Neural Metaphor Detection in Context

By Ge Gao, Yejin Choi and Luke Zettlemoyer (2018)

Stephan Detert

Ruprecht-Karls-Universität Heidelberg

19.06.2019

Background

Idea of Gao et al. (2018)

- Use standard Bi-LSTM model
  Bi-LSTM is already proven to perform well in VUA shared task 2018
- Idea: Combine LSTM approach with neural contextualized word representation

Leong et al.: Report on 2018 VUA Metpahor Detection Shared Task

Idea: Share knowledge about best architectures among growing Metaphor Detection researcher community.

- Task: Metaphor recognition on **all POS** or **verbs**
- Training phase: Training dataset is published participants decide how to train on this dataset (cross validation, generating sub-set as development set)
  Result: $N = 12$ trained systems are ready for testing
- Evaluation with easy accessible framework on common dataset
- Teams get test dataset and perform predictions on it
  Result: Predictions are submitted and automatically compared against true test labels

Approaches - Overview

Team	Word Embeddings	Dictionary-based	Linguistic	CRF	RNN	CNN	LSTM	Bi-LSTM	Di-LSTM	Context
THU NGN	X					X		X		
OCOTA	X		X				X	X		
bot.zen	X				X					
ZIL IPIPAN		X					X			
DeepReader	X		X						X	
Samsung_ RD_ PL	X			X						X
MAP	X			X				X		
nsu_ ai			X	X						

Features for metaphor detection tasks

- Concreteness/abstractness (Turney et al., 2011)
- Imaginability (Boradwell et al., 2013, Strzalkowski et al., 2013)
- Feature norms (Bulat et al., 2017)
- Sensory features (Tekiroglu et al, 2015; Shutova et al., 2016)
- Bag-of-words features (Köper and im Walde, 2016)
- Semantic class (Hovy et al., 2013; Tsvetkov et al., 2014)
- **Embedding-based approaches** (Köper and im Walde, 2017; Rei et al., 2017)

Trends in system design

- All submitted systems but one are based on NN architecture
- Use of explicit linguistic features
- Broad variety of corpora used to generate embeddings

Comparison of approaches

Rank	Team	P	R	F1	Approach
			All POS (Overall)		
1	THU NGN	0.608	0.700	0.651	word embeddings + CNN + Bi-LSTM
2	OCOTA	0.595	0.680	0.635	word embeddings + Bi-LSTM + linguistic
3	bot.zen	0.553	0.698	0.617	word embeddings + LSTM RNN
4	Baseline 2	0.510	0.696	0.589	UL + WordNet + CCDB + Logistic Regression
5	ZIL IPIPAN	0.555	0.615	0.583	dictionary-based vectors + LSTM
6	Baseline 1	0.521	0.657	0.581	UL + Logistic Regression
7	DeepReader	0.511	0.644	0.570	word embeddings + Di-LSTM + linguistic
8	Samsung_RD_PL	0.547	0.575	0.561	word embeddings + CRF + context
9	MAP	0.645	0.459	0.536	word embeddings + Bi-LSTM + CRF
10	nsu_ai	0.183	0.111	0.138	linguistic + CRF

Figure: Team scores ranked by $F1$

Source: *[5]*

Comparison of approaches *THU NGN* vs. *MAP*

	P	R	F1
THU NGN	0.608	0.700	0.651
MAP	0.645	**0.459**	0.536

Both approaches use word embessings, Bi-LSTM
Further comparison:

- Both use word2vec
- Both use additional features like POS tags
- *THU NGN* uses CNN
- *THU NGN* uses ensemble method
- *MAP* uses CRF

Authors of the VUA evaluation paper conclude, that using Softmax instead of CRF improves recall rate *R*.

Conclusion

- Metaphor detection for **verbs** is easier for current approaches. Performance on all parts of speech is worse.
- There are severe **genre-based gaps** in performance accrosss different genres.
- Traditional baseline classifiers relying on **feature engineering** are **not far behind** deep learning approaches. Combining NNs with explicit linguistic features may be promising approach for the future.

What is context?

- Verb, target word (Turney et al.)
- SVO triples (Shutova et al.)
- Full sentence (Köper and im Walde, 2017; Turney et al., 2011; Jang et al., 2016)

Two task formulations

Sequence labeling task: Every word in a sentence is target word.

