

Linguistic Interpretation of Neural Networks

David Mareček

📅 January 6, 2022



EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

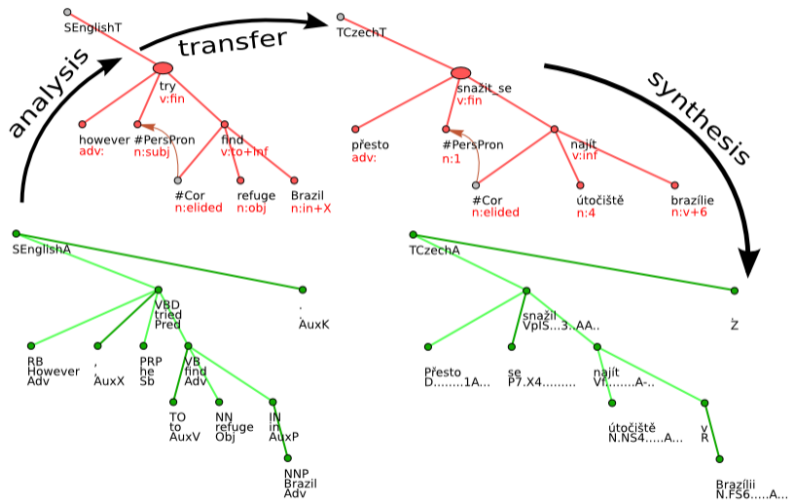


unless otherwise stated

Motivation – Tree-based Machine Translation

In 2010:

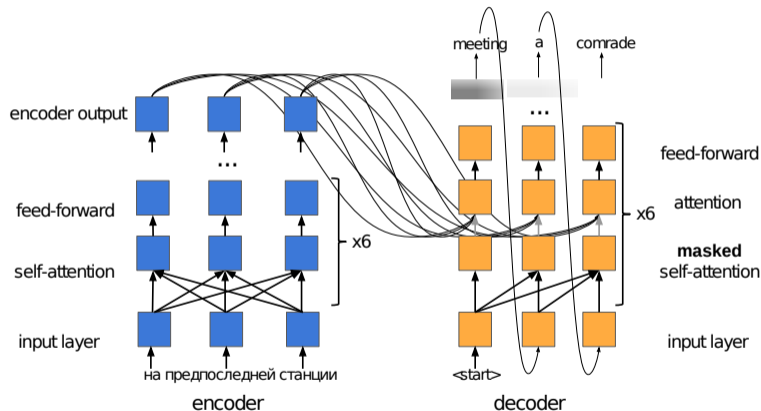
- Phrase-based or tree-based machine translation systems.
- The task was separated into many subtasks: *tokenization, morphological analysis, parsing, semantics, transfer, generation.*
- Easy to analyze the pipeline, easy to track errors.



Motivation – Neural Machine Translation

In 2020:

- Neural machine translation (Transformer).
- Trained in end-to-end fashion on raw parallel texts.
- All the analyses, transfer and generation is learned latently.
- Hard to interpret, hard to track errors.



Motivation – Syntactic Analysis in Machine Translation

2010:

- We have treebanks (manually annotated syntactic trees), up to 1 million words per language and we are able to build syntactic tree for any sentence.
- However, tree-based systems using such syntactic trees in e.g. machine translation were outperformed by a simple and more robust phrase-based systems.
- Why? Such trees were not suitable for these tasks. The tree structure is probably too simple to represent all relations in the sentence.

2020:

- The translation quality on some of the datasets is superhuman.
- We have a self-attention mechanism, which is able to learn how to connect and combine words to build a sentence representation.
- However, each token is represented by high-dimensional vectors in 6 layers, the tokens are combined by 96 different weight-matrices.
- We know that some structures are learned, but it is hard to grasp them, to visualize them.

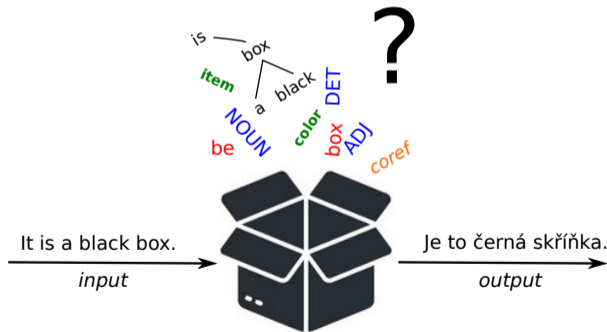
Motivation

- Deep neural networks have rapidly become a central component in many NLP systems.
- They do not have any explicit knowledge of linguistic abstractions.
- End-to-end-trained models are black boxes that are very hard to interpret.

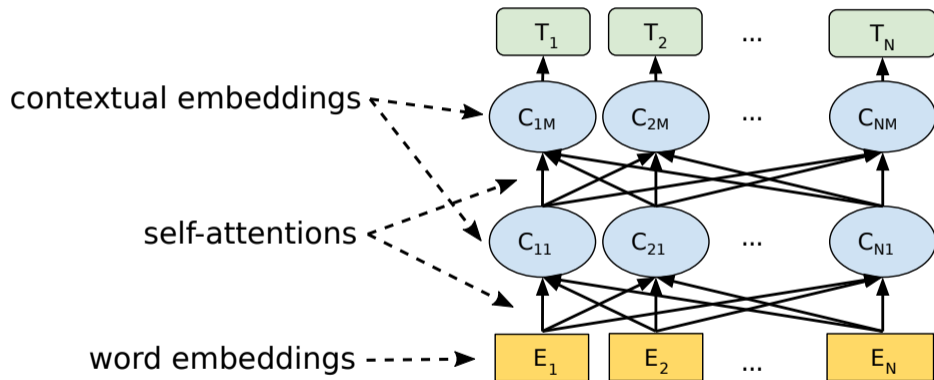


Motivation

- How do they work? What emergent abstractions can we observe in them?
- How can we interpret their internal representations?
- Are the emergent structures and abstractions similar to classical linguistic structures and abstractions?



