INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN PYTHON

Ingo Kleiber

 @KleiberIngo

 @IngoKleiber

mastodon.social

Who’s That?

Ingo (Kleiber)

- (Computational) Linguist & teacher educator at Heidelberg University (HSE)
- Interested in a wide range of (often unrelated) things such as (digital) education, languages, coffee, photography, artificial intelligence, (political) philosophy, economics, …
- Not a programmer; similarly to the fact that you’re not an ‘e-mailer’

Today

1. Introduction: Natural Language Processing (NLP)
2. Corpus Linguistics
3. Common Problems & Use Cases
4. Fundamentals of Linguistics
5. Exploring Some Basic Methods/Approaches in Python
6. State-of-the-Art ML/AI Approaches
7. What's Next?

Today's Aims

You will be able to …

- describe what corpus linguistics and NLP are
- discuss common use cases, applications, and challenges of NLP
- use basic linguistic terminology in order to talk about natural language
- to use Python (and spaCy) in order to perform basic NLP tasks
- list some current / state-of-the-art AI/ML approaches to NLP

Code Along!

If you like, you can **code and experiment along**!

https://github.com/IngoKl/36c3-workshops

(then use *Binder)*

Python Libraries

For this workshop, we will be using a number of third party Python libraries.

Most importantly:

- **spaCy** (spacy.io) – a modern natural language processing library as well as a collection of pre-trained models
- **gensim** (radimrehurek.com) – a library specialized on topic modelling and semantic similarities
- **scikit-learn** (scikit-learn.org)

Toy/Demo Corpus

I've compiled a toy/demo corpus for this workshop called ***BROWN-a1-a44-modified.txt***.

- Sample (subset a1 to a44 → PRESS) of the BROWN corpus
- Tags/Annotation has been stripped; some basic data preprocessing and cleaning

1. Natural Language Processing (NLP)

We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like speech and language processing, human language technology, natural language processing, computational linguistics, and speech recognition and synthesis. The **goal of this new field** is to get computers to **perform useful tasks involving human language**, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

(Jurafsky and Martin 2008: 1)

1. Natural Language Processing (NLP)

The **goal of this new field** is to get computers to **perform useful tasks involving human language**, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

(Jurafsky and Martin 2008: 1)

versus (?)

Corpus Linguistics

Computational Linguistics

Text Mining

2. Corpus Linguistics

“We could reasonably define corpus linguistics as dealing with some set of machine-readable texts which is deemed an appropriate basis on which to study a specific set of research questions.” (McEnery and Hardie 2012: 1)

“Corpus linguistics is the investigation of linguistic research questions that have been framed *in terms of the conditional distribution of linguistic phenomena* in a *linguistic corpus*.” (Stefanowitsch Forth.: 54)

“I would contend that that corpus linguistics represents both a **new method** (in terms of computer-aided descriptive linguistics) and a **new research discipline** (in terms of a new approach to language description)” (Mukherjee 2005: 86)

→ The collection and analysis of large and systematic collections of linguistic data (= corpora)

3. Common Problems & Use Cases

Some Common Use Cases

- Parsing & Tagging
- Speech Recognition
- Machine Translation
- Text Summarization
- Question Answering
- Information Extraction
- Text Classification
- Language Generation
- …

