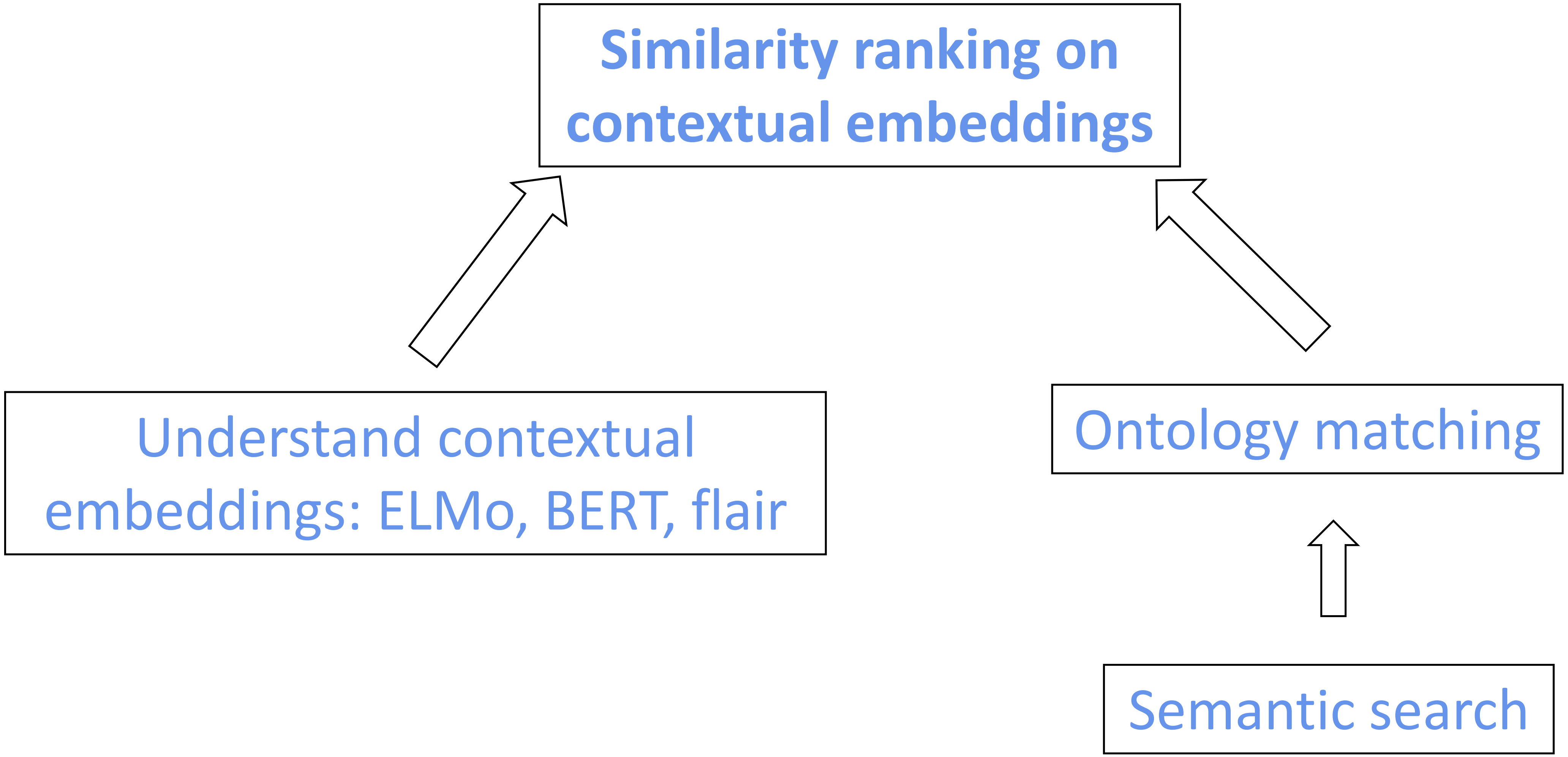
Semantic search and similarity ranking

Ane Berasategi

18. July 2019

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN


Similarity ranking on contextual embeddings
Understand contextual embeddings: ELMo, BERT, flair
Ontology matching
Semantic search

Why is this interesting?

- Contextual embeddings have achieved unprecedented results in many tasks
- 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

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?

1.1. 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: Contextualized word embeddings