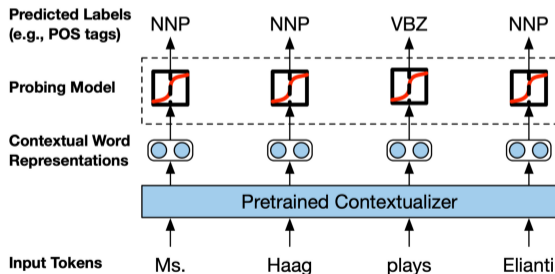
What We Can Analyze



Methods

Supervised Methods: Probing

- Training supervised classifiers predicting linguistic features (e.g. POS tagger) on top of the internal representations.
- We assume that when probing classifier accuracy is high the networks encodes linguistic abstraction well.



Liu et al. (2019): "Linguistic Knowledge and Transferability of Contextual Representations"

Supervised vs. Unsupervised Methods

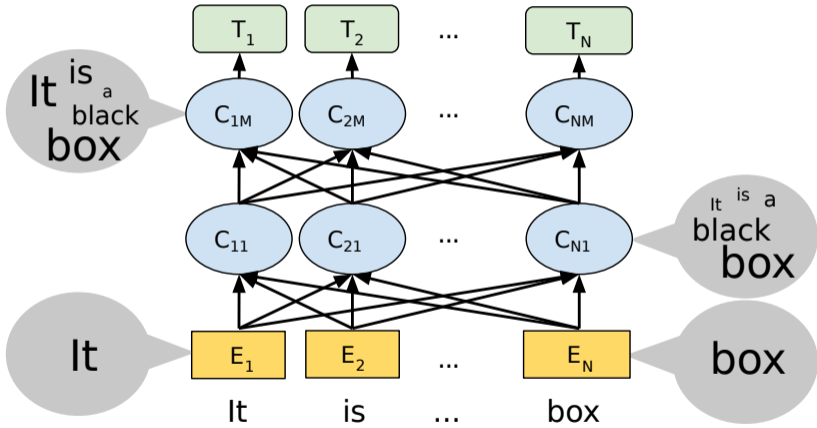
Supervised:

- Requires data annotated for the studied property.
- We can only reveal the kind of information that we have previously decided to look for.
- Retroactively affirms the correctness of the conceptualization and design decisions.

Unsupervised:

- Clustering, Component analysis, Structural induction from attentions
- We analyze the features that emerge in the representations, and only then we try to map them to existing linguistic abstractions.
- Complicated evaluation.

Words versus States



Words versus Subwords

The models typically operate on subword units.

Many linguistic formalisms are based on words (POS tags, dependency trees, ...)

- Words to Subwords – modify the linguistic abstraction to apply to subwords
- Subwords to Words – reconstruct word representations from the subword representations
- Fully Word-Based Approach – train the model on words

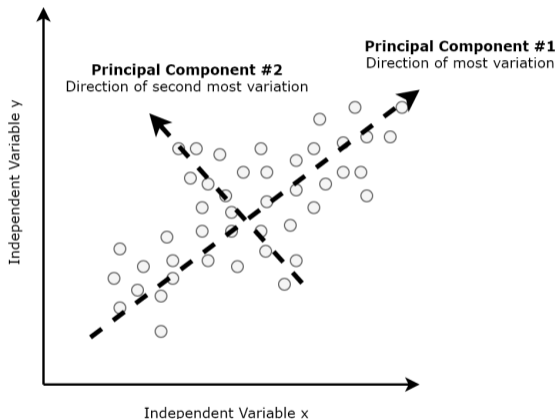
Word Embeddings

Word Embeddings

- A vector for each word (e.g. 100 dimensional, i.e. each word associated with a list of 100 real numbers)
- Learned in an unsupervised way from large plaintext corpora
- Observes the distributional hypothesis: words that appear in similar context have similar embeddings
- word2vec > RNN > attention > Transformer

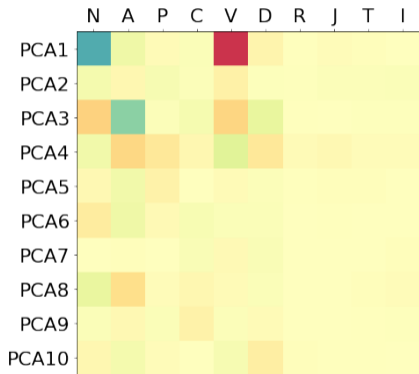
Principal Component Analysis (PCA)

- Transformation to another orthogonal basis set
- 1st principal component has the largest possible variance across the data
- Each other principal component is orthogonal to all preceding components and has the largest possible variance.
- If something correlates with the highest principal components its possibly very important for the NLP task.