Some Challenges / Problems

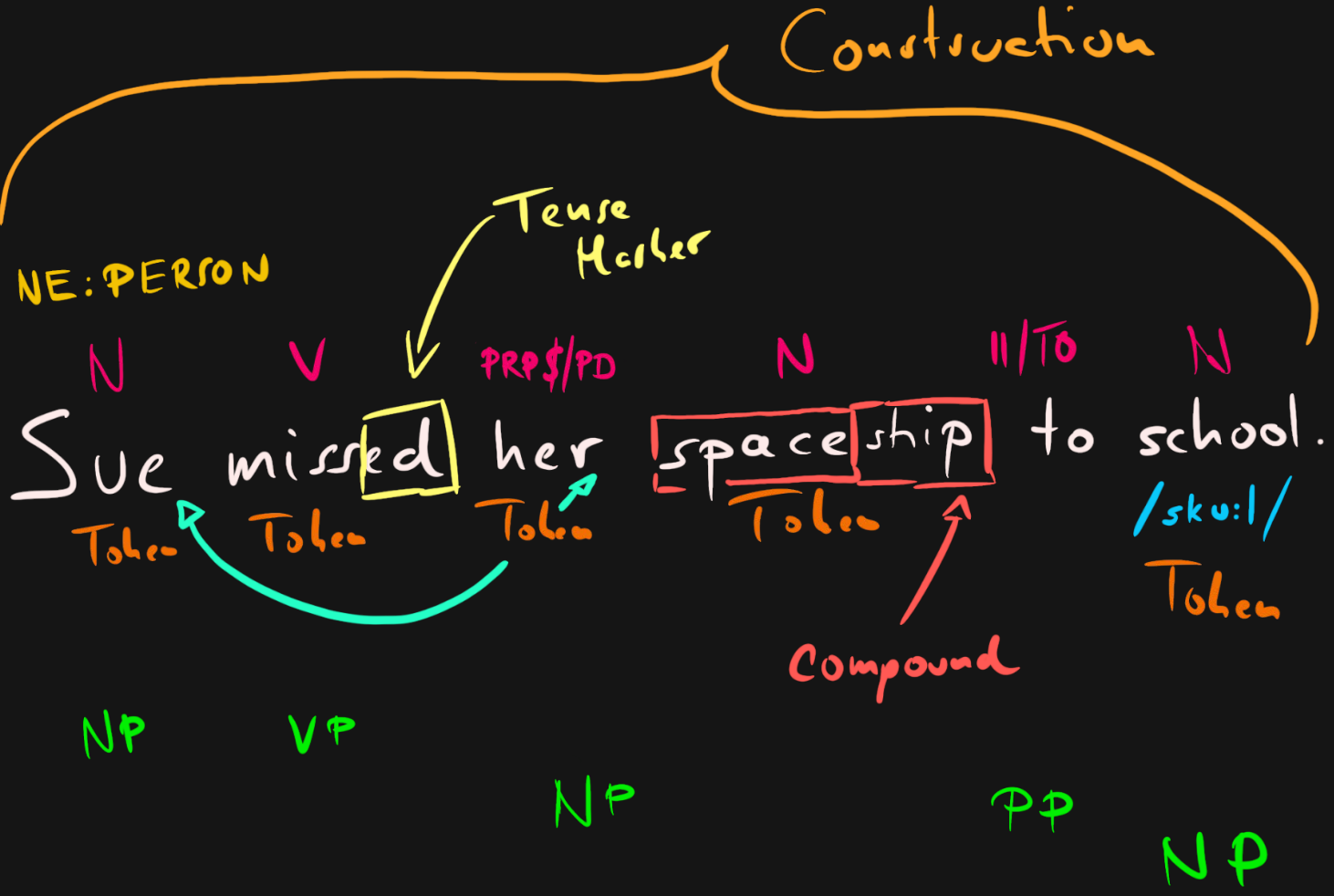
- Complexity of language and use
- Language is extremely context sensitive
- Multi- and Translingualism
- Model/data availability (English bias)
- Automatic tagging/parsing is still hard
- #ethNLP (ethical challenges)
- …

4. Fundamentals of Linguistics

Linguistics = the study of language(s)

