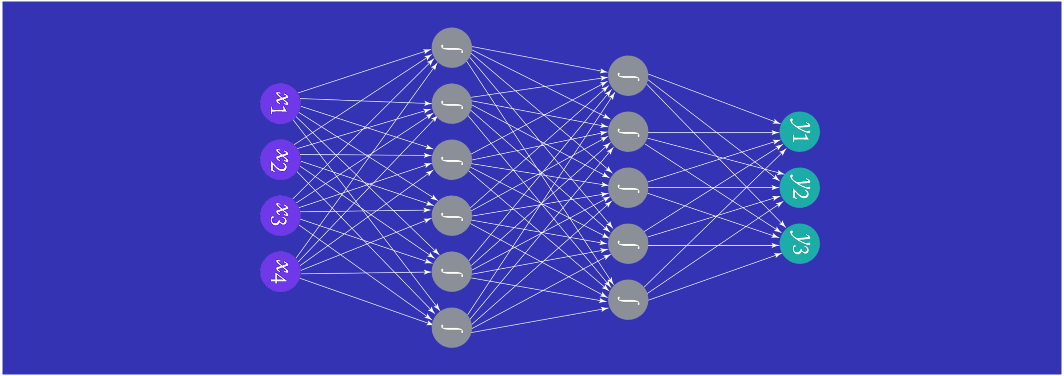
Neural networks in NLP: hype or more?

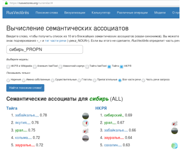


Andrey Kutuzov
University of Oslo, Language Technology Group
4-th Kolmogorov seminar in computational linguistics, Moscow, 25 April 2019

Who's here?

- I am Andrey Kutuzov.
- Master degree in computational linguistics from NRU HSE.
- Industrial past: *Lionbridge*, *Mail.ru Search*.
- Now doing my PhD and teaching at the University of Oslo.
  - Deep learning, computational linguistics...
- One of the maintainers of the *RusVectōrēs* project (word embeddings for Russian) [Kutuzov and Kuzmenko, 2017].
- Contributing to the *Gensim* library [Řehůřek and Sojka, 2010].

https://www.mn.uio.no/ifi/english/



https://rusvectores.org

https://github.com/RaRe-Technologies/gensim

What I will talk about?

- There are 4 take-home messages in the talk.
- Look for slides with large friendly red letters:

Take-home message

Like this one

About NLP

Linguistics

About NLP

Linguistics

Statistics

About NLP

Linguistics

Statistics

Deep learning

About NLP

- Computational Linguistics (CL);

About NLP

- Computational Linguistics (CL);
- Natural Language Processing (NLP);

About NLP

- Computational Linguistics (CL);
- Natural Language Processing (NLP);
- Natural Language Understanding (NLU);

About NLP

- Computational Linguistics (CL);
- Natural Language Processing (NLP);
- Natural Language Understanding (NLU);
- More or less the same field:

About NLP

- Computational Linguistics (CL);
- Natural Language Processing (NLP);
- Natural Language Understanding (NLU);
- More or less the same field:
    1. academic studying of language from computational point of view;

About NLP

- Computational Linguistics (CL);
- Natural Language Processing (NLP);
- Natural Language Understanding (NLU);
- More or less the same field:
  1. academic studying of language from computational point of view;
  2. solving practical language tasks:
     - analysis (natural language understanding)
     - synthesis (‘natural’ language generation)

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;

Why NLP/CL is so important recently?

- Data — 'XXI century oil';
- Everyone wants to process big data (especially in IT);

Why NLP/CL is so important recently?

- Data — 'XXI century oil';
- Everyone wants to process big data (especially in IT);
    - Data is often found in texts.

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)
  - machine translations

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
    - Data is often found in texts.
- NLP is now inside every smart-phone:
    - search engines (information retrieval)
    - machine translations
    - sentiment analysis

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
    - Data is often found in texts.
- NLP is now inside every smart-phone:
    - search engines (information retrieval)
    - machine translations
    - sentiment analysis
    - spam detection

Why NLP/CL is so important recently?

- Data — 'XXI century oil';
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)
  - machine translations
  - sentiment analysis
  - spam detection
  - speech recognition

Why NLP/CL is so important recently?

- Data — 'XXI century oil';
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)
  - machine translations
  - sentiment analysis
  - spam detection
  - speech recognition
  - chat bots

Why NLP/CL is so important recently?

- Data — ‘XXI century oil’;
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)
  - machine translations
  - sentiment analysis
  - spam detection
  - speech recognition
  - chat bots
  - summarization

Why NLP/CL is so important recently?

