Searching for Linguistic Structures in Neural Networks

David Mareček

University of Helsinki, September 12th, 2019

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

Outline

My Research Story

Inspecting Word Embeddings using PCA

Looking for Syntax in Transfomer's Self-Attentions

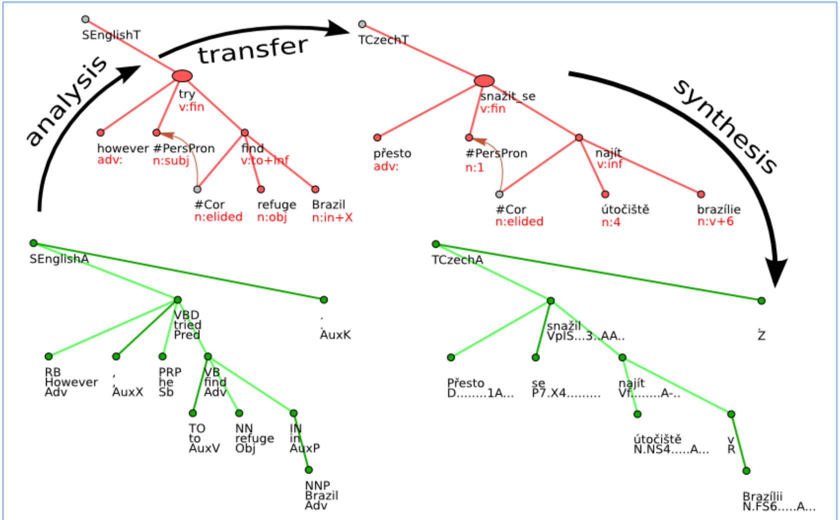
My Research Story

Well, I am from Prague ...

I started with tree-based MT

2008 – 2011

- TectoMT system
- English → Czech syntax-based MT system
- source language parsing
- alignment of syntactic trees



TectoMT: a peek under the hood

Supervised Dependency Parsing

2008 – 2011

- I believed that parsing of a natural language is the key element for every NLP application (*machine translation, question answering, natural language understanding*).
- But even though we worked hard and kept improving our syntax-based MT system, the phrase-based MT system Moses was still better and was improving faster.
- Why? Where was the problem? Maybe the annotations based on linguistic theories were not suitable for MT?
- Moreover, there were substantial differences in annotation styles of various language treebanks (before Universal Dependencies).

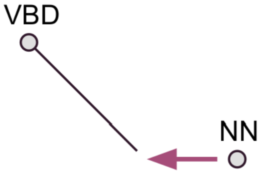
Unsupervised Dependency Parsing

2011 – 2016

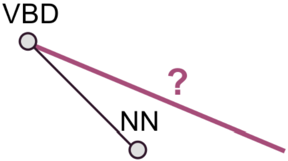
- What about unsupervised learning of trees?
- It was very popular in that time to make anything in unsupervised fashion.
- Parsing without any manually annotated treebanks.
- Not burdened by any linguistic theories, language universal.
- And maybe the tree structures learned by unsupervised parsers are more suitable for NLP tasks.

Unsupervised Dependency Parsing

- Dependency model with Valence (DMV)
  - generative model, which is able to generate all possible projective dependency trees
- Gibbs sampling
  - random initialization of the trees
  - select one sentence, and train the model (collect counts) on all other trees in the corpus
  - sample a new tree on that sentence based on the model
  - repeat until convergence
- This produced quite nice dependency trees.
- But it didn't worked well when used directly in NLP tasks.



P_{choose}(NN | VBD, right)

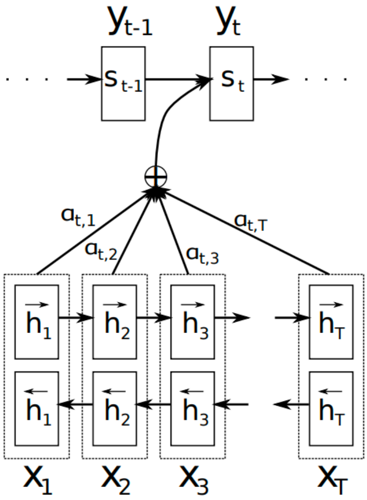


P_{stop}($\neg$STOP | VBD, right, 0)

Advance Neural Machine Translation

since 2014

- Moses system outperformed by NMT
- RNN based NMT use "memory" that could keep information about any other word in the sentence. It is therefore able to learn syntactic relations.
- Transformer NMT uses self-attentions, in which any word can look anytime at any other word in the sentence.
- These approaches can easily learn a kind of latent unsupervised structure of the sentence tailored exactly for the machine translation task.



LSD project

since 2018

Linguistic **S**tructure representation in **D**eep networks

- National Science Foundation of Czech Republic
- 2018 – 2020

Goals:

- Word embeddings and DNNs perform great.
- They do not have any explicit knowledge linguistic abstractions.
- How do they work? What abstractions can we observe in them? How do we interpret them?
- Are the emergent structures similar to classical linguistic structures?

LSD team

David Mareček

Jindřich Libovický

Rudolf Rosa

Tomáš Musil

Selected LSD results

Inspecting Word Embeddings using Principal Component Analysis (Musil, 2019)

- What features are important for word embeddings of various NLP tasks?

Derivational Morphological Relations in Word Embeddings (Musil et al., 2019)

- Unsupervised clustering of word-embedding differences captures derivational relations.

Neural Networks as Explicit Word-Based Rules (Libovický, 2019)

- We interpret a convolutional network for sentiment classification as word-based rules.

Looking for Syntax in Transformer Self-Attentions (Mareček and Rosa, 2019, 2018)

- Building constituency trees from multi-head self-attentions.