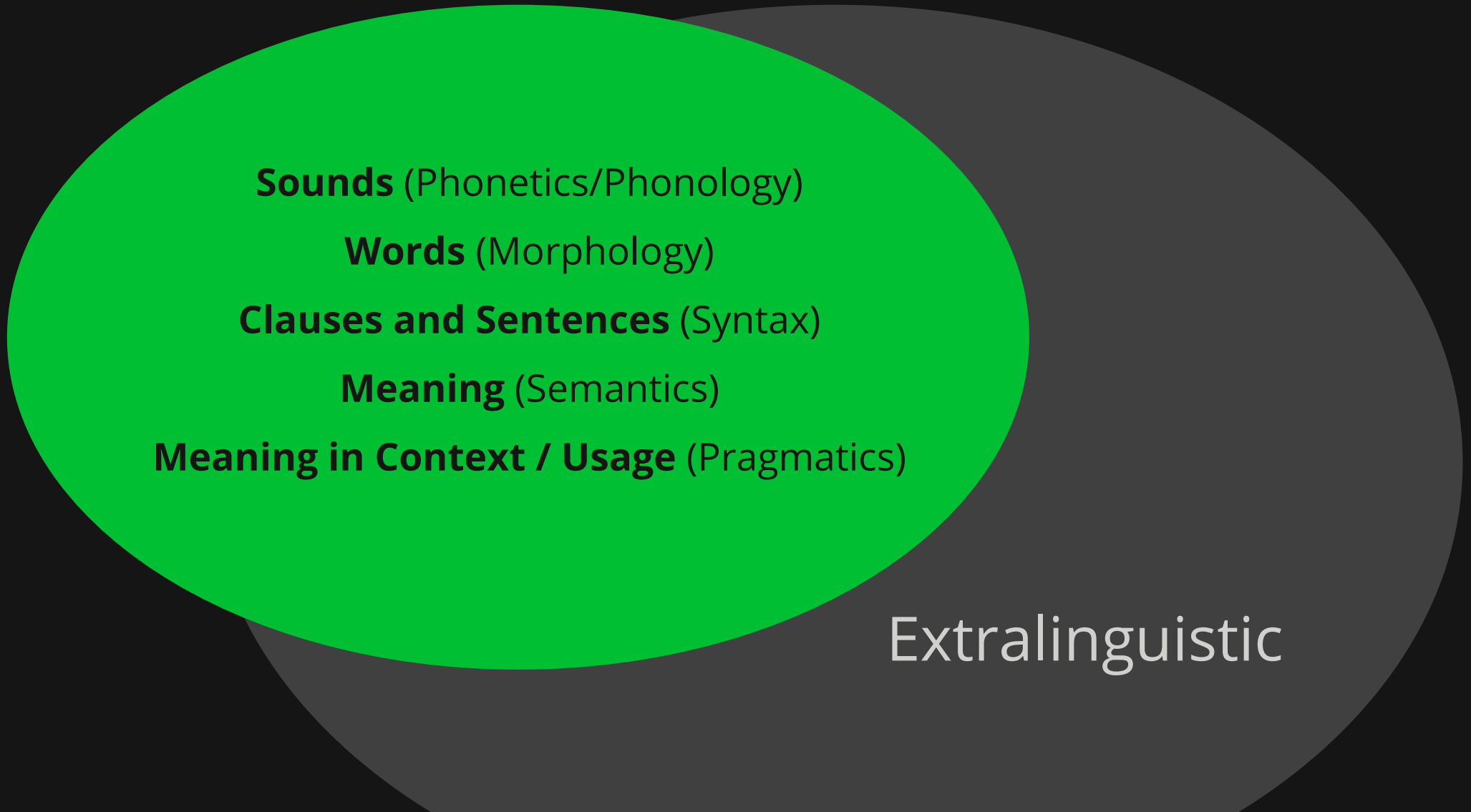
"Underlying the two-year-old's communicative activity is the capacity to develop a complex system of sounds and structures, plus computational procedures, that will allow the child to produce extended discourse containing a potentially infinite number of novel utterances. No other creature has been observed "using language" in this sense. **It is in this more comprehensive and productive sense that we say that language is uniquely human**."
(Yule 2014: 21)

A Linguists Perspective



... and many other things such as semantic, pragmatic, or discursive features.