Make	the	people's	heart	glow
↑	↑	↑	↑	↑

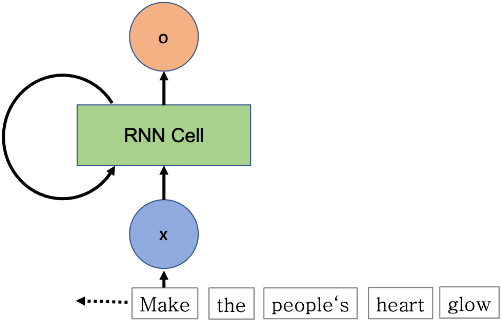
Classification model: Only a single target **verb** per sentence is labeled.

Make	the	people's	heart	glow
				↑

The sequence labeling generalizes the classification task, classifications can be derived from sequence labeling.
BUT: We will observe differences in performance.

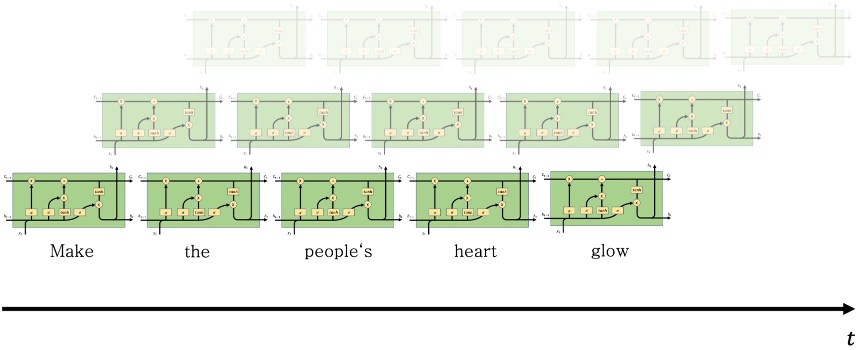
RNN Architecture

RNNs handle tokens from input sequence by keeping information in memory



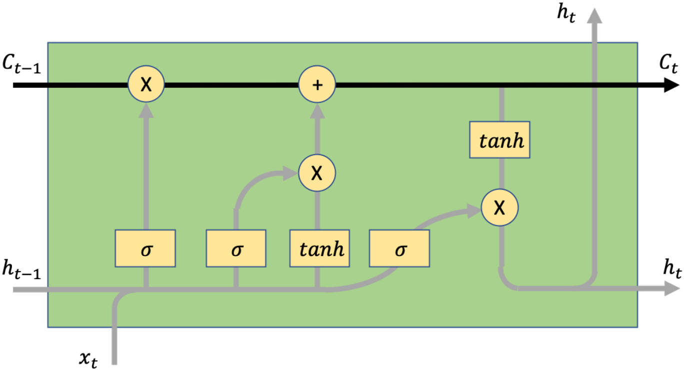
Sub-class of Recurrent Neural Networks (RNNs): LSTMs

LSTM: Layer Architecture [4]



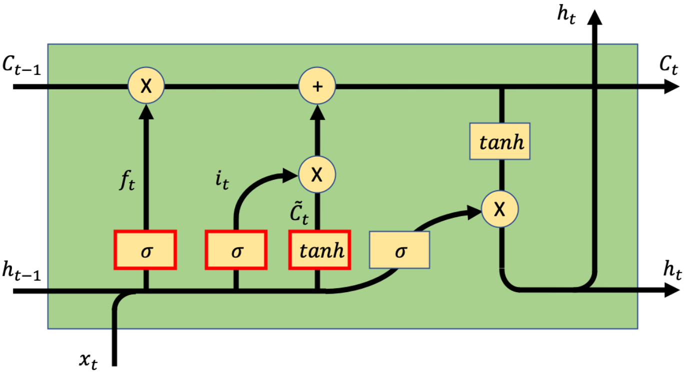
LSTMs process token sequences. Multi-layer architectures are possible.