Inspecting Word Embeddings using PCA

Idea

- Assume a word-embedding vector space learned by a neural network solving some NLP task (machine translation, sentiment analysis, NLU, word2vec)
- *Question:* Do the embedding vectors encode linguistic features like part-of-speech, gender, number, tense, named entities, derivational relations, etc?
- We would like to get something like: 2nd position encodes grammatical number, 14th position encodes abstractness, 138th position encodes colours of objects, etc.
- Not as simple:
  - Each training ends up with complete different embeddings.
  - Possible linguistic features may correlate with any linear combination.

Probing

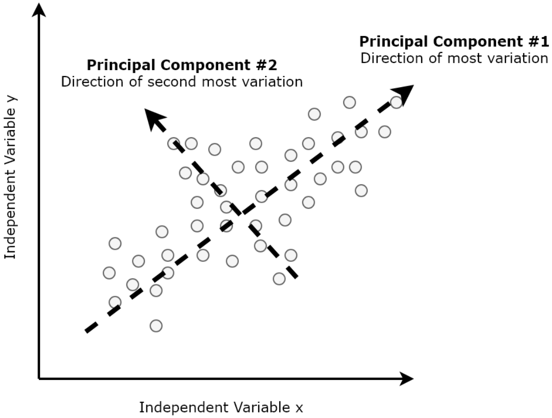
- Multilayer perceptron predicting POS from word-embeddings
- Supervised training on annotated data
- Disjoint train and test vocabularies

model	accuracy	dev.
NMT RNN encoder	94.69 %	± 0.93 %
NMT RNN decoder	**97.77 %**	± 1.16 %
NMT Transformer encoder	96.37 %	± 1.49 %
NMT Transformer decoder	93.36 %	± 3.86 %
word2vec	95.01 %	± 1.94 %

- But does the network really needs part-of-speech?
- Or it only learns it somehow form many other more important features?

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.

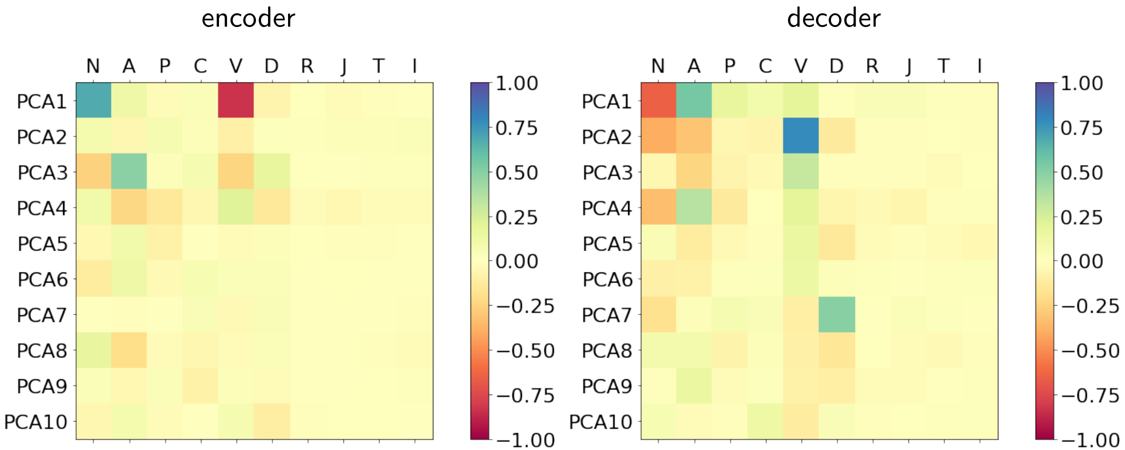


Experiment setting

We experiment with several different Czech word-embeddings spaces:

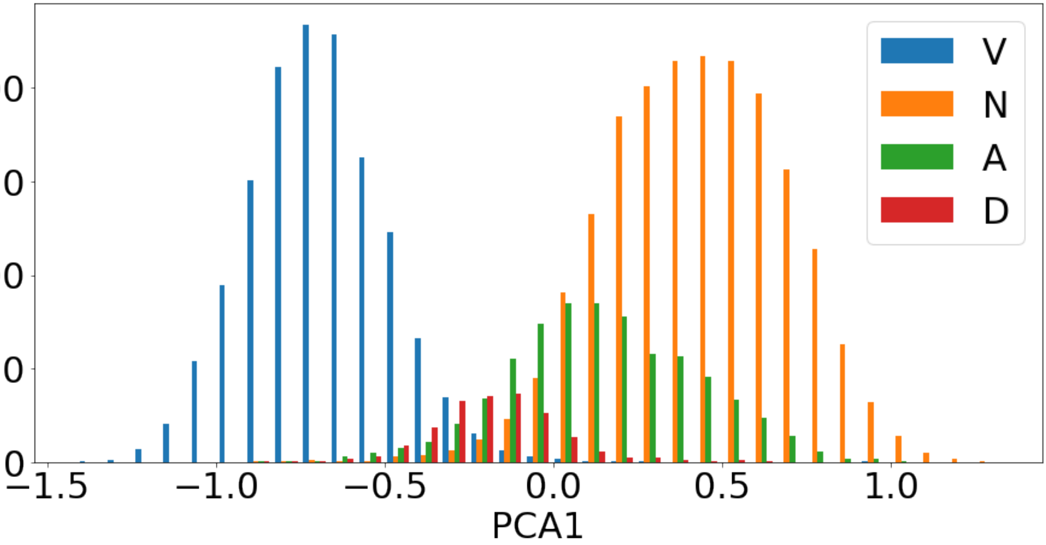
- Neural machine translation (Sutskever et al., 2014), with LSTM cells with hidden-state size of 1024, word-embedding dimension 512, trained on English $\leftrightarrow$ Czech fiction data. We examine both Czech encoder and Czech decoder embeddings.
- Word2Vec (Mikolov et al., 2013), with dimension 512, trained on the Czech side of the same parallel corpus, window size 11, negative sampling 10
- Sentiment analysis on Czech, CNN trained on ČSFD database of movies user comments and rankings