Levels of Linguistic Analysis



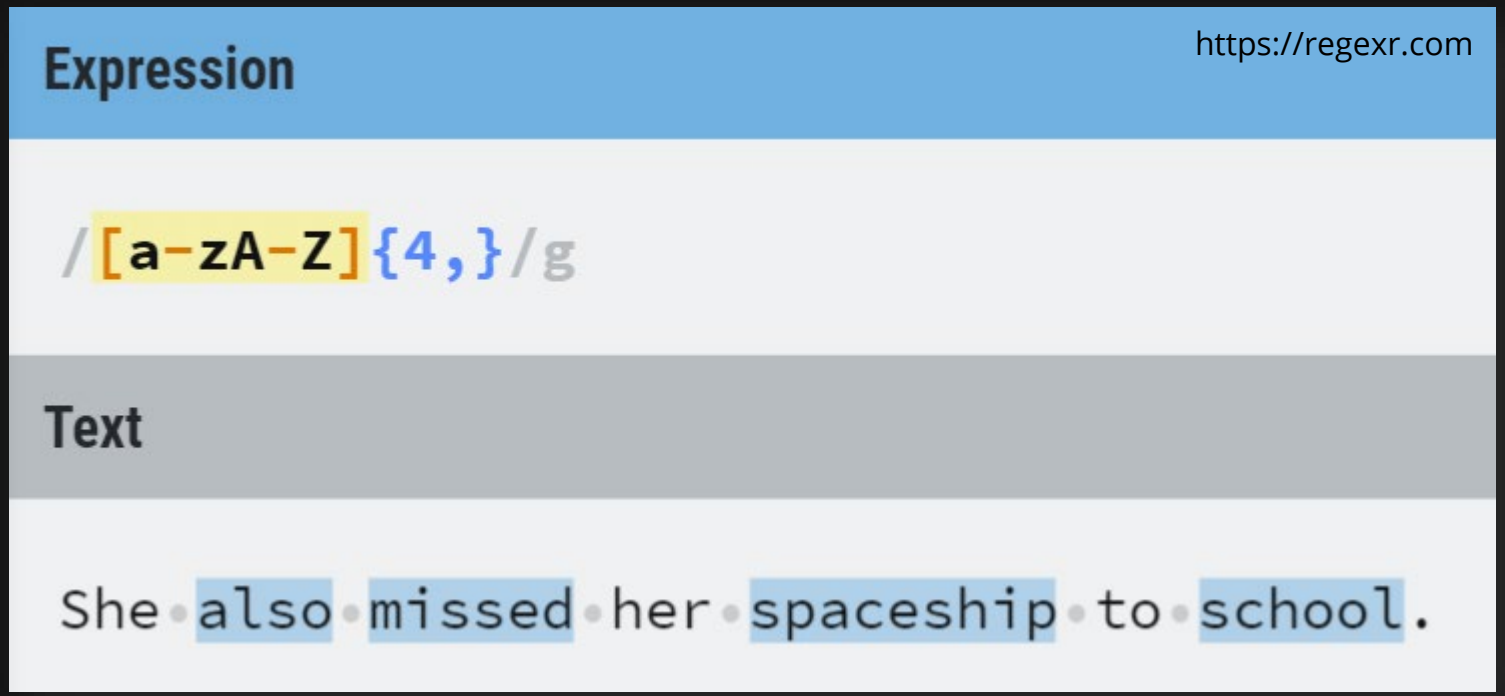
5. Exploring Some Basic Methods/Approaches in Python

Since we cannot possibly survey all major theoretical and practical approaches in two hours, we will simply explore and discuss some key methods from both CL and NLP!

- Regular Expressions
- Tokenization
- Stemming and Lemmatization
- Frequency Analysis
- (Simple) N-Gram Language Models
- PoS Tagging
- Dependency Grammar / Parsing
- Named Entity Recognition (NER)
- Vectorization (CountVectorizer)
- Word Embeddings (Word2Vec)

Regular Expressions

→ In plain English, a regular expression (RegEx) is a search pattern



"I need all words with four or more characters."

[a-zA-Z] – match a letter

{4,} – match four or more of the previous

Tokenization

→ Segmenting a string (e.g. a sentence) into tokens – essentially word forms.

→ Non-trivial because of (very common) cases such as multi-word units.

She was sad because she missed her spaceship to school.

10 tokens and 9 types

She also didn't have her school books!

Tokenization – n-Grams

→ n-grams are sequences of words (or other units such as characters)

→ word-based n-grams ‘carry’ both lexical as well as grammatical information

She was sad because she missed her spaceship to school.

2-gram / bigrams	**3-gram / trigram**
She was	She was sad
was sad	was sad because
sad because	sad because she
…	…

Stemming & Lemmatization

Stemming: reducing inflected (or derived) word forms to their stem

Lemmatization: grouping word forms by identifying their lemma (= 'dictionary form'). In comparison to stemming, a lemmatizer has additional (contextual) knowledge.

Word Form	Stemming	Lemmatization
hacked	hack	hack
better	better	good

Frequency Analysis

- How often is a particular word / n-gram / grammatical construction being used in a text or corpus?
- Compared to a reference, is a particular item being over-/underused?

