






NLP News

Oct 2019

Adrian M.P. Brasoveanu, MT

Introduction

-  **François Chollet** @fchollet · Oct 18
There's a big difference between data-efficiency and information-efficiency. We could make considerable progress on the former without even starting to approach the latter – and never realize our fundamental mistake.
-  **François Chollet** @fchollet · Oct 18
Abstraction is the golden path to information efficiency. But all kinds of hacks can get you data efficiency.
-  **François Chollet** @fchollet

The difference between data & information:

- Data is plentiful & noisy, the large majority of data is not information
- Information is what makes data useful
- We are bad at extracting info from data. We only manage to distill & use a small fraction of what is really available

12:26 AM · Oct 19, 2019 · [Twitter Web App](#)

31 Retweets 116 Likes

-  **François Chollet** @fchollet · Oct 19
Replying to @fchollet
Working on "data efficiency" typically means finding new hacks to increase the amount of information we can extract from the same data. But "information efficiency" is an entirely unrelated concept: how to efficiently and effectively make use of information

YouTube playlist: fast.ai Code-First Intro to Natural Language Processing

Search

Home

Trending

Subscriptions

Library

What is NLP? (NLP video 1) Rachel Thomas 22:42

Topic Modeling with SVD & NMF (NLP video) Rachel Thomas 1:06:40

Topic Modeling & SVD revisited (NLP video 3) Rachel Thomas 33:06

Sentiment Classification with Rachel Thomas 58:20

Sentiment Classification with Rachel Thomas 51:29

Programming PyTorch for Deep Learning Creating and Deploying Deep Learning Applications Ian Pointer

fast.ai Code-First Intro to Natural Language Processing

19 videos • 71,531 views • Last updated on Jul 12, 2019

PLAY ALL

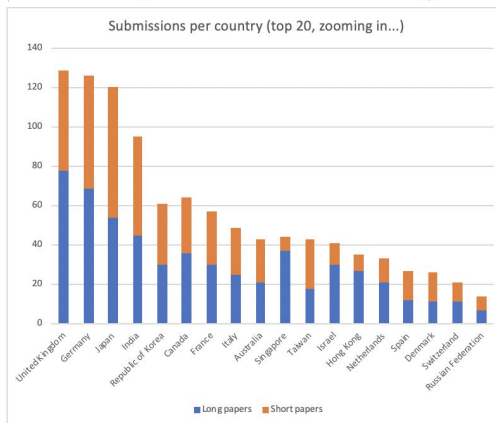
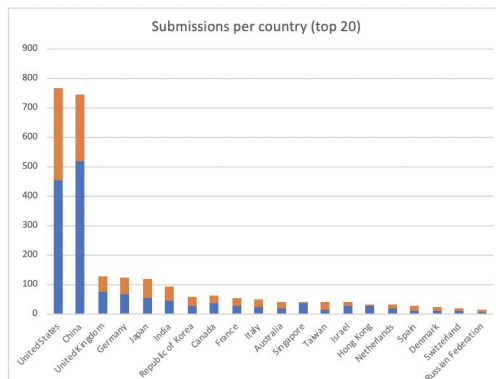
This is the playlist for the fast.ai NLP course, originally taught in the USF MS in Data Science program during May-June 2019. The course covers a blend of traditional NLP topics (including regex, SVD, naive bayes, tokenization) and recent neural network

O'REILLY*

Generative Deep Learning Teaching Machines to Paint, Write, Compose and Play David Foster

