Semantic search and similarity ranking

Ane Berasategi

18. July 2019

LMU
LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Explore contextual embeddings

Understand contextual embeddings: ELMo, BERT, flair

History of embeddings

Ontology matching

Semantic search

Why is this interesting?

- Contextual embeddings have achieved unprecedented results in many tasks
    - Natural language understanding
    - Question answering
    - NER
- But what are they, how do they work, how do they represent language?
- Do they follow (my) intuition on sentence similarity?
    - A invited B for lunch vs.
    - A did not invite B for lunch vs.
    - B invited A for lunch

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Similarity ranking

- Word vectors
- Contextualized word embeddings
- ELMo
- BERT
- Flair

Part 3: Experiments

- Word order
- Lexical similarity
- Synonyms
- Out of vocab words

Part 4: Conclusion

1. Semantic search

- **Lexical** search: literal matches of the query words
  - Anthony Hopkins <u>age</u> → relevant results
  - <u>How old</u> is Anthony Hopkins? → not relevant results
- **Semantic** search: search with meaning, understand the intention of the user
  - Why is my laptop overheating?
  - How many continents are there in the world?

1. Semantic search

- **Lexical** search: literal matches of the query words
  - Anthony Hopkins <u>age</u> → relevant results
  - <u>How old</u> is Anthony Hopkins? → not relevant results
- **Semantic** search: search with meaning, understand the intention of the user
  - Why is my laptop overheating?
  - How many continents are there in the world?
    - Correct answer: 6

1.1. Ontology matching

The search engine has a huge knowledge graph / ontology with past searches

- Ontology: a representation of semantic relations between documents.

Pipeline:

1. New query arrives
2. Query broken into root terms: POS tagging removal, NER, conversion to embeddings, etc
3. Return the closest/more relevant/semantically most similar documents from the ontology (similarity ranking)

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Similarity ranking

- Word vectors
- Contextualized word embeddings
- ELMo
- BERT
- Flair

Part 3: Experiments

- Word order
- Lexical similarity
- Synonyms
- Out of vocab words

Part 4: Conclusion

2.1. Word vectors: history

1980s: LSA

↓

2000: Bengio et al, distributed representation for words

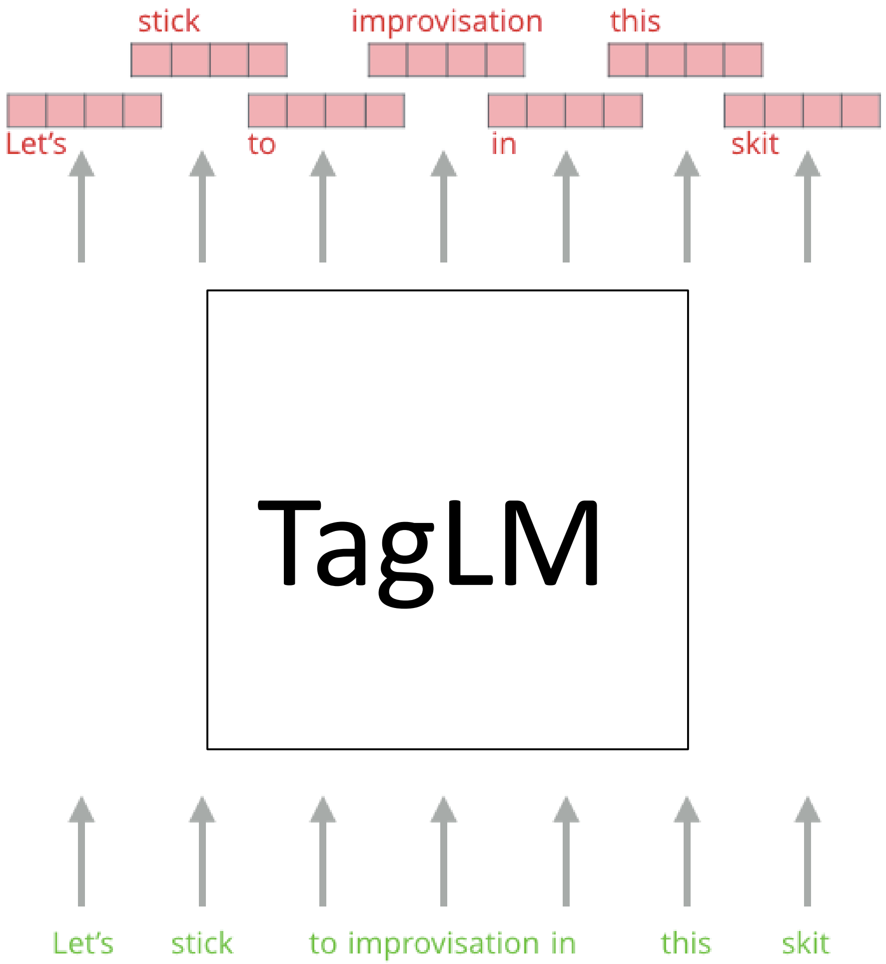
↓

2013: Mikolov et al, word2vec

- Pre-trained word embeddings became the norm (word2vec, GloVe, FastText) as input to NNs
- Each word gets an embedding vector → Irrelevant of the context, part of speech, polysemy

2.2. Contextualized word embeddings

- 2017, Peters et al.: give the words an embedding vector based on its context, in order to:
  - capture word meaning in that context
  - capture other contextual information
- TagLM: **semi-supervised** approach to add contextual embeddings to word embeddings from **bidirectional language models**
  - ForwLM + backLM embeddings has better performance than just forwLM embeddings
- Output from TagLM: a single context-independent representation for each word, the output layer of the LSTM.