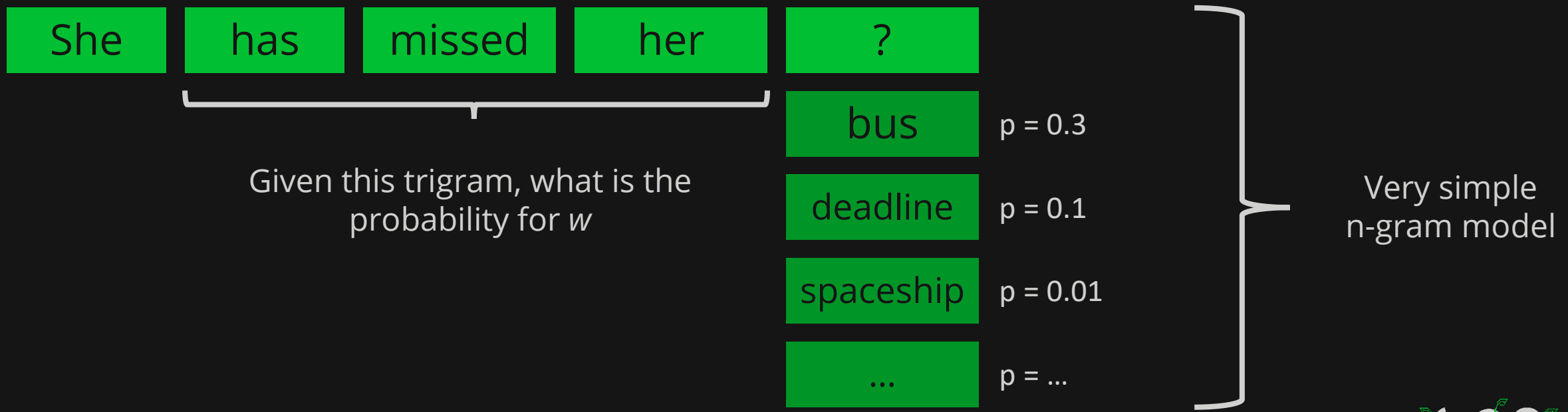
Frequency Table for COCA

Frequency			
rank	word/lemma	PoS	frequency
1	the	a	22038615
2	be	v	12545825
3	and	c	10741073
4	of	i	10343885
5	a	a	10144200
6	in	i	6996437
7	to	t	6332195
8	have	v	4303955
9	to	i	3856916
10	it	p	3872477

wordfrequency.info

(Simple) Language Models

→ Most language models try to predict which word(s) could come next given a previous sequence of words.