Long-Short-Term-Memory Architecture [4]



C: **Cell state**, memory, running through all blocks
Writing to memory through gatings

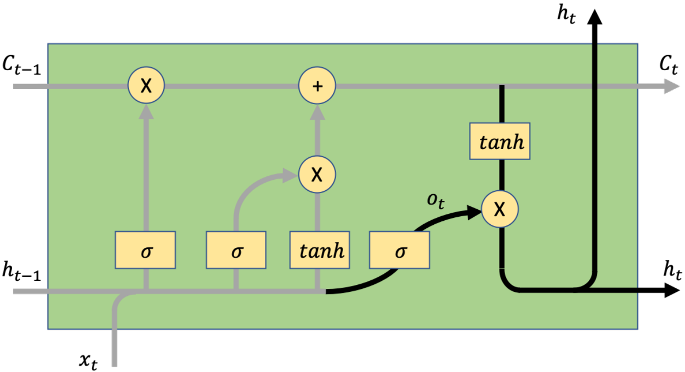
Long-Short-Term-Memory Architecture [4]



Forgetting function:
weight matrix W_f, bias b_f
$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$

Add new values to memory:
$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$
$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$

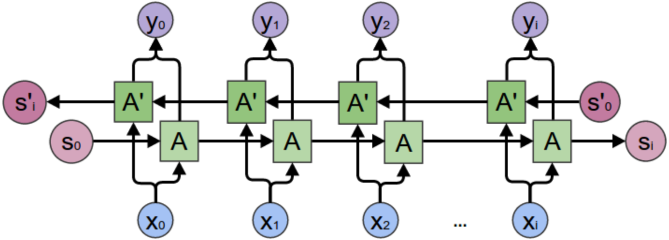
Long-Short-Term-Memory Architecture [4]



Input - output gate:
$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$

Process output components:
$h_t = o_t * \tanh\left(C_t\right)$

Bidirectional LSTM



Source: *[7]*

Pre-Processing

Open-source NLP library spaCy

- Lemmatization
- Tokenization
- Part-of-speech tagging

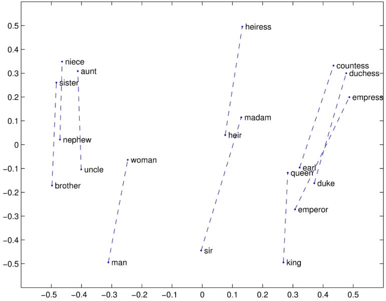
Sentences are encoded by two concatenated vectors

For the task of word sense disambiguation, the combination of two embedding variations has been proven. (Birke and Sakar)

1. Pre-trained word embeddings (GloVe) w_i
2. Embeddings from language Models (ELMo) e_i

Global Vectors for Word Embeddings (GloVe) [8]

- Word-based representation algorithm
- Representation vectors based on co-occurence of words in training corpus
- **Learning objective:** Dot product of two vectors = log probability of two words' co-occurence
- GloVe performs well on word analogy tasks



Source: *[8]*

Embeddings from language Models (ELMo) [1]

New about ELMo: Derived from whole context sentence! ELMo vector covers...

- Complex characteristics of word usage (syntax and semantics)

Example

1. I withdraw money in the **bank**.
2. She had a nice walk along the river **bank**.

Bank has different word embeddings in ELMo

Using ELMo, textual entailment, question answering and sentiment analysis improve (up to 20 %).

Language Models (LM) [1]

- Predict token based on left context and right context

$$\underbrace{\text{My dog}}_{\text{left context}}\ \underbrace{\text{barks}}_{\text{target}}\ \underbrace{\text{at the mailman}}_{\text{right context}}$$

Language Models (LM)

- Predict token t_k

$$p\left(t_1, t_2, \ldots, t_N\right) = \prod_{k=1}^{N} p\left(t_k | t_1, t_2, \ldots, t_{k-1}\right)$$

Architecture of recent state-of-the-art language models

- Get context-independent word representation $\overrightarrow{\mathbf{h}}_{k,L}^{LM}$ of t_k given $(t_{k+1}, ..., t_N)$
- Pass representation through L Layers
- At each position k each layer outputs context-dependent vector $\overrightarrow{\mathbf{h}}_{k,j}^{LM}$
- The top Layer outputs $\overrightarrow{\mathbf{h}}_{k,L}^{LM}$
- Output of top Layer applied to Softmax function to predict next token

ELMo architecture in use [1]

1. Feed context independent embeddings $t_0 \ldots t_{k-1}$ and $t_k + 1 \ldots t_N$ into RNN
2. Capture layer representations for each t_k
3. Supervised RNN forms context-sensitive representation h_k
4. The layer representations h_k are weighted, normalized, summed up and scaled to one ELMo vector:

$$\mathrm{ELMo}_k^{task} = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

ELMo-improved architectures

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Source: *[1]*

Figure: *Models enhanced by use of ELMo representation*

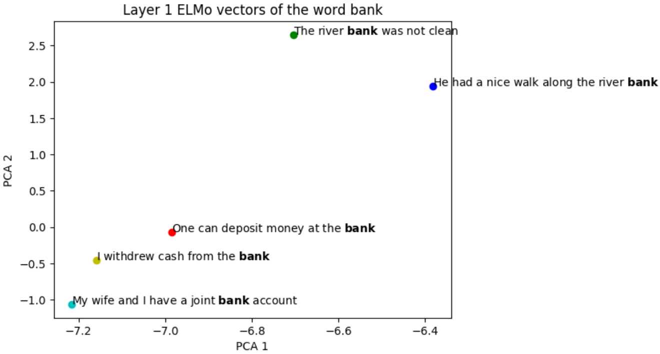
Visual interpretation of ELMo vectors [2]

Recall: There are 3 layers in ELMo

0. Character-based embedding
1. biLSTM capturing syntax (mainly)
2. biLSTM capturing semantics (mainly)

We will visualize vectors as outputs of layers **1, 2**

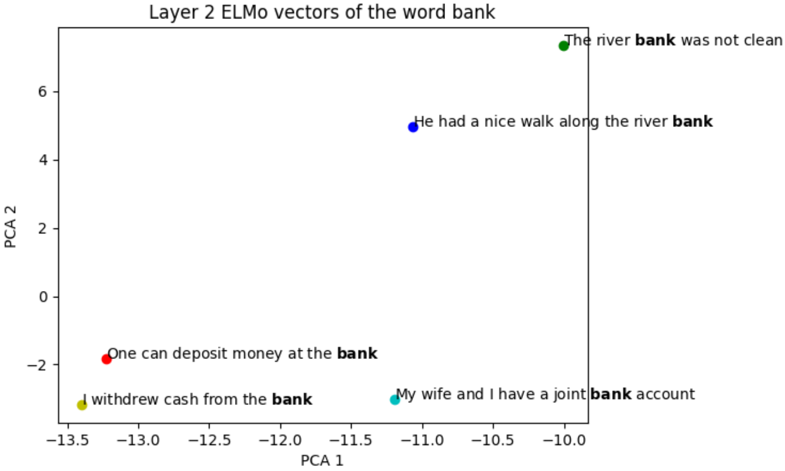
Visualization of ELMo vectors



Source: *[2]*

Figure: *PCA of layer 1*

Visualization of ELMo vectors



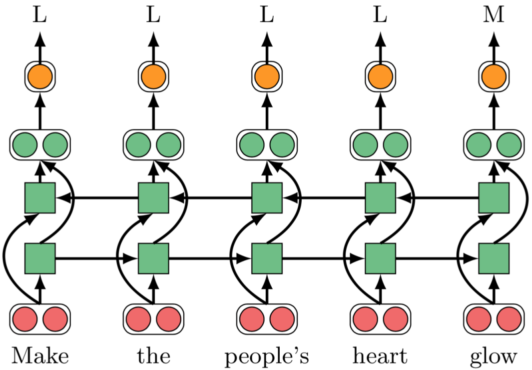
Source: *[2]*

Figure: *PCA of layer 2*

Model overview

1. Raw word encoding
2. Deep word embedding with ELMo vector e_i
3. Pre-trained word embedding w_i
4. Input word representation to bidirectional LSTM
5. Feedforward neural network (otpimized for log-likelihood of gold labels)

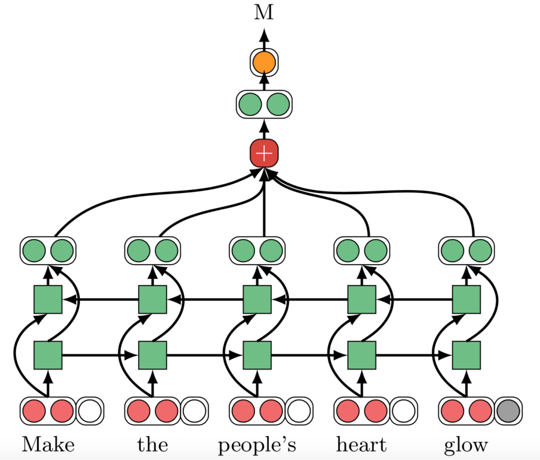
Sequence Labeling Model



Source: *[3]*

Input to model: token representation $[w_i; e_i]$

Classification Model



Source: *[3]*

Input to model: token representation $[w_i; e_i; n_i]$

Classification Model

Input to model: token representation $[w_i; e_i; n_i]$
n_i indicates, whether token is classification target