[Semi-supervised sequence tagging with bidirectional language models](), Peters et al., 2017

2.3. ELMo

- Why just the last layer of the LSTM? Use all layers + share weights between forwLM and backLM

Deep contextualized word representations, Peters et al., 2018

2.3. ELMo

- Why just the last layer of the LSTM? Use all layers + share weights between forwLM and backLM

- Higher-level layers of the LSTM capture context-dependent aspects of word meaning
- Lower-level layers of the LSTM capture aspects of syntax and can be used for POS tagging

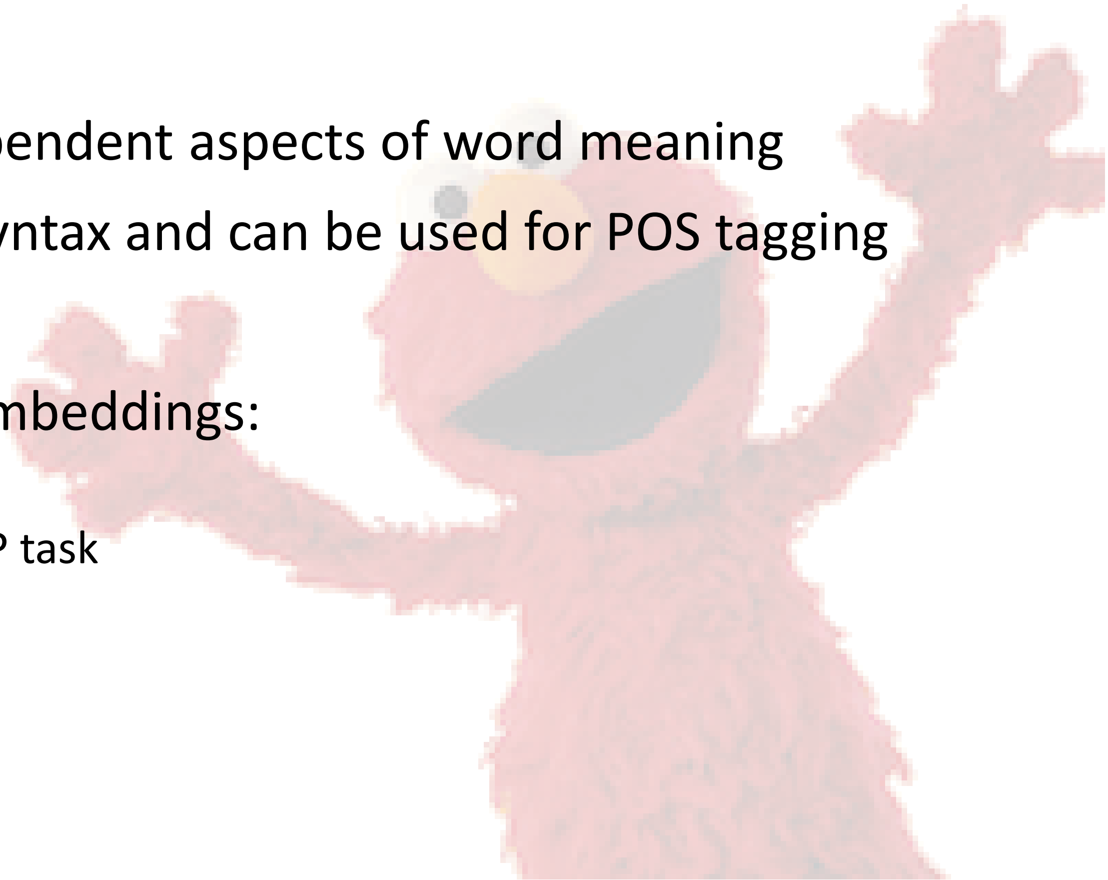
Deep contextualized word representations, Peters et al., 2018

2.3. ELMo

- Why just the last layer of the LSTM? Use all layers + share weights between forwLM and backLM

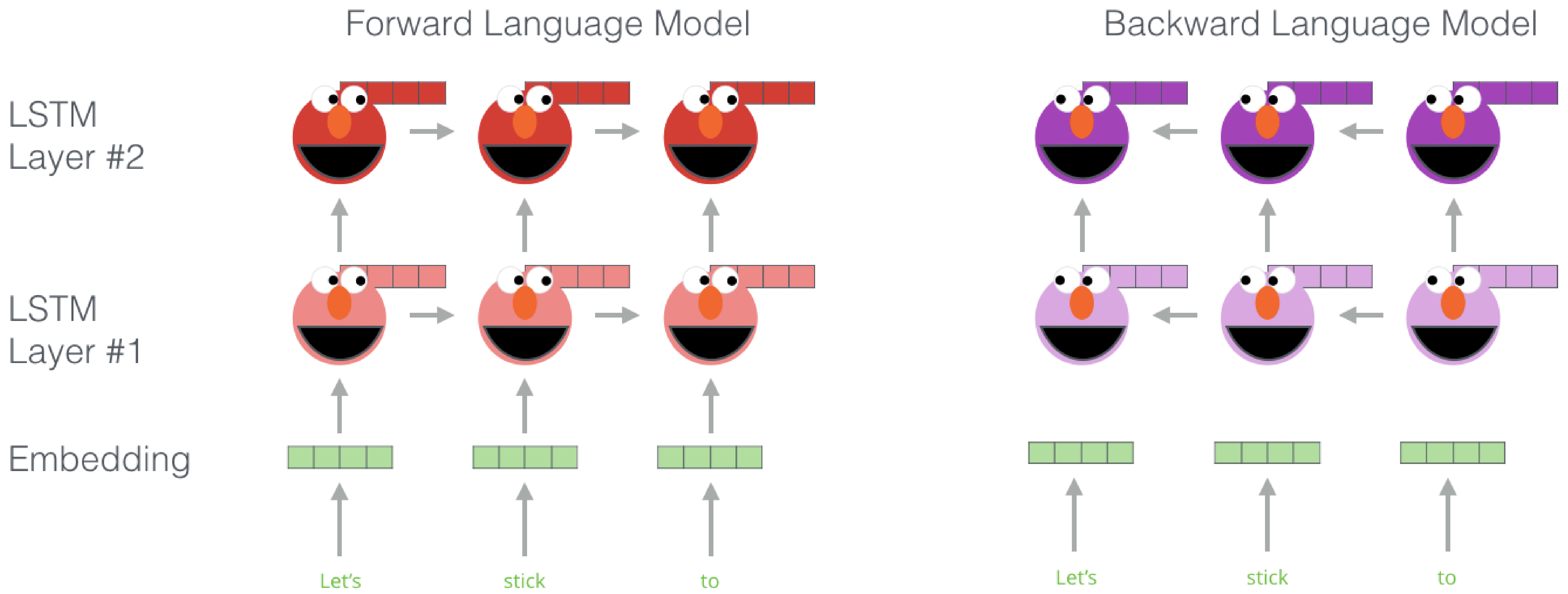
- Higher-level layers of the LSTM capture context-dependent aspects of word meaning
- Lower-level layers of the LSTM capture aspects of syntax and can be used for POS tagging