- Data — 'XXI century oil';
- Everyone wants to process big data (especially in IT);
  - Data is often found in texts.
- NLP is now inside every smart-phone:
  - search engines (information retrieval)
  - machine translations
  - sentiment analysis
  - spam detection
  - speech recognition
  - chat bots
  - summarization
  - virtual personal assistants (*Алиса*, *Siri*, *Alexa*, *Cortana*)...

Surprise: language is not simple

Yoav Goldberg Follow

Senior Lecturer at Bar Ilan University. Working on NLP. Recently with Neural Nets. Published a book about it. http://www.cs.biu.ac.il/~yogo/

Jun 9, 2017 · 14 min read

An Adversarial Review of "Adversarial Generation of Natural Language"

Or, for fucks sake, DL people, leave language alone and stop saying you solve it.

https://medium.com/@yoav.goldberg/an-adversarial-review-of-adversarial-generation-of-natural-language-409ac3378bd7

Language is not simple (especially when it has rich morphology)

- How to transform a character sequence into a machine-readable form?

Language is not simple (especially when it has rich morphology)

- How to transform a character sequence into a machine-readable form?
- Pre-processing text data:

Language is not simple (especially when it has rich morphology)

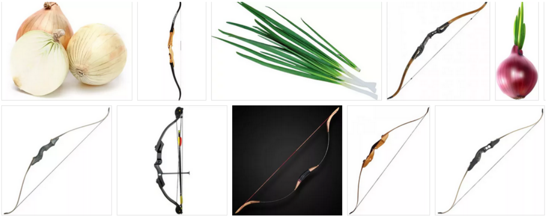
- How to transform a character sequence into a machine-readable form?
- Pre-processing text data:
  - tokenization (how to split a string into words?)

Language is not simple (especially when it has rich morphology)

- How to transform a character sequence into a machine-readable form?
- Pre-processing text data:
  - tokenization (how to split a string into words?)
  - lemmatization (what is the lemma of this word?)

Language is not simple (especially when it has rich morphology)

- How to transform a character sequence into a machine-readable form?
- Pre-processing text data:
  - tokenization (how to split a string into words?)
  - lemmatization (what is the lemma of this word?)
  - disambiguation:
    - «*Лук* (???) был просто *огонь* (???)»
  - ...and of course, many, many more.
  - Linguistic knowledge provides scaffolding for machine learning systems.



Take-home message #1

Language data is complex and specific

NLP needs linguistics

Deep Learning comes as no surprise in the history of NLP

- Contemporary NLP starts in the XX century:

Deep Learning comes as no surprise in the history of NLP

- Contemporary NLP starts in the XX century:
    - George Zipf (studied natural language statistics);

Deep Learning comes as no surprise in the history of NLP

- Contemporary NLP starts in the XX century:
  - George Zipf (studied natural language statistics);
  - Noam Chomsky (invented transformational/generative grammar);

Deep Learning comes as no surprise in the history of NLP

- Contemporary NLP starts in the XX century:
  - George Zipf (studied natural language statistics);
  - Noam Chomsky (invented transformational/generative grammar);
  - machine translation boom in the 1950s.

How we arrived to neural networks?

3 stages of the NLP history

1. 'rule-based' ('symbolic'): from the 1950s till 1990s;

How we arrived to neural networks?

3 stages of the NLP history

1. ‘rule-based’ (‘symbolic’): from the 1950s till 1990s;
2. ‘data-based’ (‘machine learning’): from the 1990s till 2000s;

How we arrived to neural networks?

3 stages of the NLP history

1. ‘rule-based’ (‘symbolic’): from the 1950s till 1990s;
2. ‘data-based’ (‘machine learning’): from the 1990s till 2000s;
3. ‘neural networks’ (‘deep learning’): from the end of 2000s till now.

How we arrived to neural networks?

3 stages of the NLP history

1. 'rule-based' ('symbolic'): from the 1950s till 1990s;
2. 'data-based' ('machine learning'): from the 1990s till 2000s;
3. 'neural networks' ('deep learning'): from the end of 2000s till now.

'Kolmogorov complexity' — the length of minimal computer program reproducing an object (model size)

- Computational power of the humanity is growing...
- ...but it doesn't help rule-based methods: they don't scale well.
- Rule-based description of an object of a given complexity does not speed up.

How we arrived to neural networks?

Instead, search and machine learning do scale extremely well!

How we arrived to neural networks?

Instead, search and machine learning do scale extremely well!

Machine learning

- In a sense, NLP today is a part of data science...

How we arrived to neural networks?

Instead, search and machine learning do scale extremely well!

Machine learning

- In a sense, NLP today is a part of data science...
- ...since it is almost entirely built around machine learning.

How we arrived to neural networks?

Instead, search and machine learning do scale extremely well!

Machine learning

- In a sense, NLP today is a part of data science...
- ...since it is almost entirely built around machine learning.
- We train our systems on large corpora.

How we arrived to neural networks?

Instead, search and machine learning do scale extremely well!

Machine learning

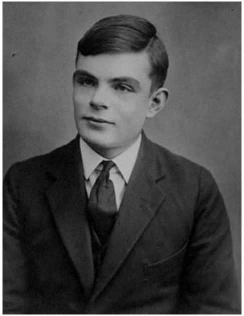
- In a sense, NLP today is a part of data science...
- ...since it is almost entirely built around machine learning.
- We train our systems on large corpora.
- Neural networks: a type of ML algorithms.

Artificial neural networks have been around for a long time

'Machines of this character can behave in a very complicated manner when the number of units is large.'

Alan Turing, 'Intelligent Machinery'

[Turing, 1948]

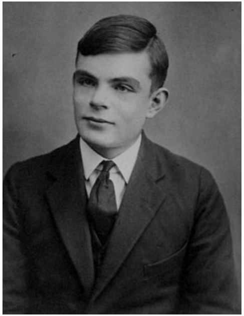


Artificial neural networks have been around for a long time

'Machines of this character can behave in a very complicated manner when the number of units is large.'

Alan Turing, 'Intelligent Machinery'

[Turing, 1948]



...but the 60s, 70s and 80s were 'AI winter'. A handful of rebels continued to ~~resist the Empire~~ play with training neural networks...

27.03.2019: Deep Learning founding fathers received 'computer science Nobel prize': ACM Turing Award

'...for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing... in computer vision, speech recognition, natural language processing, and robotics'

Neural networks revival

- ‘Deep learning’ is machine learning with multi-layered artificial neural networks and non-linear transformations.

Neural networks revival

- ‘Deep learning’ is machine learning with multi-layered artificial neural networks and non-linear transformations.
  - feedforward networks (the simplest),
  - convolutional networks (CNN),
  - recurrent networks:
    - **gated: (Bi)LSTM, GRU, etc.**
  - transformers...

Neural networks revival

- ‘Deep learning’ is machine learning with multi-layered artificial neural networks and non-linear transformations.
  - feedforward networks (the simplest),
  - convolutional networks (CNN),
  - recurrent networks:
    - **gated: (Bi)LSTM, GRU, etc.**
  - transformers...
- In nearly any complex NLP task, neural systems are better than ‘classic’ linear models.

Neural networks revival

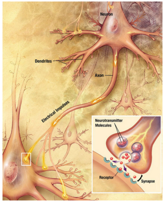
- 'Deep learning' is machine learning with multi-layered artificial neural networks and non-linear transformations.
  - feedforward networks (the simplest),
  - convolutional networks (CNN),
  - recurrent networks:
    - **gated: (Bi)LSTM, GRU, etc.**
  - transformers...
- In nearly any complex NLP task, neural systems are better than 'classic' linear models.
- '*Do we even need anything else?*'
  - Not a spoiler any more: yes, we also need linguistics.
  - Often, we also need huge amounts of training data.

Why ‘neural’?

- Our brain has 10^{11} neurons, with 10^4 connections per each.

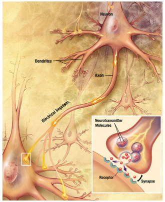
Why 'neural'?

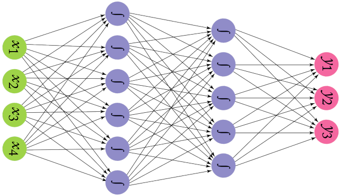
- Our brain has 10^{11} neurons, with 10^{4} connections per each.
- Neurons receive signals from other neurons, weigh them, and send new signals further down the network.



Why 'neural'?

- Our brain has 10^{11} neurons, with 10^{4} connections per each.
- Neurons receive signals from other neurons, weigh them, and send new signals further down the network.

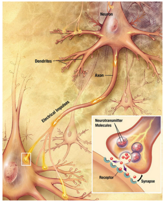


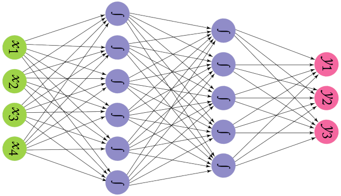


- Artificial neural networks (ANNs) try to imitate this.