Word-embeddings learned by NMT, correlation with POS tags

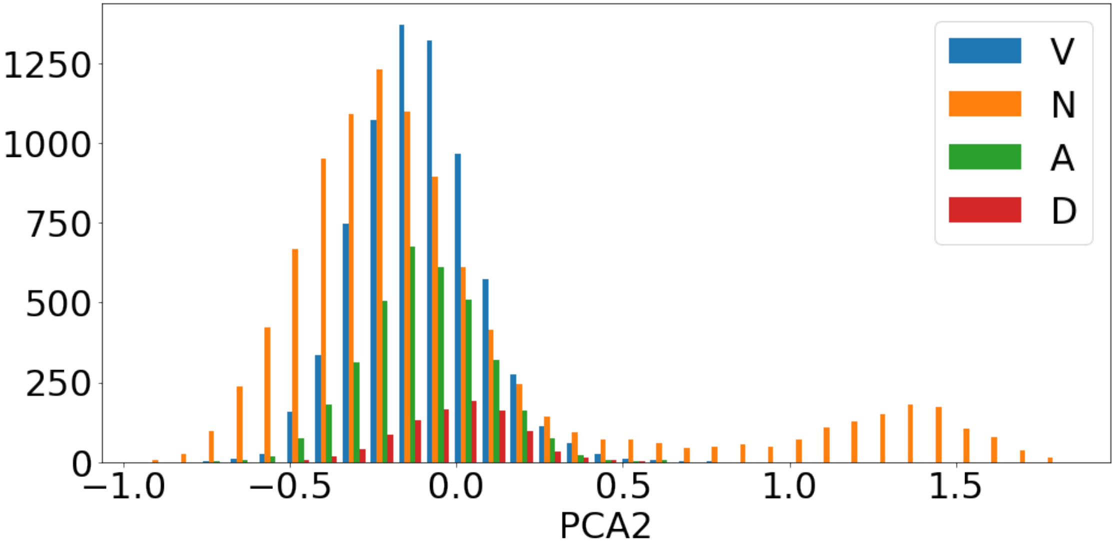


N = Nouns, A = Adjectives, P = Pronouns, C = Numerals, V = Verbs,
D = Adverbs, R = Prepositions, J = Conjunctions, T = Particles, I = Interjections

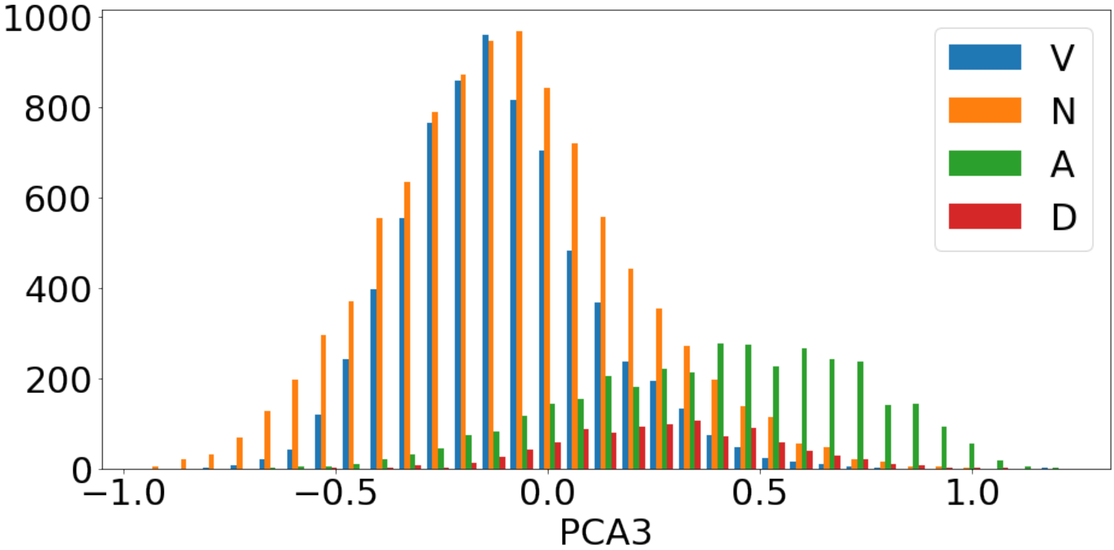
Word-embedding space learnt by NMT encoder



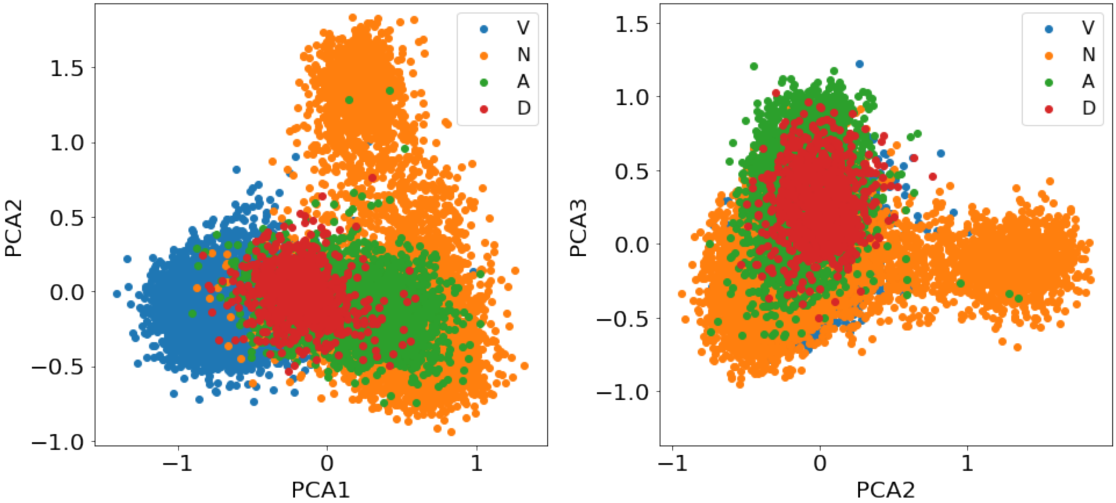
Word-embedding space learnt by NMT encoder



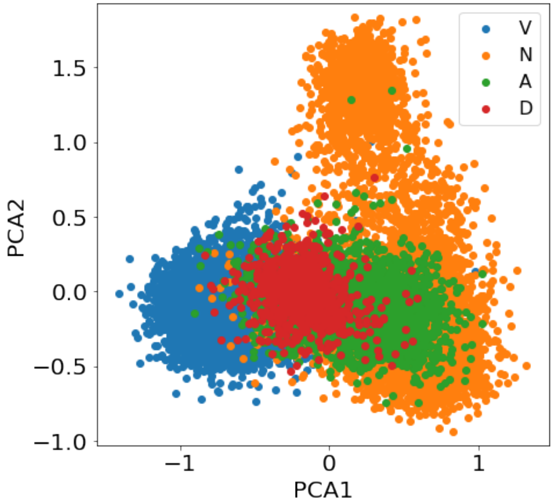
Word-embedding space learnt by NMT encoder



Word-embedding space learnt by NMT encoder



Word-embedding space learnt by NMT encoder



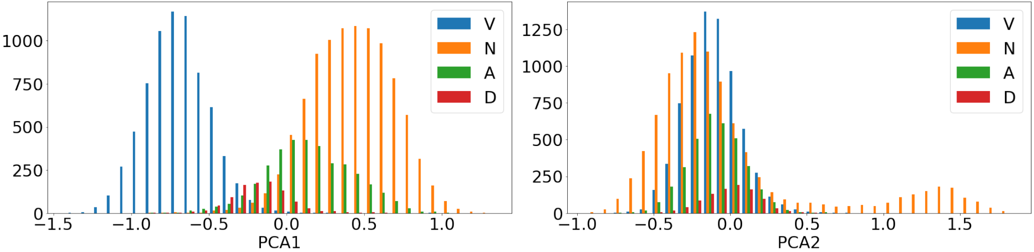
What is the separated island of Nouns visible in PCA2?

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

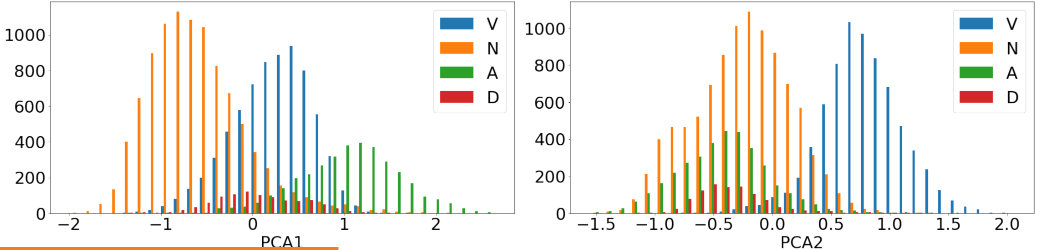
Fang, Eliáš, Još, Aenea, Bush, Eddie, Zlatoluna, Gordon, Bellondová, Hermiona

Differences between NMT encoder and decoder

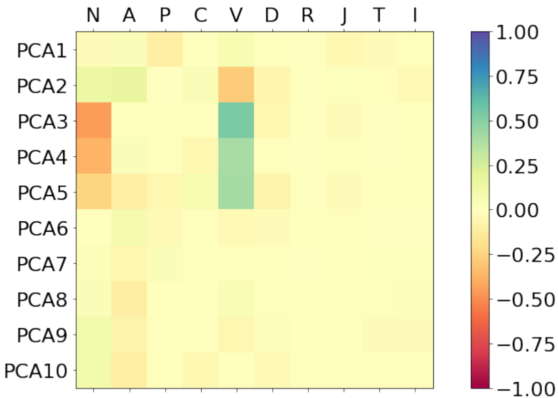
encoder:



decoder:

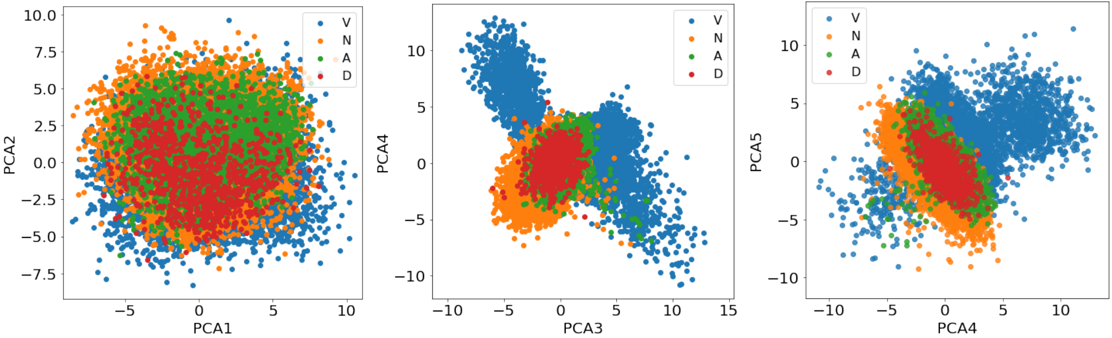


Word-embeddings learned by Word2Vec, correlation with POS tags



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

Word-embedding space learned by Word2Vec



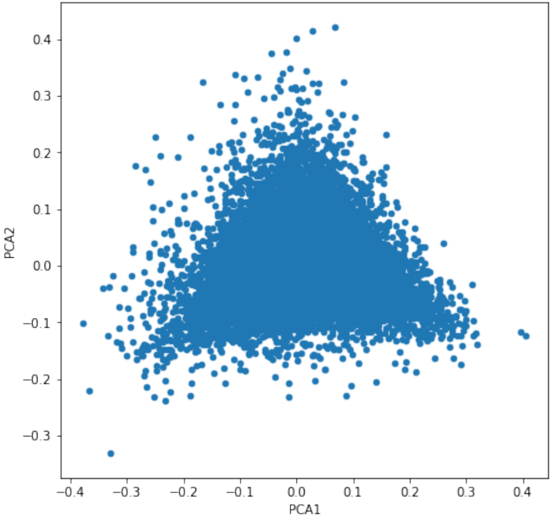
- Different structure than the NMT embeddings.
- PCA4 distinguishes infinitives and modal verbs.

Word-embedding space learnt by Sentiment Analysis

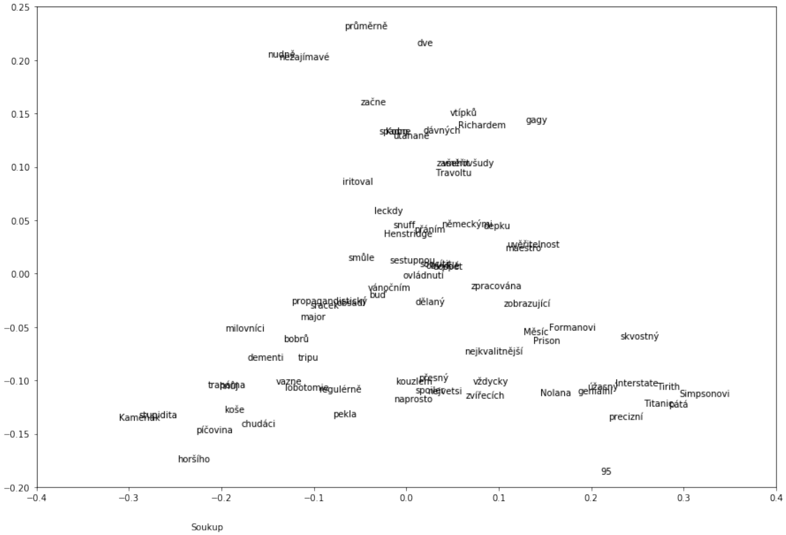
- Task: deciding whether a given text is emotionally positive, negative, or neutral.
- Trained on Czech ČSFD database (`https://www.csfd.cz/`), data were obtained from user comments and rankings of movies.
- Architecture: Convolutional neural network based on Kim (2014).

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

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

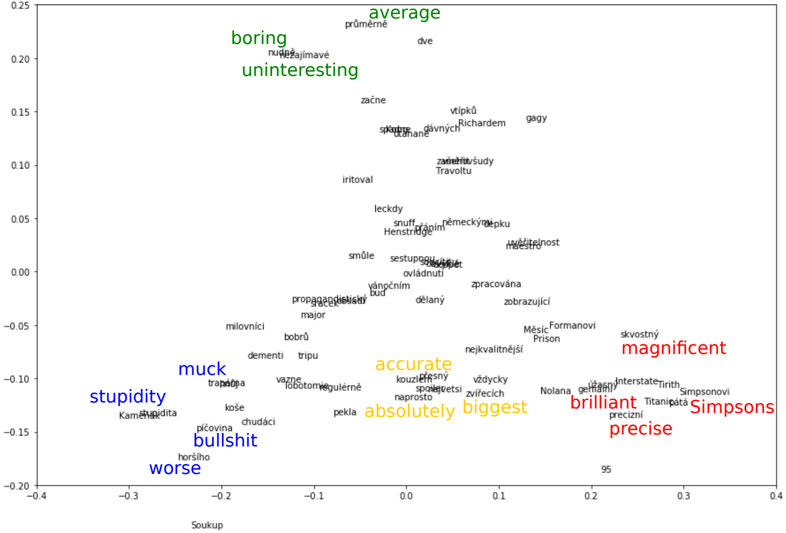


Word-embedding space learnt by Sentiment Analysis



We sampled some words from the vector space…

Word-embedding space learnt by Sentiment Analysis



↔ … polarity of the word

↕ … intensity of the word

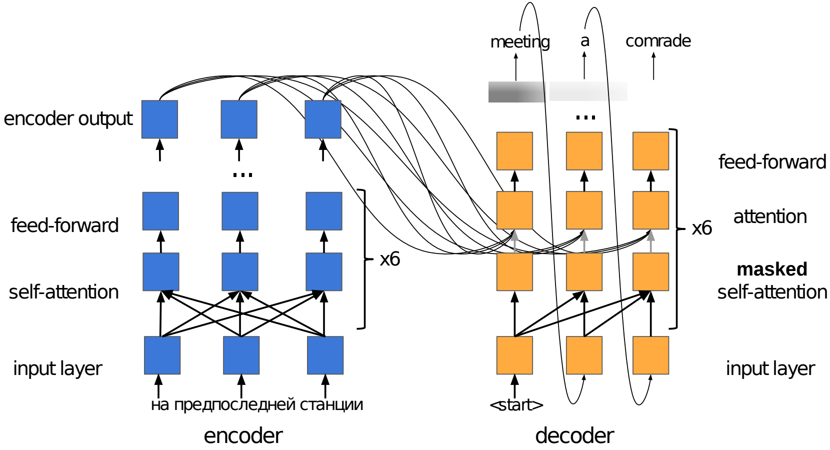
Word embedding space is shaped by the task for which it is trained.