- ELMo is a **feature-based approach** for contextual embeddings:
  - different architectures for different NLP tasks
  - The embeddings are added as additional inputs to the NLP task



Deep contextualized word representations, Peters et al., 2018

Embedding of ‘stick’ in ‘Let’s stick to’: step #1



The Illustrated BERT, ELMo, and co., Jay Alammar, 2018

Embedding of ‘stick’ in ‘Let’s stick to’: step #2

1- Concatenate hidden layers

2- Multiply each vector by a weight based on the task

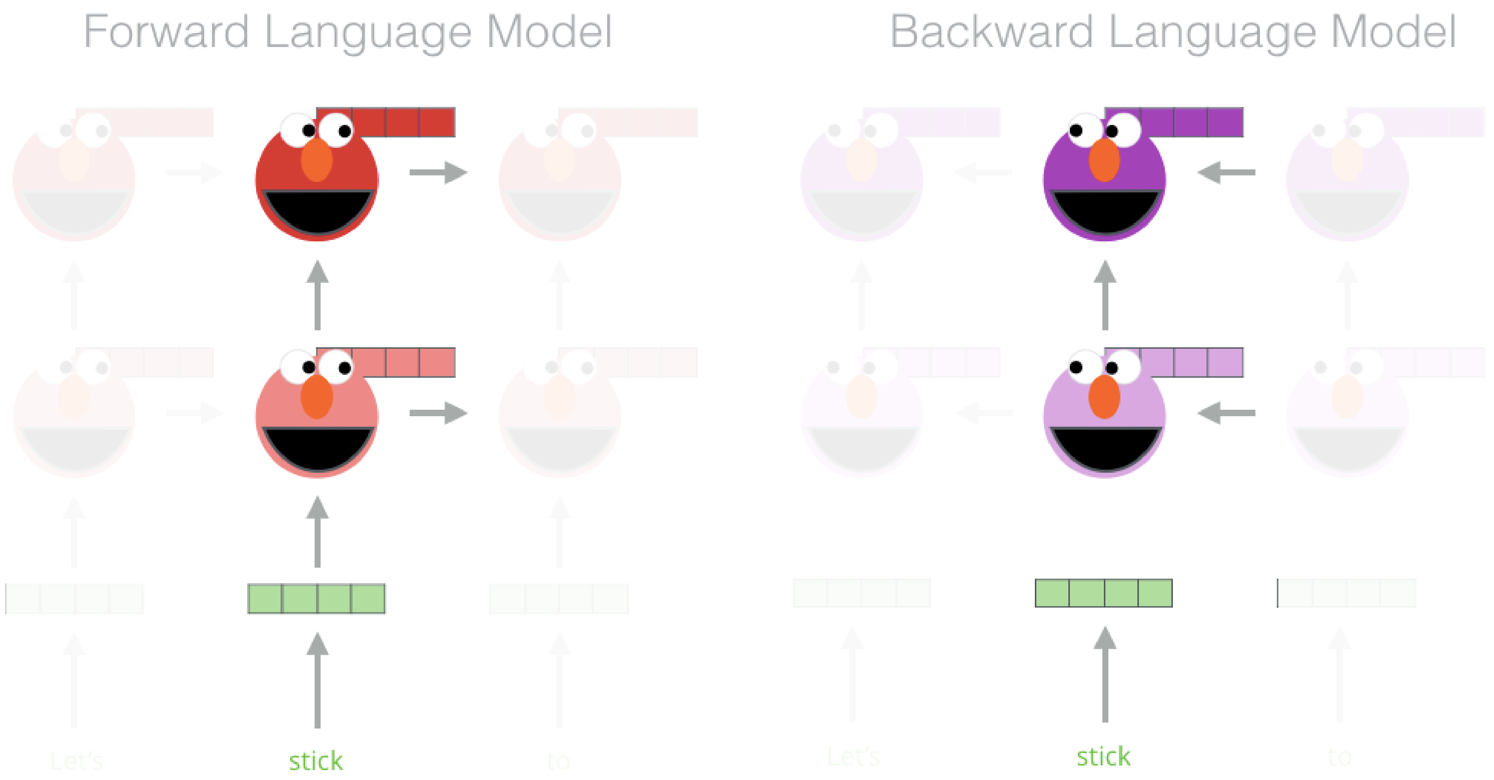
x s_2

x s_1

x s_0

3- Sum the (now weighted) vectors

ELMo embedding of “stick” for this task in this context



[The Illustrated BERT, ELMo, and co.,](#) Jay Alammar, 2018

2.4. The Transformer vs LSTM

- Recurrent, sequential models can't parallelize well → bottleneck at longer sequence lengths
- The Transformer:
  - No sequence, based solely on attention mechanisms
  - Deals with long-term dependencies better than LSTMs
  - More parallelizable than LSTMs
  - First transduction model relying entirely on **self-attention** to compute representations without using sequence-aligned networks

