What about deep NLP?

M.Pikuliak // datalys // 21.11.2018

Classical NLP vs Deep NLP

How do these two approaches to NLP differ from engineer’s perspective?

Named entity recognition

Trump defends Saudi Arabia ties despite Khashoggi murder

Named entity recognition

Trump defends Saudi Arabia ties despite Khashoggi murder

Named entity recognition

Trump defends Saudi Arabia ties despite Khashoggi murder

B-PER O B-LOC I-LOC O O B-PER O

Named entity recognition

Trump defends Saudi Arabia ties despite Khashoggi murder

B-PER O B-LOC I-LOC O O B-PER O

- **B-tags** mark the first word of any NE
- **I-tags** mark the other NE words
- **O's** are non-NE words
- **PER**sons, **LOC**ations and **ORG**anizations

Named entity recognition

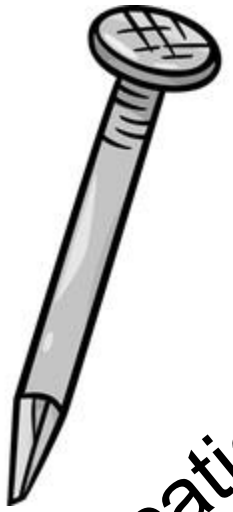
Trump defends Saudi Arabia ties despite Khashoggi murder

B-PER O **B-ORG I-ORG** O O B-PER O

???

- **B-tags** mark the first word of any NE
- **I-tags** mark the other NE words
- **O's** are non-NE words
- **PER**sons, **LOC**ations and **ORG**anizations

NER: Classical NLP



Classification problem

NER: Classical NLP

Each word is a classification sample.

We create a set of features around each word and train a classifier to predict tags from them.

Classical NLP workflow

1. Feature engineering: We identify features for each word

Feature engineering

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder
First uppercase	1	0	1	1	0	0	1	0
All uppercase	0	0	0	0	0	0	0	0

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder
First uppercase	1	0	1	1	0	0	1	0
All uppercase	0	0	0	0	0	0	0	0
Previous word	<s>	trump	defends	saudi	arabia	ties	despite	khashoggi

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder
First uppercase	1	0	1	1	0	0	1	0
All uppercase	0	0	0	0	0	0	0	0
Previous word	<s>	trump	defends	saudi	arabia	ties	despite	khashoggi
Gazetteer	1	0	1	1	0	0	0	0

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder
First uppercase	1	0	1	1	0	0	1	0
All uppercase	0	0	0	0	0	0	0	0
Previous word	<s>	trump	defends	saudi	arabia	ties	despite	khashoggi
Gazetteer	1	0	1	1	0	0	0	0
POS tags	NNP	VBZ	NNP	NNP	NNS	IN	NNP	NN

Feature engineering

Word	trump	defends	saudi	arabia	ties	despite	khashoggi	murder
Lemma	trump	defend	saudi	arabia	tie	despite	khashoggi	murder
First uppercase	1	0	1	1	0	0	1	0
All uppercase	0	0	0	0	0	0	0	0
Previous word	<s>	trump	defends	saudi	arabia	ties	despite	khashoggi
Gazetteer	1	0	1	1	0	0	0	0
POS tags	NNP	VBZ	NNP	NNP	NNS	IN	NNP	NN
First 3 characters	tru	def	sau	ara	tie	des	kha	mur
Last 2 character	mp	ds	di	ia	es	te	gi	er
…								

Classical NLP workflow

1. Feature engineering: We found features for each word
2. Feature engineering: Fixed size vector

Creating fixed size vector

{word: *kashoggi*, lemma: *kashoggi*, first_uppercase: True, all_uppercase: False, previous_word: *despite*, gazetteer: False, pos_tag: *NNP*, first_three_chars: *kha*, last_two_chars: *gi*}

- Normalize *numerical* data
- *Binary* data into 0/1
- *Strings* are categorical variables – one-hot encoding

Classical NLP workflow

1. Feature engineering: We found features for each word
2. Feature engineering: Fixed size vector
3. Use a ML algorithm to train a model

Training our model

```
clf = sklearn.svm.SVC()
clf.fit(train_inputs, labels)
clf.predict(test_inputs)
```

Training is the easy part, yay!

Training our model

```
clf = sklearn.svm.SVC(C=200.0)
clf.fit(train_inputs, labels)
clf.predict(test_inputs)
```

Training is the easy part, yay!

Maybe some hyperparameter tuning?

We need a lot of linguistic resources

- POS tagger
- Lemmatizer
- Dependency parser
- Gazetteer
- etc.

Only a handful of languages have these!