Summary

- Examining histograms of classes along the principal component is important to understand the structure of representation of information in embeddings.
- NMT models distinguished verbs from nouns and adjectives very well and also represent named entities separately in the embedding space.
- word2vec distinguishes infinitive forms and modal verbs from the rest of the verbs.
- CNN sentiment analysis naturally models emotional properties of words in the shape of the embedding space.

Looking for Syntax in Transfomer's Self-Attentions

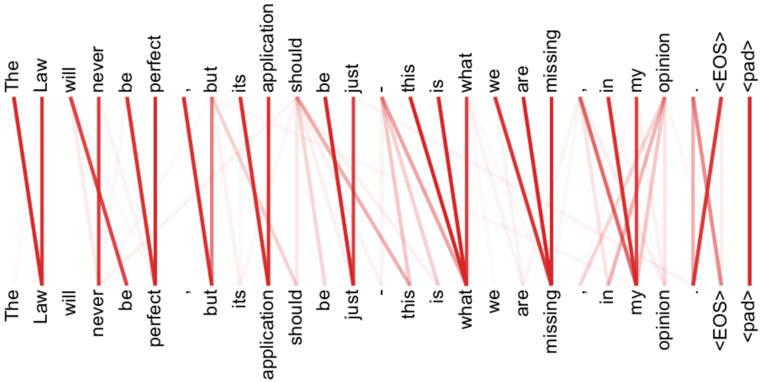
Transformer NMT



Source: https://research.jetbrains.org/files/material/5ace635c03259.pdf

Multi-headed self-attention mechanism

- Encoder has 6 layers, each one with 16 attention heads, i.e. 96 heads in total
- Each head may possibly look at all the positions (contextual representations of words) on the previous layer and returns a distribution of weights across the positions.
- But usually it looks at just one position.



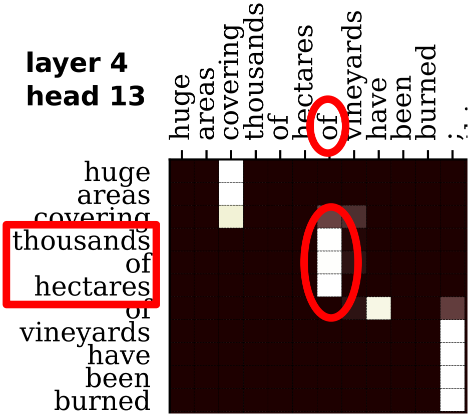
Source: Attention is all you need (Vaswani et al., 2017)

Observation

We visualize the (softmaxed) attention heads using matrices.

In many attention heads, we observe the following pattern:

- Continuous sequences of words attends to the same positions.
- They resemble syntactic phrases.
  - To what extent?
  - $\rightarrow$ That's our research question!



Experiment setup

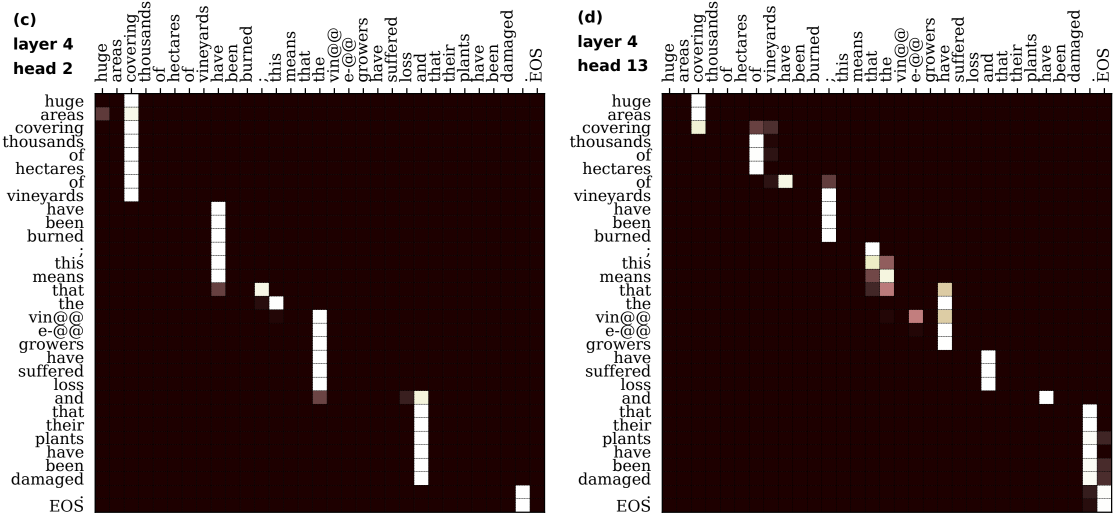
Transformer NMT system (Vaswani et al., 2017)

- Encoder: 6 layers x 16 heads
- Data from Europarl, 6 translation pairs
- French $\leftrightarrow$ English, German $\leftrightarrow$ English, French $\leftrightarrow$ German