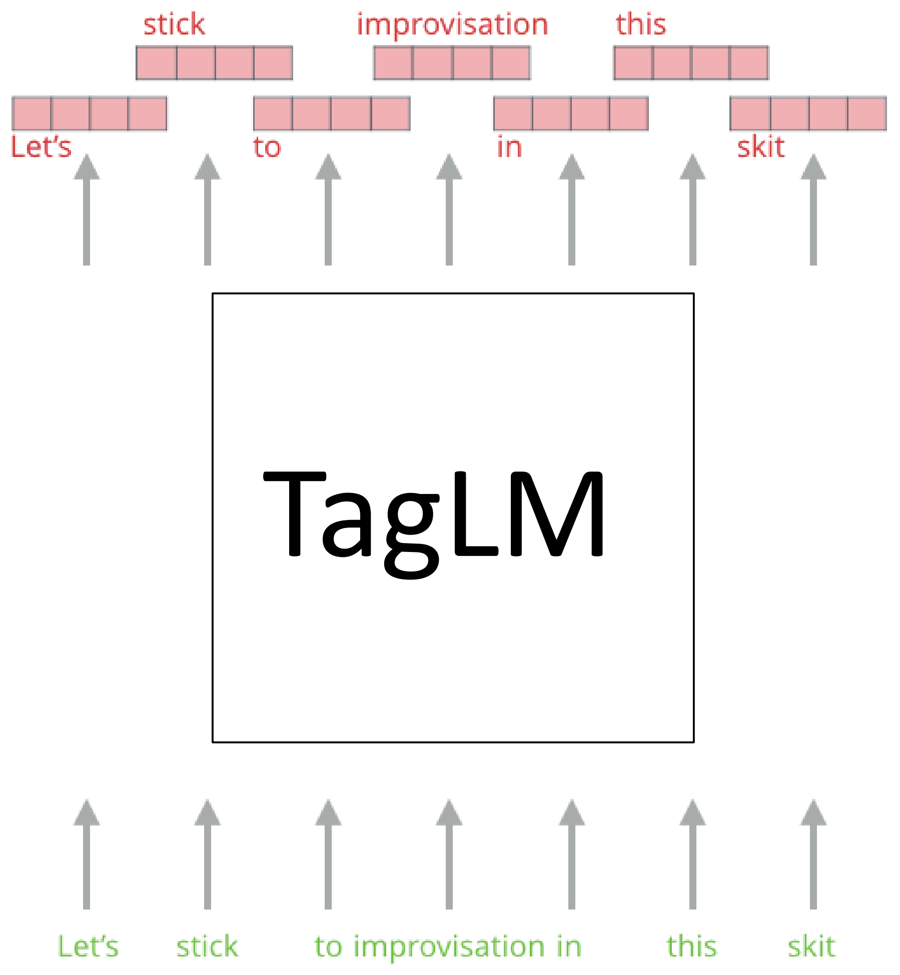
- ELMo
- BERT
- Flair

Part 3: Experiments

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

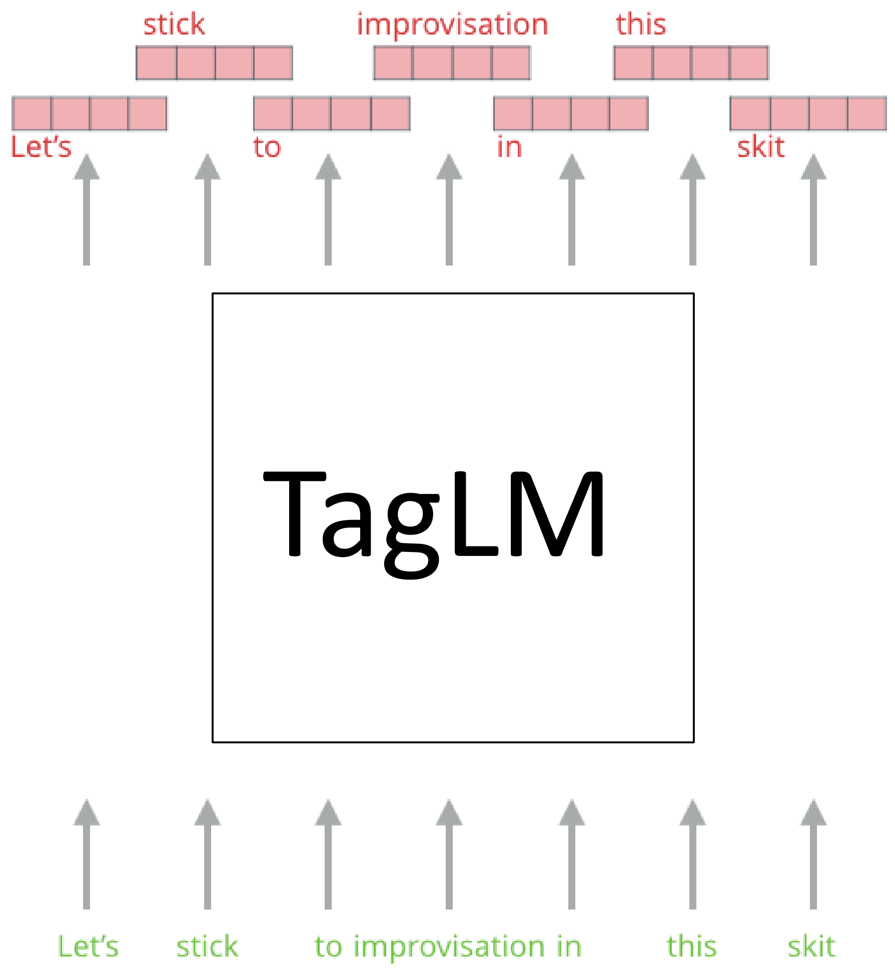
Part 4: Conclusion

2. Contextualized word embeddings



2. Contextualized word embeddings

- TagLM: **semi-supervised** approach to add contextual embeddings to word embeddings from **bidirectional language models**
  - fLM + bLM embeddings has better performance than just fLM embeddings
  - The two LMs are trained independently
- Input for TagLM: sequences of words
- Output from TagLM: a single **context-sensitive** representation for each word, the output layer of the LM/LSTM.