Why 'neural'?

- Our brain has 10^{11} neurons, with 10^{4} connections per each.
- Neurons receive signals from other neurons, weigh them, and send new signals further down the network.

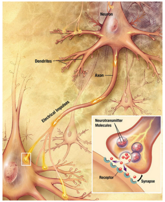


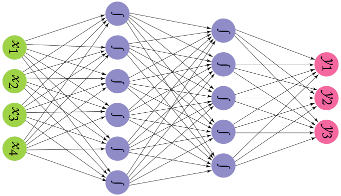


- Artificial neural networks (ANNs) try to imitate this.
- But this is not certain: in fact, we do not know for sure how our brain works.

Why 'neural'?

- Our brain has 10^{11} neurons, with 10^4 connections per each.
- Neurons receive signals from other neurons, weigh them, and send new signals further down the network.





- Artificial neural networks (ANNs) try to imitate this.
- But this is not certain: in fact, we do not know for sure how our brain works.
- Hence, neural networks are vectors, weights arrays and differentiation. Not so sexy, but true.

Why 'deep'?

- ANNs as a rule include several transformation layers (hence, *multi-layered perceptrons*, MLP).

Why ‘deep’?

- ANNs as a rule include several transformation layers (hence, *multi-layered perceptrons*, MLP).
- They were shown to be able to approximate any function.

Why 'deep'?

- ANNs as a rule include several transformation layers (hence, *multi-layered perceptrons*, MLP).
- They were shown to be able to approximate any function.
  - ...given enough layers and neurons (computational units) in these layers.

Why 'deep'?

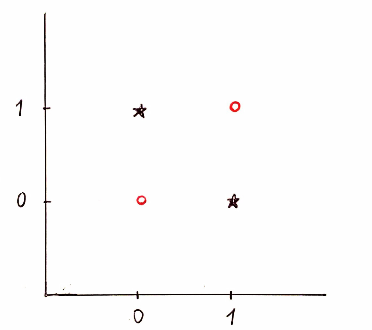
- ANNs as a rule include several transformation layers (hence, *multi-layered perceptrons*, MLP).
- They were shown to be able to approximate any function.
  - ...given enough layers and neurons (computational units) in these layers.
- Non-linear transformations between layers make it possible to process data which is not linearly separable.
- This is the reason ANNs work better.

Why 'non-linear'?

Consider this toy XOR dataset, and the non-linear transformation $\phi(x, y) = [x \times y, x + y]$:

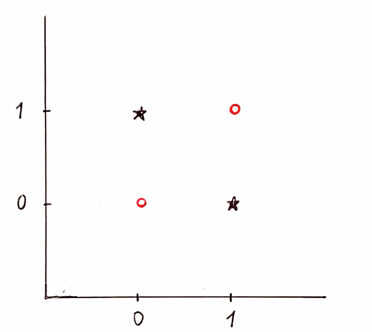
Why 'non-linear'?

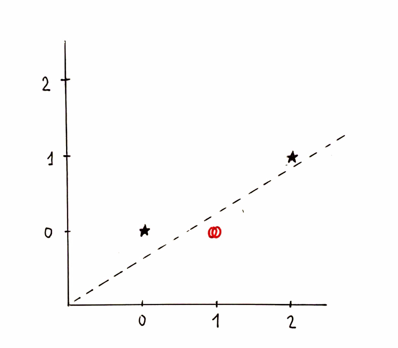
Consider this toy XOR dataset, and the non-linear transformation $\phi(x, y) = [x \times y, x + y]$:



Why 'non-linear'?

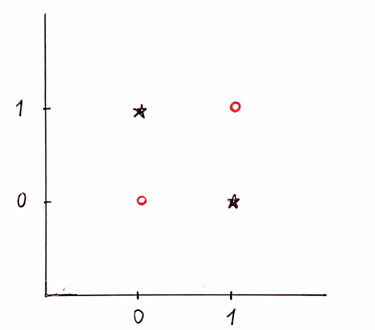
Consider this toy XOR dataset, and the non-linear transformation $\phi(x, y) = [x \times y, x + y]$:

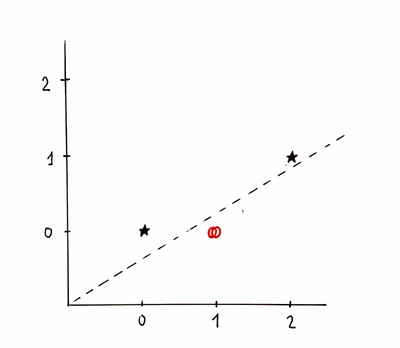




Why 'non-linear'?