Attention Is All You Need, Vaswani et al., 2018
The illustrated Transformer, Jay Alammar, 2018

2.4. BERT

2.4. BERT

Expectation

2.4. BERT

Expectation

Reality

2.4. BERT

- **B**idirectional **E**mbedding **R**epresentations from **T**ransformers
- BERT jointly conditions on both left and right context in all layers (unlike ELMo)
- BERT is fine-tuned
  - to use it with a specific task, just fine-tune all pre-trained parameters end-to-end

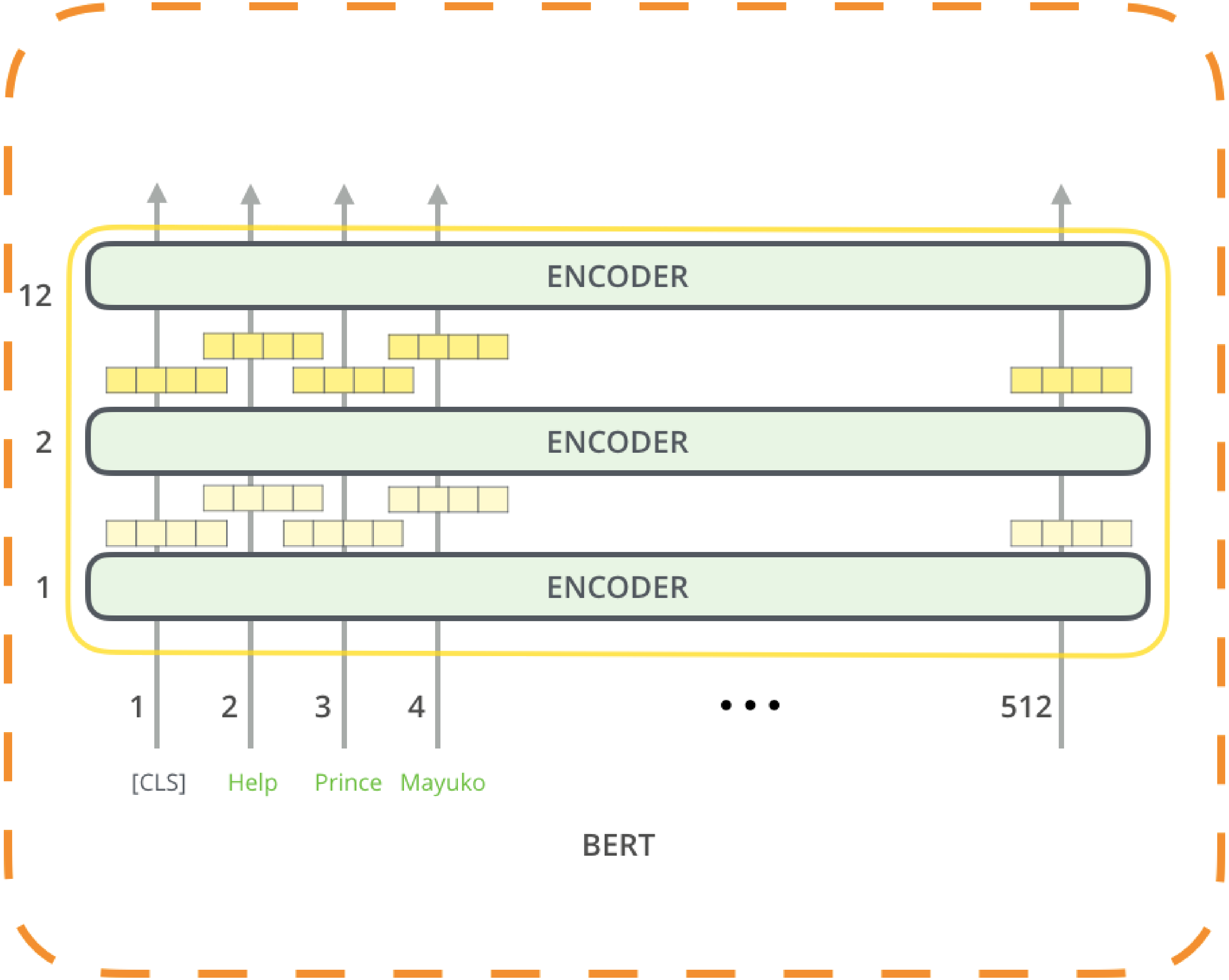
[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](), Devlin et al., 2018
[The Illustrated BERT, ELMo, and co.,]() Jay Alammar, 2018