ACL 2019 Stats

	Area	Long	Short	Total
1.	Information Extraction, Text Mining	156	93	249
2.	Machine Learning	148	73	221
3.	Machine Translation	102	105	207
4.	Dialogue and Interactive Systems	125	57	182
5.	Generation	97	58	155
6.	Question Answering	99	55	154
7.	Sentiment Analysis, Argument Mining	91	60	151
8.	Word-level Semantics	78	59	137
9.	Applications	65	72	137
10.	Resources and Evaluation	70	60	130
11.	Multidisciplinary, AC COI	70	44	114
12.	Sentence-level Semantics	70	42	112
13.	Tagging, Chunking, Syntax, Parsing	50	49	99
14.	Social Media	51	42	93
15.	Summarization	48	35	83
16.	Document Analysis	48	33	81
17.	Vision, Robotics Multimodal Grounding, Speech	56	23	79
18.	Multilinguality	43	32	75
19.	Textual Inference, Other Areas of Semantics	44	30	74
20.	Linguistic Theories, Cognitive, Psycholinguistics	39	21	60
21.	Discourse and Pragmatics	33	24	57
22.	Phonology, Morphology, Word Segmentation	26	18	44
	Total	1609	1085	2694



ACL 2019 CHAIRS BLOG

Florence, Italy July 28 - August 2

HOME / ABOUT US

Statistics on submissions

APRIL 8, 2019 / PC CHAIRS / COMMENTS OFF

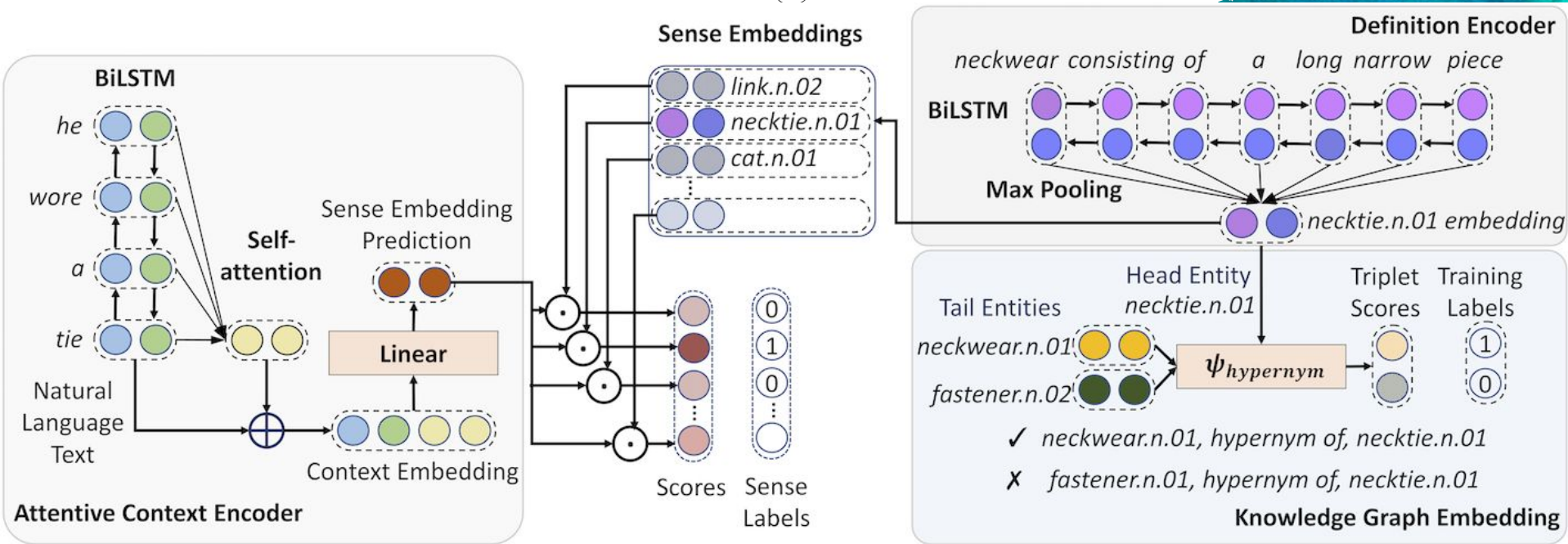
ACL 2019 received as many as 2906 submissions by the submission deadline. This constitutes more than a 75% increase over ACL 2018 and is an all-time record for ACL-related conferences! The huge logistics involved in handling these submissions explains our long silence. However, we can now finally give you some basic statistics:

POSTS

October 2019

M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13

Zero-shot WSD with Sense Definition Embeddings



Transferable Multi-Domain State Generator for TODS

× Model

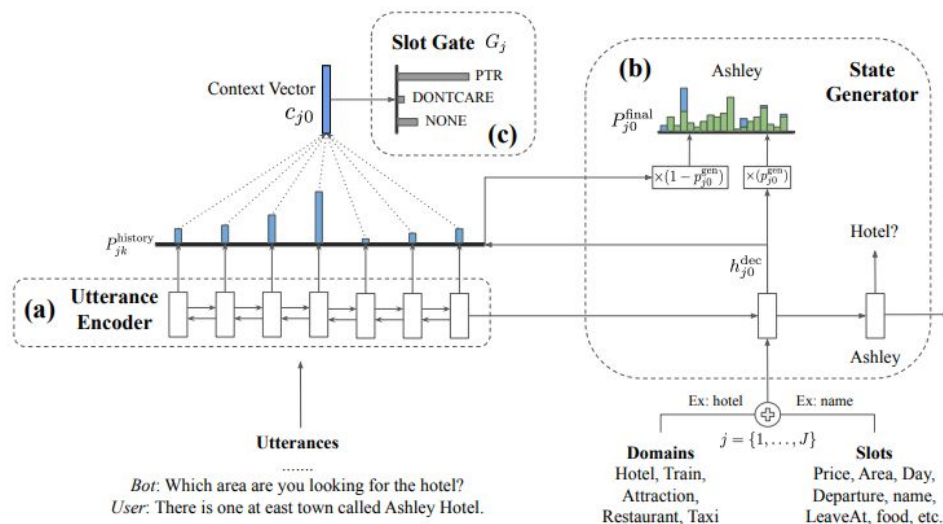
- utterance encoder
- slot gate
- state generator
- all shared across domains

× Knowledge Transfer for predicting

- domain, slot, value

× Madotto et al, ACL 2019

- <https://arxiv.org/pdf/1905.08743.pdf>
- <https://github.com/jasonwu0731/trade-dst>



Energy and Policy Considerations for NLP

- × **Cost of hardware**
- × **Cost of electricity**
- × **Cloud compute time**
- × **Carbon footprint**
- × **Cost of training**
- × **Cost of development**
- × **Emma Strubell et al**
 - ACL 2019 paper
 - <https://arxiv.org/pdf/1906.02243.pdf>