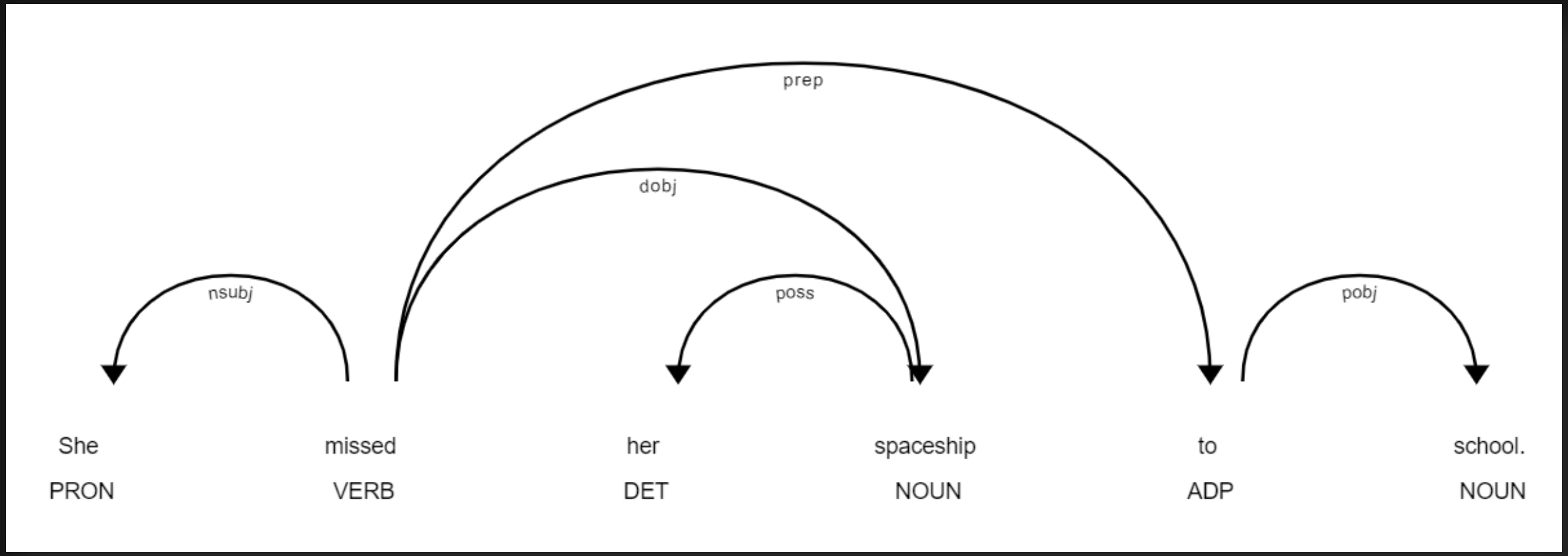


Part-of-Speech Tagging

→ It is very often useful to tag words according to their grammatical category (e.g. noun, verb, adjective, …)

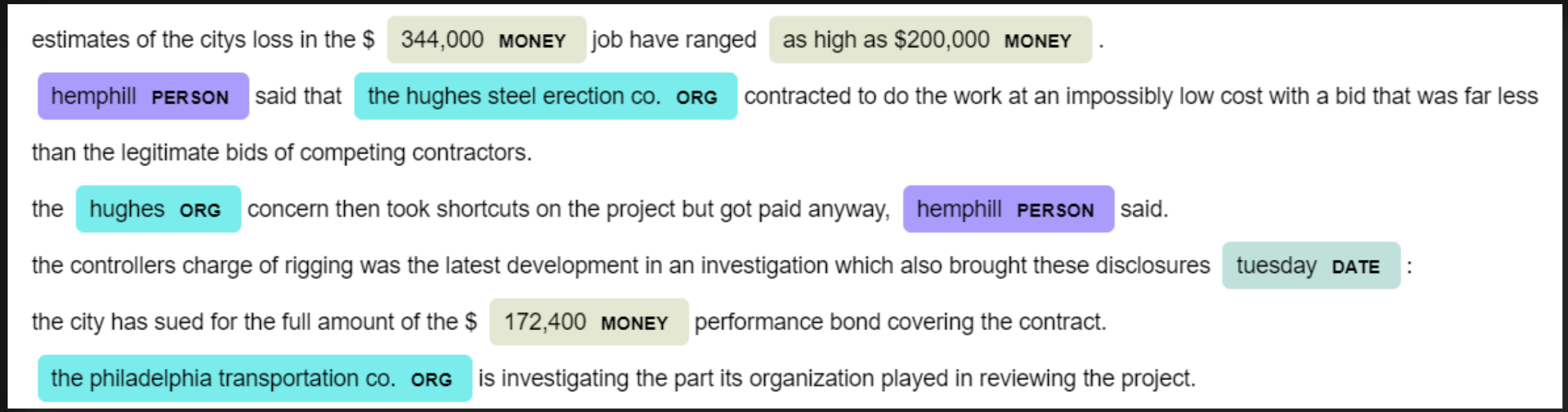
She	Pronoun
missed	Verb
her	Determiner
spaceship	Noun
to	Adposition (Preposition)
school	Noun

Dependency Grammar / Parsing



Named Entity Recognition (NER)

→ In NER, we use pre-trained models in order to label entities (e.g. people or places) in a text or corpus.



CountVectorizer

→ For many applications (e.g. using text as a feature for ML), we need to vectorize language. A very simple approach is a CountVectorizer.

