Neural Metaphor Detection in Context

Stephan Detert
Ruprecht-Karls-Universität Heidelberg

19.06.2019
1. VUA shared task
2. Introduction
3. LSTM
4. Embeddings
5. Model
6. Datasets
7. Experiments
Background


- 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
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

<table>
<thead>
<tr>
<th>Team</th>
<th>Word Embeddings</th>
<th>Dictionary-based</th>
<th>Linguistic</th>
<th>CRF</th>
<th>RNN</th>
<th>CNN</th>
<th>LSTM</th>
<th>Bi-LSTM</th>
<th>Di-LSTM</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>THU NGN</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OCOTA</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>bot.zen</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZIL IPIPAN</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepReader</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samsung RD PL</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nsu ai</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

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

**Figure:** Team scores ranked by $F1$

*Source: [5]*
Comparison of approaches *THU NGN* vs. *MAP*

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>THU NGN</td>
<td>0.608</td>
<td>0.700</td>
<td>0.651</td>
</tr>
<tr>
<td>MAP</td>
<td>0.645</td>
<td><strong>0.459</strong></td>
<td>0.536</td>
</tr>
</tbody>
</table>

Both approaches use word embeddings, 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 $R$. 
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 across 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.
RNNs handle tokens from input sequence by keeping information in memory

Sub-class of Recurrent Neural Networks (RNNs): LSTMs
LSTMs process token sequences. Multi-layer architectures are possible.
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 $b_f$

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

Add new values to memory:

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh (W_C \cdot [h_{t-1}, x_t] + b_C)$$
Long-Short-Term-Memory Architecture [4]

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

Process output components:
\[ h_t = o_t \times \tanh(C_t) \]
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-occurrence of words in training corpus
- **Learning objective:** Dot product of two vectors $= \log$ probability of two words’ co-occurrence
- GloVe performs well on word analogy tasks

![Graph showing word vectors](image)

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

**My dog barks at the mailman**

- left context
- target
- right context
**Language Models (LM)**

- Predict token $t_k$

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

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

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

1. Feed context independent embeddings $t_0...t_{k-1}$ and $t_k+1...t_N$ 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:

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

<table>
<thead>
<tr>
<th>TASK</th>
<th>PREVIOUS SOTA</th>
<th>OUR BASELINE</th>
<th>ELMo + BASELINE</th>
<th>INCREASE (ABSOLUTE/RELATIVE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD</td>
<td>Liu et al. (2017)</td>
<td>84.4</td>
<td>81.1</td>
<td>85.8</td>
</tr>
<tr>
<td>SNLI</td>
<td>Chen et al. (2017)</td>
<td>88.6</td>
<td>88.0</td>
<td>88.7 ± 0.17</td>
</tr>
<tr>
<td>SRL</td>
<td>He et al. (2017)</td>
<td>81.7</td>
<td>81.4</td>
<td>84.6</td>
</tr>
<tr>
<td>Coref</td>
<td>Lee et al. (2017)</td>
<td>67.2</td>
<td>67.2</td>
<td>70.4</td>
</tr>
<tr>
<td>NER</td>
<td>Peters et al. (2017)</td>
<td>91.93 ± 0.19</td>
<td>90.15</td>
<td>92.22 ± 0.10</td>
</tr>
<tr>
<td>SST-5</td>
<td>McCann et al. (2017)</td>
<td>53.7</td>
<td>51.4</td>
<td>54.7 ± 0.5</td>
</tr>
</tbody>
</table>

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

Layer 2 ELMo vectors of the word bank

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 (optimized for log-likelihood of gold labels)
**Sequence Labeling Model**

Source: [3]

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

Input to model: token representation \([w_i; e_i; n_i]\)

Source: [3]
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 \(a_i\)
   
   \[a_i = \text{SoftMax}_i (W_a h_i + b_a)\]

   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

- **MOH-X**
  - Subset of MOH: argument of verb is extracted

- **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 across 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 labeled

("PRON", "VERB", "PART", "PRON", "ADP", "DET", "NOUN", "PUNCT"), He M- turned M-on me like a M-snake
## Dataset statistics