Consider this toy XOR dataset, and the non-linear transformation $\phi(x, y) = [x \times y, x + y]$:





ANNs learn such transformations.

Take-home message #2

Deep Learning is a logical development of NLP

Representation learning

- ANNs are very efficient in learning optimal vector representations for various entities.

Representation learning

- ANNs are very efficient in learning optimal vector representations for various entities.
- Of ultimate importance in NLP: language has a lot of complex objects, and we need ways to calculate similarities:
  - sounds,
  - characters,
  - words,
  - sentences, documents... even languages themselves.

Representation learning

- ANNs are very efficient in learning optimal vector representations for various entities.
- Of ultimate importance in NLP: language has a lot of complex objects, and we need ways to calculate similarities:
  - sounds,
  - characters,
  - words,
  - sentences, documents... even languages themselves.
- Such representations are called embeddings (e.g., word embeddings).

word2vec: representing words with vectors is commonplace now

(Dmitry Malkov)

Deep learning has changed NLP substantially

- Much less manual feature engineering: models find useful feature combinations themselves.

Deep learning has changed NLP substantially

- Much less manual feature engineering: models find useful feature combinations themselves.
- Extracting useful signal from unannotated texts.

Deep learning has changed NLP substantially

- Much less manual feature engineering: models find useful feature combinations themselves.
- Extracting useful signal from unannotated texts.
- Pre-trained models: easily fine-tuned for your tasks.

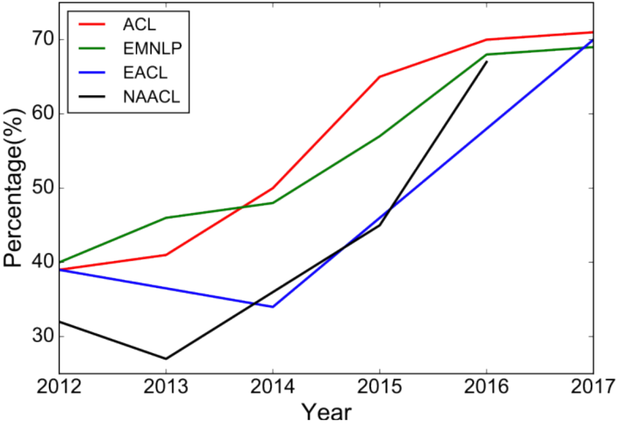
Deep learning has changed NLP substantially

- Much less manual feature engineering: models find useful feature combinations themselves.
- Extracting useful signal from unannotated texts.
- Pre-trained models: easily fine-tuned for your tasks.
- Elementary 'bricks' are combined into different complex architectures.

Deep learning has changed NLP substantially

- Much less manual feature engineering: models find useful feature combinations themselves.
- Extracting useful signal from unannotated texts.
- Pre-trained models: easily fine-tuned for your tasks.
- Elementary 'bricks' are combined into different complex architectures.
- Hardware today is comparatively cheap and powerful (GPUs, TPUs).

Steamroller, or DL at the NLP conferences



[Young et al., 2018]

Regular NLP revolutions and new SOTA achievements

- Multi-task learning;

Regular NLP revolutions and new SOTA achievements

- Multi-task learning;
- Generative Adversarial Networks (GANs);

Regular NLP revolutions and new SOTA achievements

- Multi-task learning;
- Generative Adversarial Networks (GANs);
- Pre-trained language models:

Regular NLP revolutions and new SOTA achievements

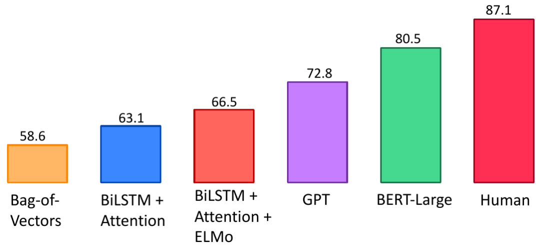
- Multi-task learning;
- Generative Adversarial Networks (GANs);
- Pre-trained language models:
  - Contextualized word embeddings (ELMo) [Peters et al., 2018];

Regular NLP revolutions and new SOTA achievements

- Multi-task learning;
- Generative Adversarial Networks (GANs);
- Pre-trained language models:
  - Contextualized word embeddings (ELMo) [Peters et al., 2018];
  - Transformers (BERT) [Devlin et al., 2018]...