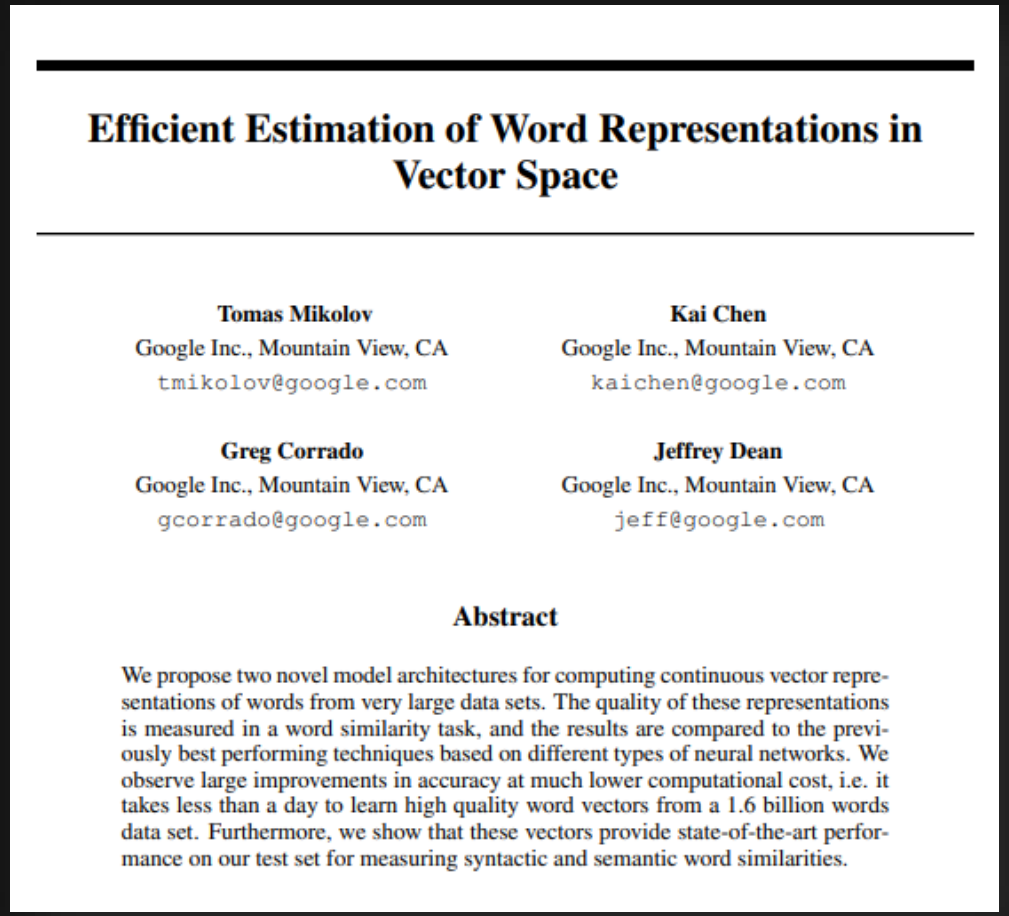
She missed her spaceship to school.
The spaceship usually is a little bit late.

0 1 0 0 0 1 1 1 1 0 1 0
1 0 1 1 1 0 0 0 1 1 0 1

Index	Word Form (lower)
0	bit
1	her
2	is
3	late
4	little
5	missed
	...

Word Embeddings (Word2Vec)

→ Word2Vec is a (now) old, but seminal approach by Tomas Mikolov et al.