1. LSTM gives contextualized representation h_i
2. Tokens in context sentence are weighted by attention a_i
   $a_i = SoftMax_i\left(W_a h_i + b_a\right)$ Weights W_a and bias b_a are learnt parameters
3. Introduce weighted sum c: $c = \sum_{i=1}^{n} a_i h_i$
4. Feed c to feedforward network to compute the label scores for target verb.

Datasets

- MOH
  - Extract example sentences for WordNet instances
  - Label them manually (CrowdFlower)
  - Higher metaphor density than natural likelihood in running text

`communicate,The rooms communicated,1`

- MOH-X
  - Subset of MOH: argument of verb is extraceted

`workers,abuse, This boss abuses his workers`

- TroFi
  - 50 verb clusters with literal/non-literal usage
  - Higher metaphor density (see MOH-X)

```
CLUSTER: absorb, IDX: 12, LABEL: 0, 'Vitamins cold
be passed right out of the body without being
absorbed'
```

Datasets

- VUA
  - 117 fragments sampled accross genres in British National Corpus: Academic, News, Conversation, Fiction
  - Same number of tokens for each genre
  - Over 2K unique verbs
  - All words in sentence are labled

```
('PRON', 'VERB', 'PART', 'PRON', 'ADP', 'DET',
'NOUN', 'PUNCT'),He M-turned M-on me like a
M-snake
```

Dataset statistics

	# Expl.	% Metaphor	# Uniq. Verb	Avg # Sent. Len
MOH-X	647	49%	214	8.0
MOH	1,639	25%	440	7.4
TroFi	3,737	43%	50	28.3
VUA	23,113	28%	2047	24.5

Source: *[3]*

Implementation Details

Pre-trained part

- ELMo embeddings:
  2 layers bidirectional LSTM
  Hidden state: 512 dimensions, each layer
- GloVe embeddings:
  300 dimensional vectors
  derived from pre-trained matrix

Trainable part

- LSTM sequence labeling/classification 300 dimensional hidden state
- Dropout applied on input to LSTM and feedforward layer to prevent over-fitting
- Optimizer: SGD, ADAM

Experiment Setup

Classification Experiment Setup

- MOH-X and TroFi: 10-fold cross validation
- VUA: original training/test/development split

Sequence Labeling Experiment Setup

- Use VUA as it contains labels for all POS
- Manually create training/test/development split

Comparison to other models

- **Lexical baseline**: Logistic regression
  Weights inversely proportional to class frequencies, see naive Bayes
- **Klebanov (2016)**: Logistic regression classifier
  Features: Verb lemmas, verb's semantic class from WordNet
- **Rei (2017)**: Neural similarity network
  Features: skip-gram, word embeddings
- **Köper (2017)**: Balanced logistic regression classifier
  Features: target verb lemma rated for abstractness
- **Wu (2018)**: CNN-LSTM model with weighted-softmax classifier
  Features: pre-trained word2vec, POS tags, word cluster features

Evaluation Metric

- Precision **P**
- **F1** score
- Overall **acc**uracy
- For VUA: F1 scores averaged per genre:
  - conversation
  - academic writing
  - fiction
  - news

Evaluation Results

Model	MOH-X (10 fold)				TroFi (10 fold)				VUA - Test				
	P	R	F1	Acc.	P	R	F1	Acc.	P	R	F1	Acc.	MaF1
Lexical Baseline	39.1	26.7	31.3	43.6	**72.4**	55.7	62.9	71.4	67.9	40.7	50.9	76.4	48.9
Klebanov (2016)	-	-	-	-	-	-	-	-	-	-	-	-	60.0
Rei (2017)	73.6	76.1	74.2	74.8	-	-	-	-	-	-	-	-	-
Köper (2017)	-	-	-	-	-		**75.0**	-	-	-	62.0	-	-
Wu (2018) ensemble	-	-	-	-	-	-	-	-	60.0	**76.3**	67.2	-	-
CLS	75.3	84.3	**79.1**	**78.5**	68.7	**74.6**	72.0	73.7	53.4	65.6	58.9	69.1	53.4
SEQ	**79.1**	73.5	75.6	77.2	70.7	71.6	71.1	**74.6**	**68.2**	71.3	**69.7**	**81.4**	**66.4**