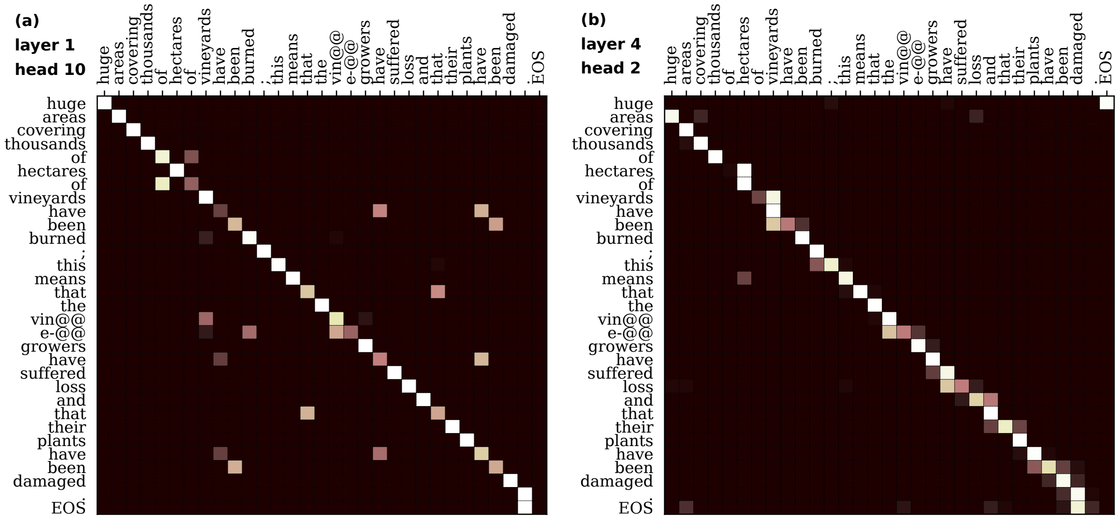
Test sentences are parsed by Stanford parser into contituency trees

- Penn Treebank, French Treebank, Negra Corpus
- used only for evaluation

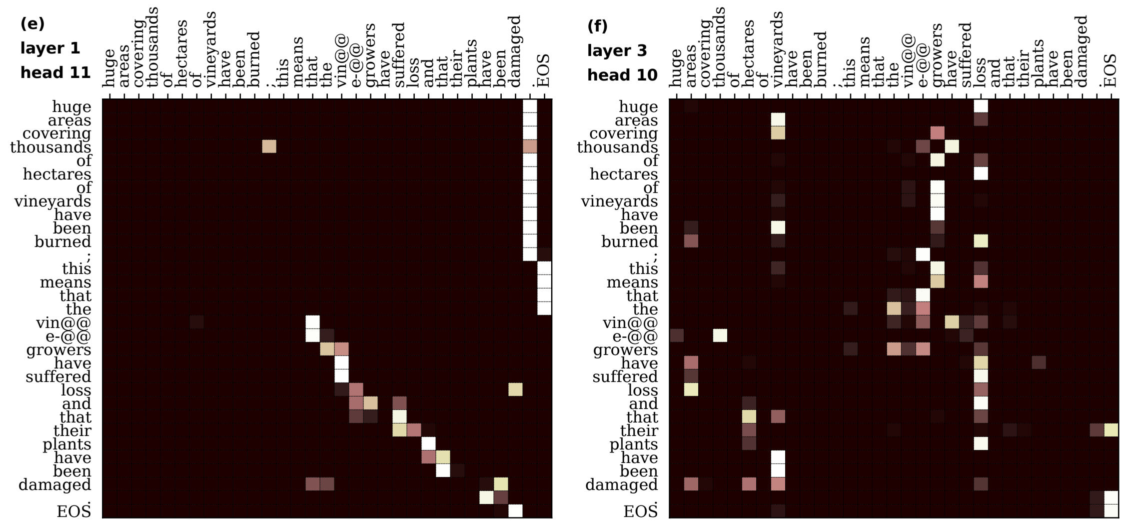
Balustrades (70% of the attention heads)



Diagonals (especially 1st layer)



The rest: attend to end, mixed, scattered...



Our Approach

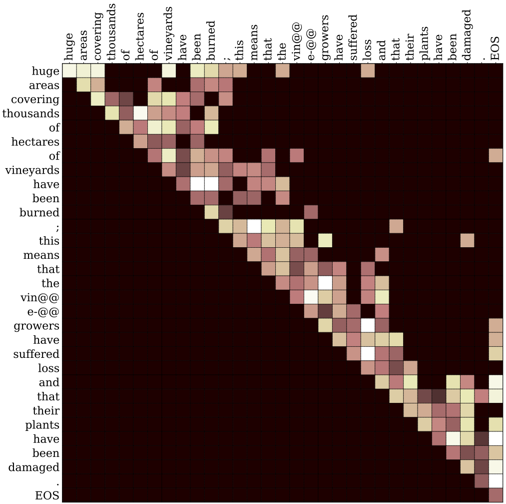
1. We collect the obtained phrases across all the heads and layers $\rightarrow$ phrase candidates
2. Using the phrase candidates, we build constituency trees
   - Linguistically uninformed algorithm
3. We compare our trees with the standard syntactic trees obtained by Stanford parser

Phrase candidates

We take all phrases (balusters) of length ≥ 2 from all 96 heads across layers.