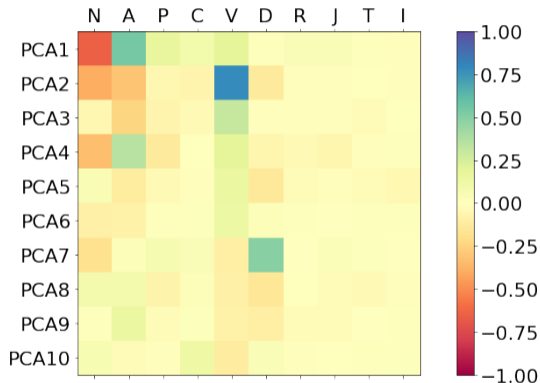


Word-embeddings learned by NMT, correlation with POS tags

encoder

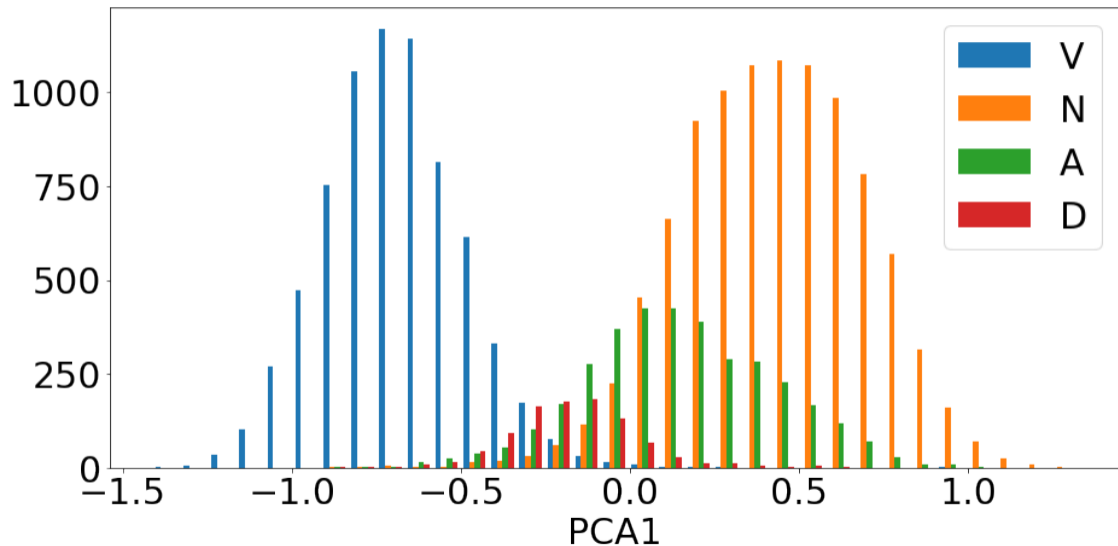


decoder

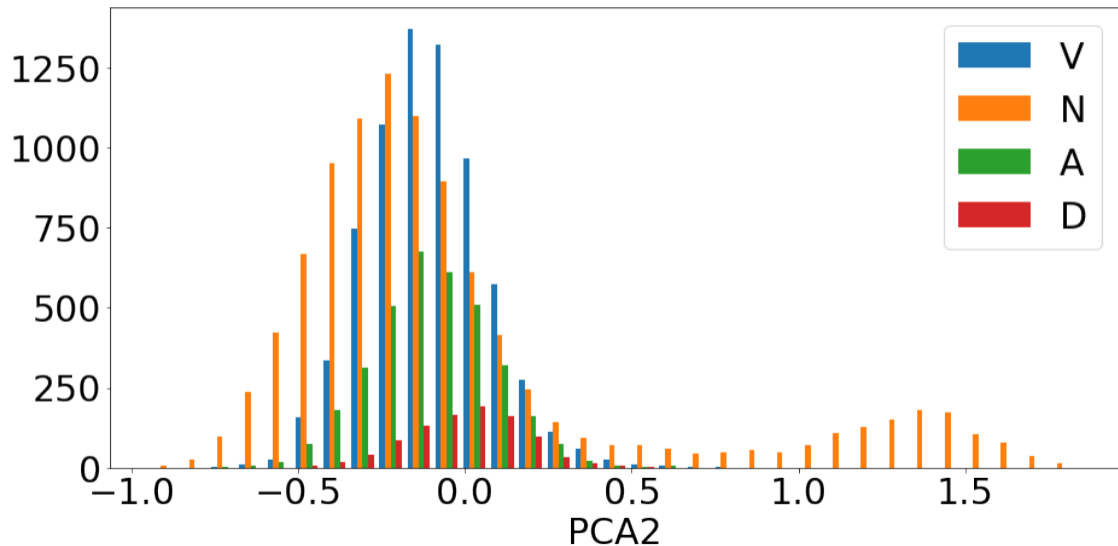


N = Nouns, A = Adjectives, P = Pronouns, C = Numerals, V = Verbs,
D = Adverbs, R = Prepositions, J = Conjunctions, T = Particles, I = Interjections

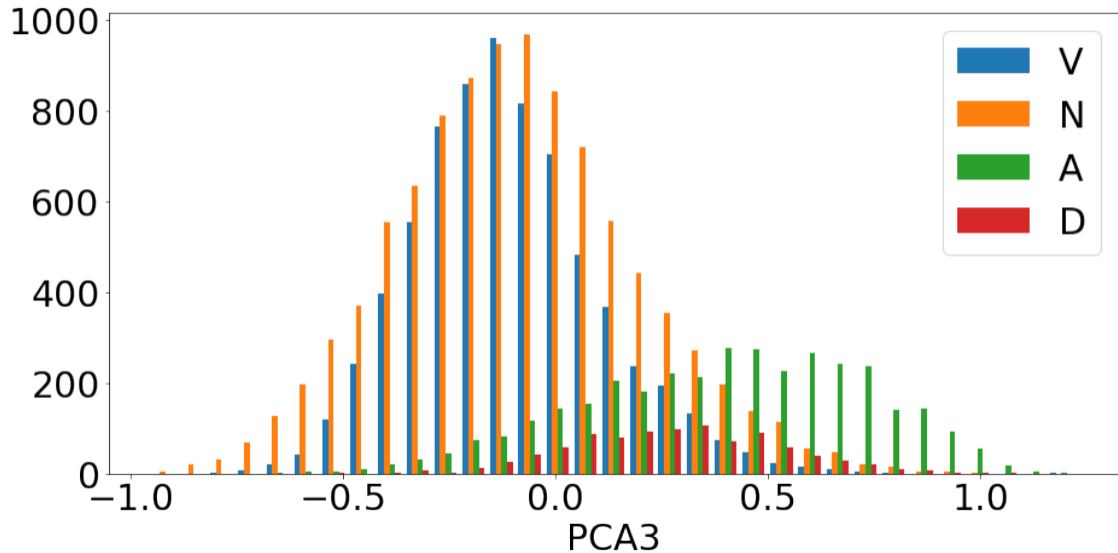
Word-embedding space learnt by NMT encoder



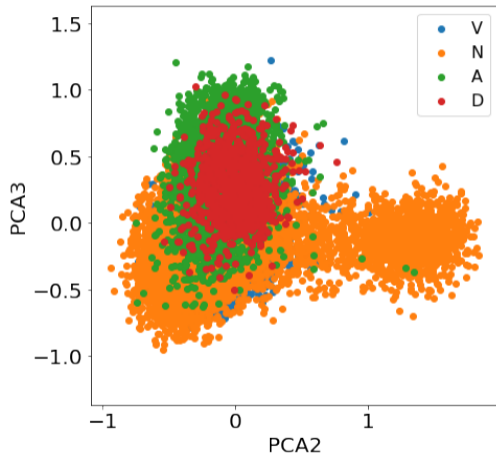
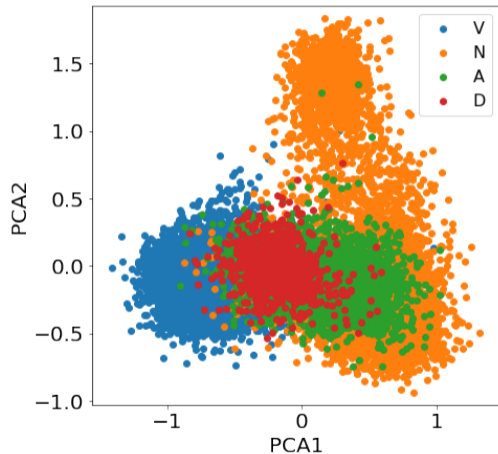
Word-embedding space learnt by NMT encoder



