Contextualized Embeddings
Relation Extraction
Yoav Goldberg
NLP, ML and Features

- NLP as ML:
  - Transform text into a feature vector.
  - Feed into a classifier.
  - Predict.
NLP, ML and Features

• NLP as ML:
  • Transform text into a feature vector.
  • Feed into a classifier.  
  • Predict.  

The classifier can be complex, output can be structured.
NLP, ML and Features

• NLP as ML:
  
  • Transform text into a feature vector.  how?
  
  • Feed into a classifier.  The classifier can be complex, output can be structured.
  
  • Predict.
Representing text as Features

\[ f(\text{It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness...}) \]
Representing text as Features

- **Indicator features** over events in the data.

  - counts
  - words, characters, ngrams, lemmas, stems, ...
Representing text as Features

"the special onion soup was not very bad."

**bag of words** features (word counts):

```python
{'the':1, 'special': 1, 'onion':1, 'soup':1, 'was':1, 'not':1, 'very': 1, 'bad':1}
```
Representing text as Features

"the special onion soup was not very bad"

ngrams:

['the', 'special', 'onion', 'soup', 'was', 'not', 'very', 'bad',
'the special', 'special onion', 'onion soup', 'soup was',
'was not', 'not very', 'very bad', 'the special onion', 'special onion soup',
'onion soup was', 'soup was not', 'was not very', 'not very bad']
Representing text as Features

"the special onion soup was not very bad"

ngrams:

unigrams: ['the', 'special', 'onion', 'soup', 'was', 'not', 'very', 'bad', 'the special', 'special onion', 'onion soup', 'soup was', 'was not', 'not very', 'very bad', 'the special onion', 'special onion soup', 'onion soup was', 'soup was not', 'was not very', 'not very bad']

bigrams:

trigrams:
Features as Symbols, 1-hot encoding

- Lookup Table for "book" with count 3943
- Lookup Table for "very bad" with count 10234
- Lookup Table for "next-tag=VBD" with count 15677
Features as Symbols, 1-hot encoding

Sparse Vector

\[(0, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0)\]

- `3943`: Lookup Table, "book"
- `10234`: Lookup Table, "very bad"
- `15677`: Lookup Table, "next-tag=VBD"
Features as Symbols, 1-hot encoding

Sparse Vector

(0, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)

What are some pros and cons of this approach? Discuss.
Distributional Representation of Words

- Represent a word by the environments it occurs in.
Consider the columns of $W_3$.

Consider the rows of $E$.
Word Embeddings

\[ \mathbf{v}_{book} = \mathbf{E}[book] \]

What are some pros and cons of this approach? Discuss.
Feature Embeddings

\[ \mathbf{v}_{book} = \mathbf{E}[book] \]
Feature Embeddings

\[ v_{book} = E[book] \]

What are some pros and cons of this approach? Discuss.
Combining Vectors

\[
\mathbf{v}_I \quad \mathbf{v}_{read} \quad \mathbf{v}_a \quad \mathbf{v}_{book} \quad \mathbf{v}_{about}
\]

Lookup Table

I

read

a

book

about
Combining Vectors

I read a book about

concatenate

\[ \text{I read a book about} \]

\[ \text{concatenate} \]

\[ V_I \quad V_{\text{read}} \quad V_a \quad V_{\text{book}} \quad V_{\text{about}} \]

\[ \text{Lookup Table} \quad \text{Lookup Table} \quad \text{Lookup Table} \quad \text{Lookup Table} \quad \text{Lookup Table} \]

\[ \text{I} \quad \text{read} \quad \text{a} \quad \text{book} \quad \text{about} \]
Combining Vectors

I read a book about

\[ \text{sum} \]

\[ v_I + v_{\text{read}} + v_a + v_{\text{book}} + v_{\text{about}} \]

Lookup Table

I

Lookup Table

read

Lookup Table

a

Lookup Table

book

Lookup Table

about
Combining Vectors

I read a book about

\[ \text{sum (or average)} \]

\[ \mathbf{v}_I + \mathbf{v}_{\text{read}} + \mathbf{v}_a + \mathbf{v}_{\text{book}} + \mathbf{v}_{\text{about}} \]

Lookup Table

I

read

a

book

about
Combining Vectors

book a about read I

\[ V_I + V_{\text{read}} + V_a + V_{\text{book}} + V_{\text{about}} \]

sum (or average)

Lookup Table

I

Lookup Table

read

Lookup Table

a

Lookup Table

book

Lookup Table

about
Combining Vectors

**Concatenate**

I read

I read a

I read a book

I read a book about

**Sum (or average)**

"cbow"

I read

I read a

I read a book

I read a book about

I book a read about book about read I a I a about book read a read about book I ...

more words = longer vectors

order invariant
Word Embeddings

• Translate each word in the (fixed) vocabulary to a vector.
  • Typical dimensions: 100-300
  • Translation is done using a lookup table.
  • Can be "pre-trained" (word2vec, glove)

• Dealing with "infinite" vocabularies?
Solutions for "Infinite" Vocabularies

- \{\text{characters}\}, \{\text{word pieces, bpe}\}, \{\text{fastText}\}

\[
\text{dinosaur} = d \ i \ n \ o \ s \ a \ u \ r
\]

\[
\text{dinosaur} = \text{dino} \ #\text{sa} \ #\text{ur}
\]

\[
\text{dinosaur} = \text{dina} + \text{no} + \text{sau} + \text{os} + \text{n} + \text{aur}
\]
Solutions for "Infinite" Vocabularies

- \{characters\}, \{word pieces, bpe\}, \{fastText\}

\[
\text{dinosaur} = \text{d i n o s a u r}
\]

\[
\text{dinosaur} = \text{dino } \#sa \#ur
\]

\[
\text{dinosaur} = \text{dinosa } + \text{inosau } + \text{nosaur } + \\
\text{dino } + \text{inos } + \text{nosa } + \text{osau } + \text{saur } + \\
\text{din } + \text{ino } + \text{nos } + \text{osa } + \text{sau } + \text{aur}
\]

Pros/cons of each approach? Are there better ways to do it?
Can be thought of as a Black Box

Text encoding

Classifier
All the ML-based algorithms we saw before can be transformed to use feature embeddings.

Can be thought of as a Black Box
All the ML-based algorithms we saw before can be transformed to use feature embeddings.

- Text classification?
- PP-attachment?
- Sequence Tagging?
- Dependency Parsing?
- Phrase-based parsing?

Text encoding

Can be thought of as a Black Box
Problems with Word Embeddings

What are some shortcomings of word embeddings?
Problems with Word Embeddings

learned similarities may not be ideal

Where will this be desired? where will this be bad?
Problems with Word Embeddings

Polysemy

orange
Problems with Word Embeddings

Polysemy

red

orange

banana

apple
Problems with Word Embeddings

Polysemy

red
orange
banana
apple
microsoft
Problems with Word Embeddings

Polysemy

red

orange

banana

apple

microsoft

book
Problems with Word Embeddings

Polysemy

red
orange
banana
apple
microsoft
order
book
magazine
Problems with Word Embeddings

Polysemy
Problems with Word Embeddings

Polysemy
Contextualized word embeddings
Contextualized word embeddings

Use a contextualizing function to represent each word in context as a vector.
Contextualized word embeddings

(in reality, we don't really get rid of the lookup table)
Contextualized word embeddings

Use a contextualizing function to represent each word in context as a vector.

what does this solve? what doesn't this solve?
Contextualized word embeddings

All the ML-based algorithms we saw before can be transformed to use contextualized embeddings.

- Text classification?
- PP-attachment?
- Sequence Tagging?
- Dependency Parsing?
- Phrase-based parsing?
Contextualized word embeddings

All the ML-based algorithms we saw before can be transformed to use contextualized embeddings.

- Text classification?
- PP-attachment?
- Sequence Tagging?
- Dependency Parsing?
- Phrase-based parsing?

End-to-end training: the contextualized can be "fine-tuned" with the task (true also for word embeddings)
Contextualized word embeddings

All the ML-based algorithms we saw before can be transformed to use contextualized embeddings.

- Text classification?
- PP-attachment?
- Sequence Tagging?
- Dependency Parsing?
- Phrase-based parsing?

End-to-end training: the contextualized can be "fine-tuned" with the task (true also for word embeddings)
Contextualized vs Static word embeddings

• Is there still a use for static word embeddings?

• What are some pros / cons of contextualized embeddings compared to static embeddings?

• When will you use each?
Contextualizers

- Based on recurrent neural networks. (BiRNN / BiLSTM)
- Based on neural attention mechanism. (Transformer)
Recurrent Neural Network: RNN

I read a book about

RNN cell → RNN cell → RNN cell → RNN cell → RNN cell

Lookup Table → Lookup Table → Lookup Table → Lookup Table → Lookup Table

I, read, a, book, about
Bi-RNN / BiLSTM

keep intermediate vectors

I

I read

I read a

I read a book

I read a book about

RNN cell

RNN cell

RNN cell

RNN cell

RNN cell
Bi-RNN / BiLSTM

add right-to-left RNN (bi-RNN)

I read a book about
Bi-RNN / BiLSTM

add right-to-left RNN
(bi-RNN)

I read a book about
Bi-RNN / BiLSTM

add right-to-left RNN (bi-RNN)

I read a book about
Bi-RNN / BiLSTM

add right-to-left RNN (bi-RNN)

I read a book about

I read a book about

I read a book about
"Attention" Mechanism

I read a book about

weighted sum
Transformer

replace RNN with attention-based mechanism

• Main concepts to know:
  • Self-attention
  • Multi-head attention

• Also think about: why do this? what is the motivation?
Transformer

Self attention

each token attends to all tokens in previous layer
Transformer

Self attention
Transformer

Self attention
Transformer

multi-head attention

one attention pattern
Transformer

multi-head attention

another attention pattern
Transformer

multi-head attention

why choose if we can just have several?
Transformer

multi-head attention

why choose if we can just have several?
Contextualizer Training

• How do we train the contextualizes?

  • As part of a supervised task ("fine tuning").

  • Using large corpora and "self supervision" ("pre-training").
Contextualizer Training

• How do we train the contextualizer?
  
  • As part of a supervised task ("fine tuning").
  
  • Using large corpora and "self supervision" ("pre-training").
Contextualizer Training

- **Language modeling (LM) objective:**
  - Predict the *next word*.

- **Masked language modeling (MLM) objective:**
  - Hide a word and attempt to recover it.
Contextualizer Training

- **Language modeling (LM) objective:**
  - Predict the **next word**.

- **Masked language modeling (MLM) objective:**
  - Hide a **word** and attempt to recover it.
Masked Language Model

BERT, XLNET, RoBERTA, ...

Contextualizer
Masked Language Model

BERT, XLNET, RoBERTA, ...

???

$V_I$  $V_a$  $V_{book}$  $V_{about}$

Contextualizer

I  [MASK]  a  book  about
Masked Language Model

BERT, XLNET, RoBERTA, ...

Contextualizer

V\_I \uparrow \quad V\_\text{read} \uparrow \quad V\_a \uparrow \quad V\_\text{book} \uparrow \quad \text{???

I \uparrow \quad \text{read} \uparrow \quad a \uparrow \quad \text{book} \uparrow \quad \text{[MASK]}
Masked Language Model

BERT, XLNET, RoBERTA, ...

Contextualizer

\[ V_I \quad V_{read} \quad V_{book} \quad V_{about} \]

I \quad read \quad [MASK] \quad book \quad about
Masked Language Model

SpanBERT

Contextualizer

$V_I$  $V_{book}$  $V_{about}$

I  [MASK]  [MASK]  book  about
Recap of what we've seen

- Represent **words as vectors**, feed into an ML algorithm.

- From **one-hot/symbols** to **static embeddings** to **contextualized embeddings**.

- **Language modeling** and **masked language modeling** objectives for training contextualizers.
Software

Allen NLP

A natural language processing platform for building state-of-the-art models.

Huggingface

Transformers

State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0
Using contextualizers as LMs

Demo
Some active research topics on contextualizers

- What is encoded in the vectors?
- What isn't encoded in the vectors?
- What are the differences / similarities between different contextualizers?
- What is the most effective training method?
- Dealing with multi-word units?
- Dealing with sub-word units?