2.4. BERT

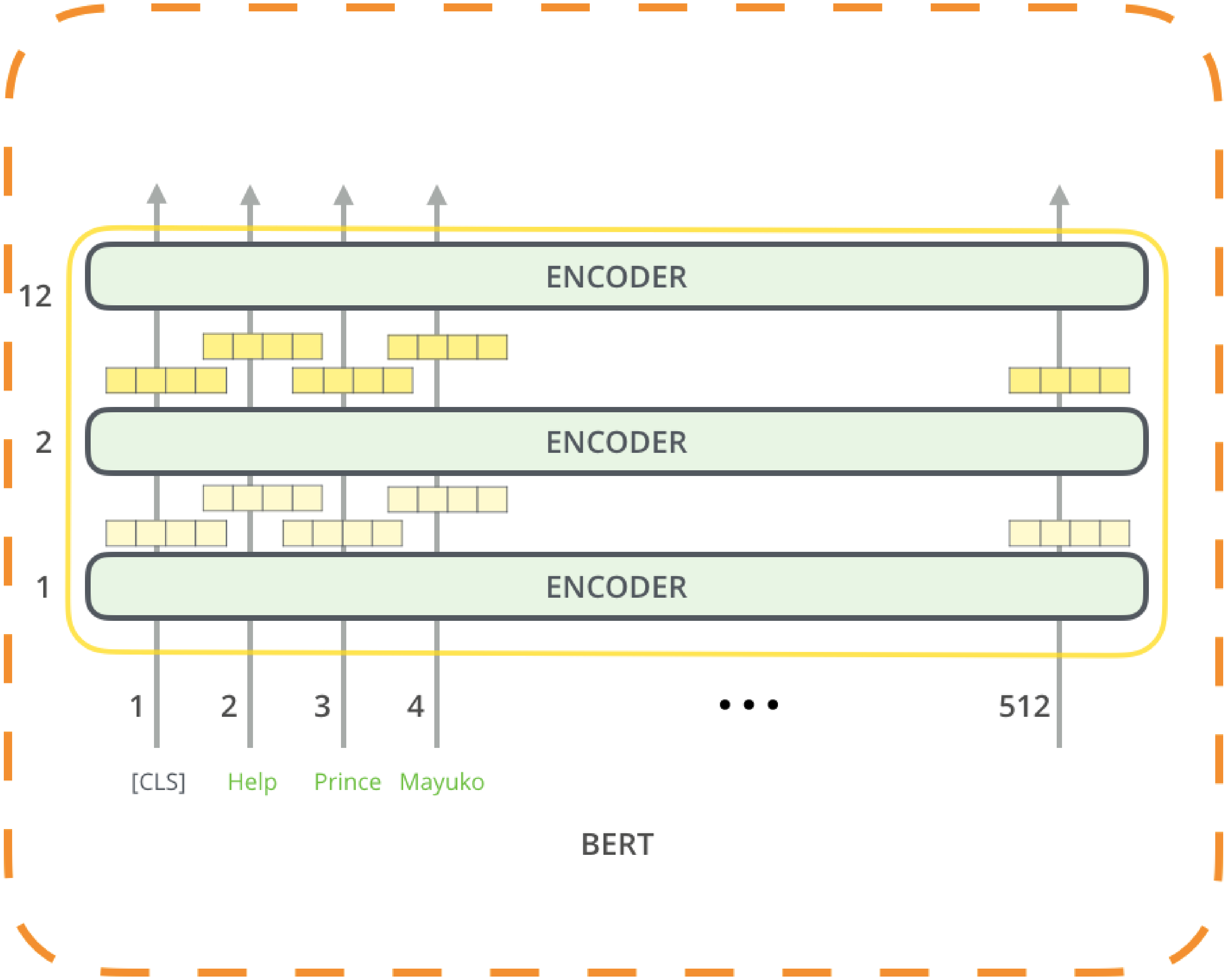
- **B**idirectional **E**mbedding **R**epresentations from **T**ransformers
- BERT jointly conditions on both left and right context in all layers (unlike ELMo)
- BERT is fine-tuned
  - to use it with a specific task, just fine-tune all pre-trained parameters end-to-end
- **Pre-trained BERT can be used to create contextualized word embeddings**

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al., 2018
The Illustrated BERT, ELMo, and co., Jay Alammar, 2018

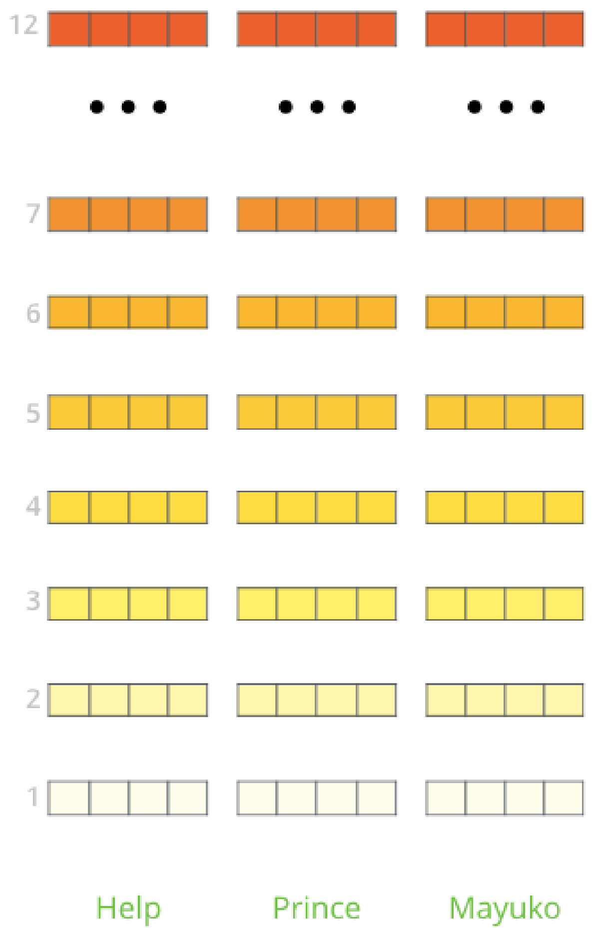
Generate Contexualized Embeddings



Generate Contexualized Embeddings



The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

Which contextualized embedding do we take for ‘Help’ in this context?