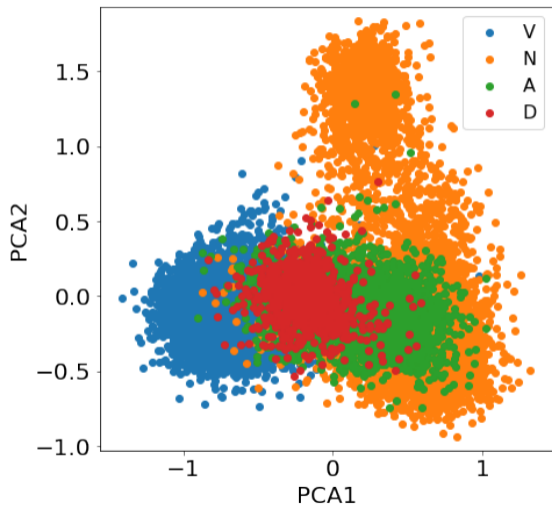
Word-embedding space learnt by NMT encoder



Word-embedding space learnt by NMT encoder



Word-embedding space learnt by NMT encoder



What is the separated island of Nouns visible in PCA2?

When we take a sample of words from this cluster, it contains almost exclusively named entities:

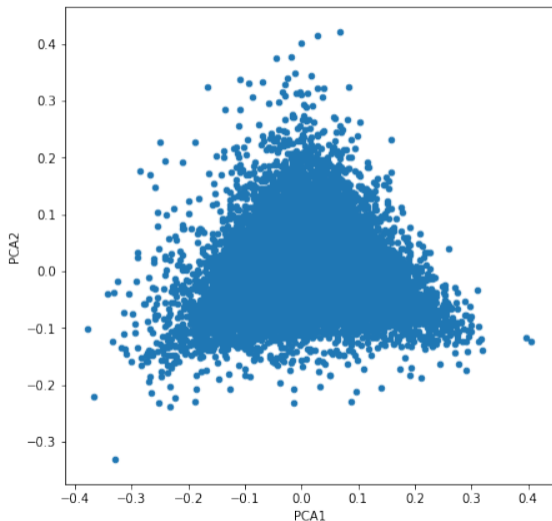
Fang, Eliáš, Još, Aenea, Bush, Eddie, Zlatoluna, Gordon, Bellondová, Hermiona

Word-embedding space learnt by Sentiment Analysis

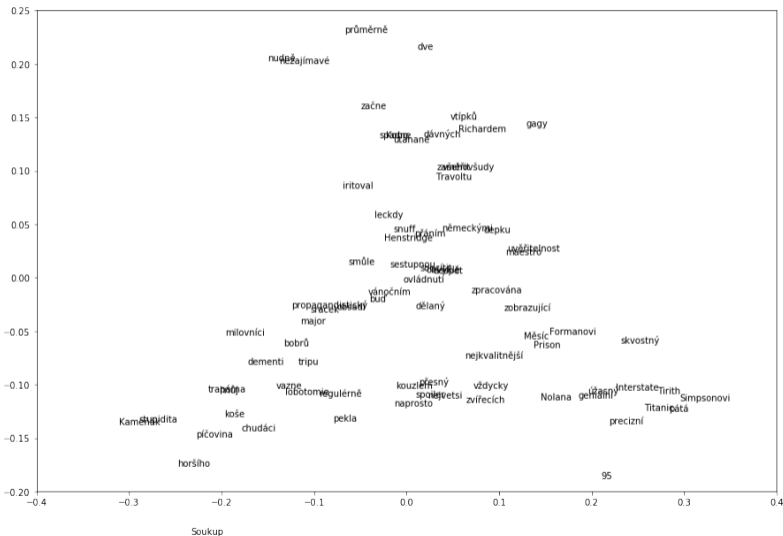
- Task: deciding whether a given text is emotionally positive, negative, or neutral.
- Trained on Czech ČSFD database (<https://www.csfd.cz/>), data were obtained from user comments and rankings of movies.
- Architecture: Convolutional neural network based on Kim 2014.

Neg: *"Very boring. I felt asleep."*

Pos: *"Great movie with super effects!!!"*

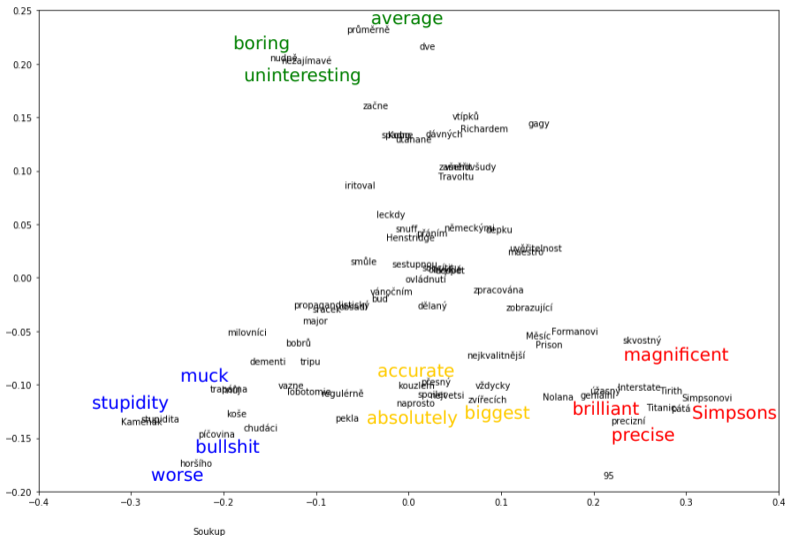


Word-embedding space learnt by Sentiment Analysis



We sampled some words from the vector space...