Source: *[3]*

- Classification performs better on smaller sentences (MOH-X)
- Köper et al. outperform both models for TroFi.
  Interpretation: Abstractness and imaginability ratings of surrounding words correlate to metaphor labels
- On VUA dataset the sequence classifier performs better
  Interpretation: Prediciting labels on all POS helps to classify target

Comparison

- The paper's approach performs comparably well on all datasets.
- For TroFi and MOH-X, the classification task performs better
- In VUA, where all words are labeled, sequence classifier is preferred

Comparison with *THU NGN*

Model	P	R	F1	Acc.
Lexical Baseline	68.6	45.2	54.5	90.6
Wu (2018) ensemble	60.8	70.0	65.1	-
Ours (SEQ)	**71.6**	**73.6**	**72.6**	**93.1**

Figure: Performance on the VUA sequence labeling test set for all POS tags

Source: *[3]*

Using ELMo improves state-of-the-art model (by Wu et al., 2018)

Effects of Contextual Word Representation

Model	P	R	F1.	Acc.
SEQ	**68.3**	**72.0**	**70.4**	**83.5**
-ELMo	59.4	64.3	61.7	78.2
CLS	52.4	63.0	57.3	74.3
-ELMo	52.0	48.7	50.8	74.1

Figure: Ablation study on VUA development set

Source: *[3]*

Sequence Labeling in Detail

POS	#	% metaphor	P	R	F1.
VERB	**20K**	18.1	68.1	71.9	69.9
NOUN	20K	13.6	59.9	60.8	60.4
ADP	13K	**28.0**	**86.8**	**89.0**	**87.9**
ADJ	9K	11.5	56.1	60.6	58.3
PART	3K	10.1	57.1	59.1	58.1

Source: *[3]*

- Performance on POS tags with more training data is higher
- POS tags as part of multi-word expressions are difficult to classify: 'Put **down** the disturbances'

Error Analysis

100 errors occuring in the best model tested on the VUA development set were analysed: Metaphor classes in VUA sould help analysing: *direct metaphor, indirect metaphor, implicit metaphor, personification, borderline case*

False positives / false negatives

- 31 / 33 % depend on implicit arguments (not in context)
- 20 / 50 % borderline cases
- - / 18 % personifications
- 15 / - % have long range dependencies ($>$ 4 words)
- 10 / - % arguments with rare word sense

For false negatives as well as for false positives borderline cases are crucial: Metaphor annotation still is a subjective task.

Positives

Indirect metaphor

`So they bought immunity.`

CLS: ✗ SEQ: ✗

Personification

`He thought of thick, fat, hot motorways carving up that land.`

CLS: ✗ SEQ: ✓

Direct metaphor

`In reality you just invent a tale, as if you were sitting round a fire in a cave.`

CLS: ✗ SEQ: ✗

Challenges

The model apparently does not cover...

- Borderline cases
- Long context
- Less frequently used words

For false negatives as well as for false positives borderline cases are crucial: Metaphor annotation still is a subjective task!

Discussion

Thank you.

[1] B. Beilharz. Elmo: Embeddings from language models. 2019.

[2] H. Chang. Visualizing elmo contextual vectors. *Towards DataScience*, 2019.

[3] G. Gao, E. Choi, Y. Choi, and L. Zettlemoyer. Neural metaphor detection in context. In *EMNLP*, 2018.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[5] C. W. Leong, B. B. Klebanov, and E. Shutova. A report on the 2018 vua metaphor detection shared task. 2018.

[6] S. M. Mohammad, E. Shutova, and P. D. Turney. Metaphor as a medium for emotion: An empirical study. 2016.

[7] C. Olah. Neural networks, types, and functional programming. 2015.

[8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. 2014.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[10] C. Wu, F. Wu, Y. Chen, and S. Wu. Neural metaphor detecting with cnn-lstm model. 2018.