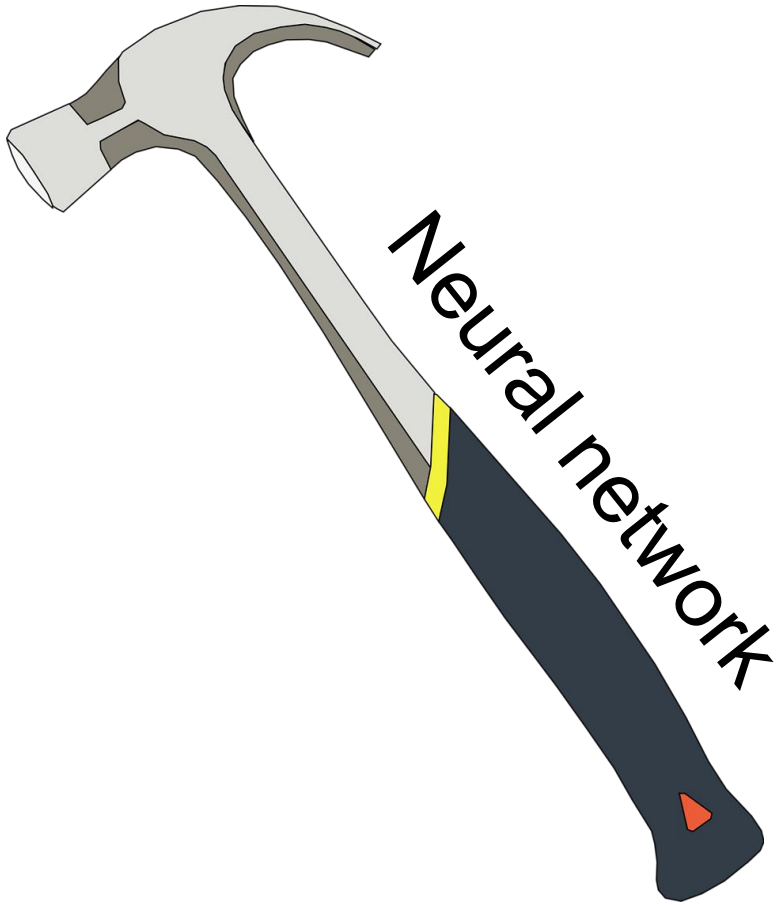
Overly engineered features make brittle solutions

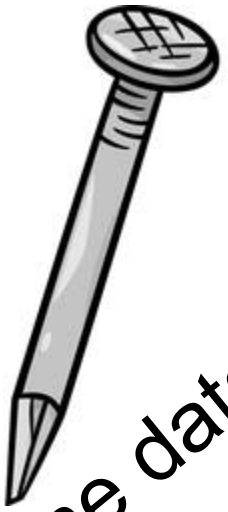
- Spurious models memorize non-relevant features and they do not generalize well
- It is in fact us who do most of the learning!
- Solutions are often over-fitted for particular task/domain/language
- People usually just check what other researchers are using

It is not optimal to make everything into fixed size vector

- Text is inherently a sequence of words with a variable length
- Most ML algorithms are not really suited for this task (vector-to-vector only)
- We throw away a lot of information to comply with them

NER: Deep NLP





Some data we can
cram into a neural network

NER: Deep NLP

We handcraft a model that takes a sentence and returns a sequence of tags.

How did deep learning change NLP?

- Last ~5 years
- Some say it’s a completely new paradigm
- It became a default approach in the academy
- All the big boys (Google, Facebook) use it

Deep NLP workflow

1. Just a pinch of pre-processing
   - Data cleaning – artifacts, strange characters and sentences
   - All numbers into one token: <NUMBER>
   - All URLs, hashtags, email addresses, etc. the same