Word-embedding space learnt by Sentiment Analysis

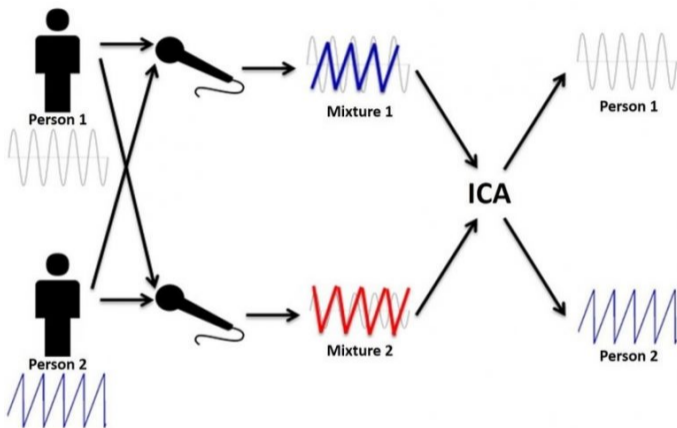


↔ ... polarity of the word

↕ ... intensity of the word

Word embedding space is shaped by the task for which it is trained.

Independent Component Analysis



- Iteratively find components that are as non-gaussian as possible

Independent Component Analysis

Semantic category: words with similar semantic content (e.g., law and justice) from various syntactic categories (in this case predominantly nouns in nominative and genitive morphological case):

zákona, Unie, členských, zákon, stanoví, Komise, zákony, soud, zákonů, zákonem, Evropské, práva, práv, ustanovení, nařízení, porušení, soudu, tj, souladu, podmínek

Glosses: *law*_{noun gen. sg.}, *union*_{noun nom. sg.}, *member*_{adj. gen. masc.}, *law*_{noun nom. sg.},
*determines*_{verb}, *committee*_{noun nom. sg.}, *laws*_{noun nom. pl.}, *court*_{noun nom. sg.}, *laws*_{noun gen. pl.},
*law*_{noun inst. sg.}, *european*_{adj. gen. fem. sg.}, *rights*_{noun nom. pl.}, *rights*_{noun gen. pl.}, *provision*_{noun sg.},
*regulation*_{noun sg.}, *violation*_{noun sg.}, *court*_{noun gen. sg.}, *ie*_{shortcut}, *compliance*_{noun gen. sg.},
*conditions*_{noun gen. pl.}

Independent Component Analysis

Semantic and syntactic category: words that are defined both semantically and syntactically, in this case, predominantly verbs associated with *going somewhere* in the past tense masculine:

šel, zašel, zajít, jít, spěchal, šla, zavedl, vešel, dopravit, nešel, vrátil, poslal, vydal, šli, poslat, přišel, odjel, přijel, jel, dorazil

Glosses: *went*_{verb masc.}, *went down*_{verb masc.}, *go down*_{verb inf.}, *go*_{verb inf.}, *hurried*_{verb masc.},
*went*_{verb fem.}, *led*_{verb masc.}, *entered*_{verb masc.}, *transport*_{verb inf.}, *didn't go*_{verb masc.},
*returned*_{verb masc.}, *sent*_{verb masc.}, *issued*_{verb masc.}, *went*_{verb masc. pl.}, *send*_{verb inf.}, *came*_{verb masc.},
*left*_{verb masc.}, *came*_{verb masc.}, *went*_{verb masc.}, *arrived*_{verb masc.}

Independent Component Analysis

Syntactic subcategory: words with specific syntactic features, but semantically diverse (in this case, adjectives in feminine singular form):

Velká, moudrá, občanská, dlouhá, slabá, čestná, železná, překrásná, hladká, určitá, marná, tmavá, hrubá, příjemná, bezpečná, měkká, svatá, nutná, volná, zajímavá

Glosses: *big*_{adj. fem.}, *wise*_{adj. fem.}, *citizen*_{adj. fem.}, *long*_{adj. fem.}, *weak*_{adj. fem.}, *honest*_{adj. fem.}, *iron*_{adj. fem.}, *beautiful*_{adj. fem.}, *smooth*_{adj. fem.}, *certain*_{adj. fem.}, *in vain*_{adj. fem.}, *dark*_{adj. fem.}, *gross*_{adj. fem.}, *pleasant*_{adj. fem.}, *safe*_{adj. fem.}, *soft*_{adj. fem.}, *holy*_{adj. fem.}, *necessary*_{adj. fem.}, *free*_{adj. fem.}, *interesting*_{adj. fem.}

Independent Component Analysis

Feature across POS categories: e.g., feminine plural form for adjectives, pronouns and verbs:

tyto, tyhle, neměly, byly, mohly, začaly, vynořily, zmizely, měly, objevily, všechny, vypadaly, nebyly, zdály, změnilly, staly, takové, podobné, jiné, tytéž

Glosses: *these*_{pron. fem. pl.}, *those*_{pron. fem. pl.}, *didn't have*_{verb fem. pl.}, *were*_{verb fem. pl.},
*could*_{verb fem. pl.}, *began*_{verb fem. pl.}, *emerged*_{verb fem. pl.}, *disappeared*_{verb fem. pl.}, *had*_{verb fem. pl.},
*discovered*_{verb fem. pl.}, *all*_{pron. fem. pl.}, *looked*_{verb fem. pl.}, *weren't*_{verb fem. pl.}, *seemed*_{verb fem. pl.},
*changed*_{verb fem. pl.}, *happened*_{verb fem. pl.}, *such*_{pron. fem. pl.}, *similar*_{adj. fem. pl.}, *other*_{adj. fem. pl.},
*same*_{pron. fem. pl.}

Independent Component Analysis

Stylistic: in this case, words that often appear in informal spoken language (often second person verbs and colloquial forms):

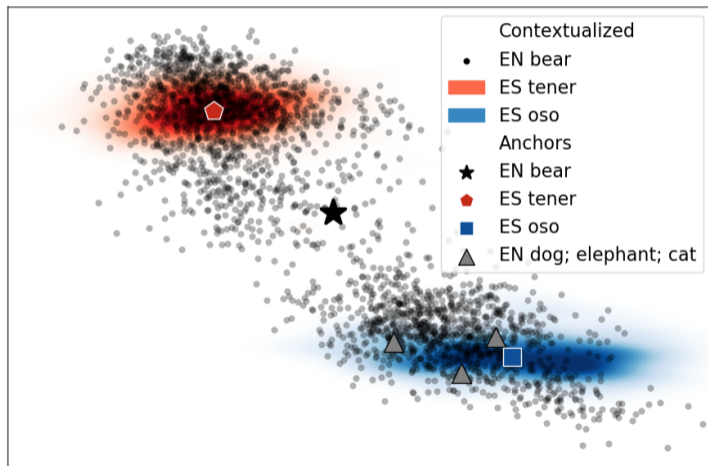
máš, bys, tý, nemáš, seš, ses, víš, Hele, kterej, sis, jseš, bejt, vo, svýho, celej, děláš, chceš, teda, každej, velkej