<table>
<thead>
<tr>
<th></th>
<th># Expl.</th>
<th>% Metaphor</th>
<th># Uniq. Verb</th>
<th>Avg # Sent. Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOH-X</td>
<td>647</td>
<td>49%</td>
<td>214</td>
<td>8.0</td>
</tr>
<tr>
<td>MOH</td>
<td>1,639</td>
<td>25%</td>
<td>440</td>
<td>7.4</td>
</tr>
<tr>
<td>TroFi</td>
<td>3,737</td>
<td>43%</td>
<td>50</td>
<td>28.3</td>
</tr>
<tr>
<td>VUA</td>
<td>23,113</td>
<td>28%</td>
<td>2047</td>
<td>24.5</td>
</tr>
</tbody>
</table>

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 $P$
- $F_1$ score
- Overall accuracy
- For VUA: $F_1$ scores averaged per genre:
  - conversation
  - academic writing
  - fiction
  - news
### Evaluation Results

<table>
<thead>
<tr>
<th>Model</th>
<th>MOH-X (10 fold)</th>
<th>TroFi (10 fold)</th>
<th>VUA - Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
</tr>
<tr>
<td>Lexical Baseline</td>
<td>39.1</td>
<td>26.7</td>
<td>31.3</td>
</tr>
<tr>
<td>Klebanov (2016)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rei (2017)</td>
<td>73.6</td>
<td>76.1</td>
<td>74.2</td>
</tr>
<tr>
<td>Köper (2017)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Wu (2018) ensemble</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLS</td>
<td>75.3</td>
<td>84.3</td>
<td>79.1</td>
</tr>
<tr>
<td>SEQ</td>
<td>79.1</td>
<td>73.5</td>
<td>75.6</td>
</tr>
</tbody>
</table>

**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: Predicting 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*

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Baseline</td>
<td>68.6</td>
<td>45.2</td>
<td>54.5</td>
<td>90.6</td>
</tr>
<tr>
<td>Wu (2018) ensemble</td>
<td>60.8</td>
<td>70.0</td>
<td>65.1</td>
<td>-</td>
</tr>
<tr>
<td>Ours (SEQ)</td>
<td><strong>71.6</strong></td>
<td><strong>73.6</strong></td>
<td><strong>72.6</strong></td>
<td><strong>93.1</strong></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1.</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ</td>
<td>68.3</td>
<td>72.0</td>
<td>70.4</td>
<td>83.5</td>
</tr>
<tr>
<td>-ELMo</td>
<td>59.4</td>
<td>64.3</td>
<td>61.7</td>
<td>78.2</td>
</tr>
<tr>
<td>CLS</td>
<td>52.4</td>
<td>63.0</td>
<td>57.3</td>
<td>74.3</td>
</tr>
<tr>
<td>-ELMo</td>
<td>52.0</td>
<td>48.7</td>
<td>50.8</td>
<td>74.1</td>
</tr>
</tbody>
</table>

Figure: Ablation study on VUA development set

Source: [3]
Sequence Labeling in Detail

<table>
<thead>
<tr>
<th>POS</th>
<th>#</th>
<th>% metaphor</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERB</td>
<td>20K</td>
<td>18.1</td>
<td>68.1</td>
<td>71.9</td>
<td>69.9</td>
</tr>
<tr>
<td>NOUN</td>
<td>20K</td>
<td>13.6</td>
<td>59.9</td>
<td>60.8</td>
<td>60.4</td>
</tr>
<tr>
<td>ADP</td>
<td>13K</td>
<td>28.0</td>
<td>86.8</td>
<td>89.0</td>
<td>87.9</td>
</tr>
<tr>
<td>ADJ</td>
<td>9K</td>
<td>11.5</td>
<td>56.1</td>
<td>60.6</td>
<td>58.3</td>
</tr>
<tr>
<td>PART</td>
<td>3K</td>
<td>10.1</td>
<td>57.1</td>
<td>59.1</td>
<td>58.1</td>
</tr>
</tbody>
</table>

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'
100 errors occurring in the best model tested on the VUA development set were analysed: Metaphor classes in VUA should 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: ✗
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.