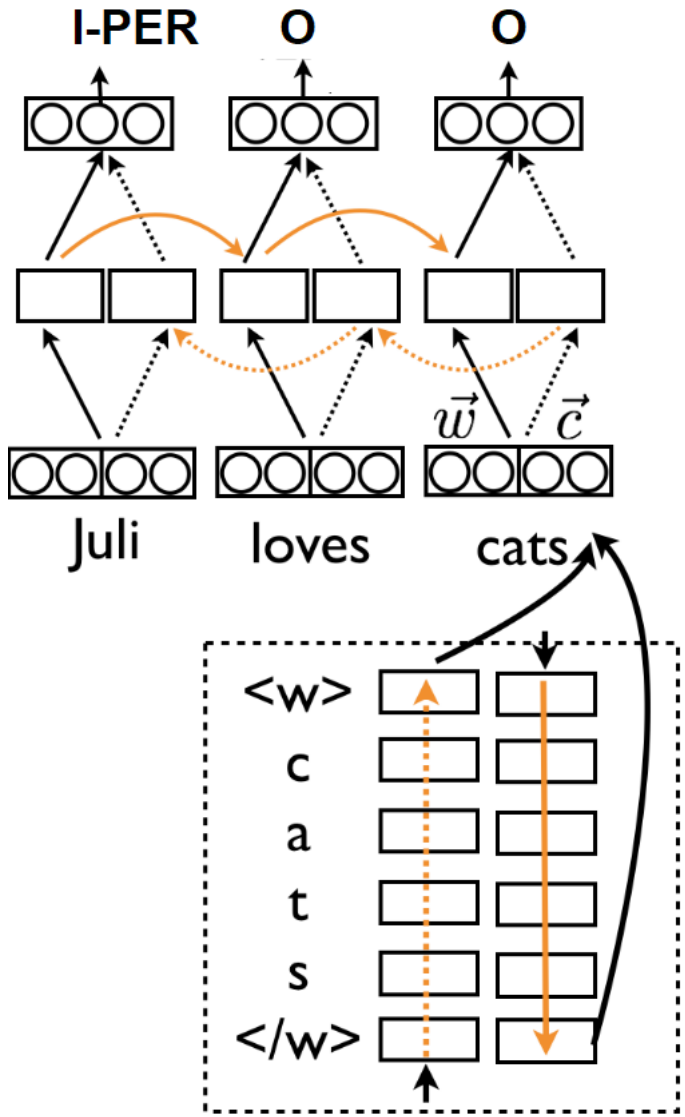
Deep NLP workflow

1. Just a pinch of pre-processing
2. Design a model

Model design

- Bi-directional character-level LSTM
- Pre-trained word embeddings
- Bi-directional word-level LSTM
- Average softmax activated cross-entropy as loss function
- Adam optimizer with dropout regularizer

But it only uses the input words!



Plank et al. - Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss (2016)

No feature engineering, purely data-driven

- Domain/language independent solutions
- No linguistic resources needed
- It’s harder to inject it with inherent bias by accident
- But it can still overfit on annotation artifacts

Question	**Answer**
How many … ?	2
What animal … ?	Dog

Agrawal et al. - Analyzing the Behavior of Visual Question Answering Models (2016)

Instead of feature engineering we now have model engineering

- Harsh learning curve for engineers (deep learning, frameworks, deep NLP knowledge)
- Even though there is actually only a handful of patterns that are used (reusability)
- More difficult hyperparameter tuning

Deep NLP is more powerful

- Empirically better results for almost all NLP tasks
- Better at modeling various input and output modalities
- Fancy learning options: pre-training, adversarial learning, transfer learning, etc.
- Can be effectively used on noisy text

Final comparison

Classical

- Heavy feature engineering
- Light model engineering
- Easy to implement models
- Often requires linguistic resources

Deep

- Almost no feature engineering
- Heavy model engineering
- Better results
- Can work even with noisy text

Deep NLP in practice

How to use deep NLP in your projects?

The hard way

- Deep learning MOOC by A. Ng
- Stanford Deep NLP course
- Visit NN Group ;-) (NN Group wiki)

The easy way

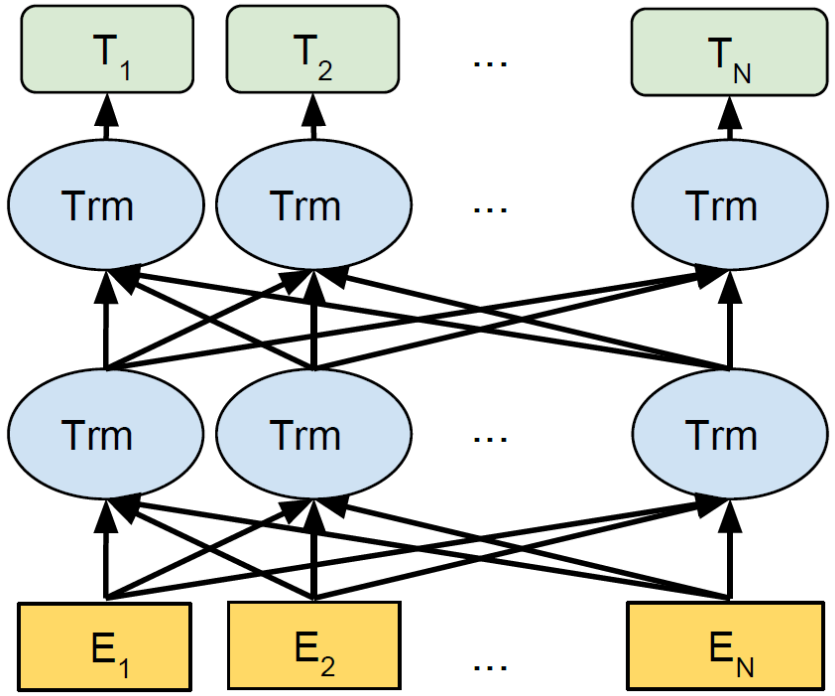
- ELMo (has Slovak model)
- BERT (has multilingual model)
- New wave of pre-trained language models (both 2018)

How to use ELMo

```
from elmoformanylangs import Embedder

e = Embedder('/path/to/your/model/')

sents = [['今', '天', '天氣', '真', '好', '阿'],
['潮水', '退', '了', '就', '知道', '誰', '沒', '穿', '褲子']]
# the list of lists which store the sentences

e.sents2elmo(sents)
# will return a list of numpy arrays
# each with the shape=(seq_len, embedding_size)
```

We get a vector representation for each word from the sentence

- It is a word representation but it encodes the information from the whole sentence
- The same word will have different representations based on the sentences it is used in
- Word representation can be used in your classical feature engineering pipeline



Average word embedding is a good sentence representation

- Average, max, min are several pooling operations we can use
- Use it instead of *tf-idf* for shorter documents

Q&A

- What use cases do you need to solve?
- What NLP obstacles did you come across in your life?
- Is deep learning going to take our jobs / kill us all?