Glosses: *have*_{verb 2nd}, *would*_{verb 2nd}, *the*_{pron. fem. gen. coll.}, *don't have*_{verb 2nd}, *are*_{verb 2nd coll.}, *have*_{verb 2nd refl.}, *know*_{verb 2nd}, *Hey*_{intj. coll.}, *which*_{pron. masc. coll.}, *have*_{verb 2nd refl.}, *are*_{verb 2nd coll.}, *be*_{verb inf. coll.}, *about*_{prep. coll.}, *your*_{pron. masc. gen. coll.}, *whole*_{adj. masc. coll.}, *do*_{verb 2nd}, *want*_{verb 2nd}, *well*_{part. coll.}, *each*_{pron. masc. coll.}, *big*_{adj. masc. coll.}

Contextual Embeddings

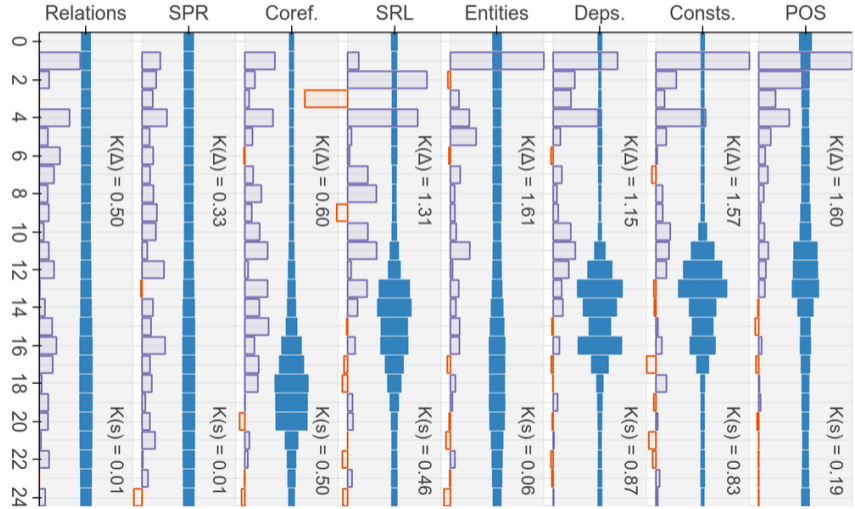
Word senses in Contextual Embeddings

Schuster et al. (2019): ELMo contextual embeddings cluster according to the various senses of the word.



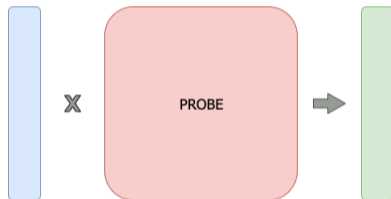
Linguistic Features in Contextual Embeddings

Tenney et al. (2017): Different linguistic features captured at different layers of BERT.

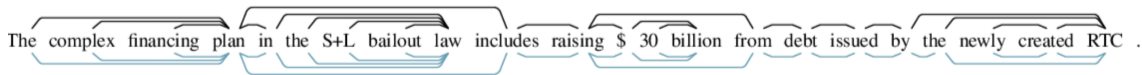


Probing for Distances in Syntactic Trees

Hewitt and Manning (2019) took the BERT contextual vectors and trained a projection matrix to obtain another vectors whose differences would approximate distances between tokens in dependency trees.



Dependency trees obtained by Minimum spanning tree gained 82.5% UAS on English PTB.



Probing for Depths in Syntactic Trees

... the same but with depths in dependency trees.



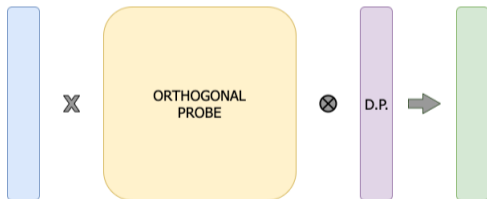
Contextual Embeddings

Separating lexical and syntactic features

Orthogonal Probing

We decompose the trained projection matrix into two matrices:

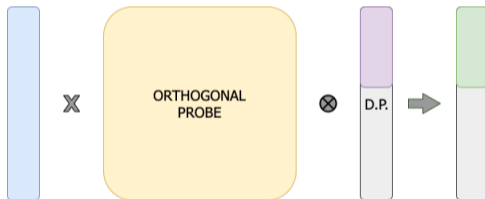
- orthogonal matrix: only rotates the vector space
- diagonal matrix: assigns weights to individual dimensions - how important they are for the probing task



Orthogonal Probing

Observation:

- Many weights trained in the diagonal matrix are close to zero.
- This method shows us which dimensions of the rotated space are useful for the probing task, e.g., syntax



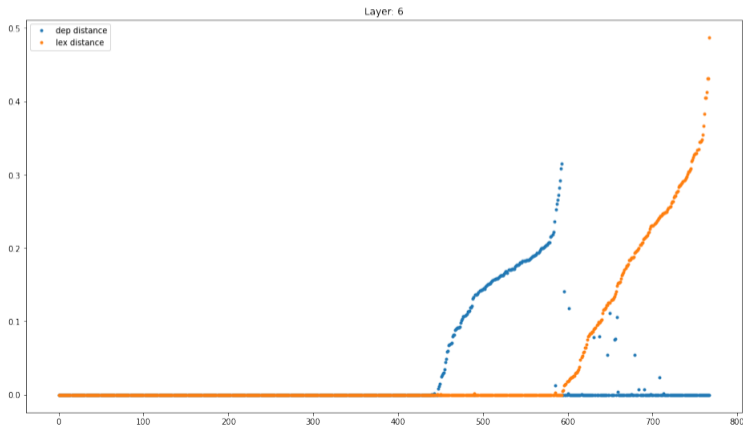
Idea:

- Probing for more tasks at once with shared orthogonal matrix.
- Could we separate the dimensions needed for specific tasks?