For each possible phrase, we compute its score:

- Average attention weight
  - individual "pixels" of the phrases may have different weight
- Sum over all heads
  - the same phrase may appear in more attention heads
- Equalize over different phrase lengths
  - shorter phrases are more frequent



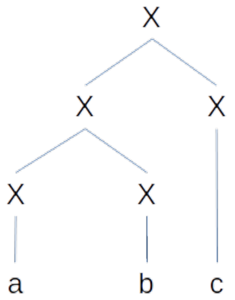
Parsing algorithm – CKY

We find the best binary constituency tree

- Tree score = the sum of scores of phrases used in the tree
- CKY algorithm (dynamic programing)
  - Finds constituency tree (set of phrases) with maximal score

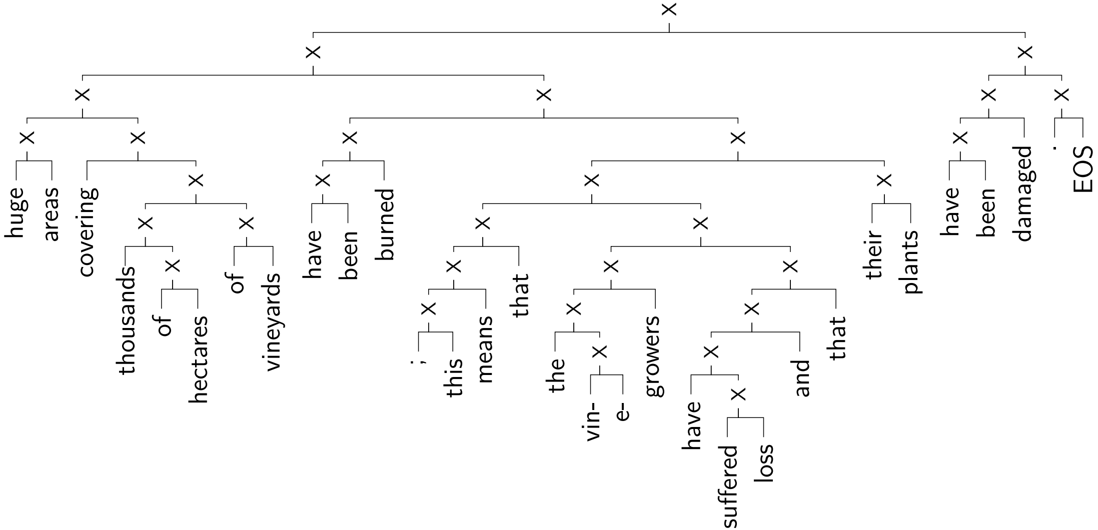
We measure F-score of the resulting trees against the "gold" trees obtained by Stanford parser.

We compare them with "balanced binary tree" baselines.

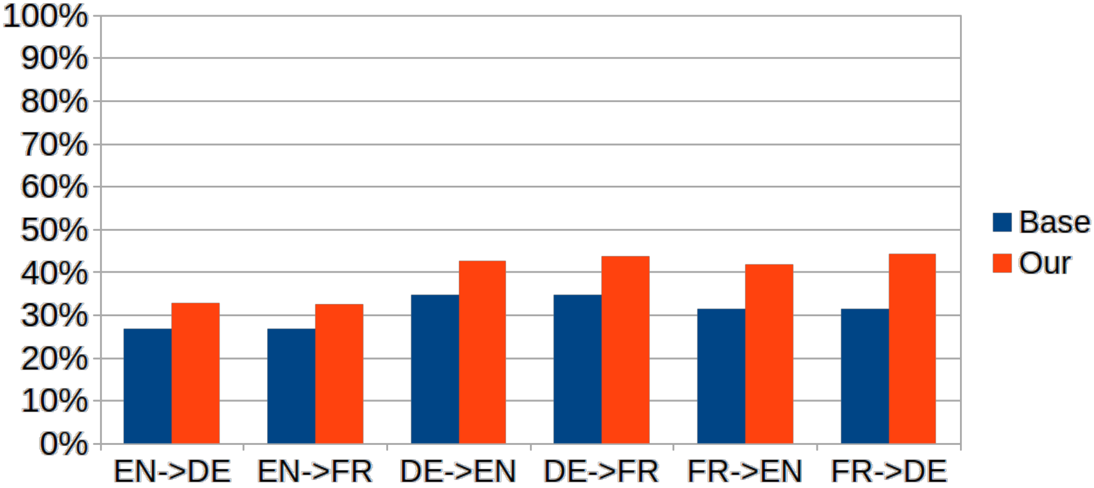


$s(T) = s(ab) + s(abc)$

Results


huge
areas
covering
thousands
of
hectares
of
vineyards
have
been
burned
;
this
means
that
the
vin-
e-
growers
have
suffered
loss
and
that
their
plants
have
been
damaged
.
EOS

Comparison to standard syntactic trees



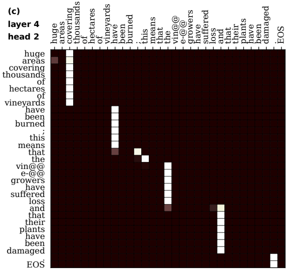
Summary

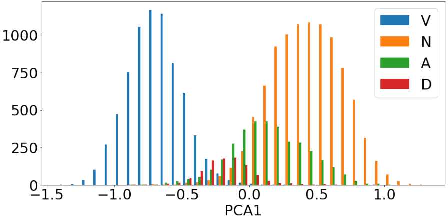
- Some syntax is learned.
- Significantly better scores than baselines.
- Still very far from the gold annotations.
- Shorter phrases very often well recognized.
- Sentence clause also very well recognized.

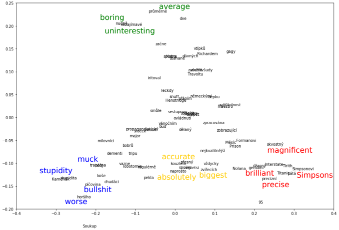
Future Work

- The encoder is probably affected by the target language.
- The idea is to train the translation into more languages (e.g. multiple decoders), so that the encoder representation is more universal.
- This could result in more syntactic behavior of the encoder.

Thank you for your attention !







http://ufal.mff.cuni.cz/david-marecek

marecek@ufal.mff.cuni.cz