- The layer(s) to use depend on the NLP task
- Some tasks look at similarities between sentences, others care more about syntax and POS tagging

- Higher-level layers of the LM capture context-dependent aspects of word meaning
- Lower-level layers of the LM capture aspects of syntax

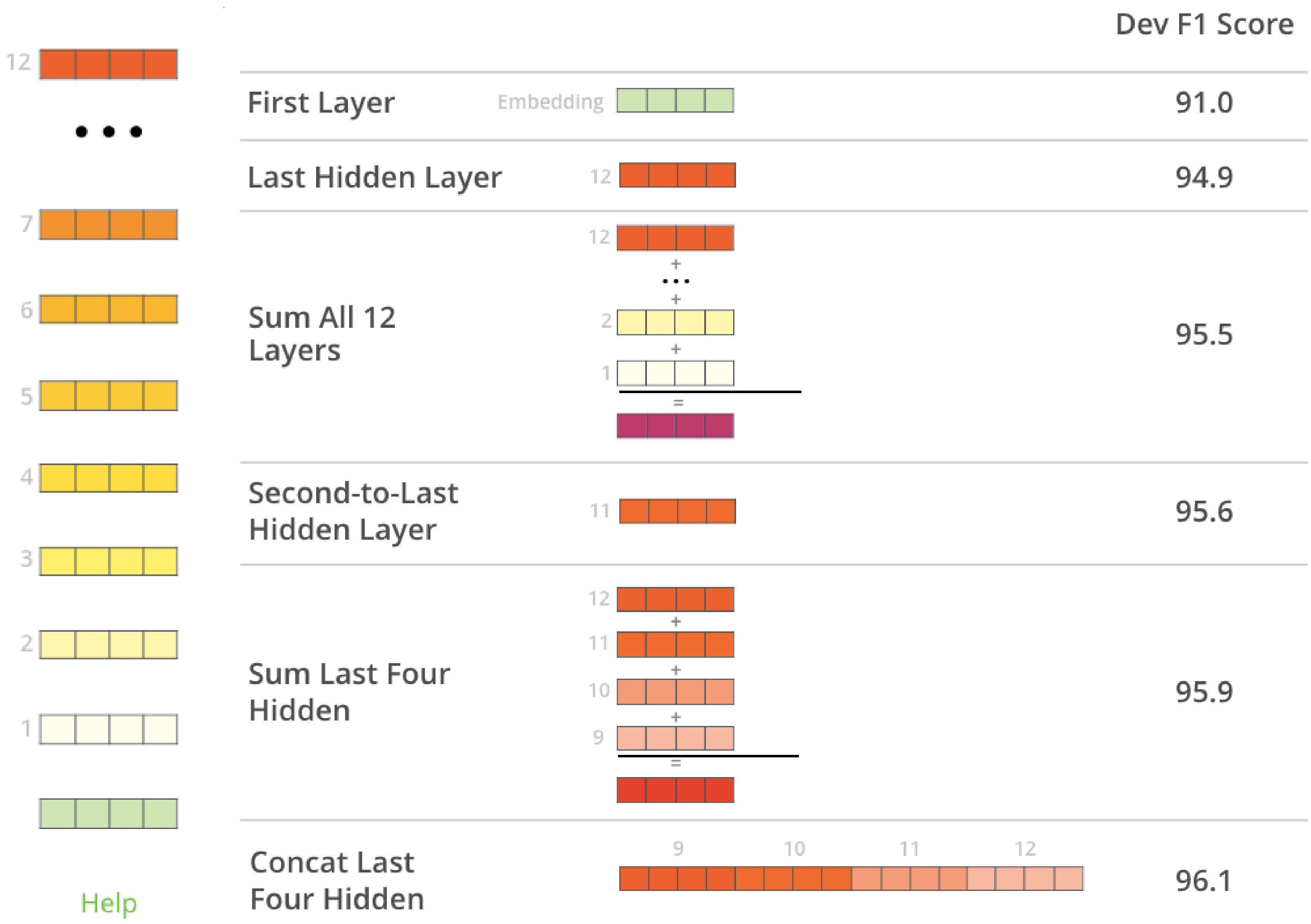
- For NER CoNLLTask 2003:

Which contextualized embedding do we take for 'Help' in this context?

- The layer(s) to use depend on the NLP task
- Some tasks look at similarities between sentences, others care more about syntax and POS tagging

- Higher-level layers of the LM capture context-dependent aspects of word meaning
- Lower-level layers of the LM capture aspects of syntax

- For NER CoNLLTask 2003:



2.5. flair

- WORK IN PROGRESS (didn’t read paper yet)
- trained without an explicit notion of what a word is.
- character-level models are shown to deal well with rare and out-of-vocabulary words and morphologically rich languages.

Contextual String Embeddings for Sequence Labeling, Akbik et al., 2018

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Similarity ranking

- Word vectors
- Contextualized word embeddings
- ELMo
- BERT
- Flair

Part 3: Experiments

- Word order
- Lexical similarity
- Synonyms
- Out of vocab words

Part 4: Conclusion

3. Experiments

- Experiment with contextual embeddings in English and multilingual (basque)
- Import ELMo, BERT-cased, BERT-uncased, and flair embeddings
  - BERT-uncased: text is lowercased before tokenisation and strips out accent markers
  - BERT paper: "uncased is typically better unless you know that case information is relevant to your task"
- For each experiment, create a reference sentence and several (dis-)similar sentences and rank them based on intuitive similarity
- Obtain contextual embeddings for each embedding type
- Calculate similarity between embedding(reference) and embedding(similar_sentence)
- Visualizations