Separating Lexical and Syntactic Features

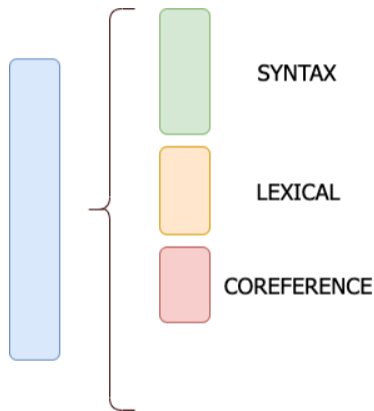
We are probing for distances between two words

- in dependency tree (syntactic features)
- in the WordNet hyperonymic tree structure (lexical features)



Orthogonal Probing - Conclusions

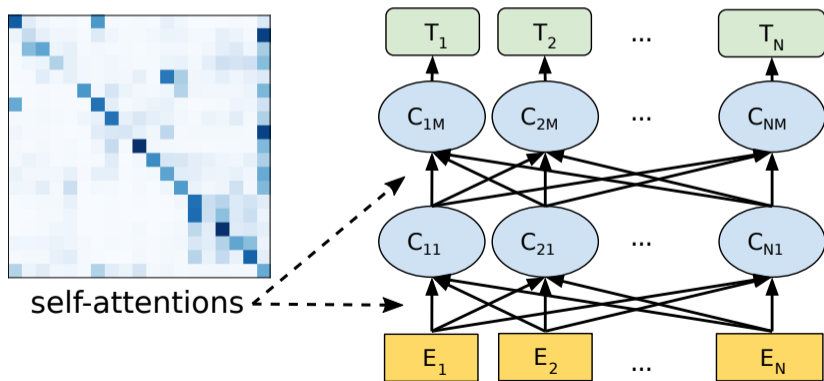
- We are able to identify syntactic and lexical dimensions of the BERT representations.
- We only need to rotate the vector space to transform the features hidden in linear combinations of dimensions into single dimensions.
- What is hidden in the rest of the dimensions?



Self-Attentions and Syntax

Self-Attentions in Transformer

Self-attentions: Weighted connections between word representations showing how much a word representation in one layer contributes to another word representation in the following layer.



Self-Attentions in Transformer – The Goals

We analyze self-attention weight matrices in:

- Neural Machine Translation Transformer Encoder (16 heads x 6 layers)
- BERTbase pre-trained model (12 heads x 12 layers)

To what extent attentions between individual (sub)word representations correspond to syntactic features?

Is it possible to extract syntactic structures from them?

Self-Attentions and Syntax
Transformer NMT Encoder and Phrase
Trees

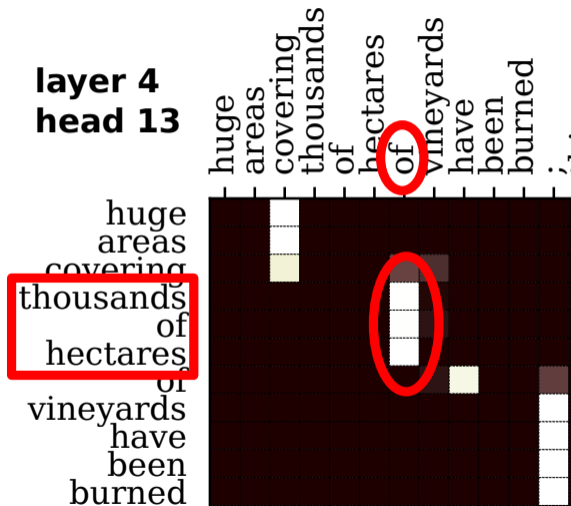
Self-Attentions in NMT Encoder and Phrase Trees

Observation: Common pattern in cca 70% of self-attention heads: “balustrades”

- Baluster: continuous sequence of words attending to the same position
- Looks like a syntactic phrase
- Usually attends to phrase boundary

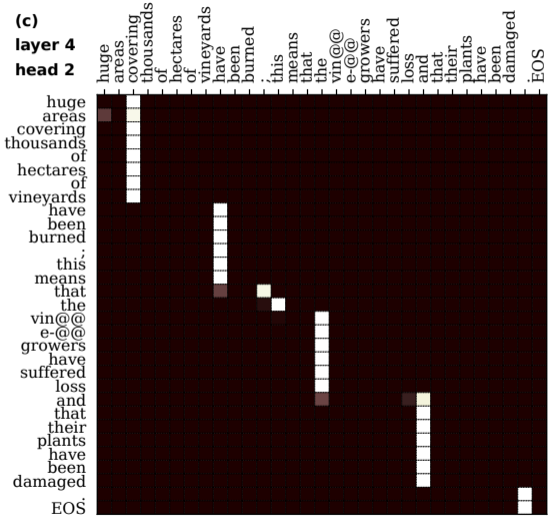
Research questions:

- Is that syntactic? To what extent?
- Could we extract phrase trees from attentions?
- How they differ from manually annotated phrase-trees?

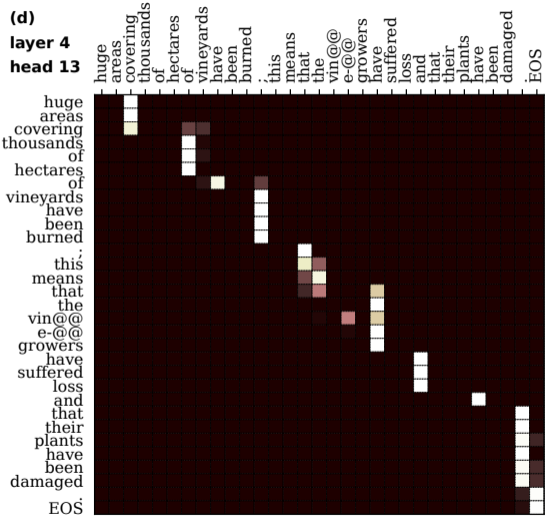


Examples of Heads with Balustrades

(c)
layer 4
head 2



(d)
layer 4
head 13



Approach and Results

1. Transformer NMT: French \leftrightarrow English, German \leftrightarrow English, French \leftrightarrow German
2. Phrase Scores:
 - based on the attention weights of “balusters”
 - collected and averaged over all heads and layers in the Encoder
3. Binary constituency trees:
 - linguistically uninformed algorithm
 - tree score = sum of phrase scores
 - CKY: find tree with maximal score

