



# NLP News & Trends

Assistant Professor Dagmar Gromann  
Centre for Translation Studies, University of Vienna  
NLP Meetup 19.06.2019  
[dagmar.gromann@univie.ac.at](mailto:dagmar.gromann@univie.ac.at)

# Additional resources

Python for Data Science Cheat Sheet 

Learn more Python for data science interactively at [www.datacamp.com](https://www.datacamp.com)



**About spaCy**

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. It's designed specifically for production use and helps you build applications that process and "understand" large volumes of text. Documentation: [spacy.io](https://spacy.io)

```
$ pip install spacy
```

```
import spacy
```

**Statistical models**

Download statistical models

Predict part-of-speech tags, dependency labels, named entities and more. See here for available models: [spacy.io/models](https://spacy.io/models)

```
$ python -m spacy download en_core_web_sm
```

Check that your installed models are up to date

```
$ python -m spacy validate
```

Loading statistical models

```
import spacy
# Load the installed model "en_core_web_sm"
nlp = spacy.load("en_core_web_sm")
```

**Documents and tokens**

Processing text

Processing text with the `nlp` object returns a `Doc` object that holds all information about the tokens, their linguistic features and their relationships

```
doc = nlp("This is a text")
```

Accessing token attributes

```
doc = nlp("This is a text")
# Token texts
[token.text for token in doc]
# ['This', 'is', 'a', 'text']
```

**Spans**

Accessing spans

Span indices are exclusive. So `doc[2:4]` is a span starting at token 2, up to - but not including! - token 4.

```
doc = nlp("This is a text")
span = doc[2:4]
span.text
# 'a text'
```

Creating a span manually

```
# Import the Span object
from spacy.tokens import Span
# Create a Doc object
doc = nlp("I live in New York")
# Span for "New York" with label GPE (geopolitical)
span = Span(doc, 3, 5, label="GPE")
span.text
# 'New York'
```

**Linguistic features**

Attributes return label IDs. For string labels, use the attributes with an underscore. For example, `token.pos_`.

Part-of-speech tags PREDICTED BY STATISTICAL MODEL

```
doc = nlp("This is a text.")
# Coarse-grained part-of-speech tags
[token.pos_ for token in doc]
# ['DET', 'VERB', 'DET', 'NOUN', 'PUNCT']
# Fine-grained part-of-speech tags
[token.tag_ for token in doc]
# ['DT', 'VBZ', 'DT', 'NN', '.']
```

Syntactic dependencies PREDICTED BY STATISTICAL MODEL

```
doc = nlp("This is a text.")
# Dependency labels
[token.dep_ for token in doc]
# ['nsubj', 'ROOT', 'det', 'attr', 'punct']
# Syntactic head token (governor)
[token.head.token_ for token in doc]
# ['is', 'is', 'text', 'is', 'is']
```

Named entities PREDICTED BY STATISTICAL MODEL

```
doc = nlp("Larry Page founded Google")
# Text and label of named entity span
[(ent.text, ent.label_) for ent in doc.ents]
# [('Larry Page', 'PERSON'), ('Google', 'ORG')]
```

**Syntax iterators**

Sentences USUALLY NEEDS THE DEPENDENCY PARSER

```
doc = nlp("This is a sentence. This is another one.")
# doc.sents is a generator that yields sentence spans
[sent.text for sent in doc.sents]
# ['This is a sentence.', 'This is another one.']
```

Base noun phrases NEEDS THE TAGGER AND PARSER

```
doc = nlp("I have a red car")
# doc.noun_chunks is a generator that yields spans
[chunk.text for chunk in doc.noun_chunks]
# ['I', 'a red car']
```

Label explanations

```
spacy.explain("RB")
# 'adverb'
spacy.explain("GPE")
# 'Countries, cities, states'
```


Visualizing

If you're in a Jupyter notebook, use `displacy.render`. Otherwise, use `displacy.render` to start a web server and show the visualization in your browser.

```
from spacy import displacy
```