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

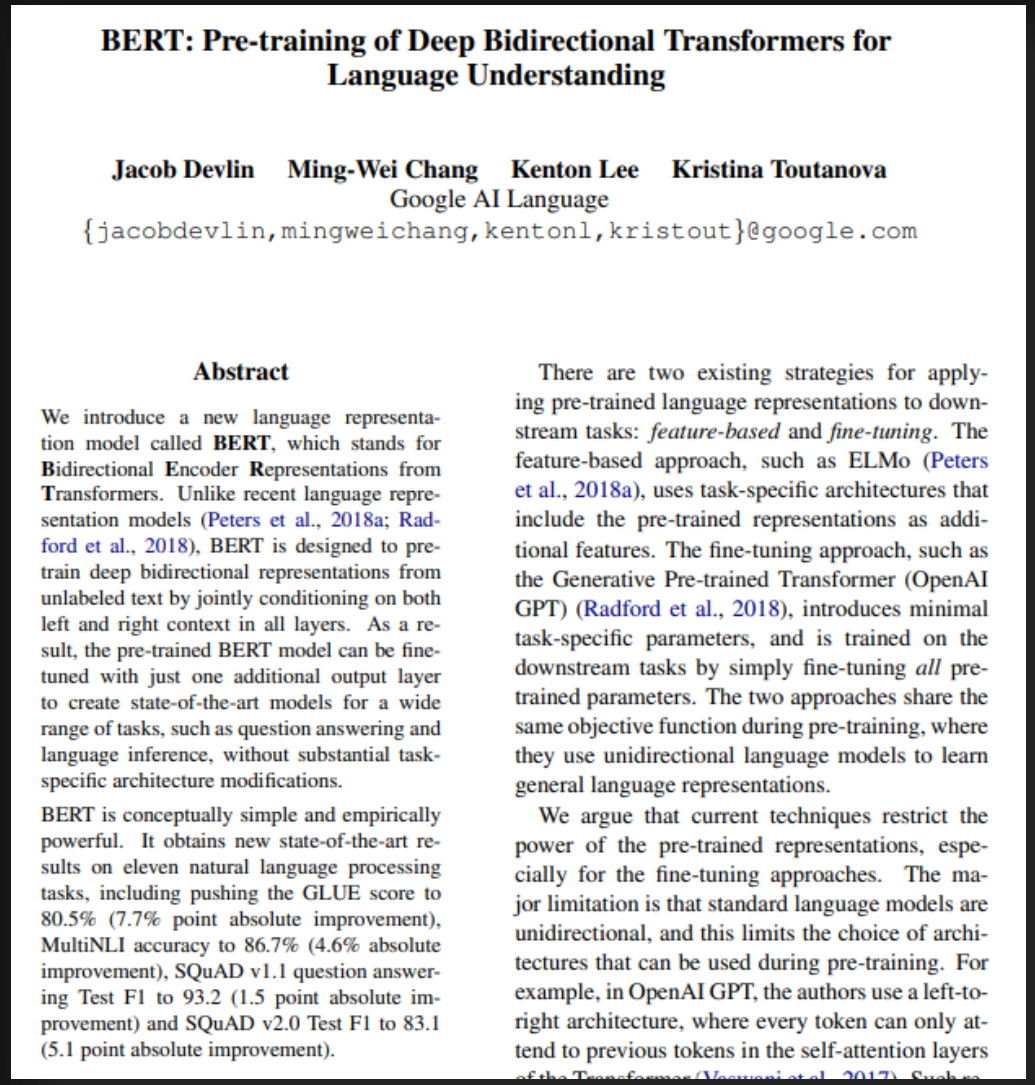
We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

Due to it producing powerful word-embedding (a language model), **Word2Vec** allows us, for example, to do operations such as:

queen + men = king (via vector operations)

Alternative: fastText, GloVe, ...

6. State-of-the-Art ML/AI Approaches



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers of the Transformer (Vaswani et al., 2017). Such ra-

BERT (also as a stand-in for various transformer based approaches) is an extremely powerful and new technique/model for NLP developed by Google.

Demo 1 (BERT):
https://www.pragnakalp.com/demos/BERT-NLP-QnA-Demo/

Demo 2 (GTP-2):
https://transformer.huggingface.co/

7. What's Next?

Three recommended (free) resources that you can consult and explore if you are interested:

1. Jurafsky's and Martin's *Speech and Language Processing*
2. Various courses and videos by Christopher Manning
3. Sebastian Ruder's *NLP-Progress*

Works Cited

- Jurafsky, Dan; Martin, James H. (2019): Speech and Language Processing. 3rd ed. Draft. 3rd. Edition. Stanford: Stanford University. https://web.stanford.edu/~jurafsky/slp3/.
- Jurafsky, Daniel; Martin, James H. (2008): Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 2nd. Edition. Upper Saddle River, NJ: Prentice-Hall (Prentice Hall Series in Artificial Intelligence). Online verfügbar unter https://www.cs.colorado.edu/~martin/slp.html.
- Mukherjee, Joybrato. 2005. *English Ditransitive Verbs: Aspects of Theory, Description and a Usage-Based Model.* Amsterdam and New York: Rodopi.
- Stefanowitsch, Anatol. Forth. *Corpus Linguistics: A Guide to the Methodology.* Textbooks in Language Sciences 8. Berlin: Language Science Press. https://paperhive.org/documents/items/Ns-5qhh9FO9s/text.
- McEnery, Tony, and Andrew Hardie. 2012. *Corpus Linguistics: Method, Theory and Practice.* Cambridge: Cambridge University Press.
- Yule, George (2014): The Study of Language. 5th. Edition. Cambridge, New York: Cambridge University Press.a