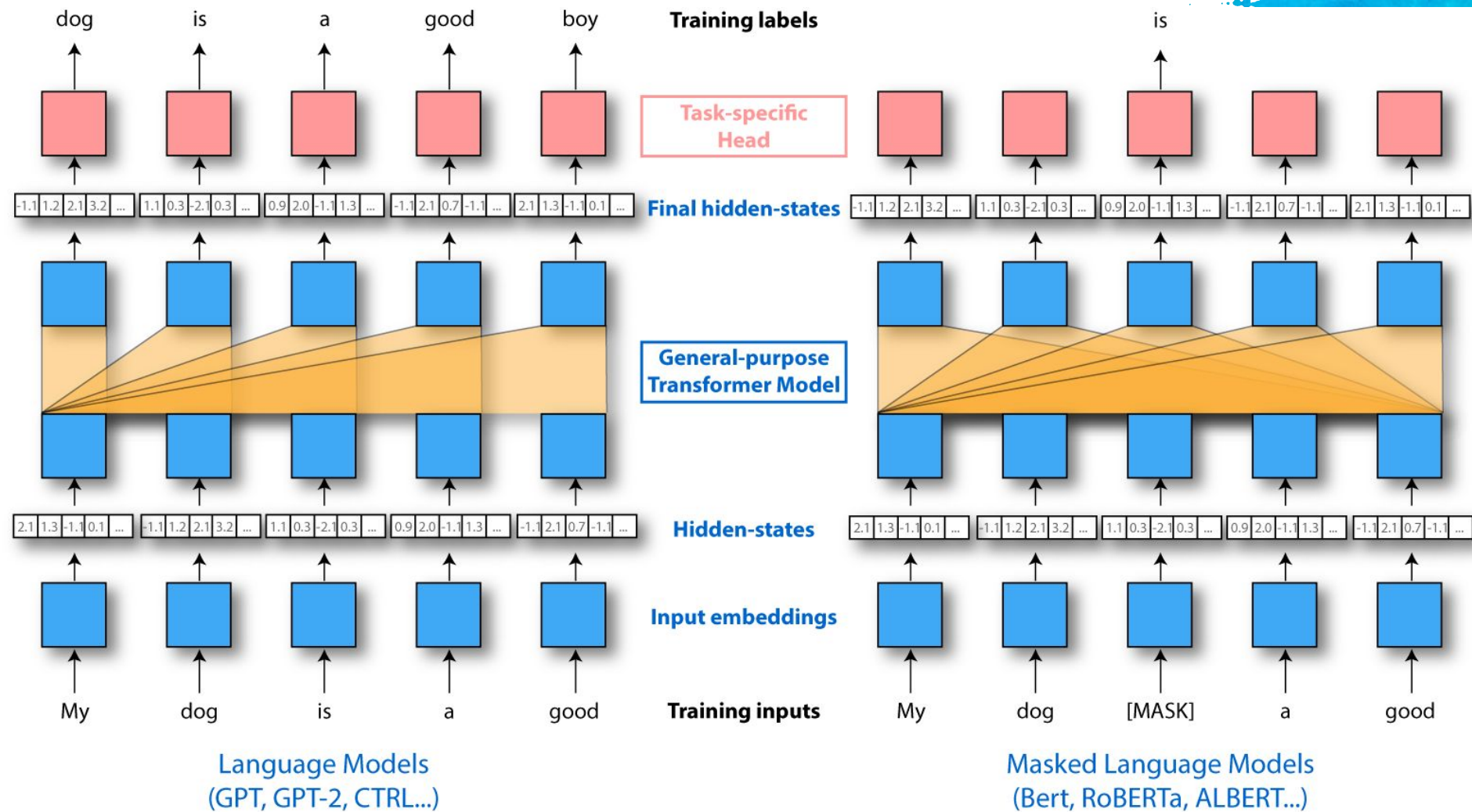
Consumption	CO ₂ e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000

Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO₂ emissions from training common NLP models, compared to familiar consumption.¹

Model	Hardware	Power (W)	Hours	kWh-PUE	CO ₂ e	Cloud compute cost
Transformer _{base}	P100x8	1415.78	12	27	26	\$41–\$140
Transformer _{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT _{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT _{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Table 3: Estimated cost of training a model in terms of CO₂ emissions (lbs) and cloud compute cost (USD).⁷ Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.



What does BERT Look at?

× Attention heads correspond to

- syntax
- coreference
- direct objects of verbs or prepositions
- determiners of nouns

× Clark, Khandelwal, Levy, Manning

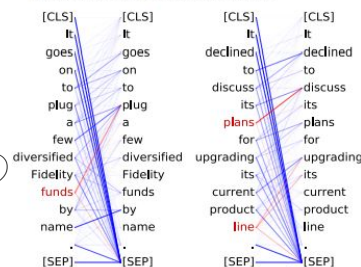
- <https://arxiv.org/pdf/1906.04341.pdf>
- <https://github.com/clarkkev/attention-analysis>