Regular NLP revolutions and new SOTA achievements

- Multi-task learning;
- Generative Adversarial Networks (GANs);
- Pre-trained language models:
  - Contextualized word embeddings (ELMo) [Peters et al., 2018];
  - Transformers (BERT) [Devlin et al., 2018]...



`https://gluebenchmark.com/`, [Wang et al., 2018]

Funny things happen sometimes

OpenAI trained a language model but didn't publish it (for fear of malicious usage):

VIEW CODE READ PAPER

Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper.

Funny things happen sometimes

OpenAI trained a language model but didn't publish it (for fear of malicious usage):

VIEW CODE READ PAPER

Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper.

NLP folks are not much grateful for that:

(((J()(J() 'yoav))))
@yoavgo
Читать

В ответ @aCraigPfeifer

we can generate all the stories on the @OpenAI blog post on a single CPU core, after training on just a handful of blogposts for several minutes.

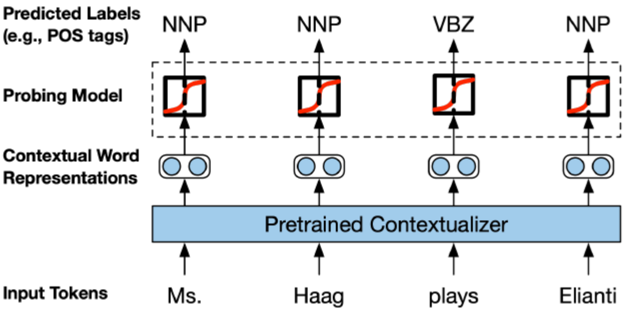
10:21 - 15 февр. 2019 г.

Problem: interpretability of predictions made by a cascade of representations

XKCD

Probing tasks

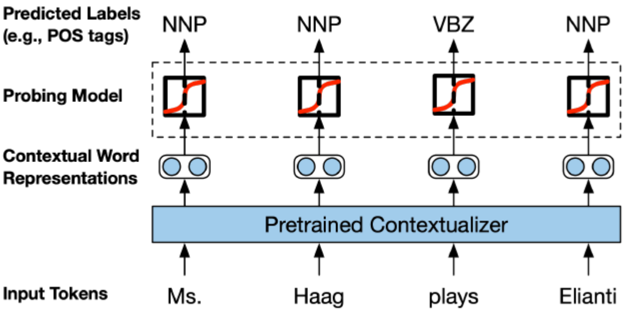
We need linguistics to diagnose: to find out what exactly our models learn:



[Liu et al., 2019], [Belinkov and Glass, 2019]

Probing tasks

We need linguistics to diagnose: to find out what exactly our models learn:



[Liu et al., 2019], [Belinkov and Glass, 2019]

NB: correlation does not imply causation!

Do deep networks help to enrich our linguistic knowledge?

Do deep networks help to enrich our linguistic knowledge?

Latent syntax trees learned by a network:

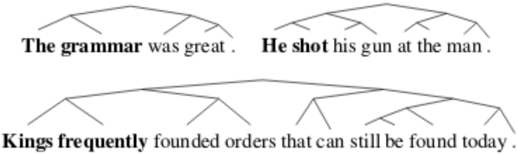


Figure 4: The ST-Gumbel models often form constituents from the first two words of sentences.

[Williams et al., 2018]

Do deep networks help to enrich our linguistic knowledge?

Latent syntax trees learned by a network:

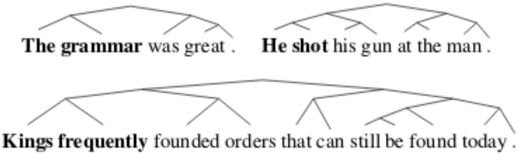
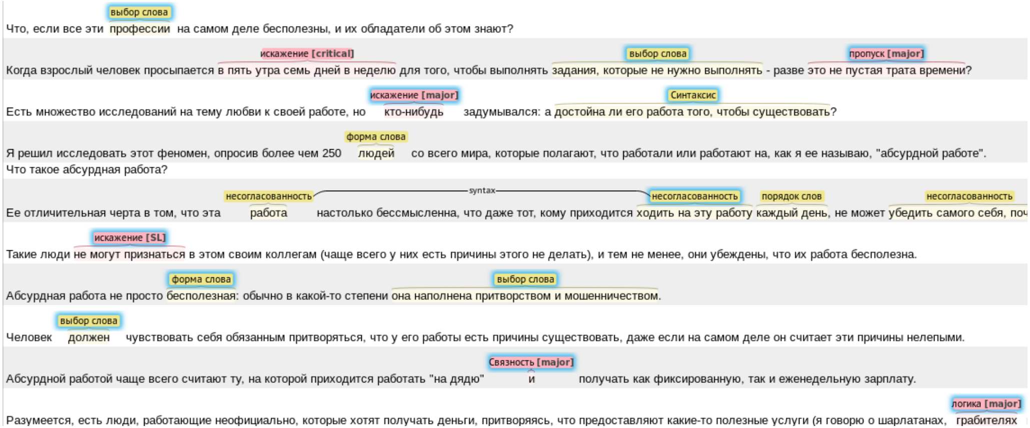


Figure 4: The ST-Gumbel models often form constituents from the first two words of sentences.

[Williams et al., 2018]

Problem: lack of annotated data



annotation of translators' mistakes (`https://rus-ltc.org`)

Take-home message #3

Deep learning in NLP is not over-hyped. Do it now!

- *But the 'classic' ML 20 years ago was not over-hyped either.*
- *Thus, DL is not some final solution to automatic language analysis and synthesis.*
- *Something different will come in the future, solving DL problems.*

Large software ecosystem (mostly Python)

- *NumPy*: multi-dimensional arrays (tensors) and other linear algebra;

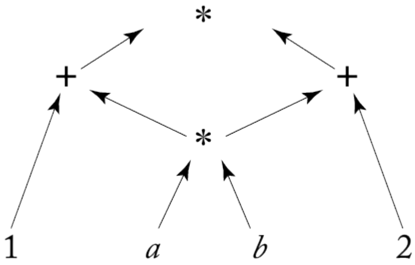
Large software ecosystem (mostly Python)

- *NumPy*: multi-dimensional arrays (tensors) and other linear algebra;
- A Law: **any IT company with more than 10K employees starts making its own DL framework**;
  - open source $\rightarrow$ everybody benefits;

Large software ecosystem (mostly Python)

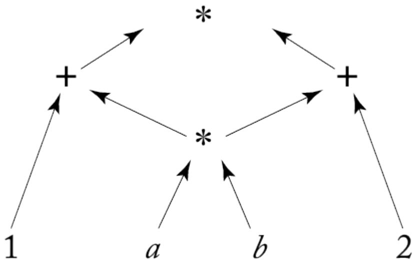
- *NumPy*: multi-dimensional arrays (tensors) and other linear algebra;
- A Law: **any IT company with more than 10K employees starts making its own DL framework**;
  - open source $\rightarrow$ everybody benefits;
- TensorFlow by *Google* [Abadi et al., 2015] `https://tensorflow.org/`
- PyTorch by *Facebook* [Paszke et al., 2017] `https://pytorch.org/`

Under the hood of frameworks: computation graph



directed acyclic graph (DAG)

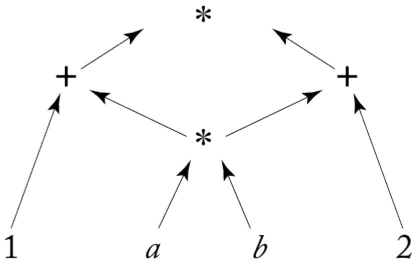
Under the hood of frameworks: computation graph



directed acyclic graph (DAG)

- Representation of a mathematical expression calculation;

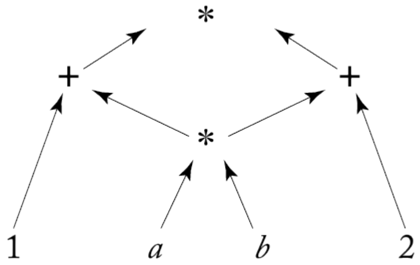
Under the hood of frameworks: computation graph



directed acyclic graph (DAG)

- Representation of a mathematical expression calculation;
- all operations and variables are nodes in the graph;

Under the hood of frameworks: computation graph



directed acyclic graph (DAG)

- Representation of a mathematical expression calculation;
- all operations and variables are nodes in the graph;
- allows to easily design arbitrarily complex and deep architectures (must be differentiable).

TensorFlow or PyTorch? What's the difference?

- TensorFlow: static graphs (compile then run).
- PyTorch: dynamic graphs (compile a new graph for each input).

TensorFlow or PyTorch? What's the difference?

- TensorFlow: static graphs (compile then run).
  - *In fact, TensorFlow supports dynamic graphs as well (eager execution).*

- PyTorch: dynamic graphs (compile a new graph for each input).

TensorFlow or PyTorch? What's the difference?

- TensorFlow: static graphs (compile then run).
  - *In fact, TensorFlow supports dynamic graphs as well (eager execution).*

- PyTorch: dynamic graphs (compile a new graph for each input).
- PyTorch is more convenient for designing new and complex architectures.

TensorFlow or PyTorch? What's the difference?

- TensorFlow: static graphs (compile then run).
  - *In fact, TensorFlow supports dynamic graphs as well (eager execution).*
- Keras: convenient wrapper for typical tasks.

- PyTorch: dynamic graphs (compile a new graph for each input).
- PyTorch is more convenient for designing new and complex architectures.
- AllenNLP: a wrapper for NLP tasks.

TensorFlow or PyTorch? What's the difference?

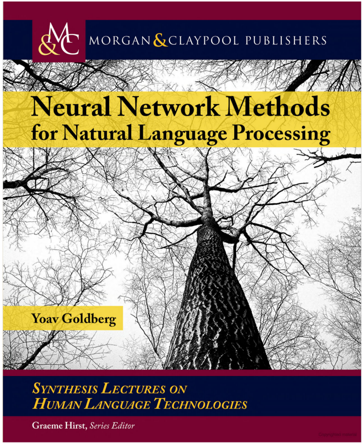
- TensorFlow: static graphs (compile then run).
  - *In fact, TensorFlow supports dynamic graphs as well (eager execution).*
- Keras: convenient wrapper for typical tasks.
- PyTorch: dynamic graphs (compile a new graph for each input).
- PyTorch is more convenient for designing new and complex architectures.
- AllenNLP: a wrapper for NLP tasks.

Both frameworks are research and production ready. Just use the one you know better.

Take-home message #4

NLP+DL frameworks: choice between *PyTorch* and *TensorFlow*.
Both are good.

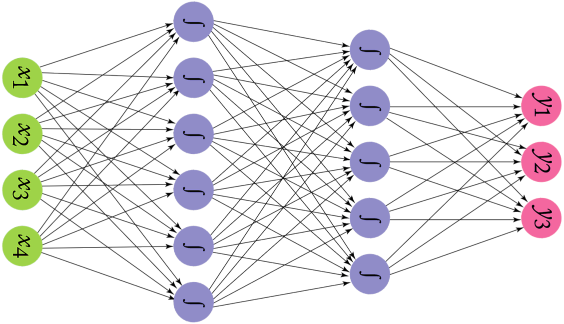
Further reading



- Yoav Goldberg: *Neural Network Methods for Natural Language Processing* [Goldberg, 2017].
- Sebastian Ruder blog (`http://ruder.io/`)

Questions?

Neural networks in NLP:
hype or more?



Andrey Kutuzov (andreku@ifi.uio.no)
Language Technology Group, University of Oslo
`https://rusvectores.org`

References I

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015).
TensorFlow: Large-scale machine learning on heterogeneous systems.
Software available from tensorflow.org.

References II

Belinkov, Y. and Glass, J. (2019).
Analysis methods in neural language processing: A survey.
Transactions of the Association for Computational Linguistics, 7:49–72.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018).
BERT: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805.

Goldberg, Y. (2017).
Neural network methods for natural language processing.
Synthesis Lectures on Human Language Technologies, 10(1):1–309.

References III

Kutuzov, A. and Kuzmenko, E. (2017).
WebVectors: A toolkit for building web interfaces for vector semantic models.

In Ignatov, D., editor, *Analysis of Images, Social Networks and Texts (AIST): 5th International Conference, Yekaterinburg, Russia*, pages 155–161, Cham. Springer International Publishing.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M., and Smith, N. A. (2019).
Linguistic knowledge and transferability of contextual representations.
arXiv preprint arXiv:1903.08855.

References IV

- Paszke, A., Gross, S., Chintala, S., and Chanan, G. (2017).
  PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration.
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018).
  Deep contextualized word representations.
  In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

References V

Řehůřek, R. and Sojka, P. (2010).
Software Framework for Topic Modelling with Large Corpora.
In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.

Turing, A. (1948).
Intelligent machinery.
Technical report, National Physical Laboratory.

References VI

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018).
GLUE: A multi-task benchmark and analysis platform for natural language understanding.
In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Williams, A., Drozdov*, A., and Bowman, S. R. (2018).
Do latent tree learning models identify meaningful structure in sentences?
Transactions of the Association for Computational Linguistics, 6:253–267.

References VII

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018).
Recent trends in deep learning based natural language processing.
IEEE Computational intelligence magazine, 13(3):55–75.