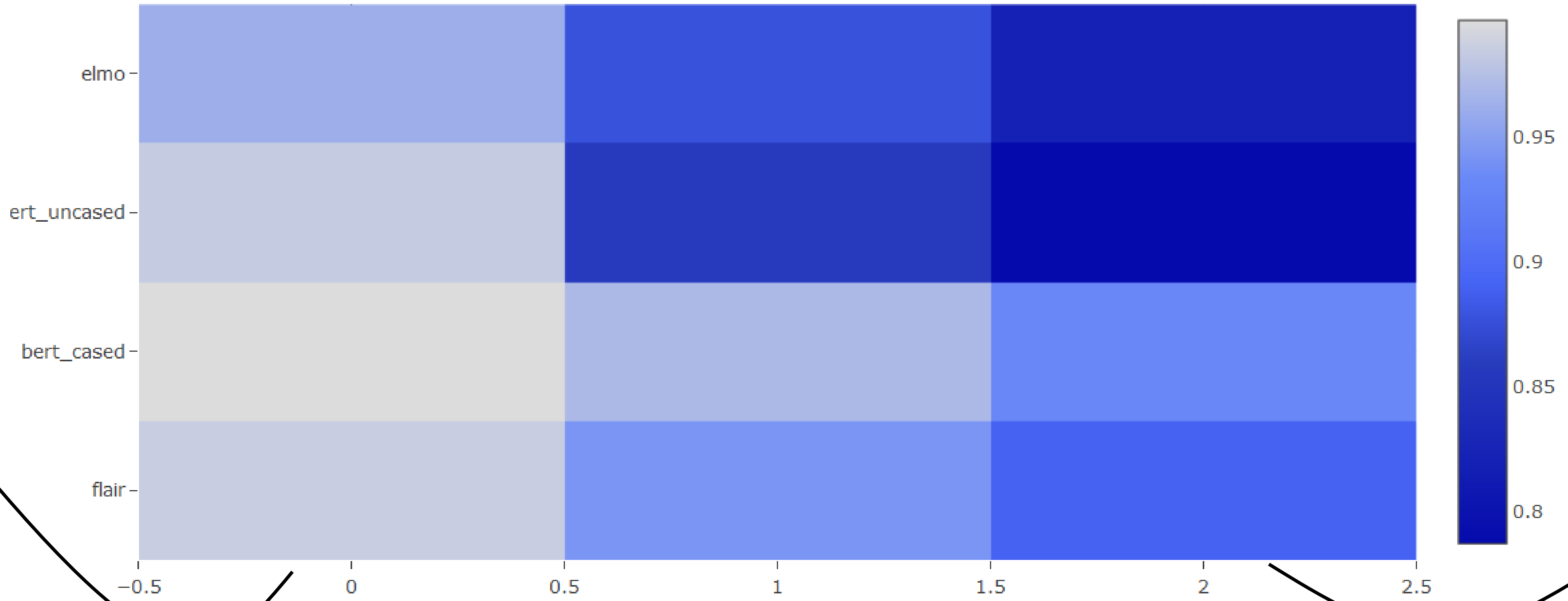
3. Experiments

	Input text type	Vocabulary size
ELMo	Words	700k words
BERT	subword	30k tokens
BERT multilingual	subword	120k tokens in total for 104 languages
flair	character	X characters

3.1. Does word order matter?

the doctor invites the patient for lunch	Change
0. the patient invited the doctor for lunch	Switch subject and object ('opposite' sentence)
1. the lunch invited the doctor for the patient	Change semantic role
2. for invited patient the doctor the lunch	Random ordering

'opposite' sentence has very high similarity

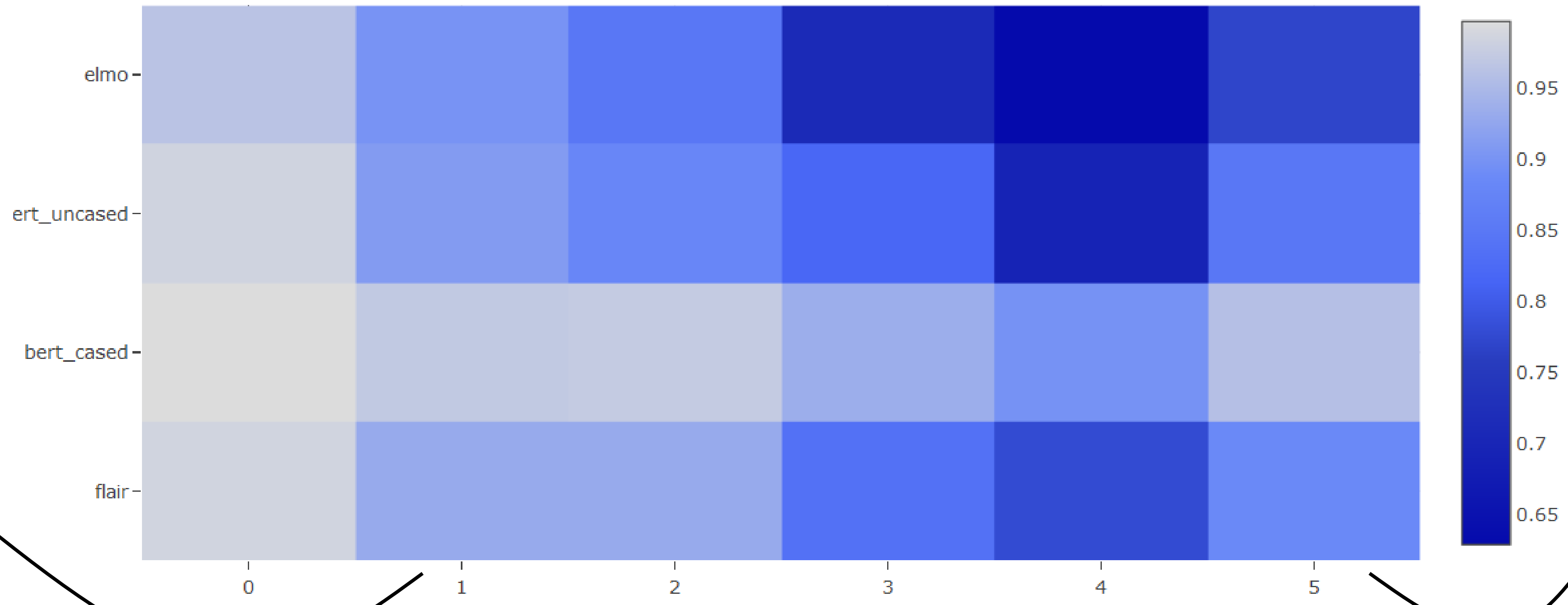


random ordering has lower similarity but still high

3.2. What is the impact of lexical similarity?

the doctor invites the patient for lunch	Change		Change
0. the patient invited the doctor for lunch	'opposite' sentence	*3. the doctor told the patient he was a fraud*	Subj. and obj. overlap
1. the doctor did not invite the patient for lunch	negation	*4. that is a matter between the doctor and the patient*	Subj. and obj. overlap
2. the child invited the grandfather for lunch	Subj. and obj. change	*5. the child and the grandfather got invited for lunch*	Invite, lunch overlap

negation sentence has very high similarity



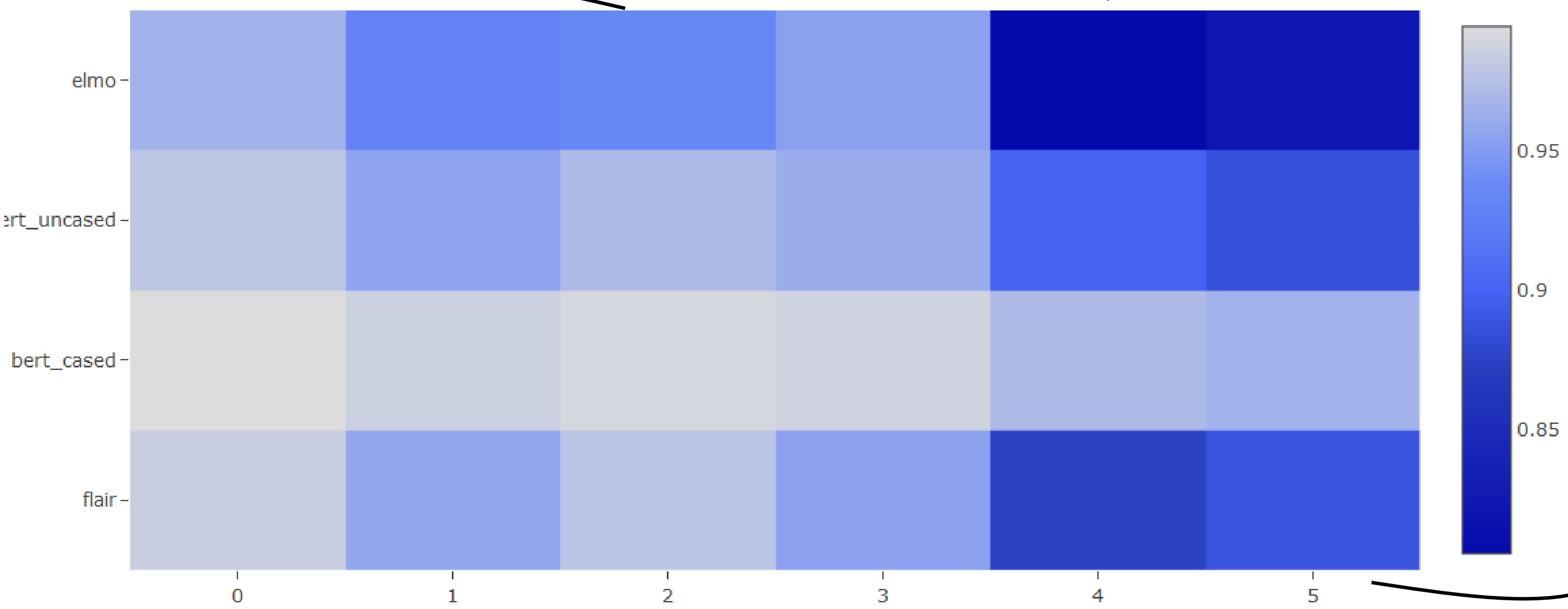
quite low result in ELMo, very high in bert_cased

invite for lunch yields a very high similarity

3.3. What is the impact of synonyms?

the doctor invites the patient for lunch	Change		Change
0. the surgeon invited the patient for lunch	Synonym of subj.	*3. the doctor invited the patient for a meal*	More general word
1. the doctor invited the sick person for lunch	More general synonym of obj.	*4. the doctor took the patient out for tea*	More general expression
2. the professor invited the patient for lunch	Synonym in different context	*5. the doctor paid for the patient's lunch*	Implicit action

unexpected result, aren't contextual embeddings supposed to generalize well to context?



2 synonyms at the same is the possible cause

The implicit action sentence doesn't have bad results

3.4. What is the impact of antonyms, meronyms…?

- Possible addition

3.5. What is the impact of out of vocabulary words?

- coming soon

3. Experiment takeaways

- Antonyms are more similar than synonyms

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Similarity ranking

- Word vectors
- Contextualized word embeddings
- ELMo
- BERT
- Flair

Part 3: Experiments

- Word order
- Lexical similarity
- Synonyms
- Out of vocab words

Part 4: Conclusion

4. Conclusion

- explain differences in results between bert_cased und bert_uncased
- why does bert_cased generally perform better
- differences between en – eu
- future work?

Semantic search and similarity ranking

Ane Berasategi

18. July 2019

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN