Deep Learning for Natural Language Processing

Subword Representations for Sequence Models

UNIVERSITY OF GOTHENBURG

CHALMERS

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Richard Johansson

richard.johansson@gu.se

how can we do part-of-speech tagging with texts like this?

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

how can we do part-of-speech tagging with texts like this?

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

can you find the named entities in this text?

In 1932 , Torkelsson went to Stenköping .

can you find the named entities in this text?

In	***19**32*	*,*	*Torkel**sson***	*went*	*to*	*Sten**köping***	*.*
	Time		Person			Location	

using characters to represent words: old-school approach

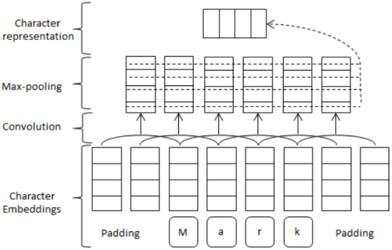
4.2.1 Spelling features

We extract the following features for a given word in addition to the lower case word features.

- whether start with a capital letter
- whether has all capital letters
- whether has all lower case letters
- whether has non initial capital letters
- whether mix with letters and digits
- whether has punctuation
- letter prefixes and suffixes (with window size of 2 to 5)
- whether has apostrophe end ('s)
- letters only, for example, I. B. M. to IBM
- non-letters only, for example, A. T. &T. to ..&
- word pattern feature, with capital letters, lower case letters, and digits mapped to 'A', 'a' and '0' respectively, for example, D56y-3 to A00a-0
- word pattern summarization feature, similar to word pattern feature but with consecutive identical characters removed. For example, D56y-3 to A0a-0

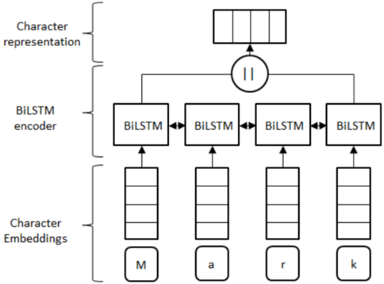
(Huang et al., 2015)

using characters to represent words: modern approaches



(a) CNN approach

(Ma and Hovy, 2016)

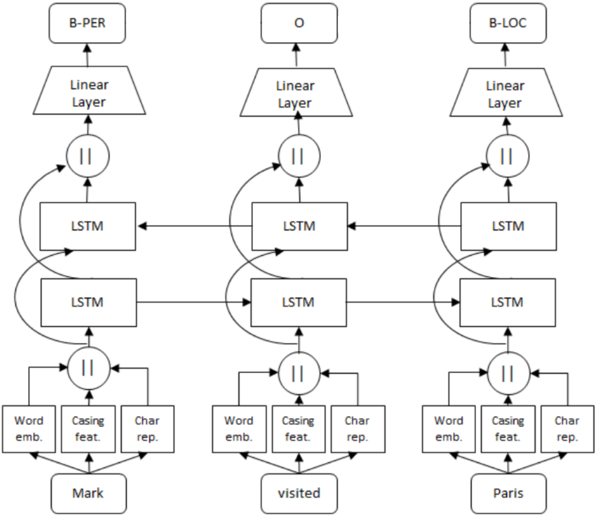


(b) BiLSTM approach

(Lample et al., 2016)

combining representations. . .

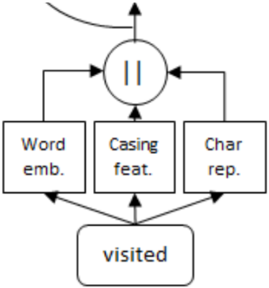
- we may use a combination of different word representations



from Reimers and Gurevych (2017)

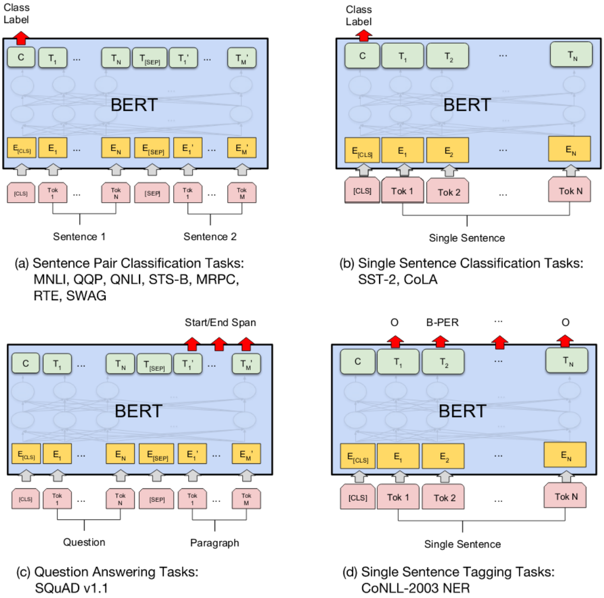
reducing overfitting and improving generalization

- character-based representations allow us to deal with words that we didn't see in the training set



- we can use **word dropout** to force the model to rely on the character-based representation
- for each word in the text, we replace the word with a dummy *"unknown"* token with a dropout probability p

recap: BERT for different types of tasks



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

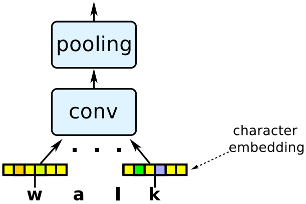
(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

recap: sub-word representation in ELMo, BERT, and friends

- **ELMo** uses a CNN over character embeddings



- **BERT** uses word piece tokenization

```
tokenizer.tokenize('In 1932, Torkelsson went to Stenköping.')

['in', '1932', ',', 'tor', '##kel', '##sson',
 'went', 'to', 'ste',  '##nko', '##ping', '.']
```

reading

- Eisenstein, chapter 7:
  - 7.1: sequence labeling as classification
  - 7.6: neural sequence models
- Eisenstein, chapter 8: applications

references

Z. Huang, W. Xu, and K. Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.

X. Ma and E. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

N. Reimers and I. Gurevych. 2017. Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. arXiv:1707.06799.