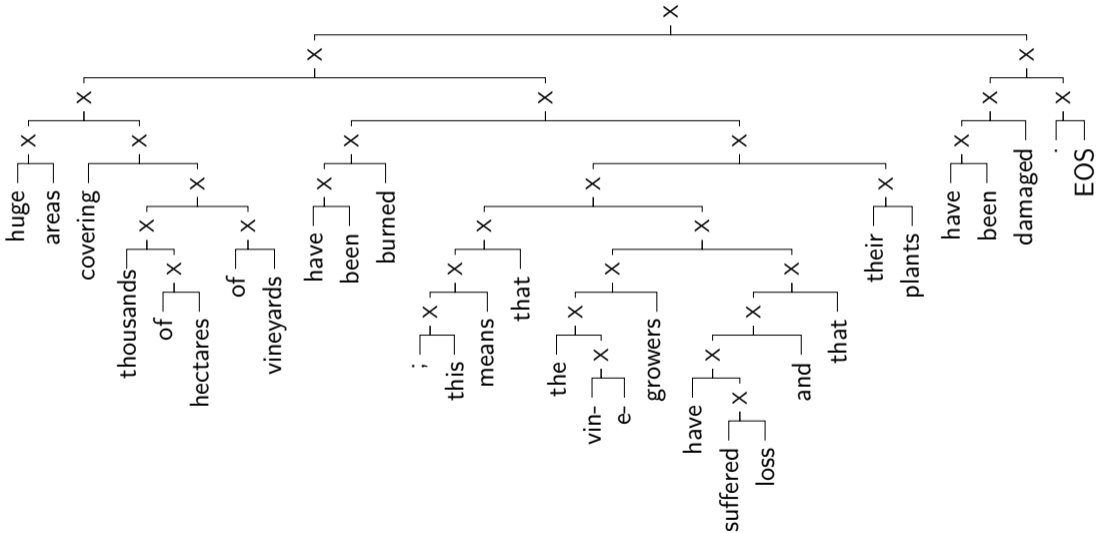
Comparison to standard constituency syntactic trees:

- we observe a 40% match, baseline has a 30% match (right-aligned balanced binary tree)

Analysis:

- The emergent structures can be seen as syntactic to some extent
- Shorter phrases are often captured
- Sentence clauses are often captured

Example of Tree



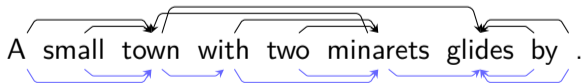
Self-Attentions and Syntax
BERT Model and Dependency Relations

BERT Model and Dependency Relations

Many previous work showed that individual BERT attention heads tend to encode particular dependency relations.

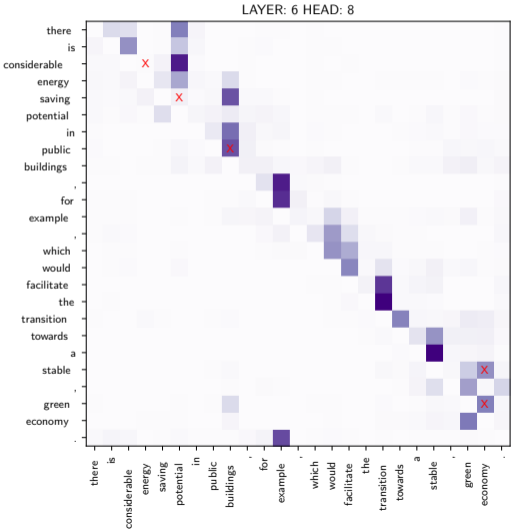
Our contributions:

- Some heads are more abstract (include more dependency relations)
- Some heads are more specific (separate one relation type into more subtypes)
- We show a method how to extract labeled dependency trees (52% UAS, 22% LAS on English UD).

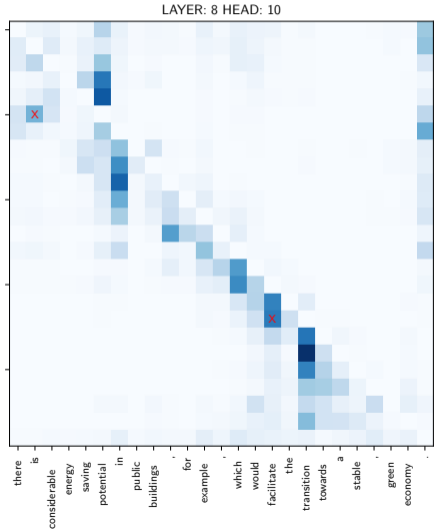


BERT model and Dependency Relations

Self-attention in a particular heads of a language model aligns with dependency relations



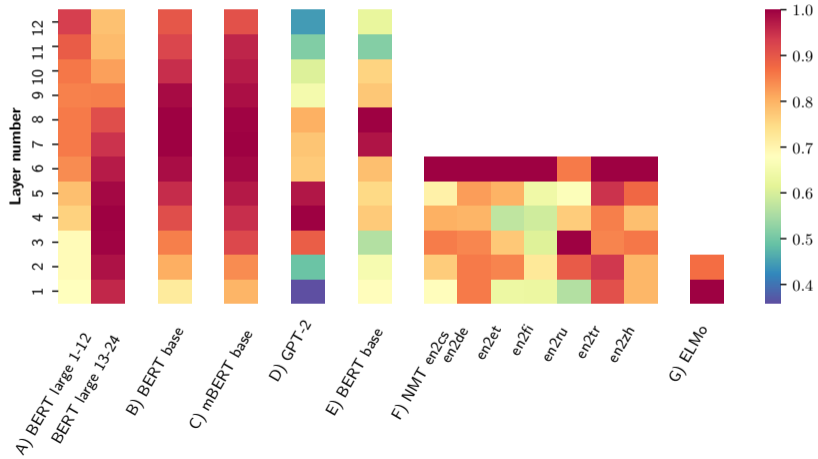
AMQD D2P



OBJ D2P

Syntactic Accuracy Across Layers

Relative syntactic information across attention models and layers



Summary

1. Advantages of unsupervised methods.
2. Word embeddings capture morphological features.
3. In contextual embeddings, the lexical and syntactic information can be separated.
4. Constituency phrases captured in NMT.
5. Dependency relations captured by individual self-attention heads.

<https://ufal.cz/courses/npfl1097>

HIDDEN IN THE LAYERS

Interpretation of Neural Networks for Natural Language Processing

David Mareček, Jindřich Libovický, Tomáš Musil, Rudolf Rosa, Tomasz Limisiewicz

http://ufallab.ms.mff.cuni.cz/~musil/lldbok/lldbok_final.pdf