× Alternatively see

- Attention is not Explanation
- Attention is not not Explanation

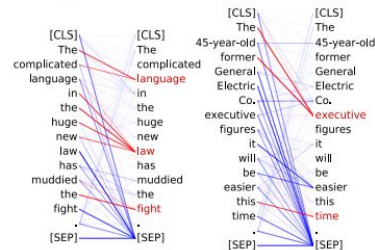
Head 8-10

- Direct objects attend to their verbs
- 86.8% accuracy at the dobj relation



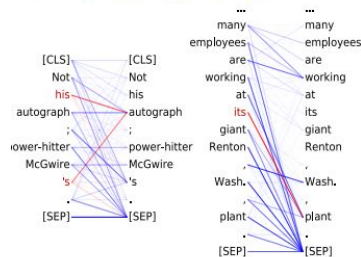
Head 8-11

- Noun modifiers (e.g., determiners) attend to their noun
- 94.3% accuracy at the det relation



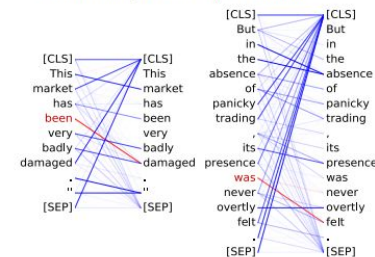
Head 7-6

- Possessive pronouns and apostrophes attend to the head of the corresponding NP
- 80.5% accuracy at the poss relation



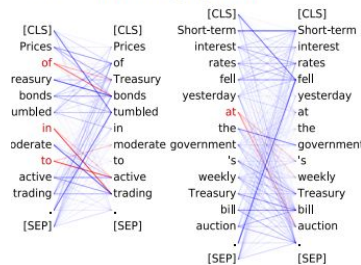
Head 4-10

- Passive auxiliary verbs attend to the verb they modify
- 82.5% accuracy at the auxpass relation



Head 9-6

- Prepositions attend to their objects
- 76.3% accuracy at the pobj relation



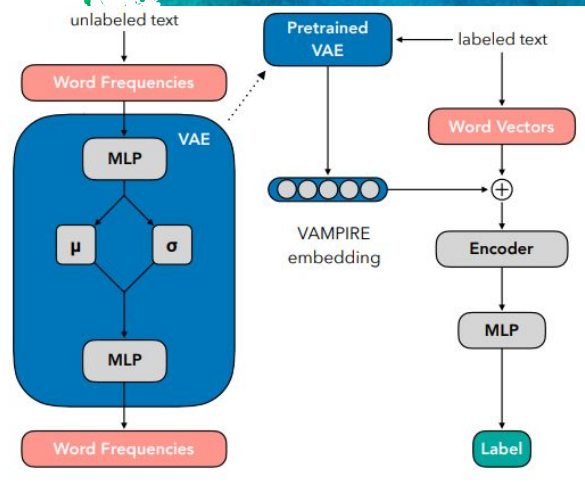
Head 5-4

- Coreferent mentions attend to their antecedents
- 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent




Transformer Updates

- × **RoBERTa** performs better than BERT
 - <https://arxiv.org/pdf/1907.11692.pdf>
- × **ALBERT** shows best results
 - <https://arxiv.org/pdf/1909.11942.pdf>
- × **DistilBERT** - less parameters, 95% performance
 - <https://arxiv.org/abs/1910.01108>
- × What about **lower resources** (e.g., less GPUs)?
 - **VAMPIRE** architecture is promising
 - <https://arxiv.org/pdf/1906.02242.pdf>



Unified Transformers

- × Unified APIs for TF and PyTorch Transformers
- × Includes most well-known Transformers
- × Generally faster than original libraries
- × SoA
- × Focus on pre-trained models
- × 2 libraries
 - <https://github.com/huggingface/transformers>
 - <https://github.com/explosion/spacy-transformers>



Transformers

build [pipelines](#) license [Apache-2.0](#) website [docs](#) release [v2.1.1](#)

State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch

🤖 Transformers (formerly known as `pytorch-transformers` and `pytorch-pretrained-bert`) provides state-of-the-art general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet, CTRL...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between TensorFlow 2.0 and PyTorch.

Features

- As easy to use as `pytorch-transformers`
- As powerful and concise as Keras
- High performance on NLU and NLG tasks
- Low barrier to entry for educators and practitioners

State-of-the-art NLP for everyone

- Deep learning researchers
- Hands-on practitioners
- AI/ML/NLP teachers and educators

Lower compute costs, smaller carbon footprint

spacy-transformers

spaCy pipelines for pretrained BERT, XLNet and GPT-2

release [v0.5.0](#) license [MIT](#) [Stars](#) 454

This package provides spaCy model pipelines that wrap Hugging Face's `transformers` package, so you can use them in spaCy. The result is convenient access to state-of-the-art transformer architectures, such as BERT, GPT-2, XLNet, etc.

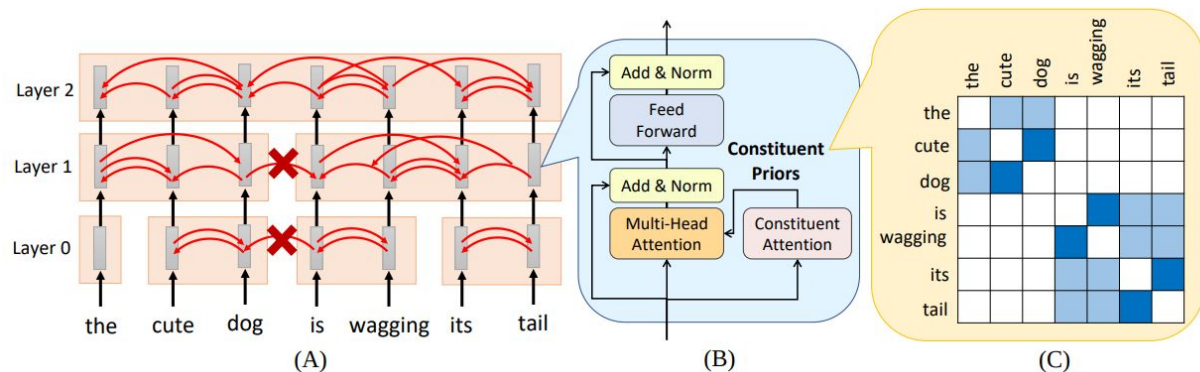
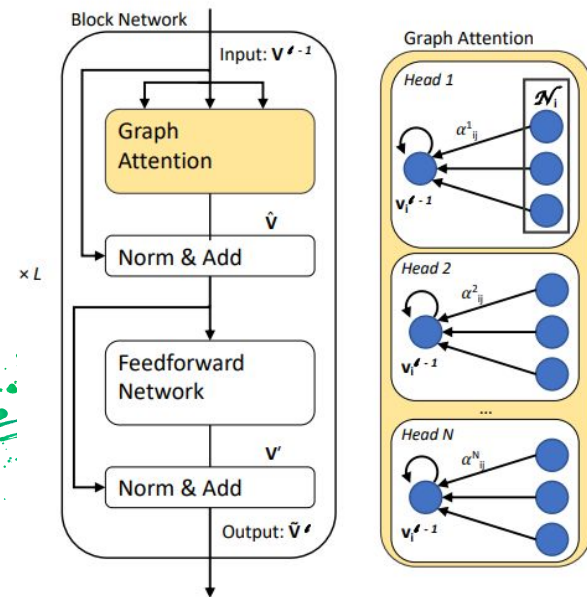
EXAMPLE

```
import spacy

nlp = spacy.load("en_trf_bertbaseuncased_lg")
doc = nlp("Apple shares rose on the news. Apple pie is delicious.")
print(doc[0].similarity(doc[7]))
print(doc._trf_last_hidden_state.shape)
```

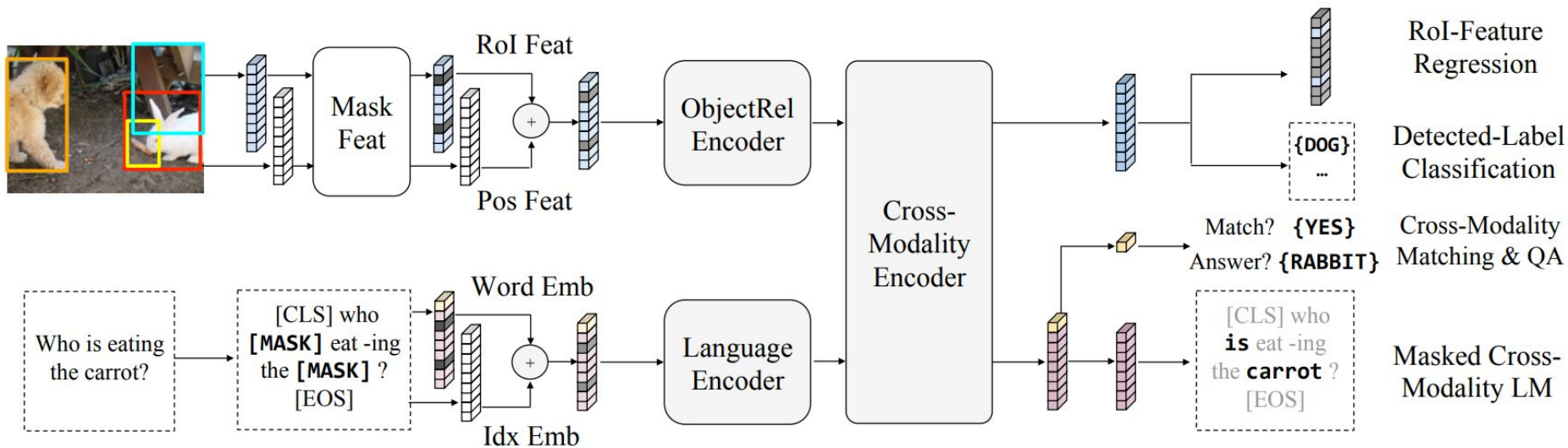
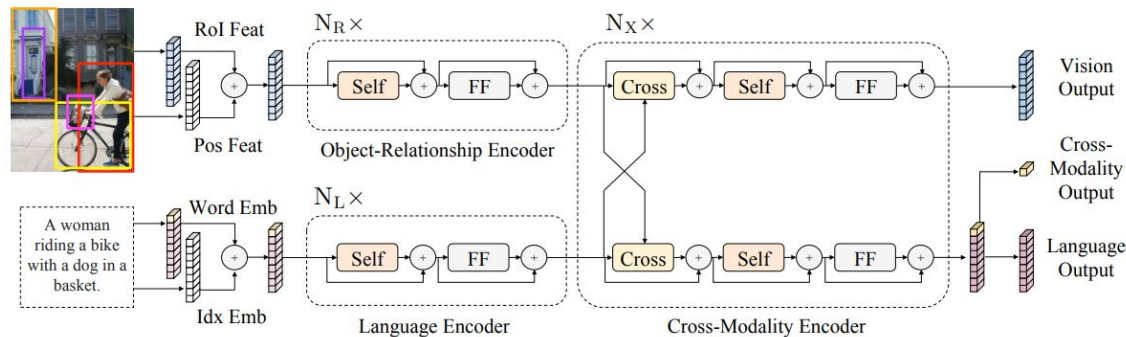
Cross Transformers

- × Why should we use different Transformers?
- × Cross-pollination
 - **Graph (NAACL)** - text generation
 - **Tree (EMNLP)** - composition, interpretation
 - **Capsule (WS)** - prediction



LXMERT

Cross-Modality





Thanks!

Any questions?

You can find me at:
[@AdrianB82](#)