Visualize dependencies

```
doc = nlp("This is a sentence")
displacy.render(doc, style="dep")
```



Visualize named entities

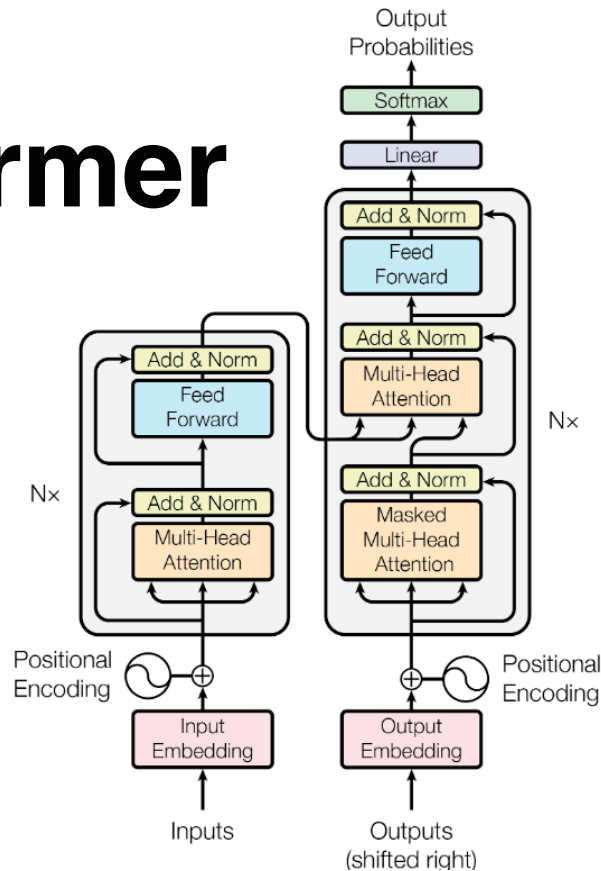
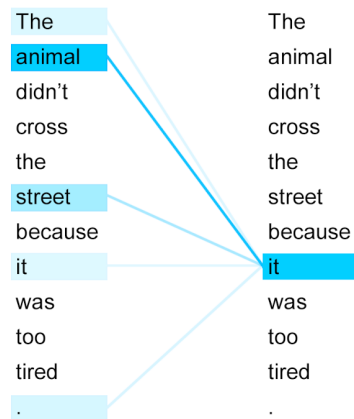
```
doc = nlp("Larry Page founded Google")
displacy.render(doc, style="ent")
```



- [SpaCy Cheat Sheet](#): SpaCy at a glance
- [NeuronBlocks](#): Neural Network Lego-like building blocks
- [PyText](#): NLP modeling framework based on PyTorch - faster experimentations at a scale
- TensorFlow Neural Machine Translation Tutorial: <https://github.com/tensorflow/nmt>
- Facebook AI Research Sequence-to-Sequence (FairSeq) toolkit in PyTorch: <https://github.com/pytorch/fairseq>

# Recap: Transformer

## Attention

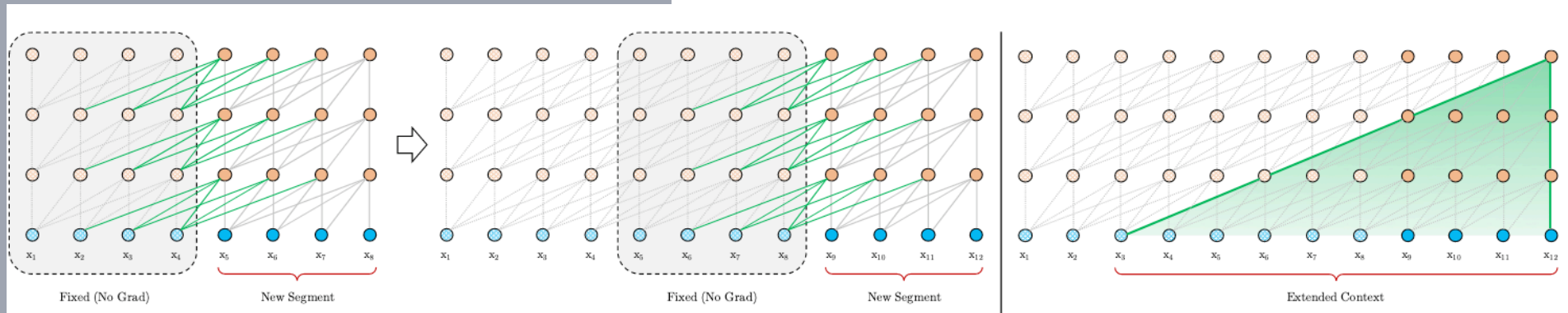


- encoder-decoder style
- no recurrence or convolutions
- sublayers joining multi-head attention with position-wise fully connected feedforward network
- residual connections

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

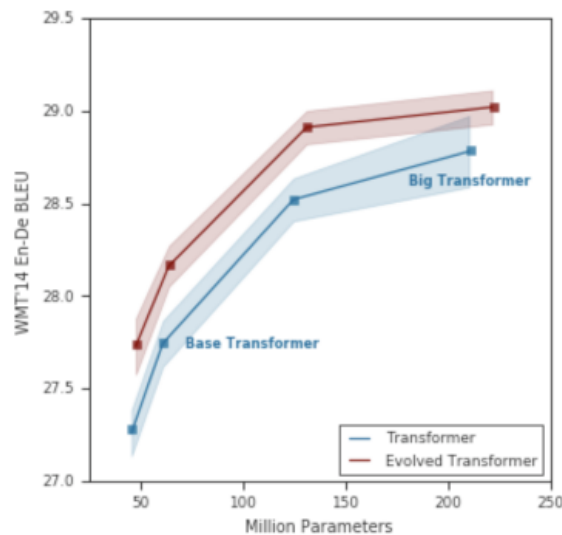
# Transformer-XL

- reintroduces recurrence on segment-level
- over two consecutive segments
- very similar to truncated Back Propagation Through Time (BPTT)



Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. [arXiv CoRR abs/1901.02860](https://arxiv.org/abs/1901.02860).

# The Evolved Transformer



Neural Architecture Search (NAS):

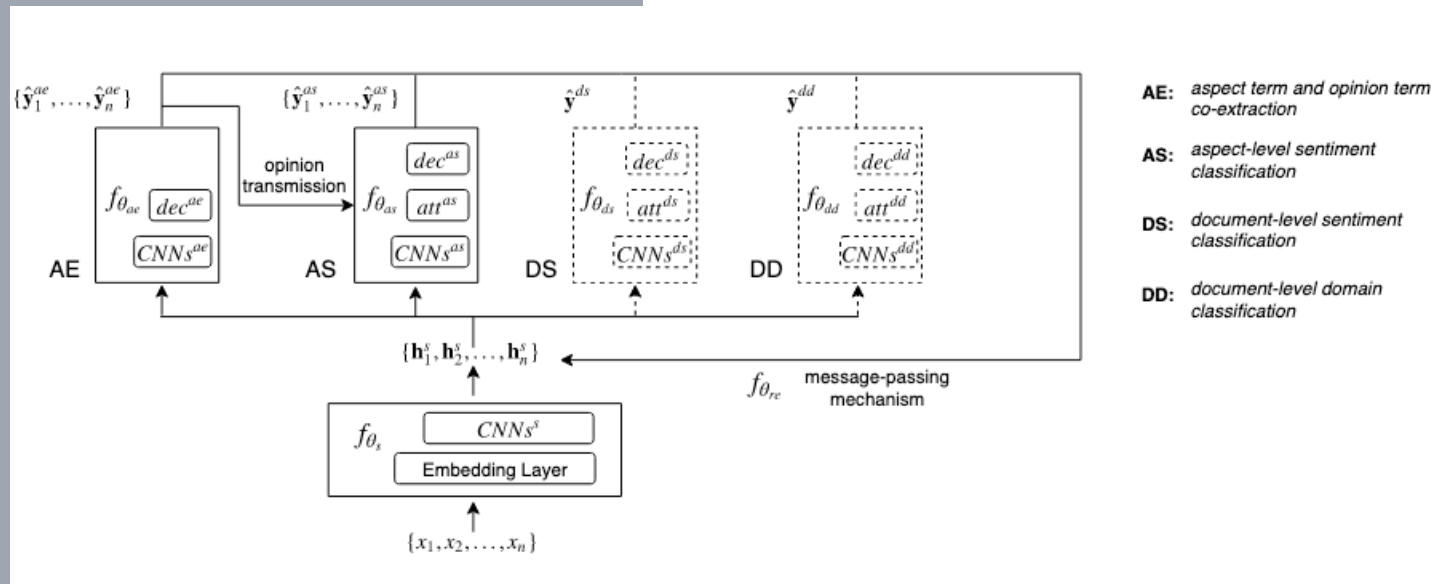
- Progressive Dynamic Hurdle (PDH): if you perform well, you may train for more steps
- Evolved Transformer: depth-wise separable convolution and self-attention



HSO, D. R., Liang, C., & Le, Q. V. (2019). [The Evolved Transformer](#). Proceedings of the 36th International Conference on Machine Learning, PMLR 97:5877-5886.

# Sentiment Analysis

Multitask: Aspect term extraction & aspect-level sentiment classification in parallel.



“Great food, but the service was dreadful.”

He, R., Lee, W.S, Ng, H.T. & Dahlmeier, D. (2019). [An Interactive Multi-Task Learning Network for End-to-End Aspect-Based Sentiment Analysis](#). *Association of Computational Linguistics (ACL) 2019*

# Self-feeding chatbot

Assumptions:

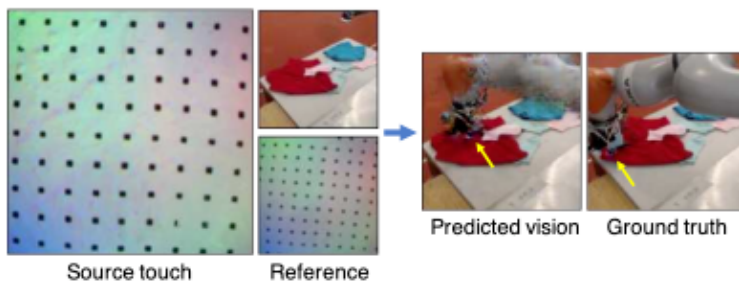
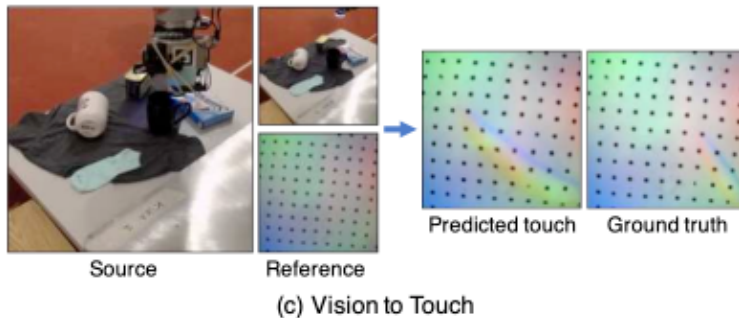
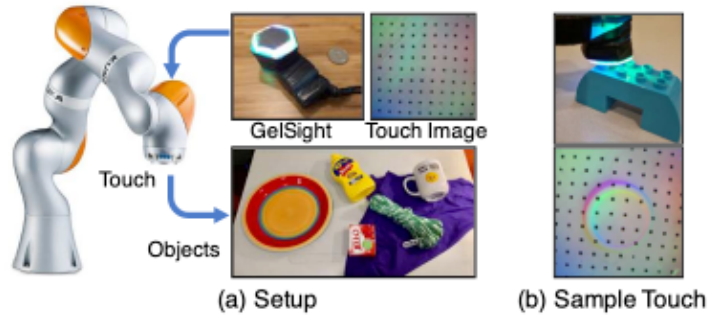
- Actively request feedback
- Overcome Dunning-Kruger effect: a bad model will also be bad at knowing it is doing a bad job

Architecture: Transformer + ranking



Hancock, B., Bordes, A., Mazare, P. E., & Weston, J. (2019). Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. *Association of Computational Linguistics (ACL) 2019*

# Touch to Vision and back



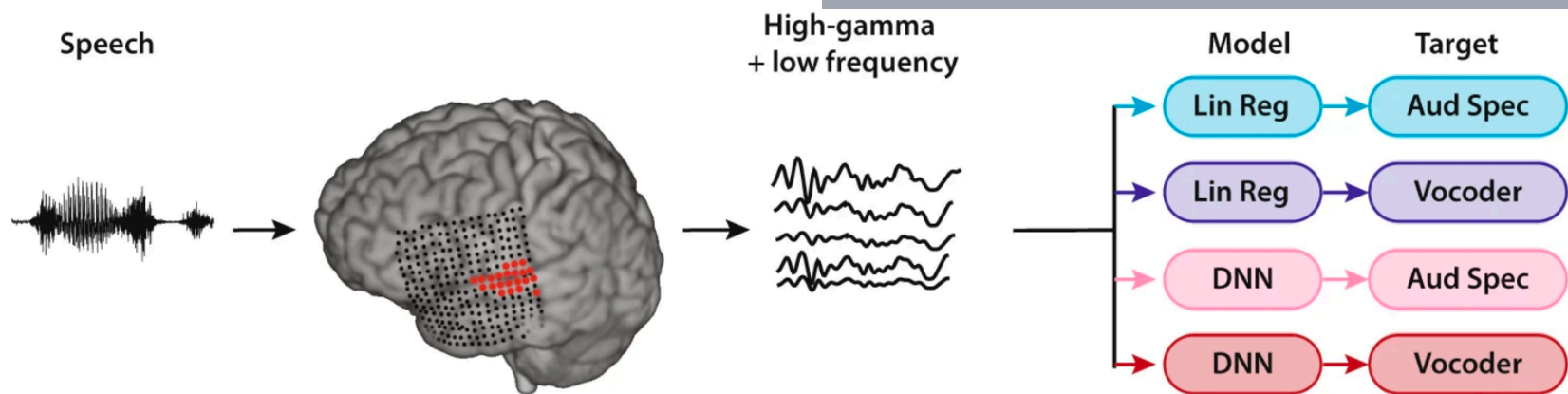
Where's the language?

- embodied cognition in cognitive linguistics
- language is based on repeated sensorimotor experiences that form patterns, e.g. CONTAINMENT

Li, Y., Zhu, J. Y., Tedrake, R., & Torralba, A. (2019). [Connecting Touch and Vision via Cross-Modal Prediction](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 10609-10618).



# Thought to Speech

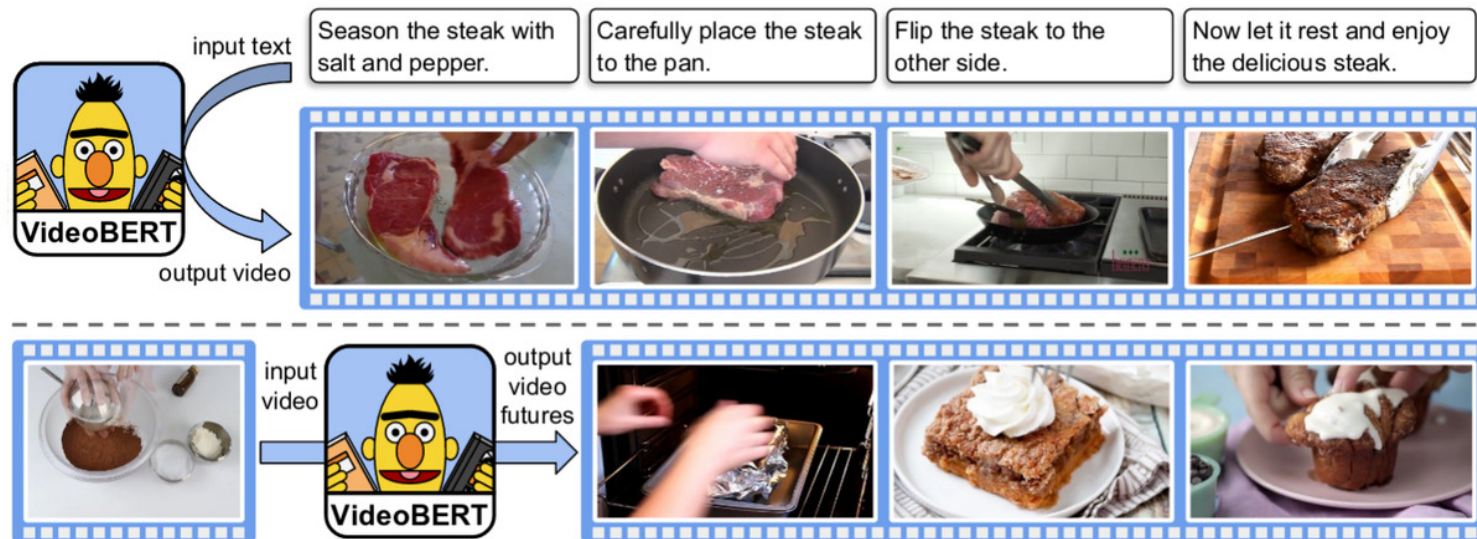


See also: Elon Musk's Neuralink advances

Akbari, H., Khalighinejad, B., Herrero, J. L., Mehta, A. D., & Mesgarani, N. (2019). [Towards reconstructing intelligible speech from the human auditory cortex](#). *Scientific reports*, 9(1), 874.

# Predicting the future

Self-supervised joint visual-linguistic model using BERT; text-to-video prediction



Sun, C., Myers, A., Vondrick, C., Murphy, K., & Schmid, C. (2019). VideoBERT: A joint model for video and language representation learning. [arXiv CoRR abs/1904.01766](https://arxiv.org/abs/1904.01766)

**General  
trend?**

Multimodal, Multisensory

# Multimodal/- sensory Bias



Source: [habibadasilva](#)

Self-driving cars more likely to overlook darker shades of skin color (Wilson et al. 2019)

Face recognition harder when female or darker skin color (Raji and Buolamwini 2019)

Wilson, B., Hoffman, J., & Morgenstern, J. (2019). Predictive Inequity in Object Detection. [arXiv CoRR abs/1902.11097](#).

Raji, I. D., & Buolamwini, J. (2019). Actionable auditing: [Investigating the impact of publicly naming biased performance results of commercial ai products](#). In *AAAI/ACM Conf. on AI Ethics and Society*.

# Multilingual Technologies

Multilingual Software Engineering

Machine Translation

Multilingual Information Management



Multilingual Communication

Applied Computer Science

Joint Master program of FH Campus  
Wien and University of Vienna

Contact:

[dagmar.gromann@univie.ac.at](mailto:dagmar.gromann@univie.ac.at)

# Online Lectures



Stanford NLP with Deep Learning (CS224N 2019) by Chris Manning and Abigail See

CMU Neural Nets for NLP (2019) by Graham Neubig

Build and Deploy an End-to-End Deep Learning System (2019) Hands-on for developers



# Thank you!

<http://www.arxiv-sanity.com/>  
<https://paperswithcode.com/>