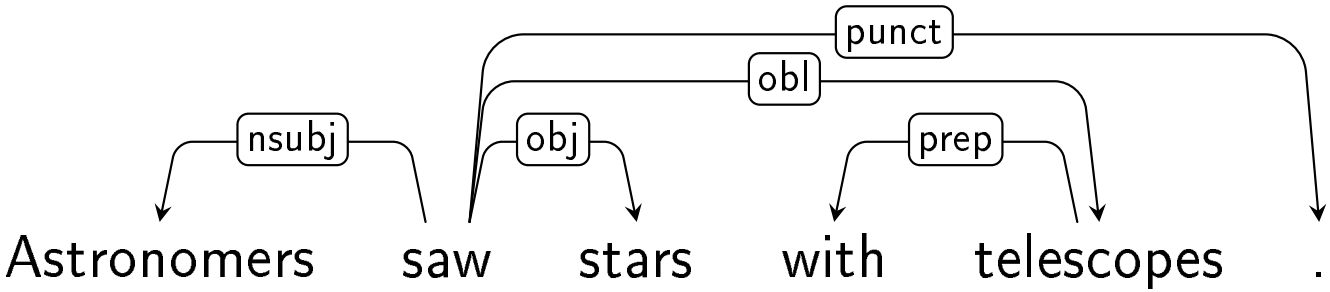
Morphological tagging

Его	DET
решение	NOUN, case=Nom, gender=Neut, number=Sing
задачи	NOUN, case=Gen, gender=Fem, number=Sing
было	AUX, mood=Ind, tense=Past, aspect=Imp
	gender=Neut, number=Sing
неправильным	ADJ, case=Ins, gender=Neut, number=Sing

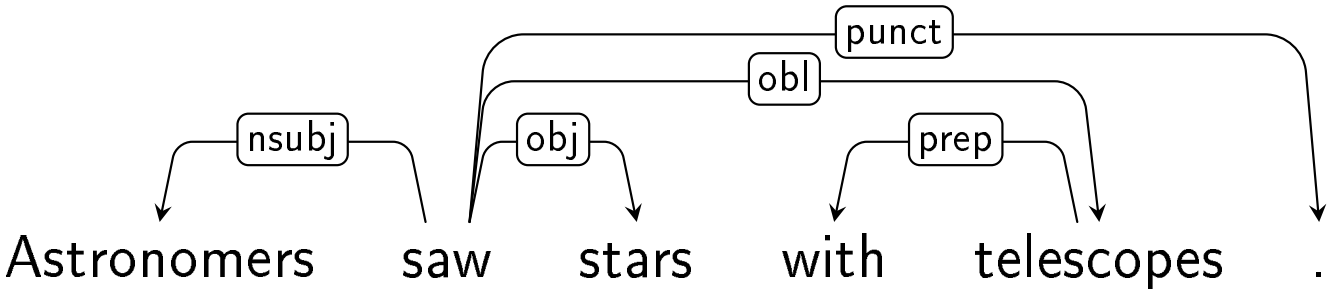
Applications:

- Downstream text processing: syntactic parsing, named entity recognition.
- Features for more complex NLP tasks.
- Automatic corpora annotation.

Syntactic parsing



Syntactic parsing



Astronomers	1	subj
saw	0	root
stars	1	obj
with	5	prep
telescopes	2	obl
.	2	punct

Syntactic parsing

- Universal Dependencies corpora (universaldependencies.org).
- 10 column format:

```
# sent_id = answers-20111108103211AA4XhnU_ans-0017
# text = Everything else should be good.
1   Everything  everything  PRON    NN  Number=Sing  5  nsubj  _ _
2   else        else        ADJ     JJ  Degree=Pos   1  amod   _ _
3   should      should      AUX     MD  VerbForm=Fin 5  root   _ _
4   be          be          AUX     VB  VerbForm=Inf 5  cop    _ _
5   good        good        ADJ     JJ  Degree=Pos   0  root   _ _
6   .           .           PUNCT   .   _            5  punct  _ _
```

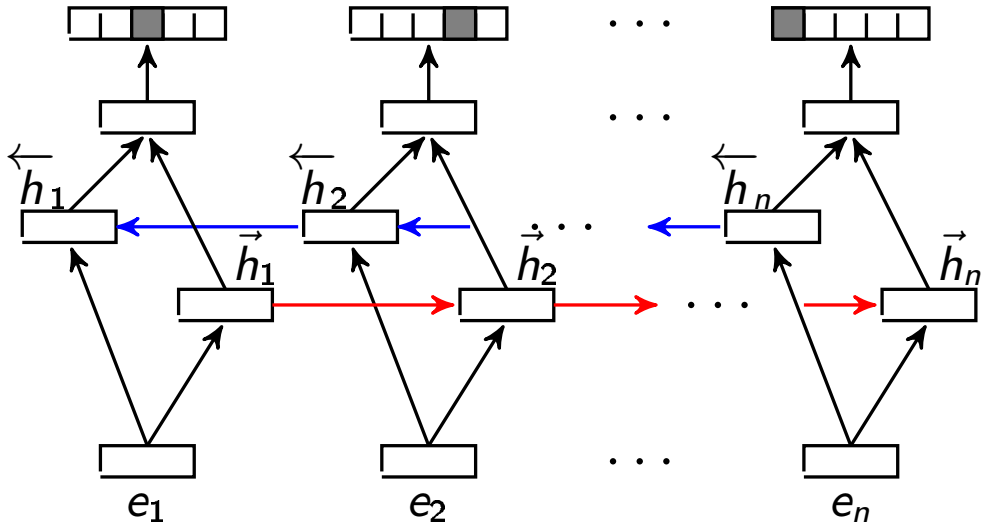
- More than 80 corpora for $>$ 50 languages.

Metrics

- Morphological tagging: tag accuracy.
- Syntactic parsing: LAS (labeled attachment score).
- Syntactic parsing: UAS (unlabeled attachment score).

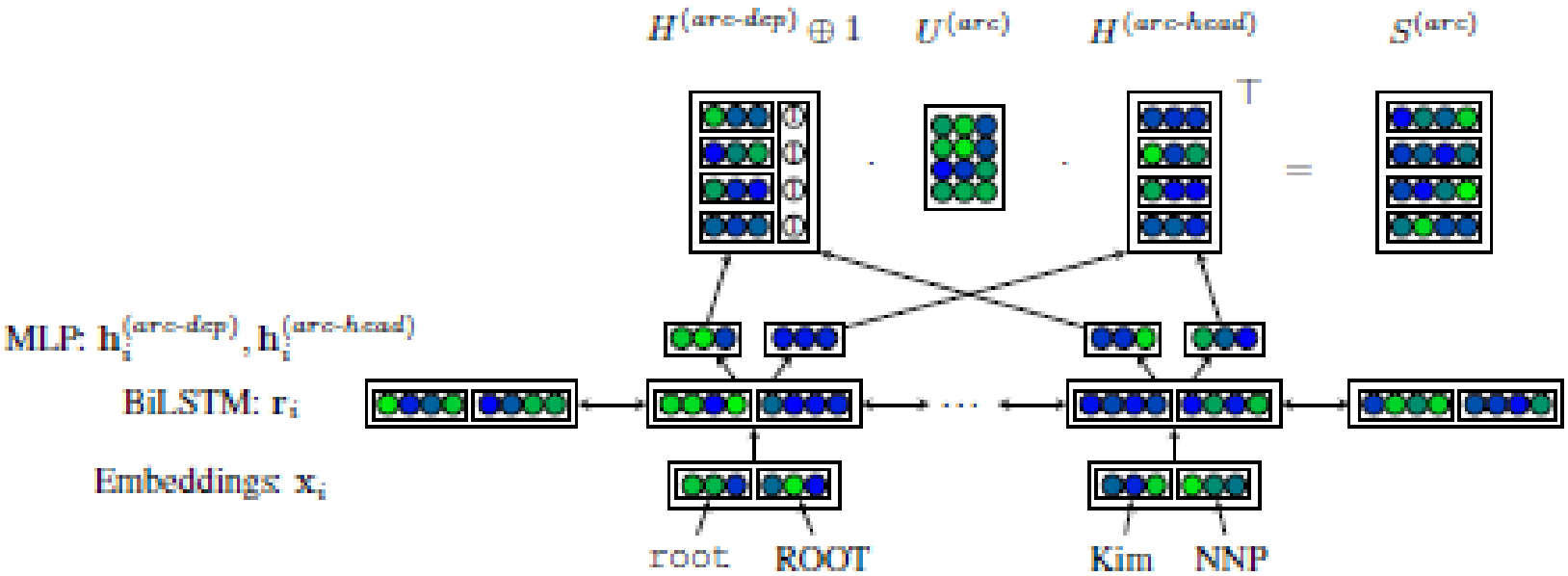
Standard architecture: morphology (Heigold, 2017)

- For word embeddings: character-level network.
- For word representations: LSTM over embeddings.ə
- No pretrained representations.



Standard architecture: syntax (Dozat, Manning, 2016)

- For word representations: pretrained embeddings + character-level network.
- For model predictions: biaffine attention.



Applying BERT

- We simply use BERT as embedder.
- A weighted average of 6 last layers representations is used for subtoken representation.
- Word embedding is the embedding of its last subtoken.
- BERT network is trained is trained together with the task-specific architecture.

Multilingual vs monolingual

Testing different BERT models on morphological and syntactic tagging.

Language	Model	Tag	Tag sent	LAS	Sent LAS	UAS	Sent UAS
Bulgarian	multi	96,99	68,28	90,07	39,16	94,1	62,54
Bulgarian	slavic	**97.27**	**70.52**	**91,47**	**45,43**	**94,86**	**65,77**
Czech	multi	96,4	64,96	91,06	44,91	93,9	58,82
Czech	Slavic	**97,05**	**69,71**	**91,68**	**46,66**	**94,4**	**60,84**
Polish	multi	93,65	67,75	94,29	73,65	96,51	83,67
Polish	slavic	**94,60**	**69,95**	**95,64**	**79,04**	**97,21**	**86,86**
Russian	multi	97,18	65,04	92,34	47,71	94,16	57,79
Russian	slavic	97,33	66,55	93,06	50,44	94,78	60,95
Russian	russian	**97,52**	**68,51**	**93,38**	**52,35**	**94,99**	**62,72**

Monolingual BERT: failure

- Belarusian dataset in UD2.3 – only 260 sentences.
- Russian and Ukrainian (closely related) – more than 54K sentences.

Training data	BERT	Tag	Tag sent	LAS	Sent LAS	UAS	Sent UAS
be	multlingual	85,09	10,29	76,34	14,71	83,72	17,65
be	Russian	80,75	4,41	45,66	1,47	57,45	4,41
be+ru+uk	multlingual	**88,57**	**19,12**	**84,8**	**16,18**	**90,74**	**33,82**
be+ru+uk	Russian	83,79	7,35	59,3	1,47	68,74	4,41

Monolingual BERT: failure

- Belarusian dataset in UD2.3 – only 260 sentences.
- Russian and Ukrainian (closely related) – more than 54K sentences.

Training data	BERT	Tag	Tag sent	LAS	Sent LAS	UAS	Sent UAS
be	multlingual	85,09	10,29	76,34	14,71	83,72	17,65
be	Russian	80,75	4,41	45,66	1,47	57,45	4,41
be+ru+uk	multlingual	**88,57**	**19,12**	**84,8**	**16,18**	**90,74**	**33,82**
be+ru+uk	Russian	83,79	7,35	59,3	1,47	68,74	4,41

- Related language data help, but not the BERT model for related languages.

Effect of segmentation

- In agglutinative languages each morpheme stands for a particular morphological feature:

  ev-ler-imiz-de-ymiș-ler
  house-PL-P1PL-LOC-COP.EV-3PL
  they apparently live in our houses

- multilingual BPE does not respect morpheme boundaries:

  ev-leri-mi-zde-ymi-ș-ler

- What if we tokenize only inside each morpheme?

Morpheme BPE: results

Language	BERT	Tag	Tag sent	LAS	Sent LAS	UAS	Sent UAS
Finnish	multilingual	93,09	60,31	86,48	51,11	90,42	63,63
Finnish	multilingual+ morpheme	94,17	64,7	86,37	49,60	90,08	61,49
Finnish	Finnish	**94,39**	**65,77**	**88,63**	**55,76**	**91,67**	**66,68**
Turkish	multilingual	88,31	38,52	64,96	22,87	74,07	36,21
Turkish	multilingual+ morpheme	**89,81**	**45,23**	64,92	21,23	73,48	34,87
Turkish	Turkish	89,33	40,31	**67,07**	**23,18**	**75,44**	**37,64**

Conclusions

- BERT is really good in processing morphology and syntax.
- Language-specific BERTs further improve performance.
- Morpheme segmentation improves morphology tagging.
- Main conclusion: respect the language data you are working on.

Thank you!
Спасибо!