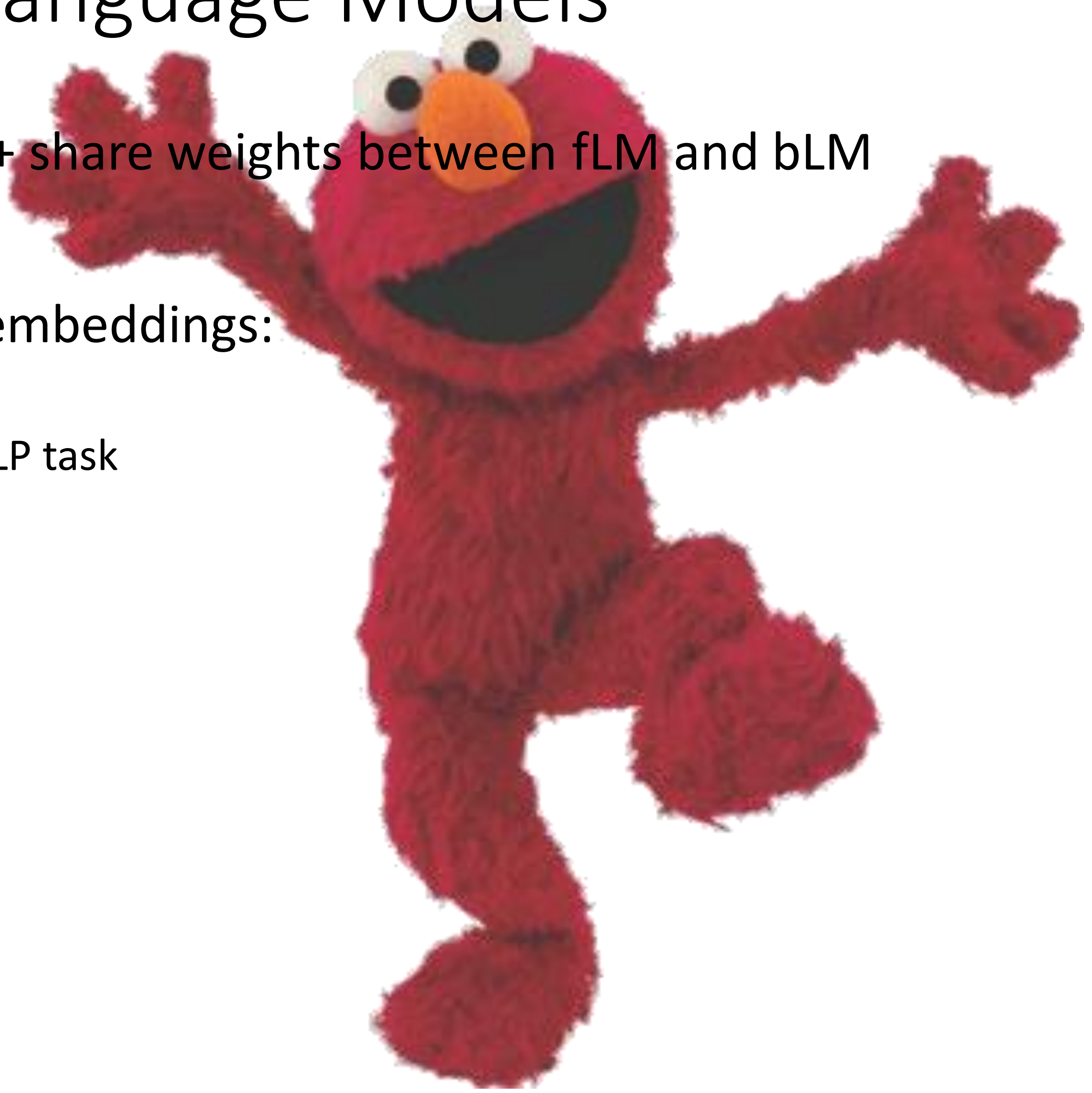


2.1. ELMo: Embeddings from Language Models

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

Deep contextualized word representations, Peters et al., 2018

2.1. ELMo: Embeddings from Language Models

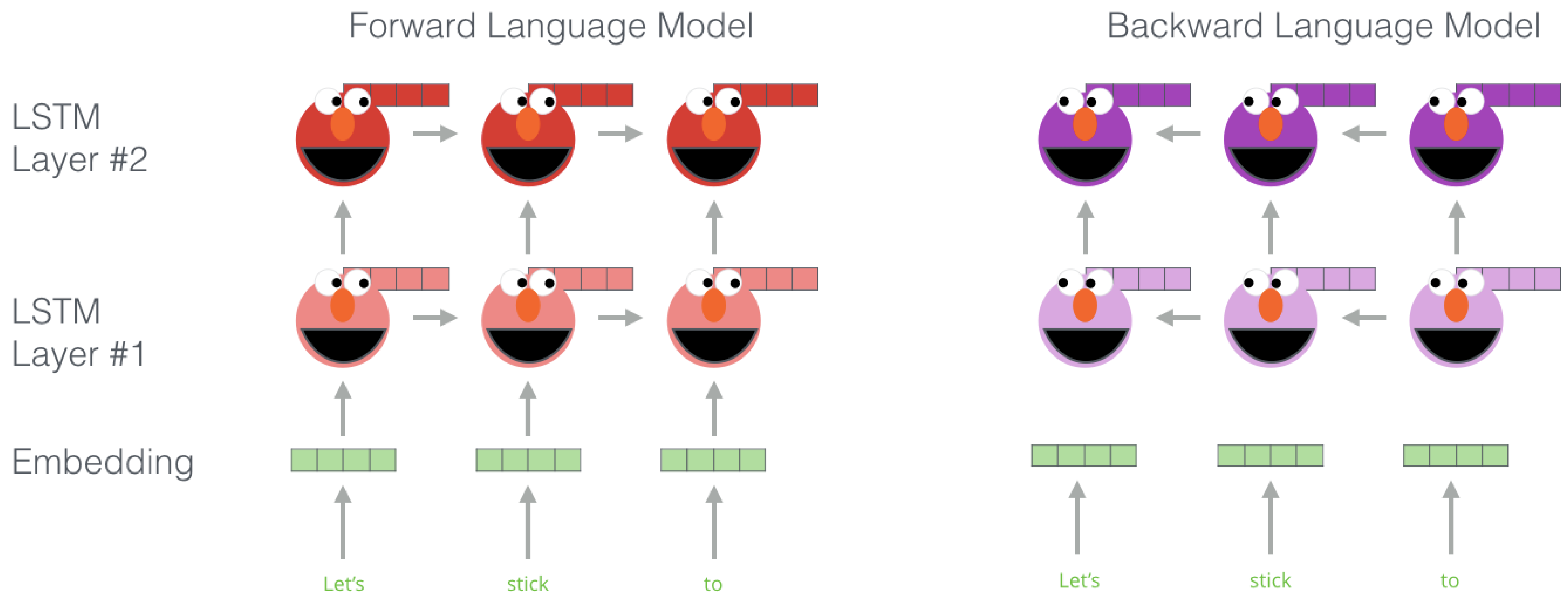


- Why just the last layer of the LSTM? Use all layers + share weights between fLM and bLM

- 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

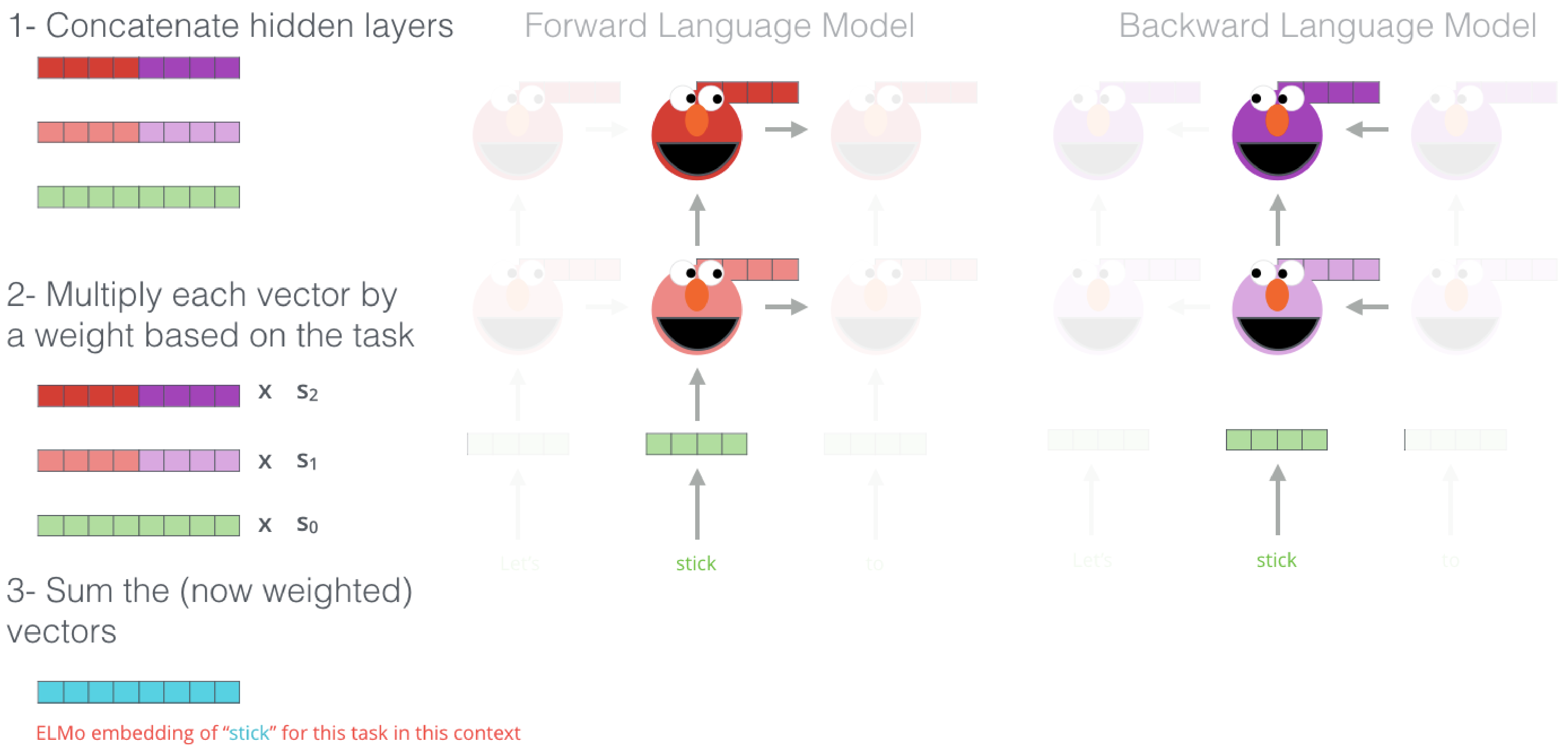
Deep contextualized word representations, Peters et al., 2018

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
Forward Language Model
Backward Language Model
2- Multiply each vector by a weight based on the task
x s_2
x s_1
x s_0
Let's
stick
to
Let's
stick
to
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

2.2. The Transformer vs LSTM

Attention Is All You Need, Vaswani et al., 2018
The illustrated Transformer, Jay Alammar, 2018
R-Transformer: recurrent neural network enhanced Transformer, Wang et al., 2019

2.2. The Transformer vs LSTM

- Problems with recurrent, sequential models such as LSTMs
  - can’t parallelize the computation procedure
  - can’t capture long-term dependencies
- The Transformer[1]
  - More parallelizable than LSTMs
  - Deals with long-term dependencies better than LSTMs
  - No sequence, based solely on **self-attention** mechanisms $\rightarrow$ no info about word order
  - Add position embeddings to get info about position
- Transformer’s problems[3]
  - Positional embeddings have limited effect
  - Transformers require huge design effort

Attention Is All You Need, Vaswani et al., 2018
The illustrated Transformer, Jay Alammar, 2018
R-Transformer: recurrent neural network enhanced Transformer, Wang et al., 2019

2.2. BERT: Bidirectional Embedding Representations from Transformers

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

2.2. BERT: **B**idirectional **E**mbedding **R**epresentations from **T**ransformers

- BERT jointly conditions on both left and right context in all layers (unlike ELMo)
- Input to BERT: sub-words
- BERT is a **fine-tuned** approach
  - to use it with a specific task, all pre-trained parameters are fine-tuned end-to-end

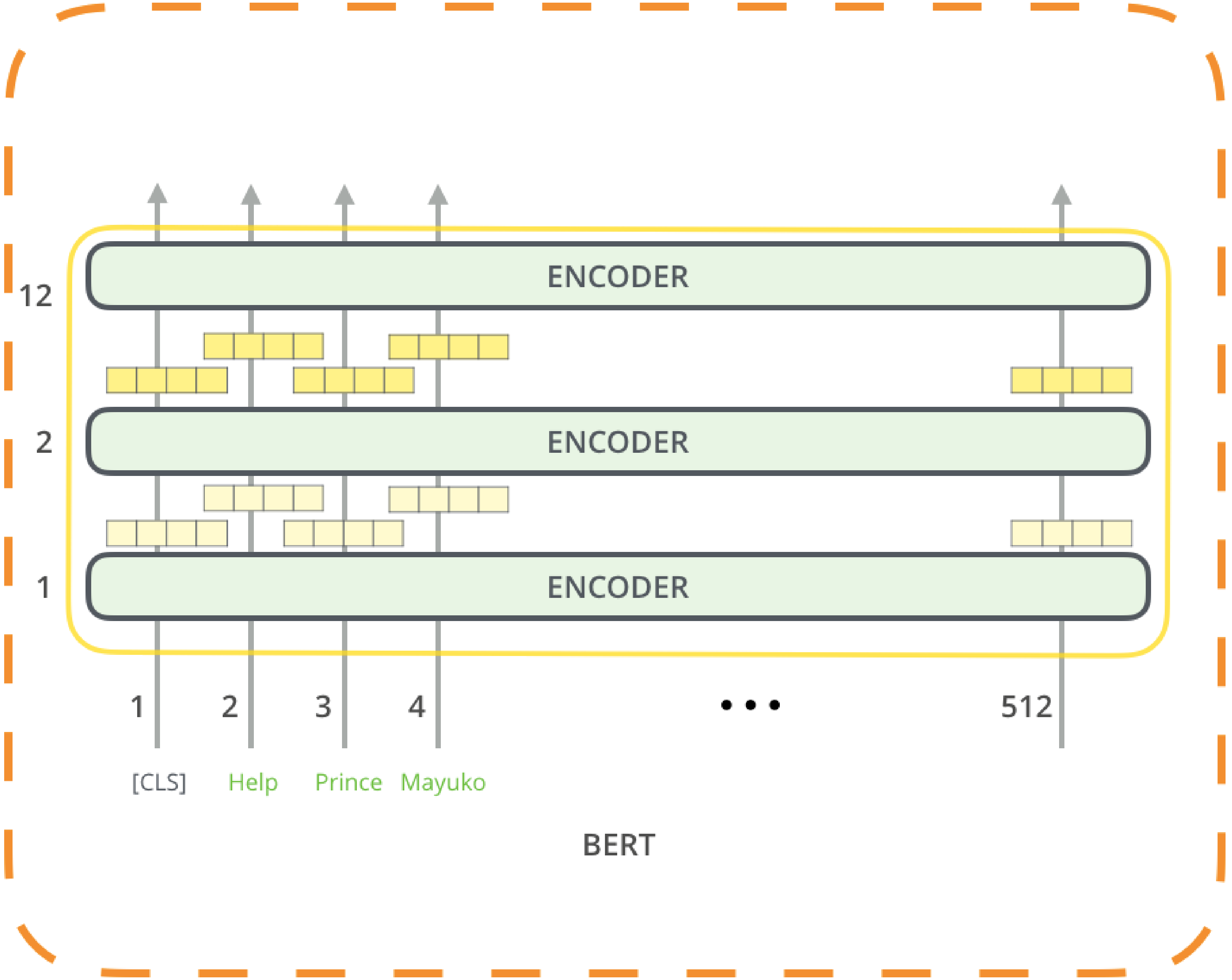
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin et al., 2018
The Illustrated BERT, ELMo, and co., Jay Alammar, 2018

2.2. BERT: Bidirectional Embedding Representations from Transformers

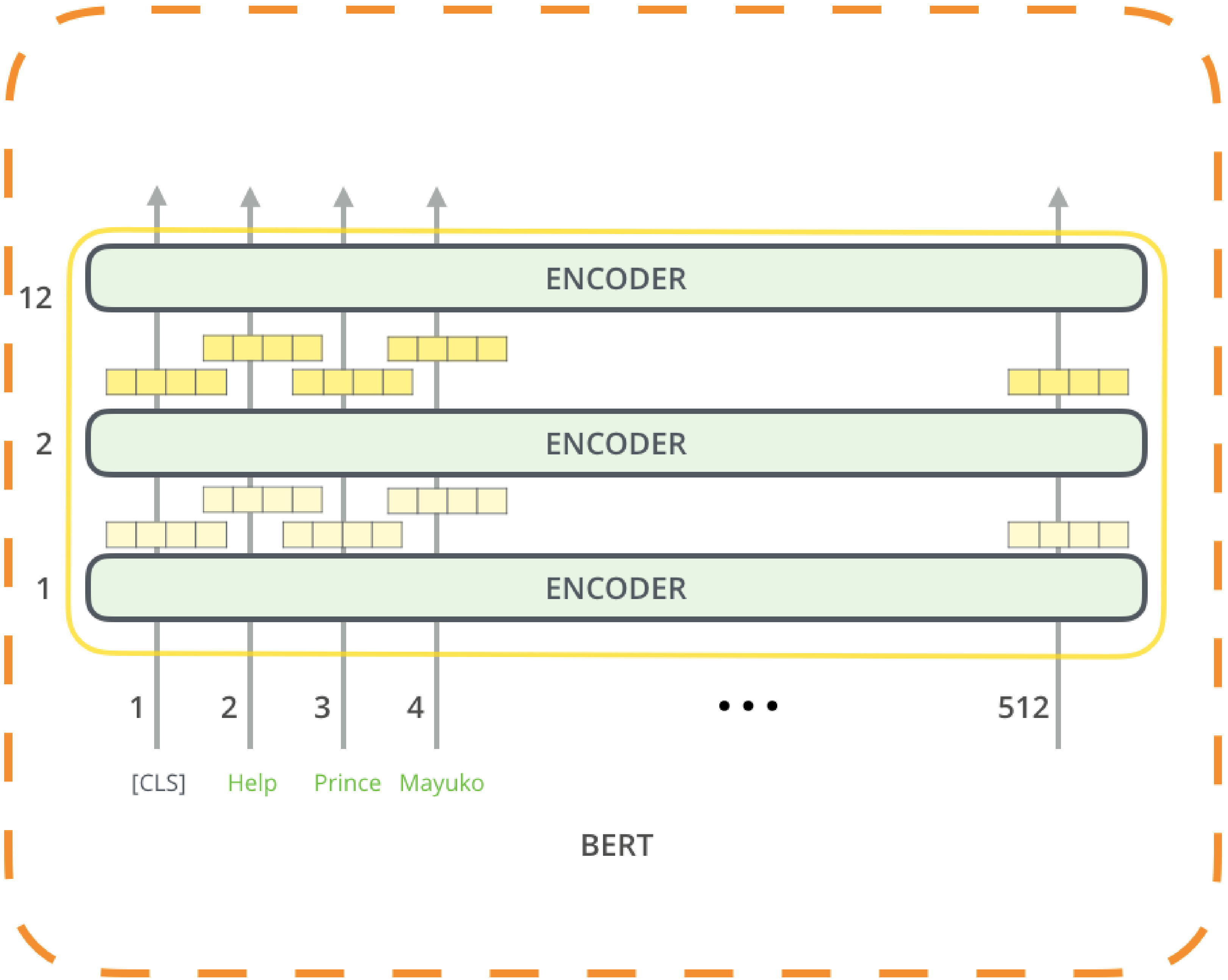
- BERT jointly conditions on both left and right context in all layers (unlike ELMo)
- Input to BERT: sub-words
- BERT is a **fine-tuned** approach
  - to use it with a specific task, all pre-trained parameters are fine-tuned end-to-end
- **Pre-trained BERT can be used to create contextualized embeddings**

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

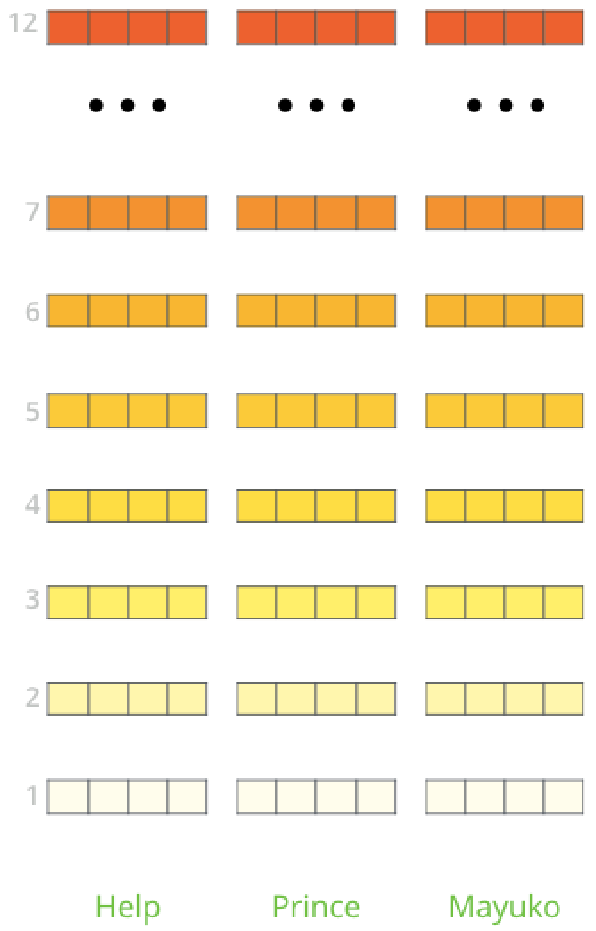
Generate Contexualized Embeddings



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

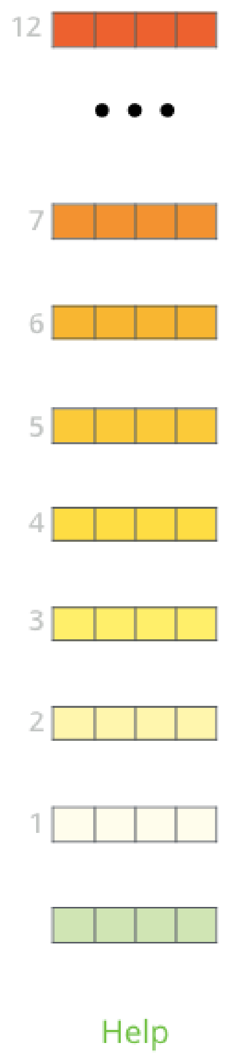
- For NER CoNLLTask 2003:

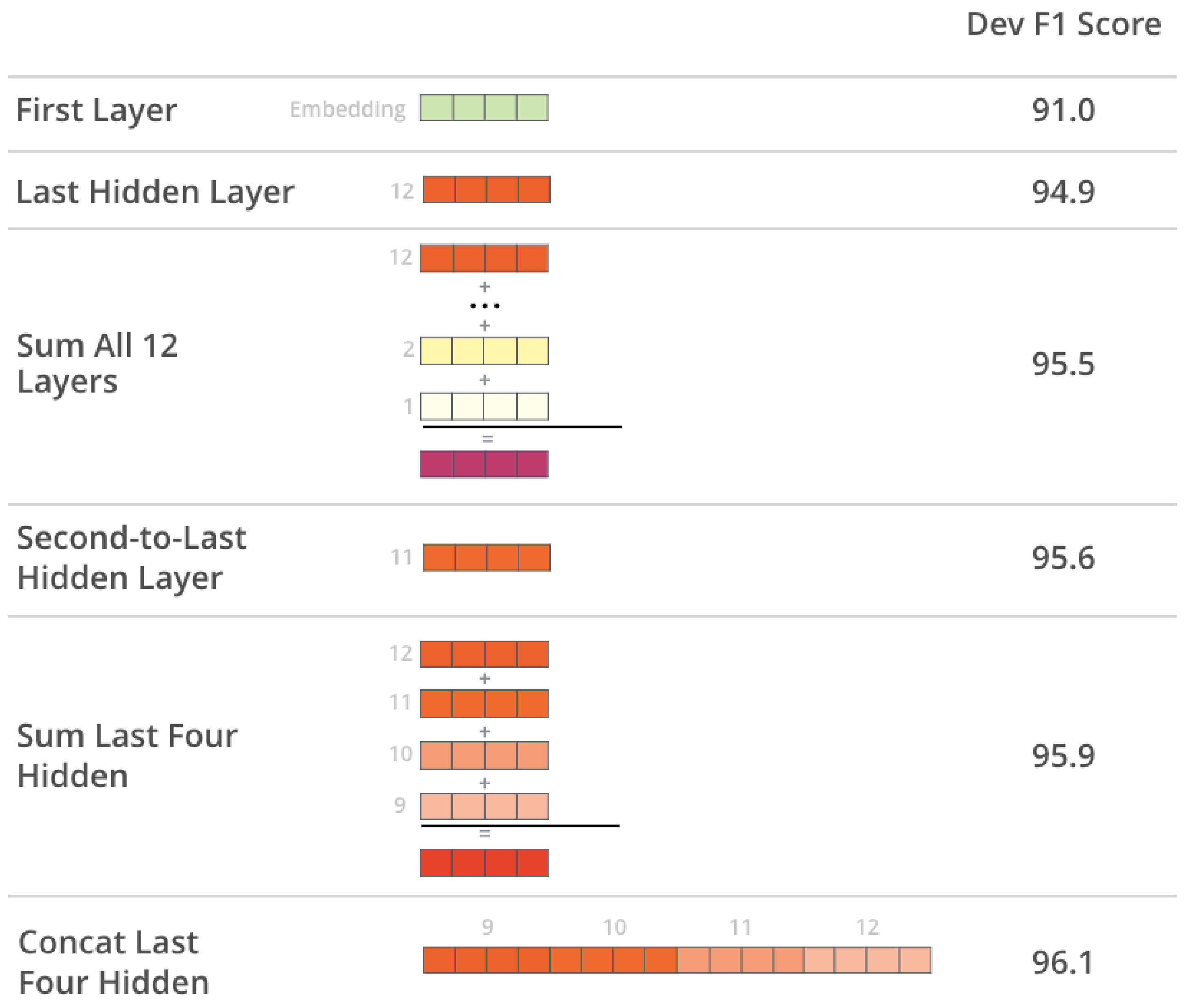
Which contextualized embedding do we take for ‘Help’ in this context?

- 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:





		Dev F1 Score
First Layer	Embedding	91.0
Last Hidden Layer	12	94.9
Sum All 12 Layers	12 + … + 2 + 1 =	95.5
Second-to-Last Hidden Layer	11	95.6
Sum Last Four Hidden	12 + 11 + 10 + 9 =	95.9
Concat Last Four Hidden	9 10 11 12	96.1

2.3. flair

- Another type of contextual string embeddings, **trained without an explicit notion of what a word is**
  - Models words as sequences of characters
  - The same word has different embeddings depending on its context
- Inputs to flair: characters
- Character-level LM is independent of tokenization and has a fixed vocabulary
- Deal with rare and out of vocabulary words better

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

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Contextualized word embeddings

- ELMo
- BERT
- Flair

Part 3: Experiments

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

Part 4: Conclusion

3. Experiments

- Experiments 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

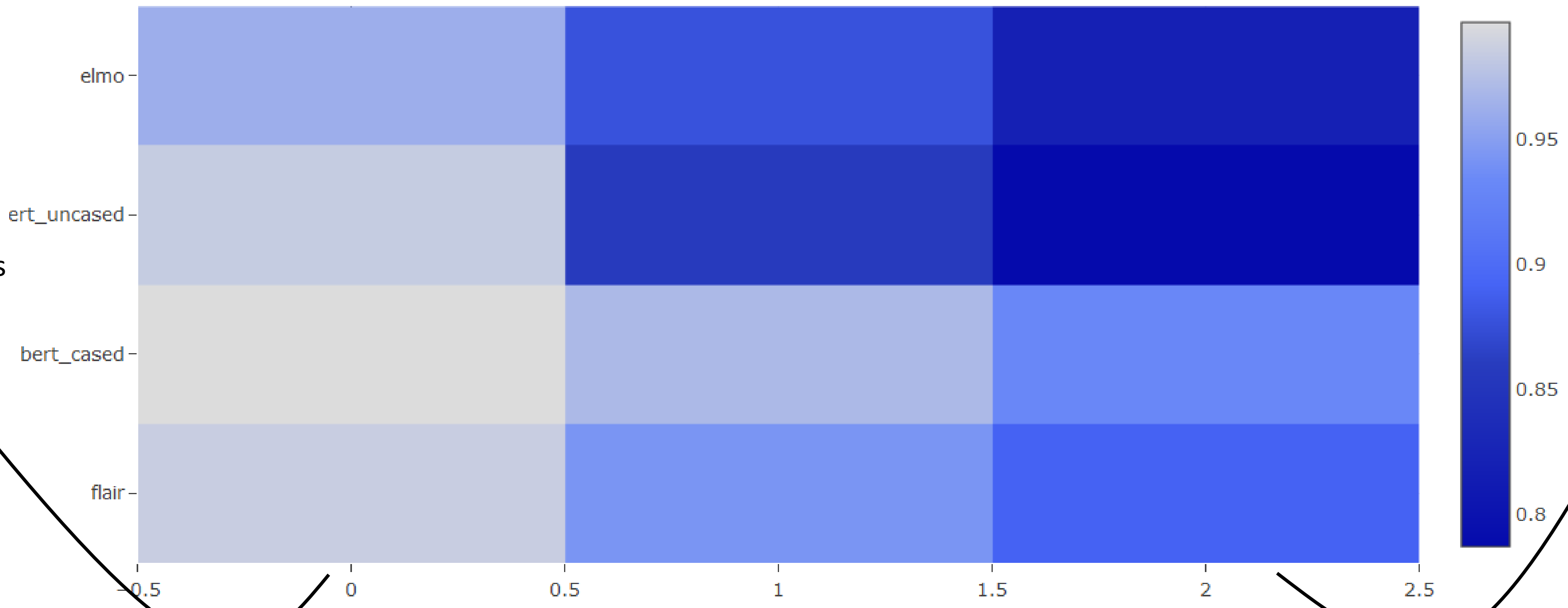
	Input text type	Vocabulary size
ELMo	word	700k words
BERT	subword	30k tokens
BERT multilingual	subword	120k tokens in total for 104 languages
flair	character	~256 characters (depending on the language)

3.1. Does word order matter?

the doctor invited 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

3.1. Does word order matter?

the doctor invited 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

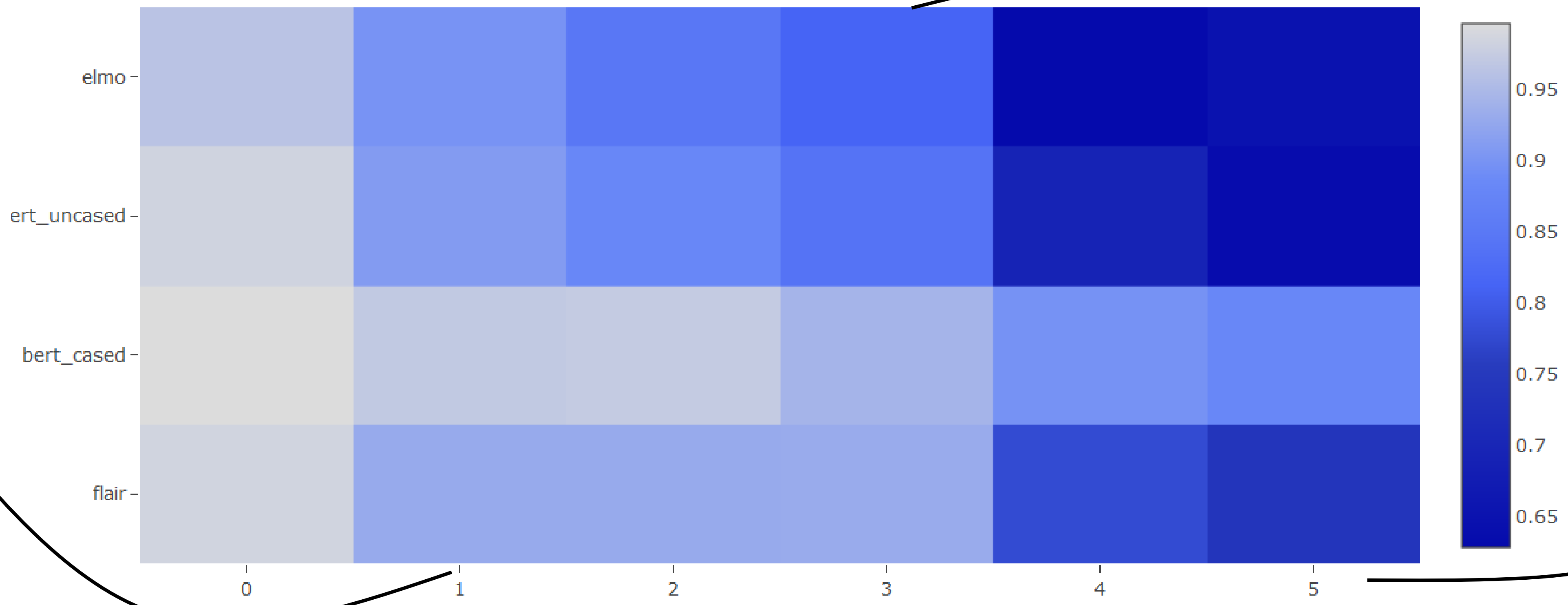


'opposite' sentence has very high similarity

random ordering has lower similarity but still high

3.2. What is the impact of lexical similarity?

the doctor invited the patient for lunch	Change		Change
0. the patient invited the doctor for lunch	'opposite' sentence	*3. the doctor killed the patient after lunch*	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. I wish I got invited for lunch*	Invite, lunch overlap



negation sentence has very high similarity

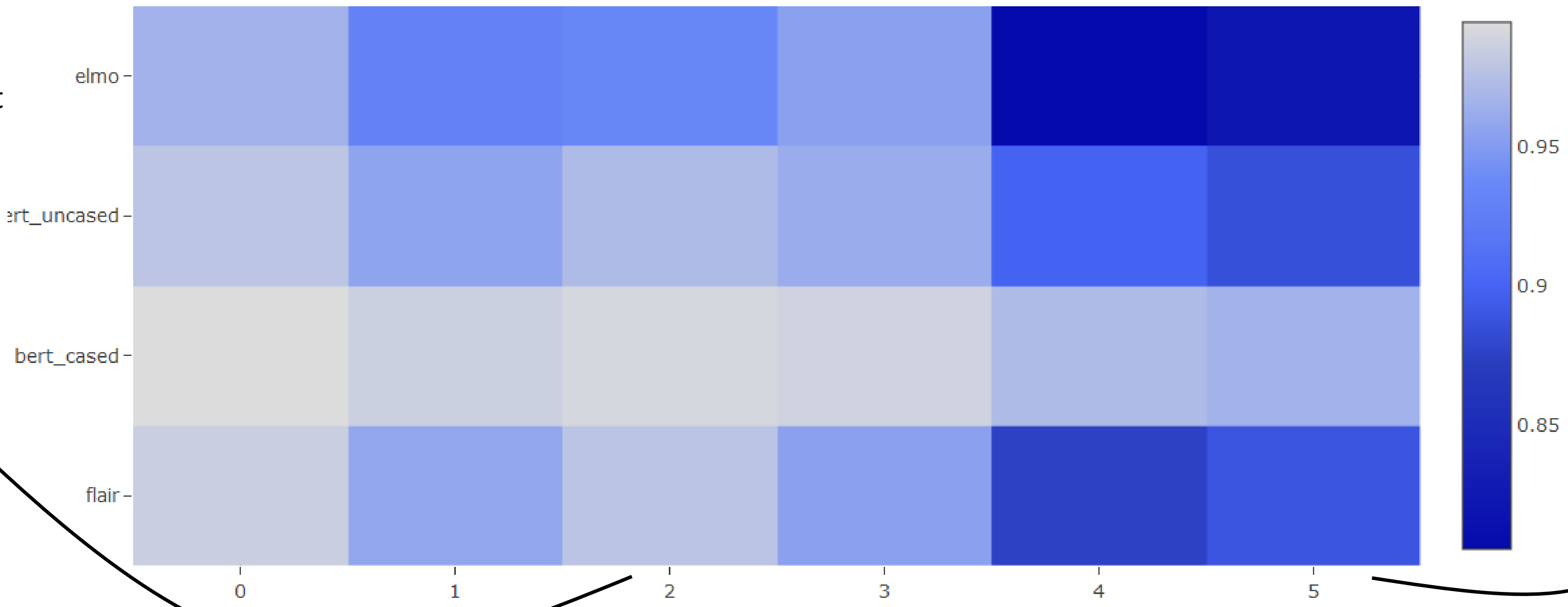
despite overlap, meaning of sentence is greatly changed

3/7 lexical overlap but much lower similarity

3.3. What is the impact of synonyms?

the doctor invited 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?



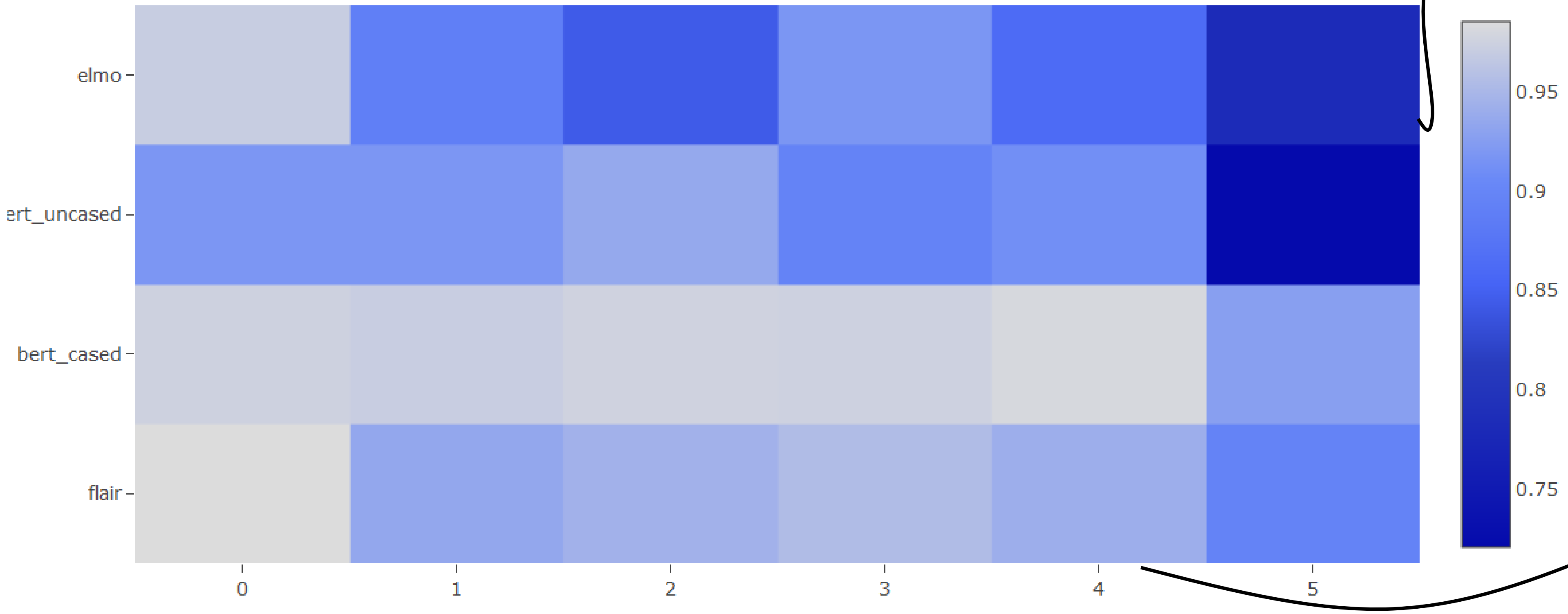
2 synonyms at the same is the possible cause

The implicit action sentence doesn't have bad results

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

the doctor invited the patient for lunch	Change		Change
0. the doctor invitted the patient for lunch	Typo	*3. the doctor invited the patient for sushi*	Specific food
1. the doctor kartoffeled the patient for lunch	Wrong word	*4. the doctor invited the patent for lunch*	Typo / different word
2. Stefan invited the patient for lunch	Proper name	*5.* the doctor invited the patent for linch	Typo / different word

slightly less similarity than the first one, but still high!



seems more logical

shouldn't this be lower?

3. Experiment takeaways

- BERT overall shows extremely high similarities
  - Mystery: especially BERT cased: lowest achieved was 85% with random sentences
- Word order is not very relevant → bag of word-like approach?
  - In Basque, random ordering has ~75% similarity in ELMo and flair and 95% in BERT
- Sentences in the same semantic space are evaluated as similar, even negation or opposite sentences
- A lexical overlap of 50% can raise similarity to 90%
  - Mystery: ELMo (word-based) shows much lower similarities than flair (character-based)
  - In Basque as well, while BERT shows similarities of 90%+ in all cases
- With synonyms that change context, similarities are still high (‘professor invited patient’)
- Typos and oov words don’t have much of an impact

Plan

Part 1: Semantic search

- Ontology matching

Part 2: Contextualized word embeddings

- ELMo
- BERT
- Flair

Part 3: Experiments

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

Part 4: Conclusion

4. Conclusion

- Semantic search → ontology matching → experiments on similarity ranking with contextual embeddings
- In my specific examples:
  - word order doesn’t seem to be relevant
  - very different sentences are evaluated as very similar
  - slightly worse results for Basque: because multilingual, low-resource language?
  - typos don’t have much impact
- Future work
  - research extreme high similarities of *bert_cased*
  - research cause of similarity differences in *bert_cased* and *bert_uncased*
  - research impact of word ordering and lexical overlap
  - research difference between word-based embeddings vs character-based
  - try more phenomena/examples
  - Visualization: T-SNE

Semantic search and similarity ranking

Ane Berasategi

18. July 2019

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN