University of Glasgow

AI4BioMed

Accelerating Cross-Encoders in Biomedical Entity Linking

Javier Sanz-Cruzado & Jake Lever

WORLD CHANGING GLASGOW

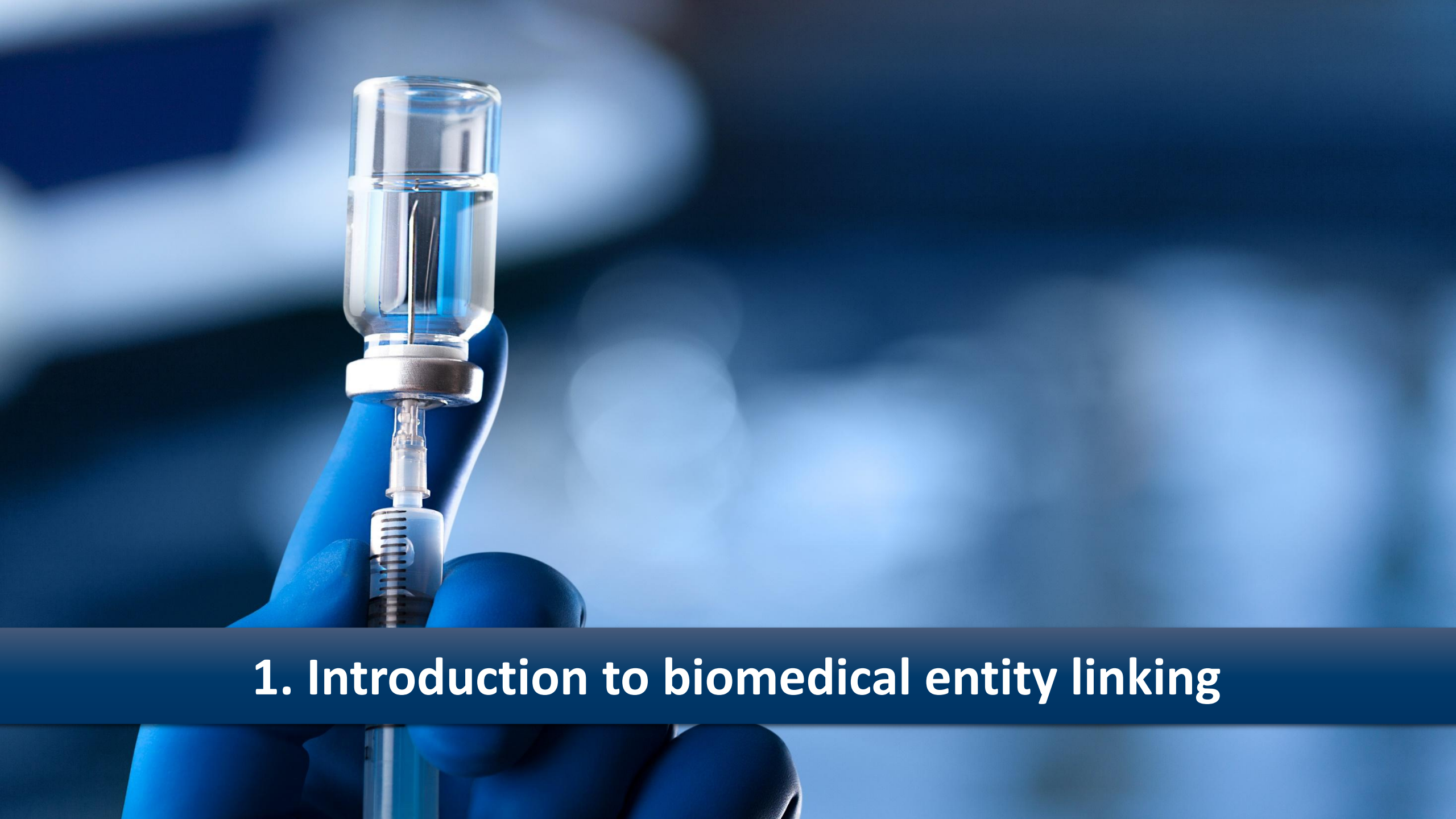
A WORLD TOP 100 UNIVERSITY

AI4Biomed Group Seminars, 2nd April 2025

University of Glasgow

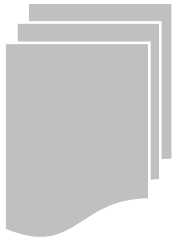
An outline of today's talk

1. Introduction to biomedical entity linking
2. Strategies for entity linking
3. Measuring the speed of cross-encoders
4. Accelerating cross-encoders
5. Experiments & evaluation
6. Conclusions



1. Introduction to biomedical entity linking

Extracting information from biomedical texts



Biomedical documents

Varicella is a highly contagious viral infection that causes an acute fever and blistered rash, mainly in children. Immunocompromised patients infected with the virus need intravenous treatment with the antiviral aciclovir.

Entities

- Varicella
- Aciclovir
- Fever

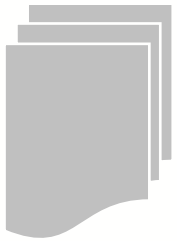
Relations

- Varicella is an infection
- Varicella causes fever
- Aciclovir treats varicella

More complex information

- Aciclovir treats varicella via intravenous treatment

Extracting entities from biomedical texts



Biomedical documents

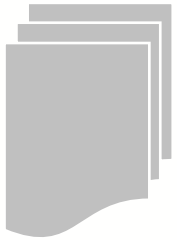
"Varicella" is a highly contagious **"viral infection"** that causes an acute **"fever"** and **"blistered rash"**, mainly in children. **"Immunocompromised patients"** infected with the **"virus"** need **"intravenous treatment"** with the **"antiviral"** **"aciclovir".**

- Identify mentions of biomedical concepts (entities) in text.
- Named entity recognition techniques can be applied.

Not the focus of this talk

But, what is the biomedical concept these mentions refer to?

What is biomedical entity linking?



Biomedical documents

"Varicella" is a highly contagious **"viral infection"** that causes an acute **"fever"** and **"blistered rash"**, mainly in children. **"Immunocompromised patients"** infected with the **"virus"** need **"intravenous treatment"** with the **"antiviral"** **"aciclovir".**

"Varicella"

?

?

?

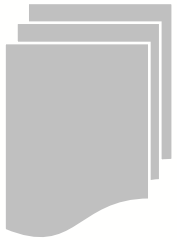
Varicella

Chickenpox

Varicella-Zoster Virus

Knowledge base

What is biomedical entity linking?



Biomedical documents

“Varicella” is a highly contagious **“viral infection”** that causes an acute **“fever”** and **“blistered rash”**, mainly in children. **“Immunocompromised patients”** infected with the **“virus”** need **“intravenous treatment”** with the **“antiviral”** **“aciclovir”.**

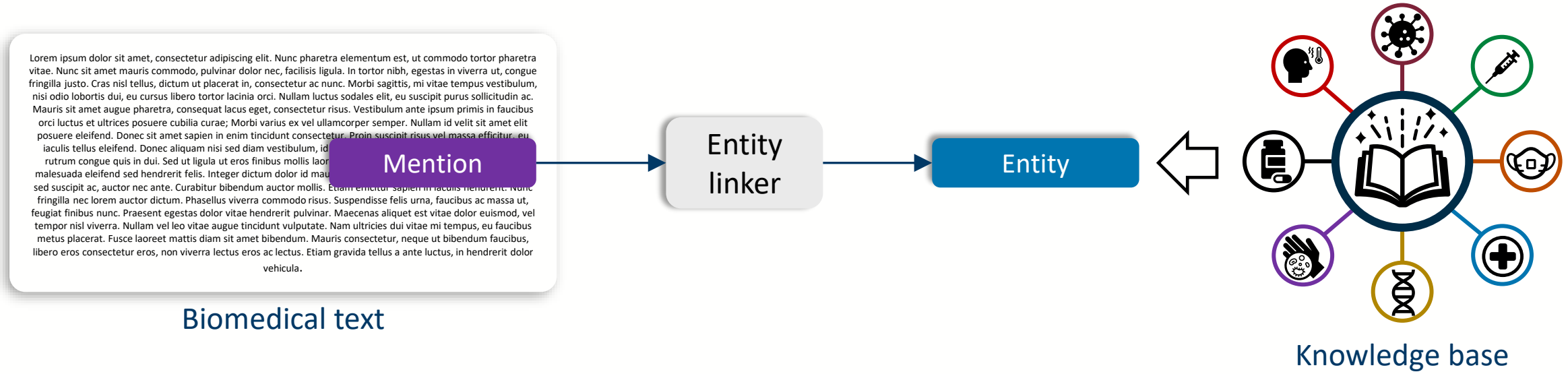
“Varicella” →

- Varicella
- Chickenpox
- Varicella-Zoster Virus

Knowledge base

What is biomedical entity linking?

Biomedical entity linking matches mentions of biomedical concepts (diseases, chemicals) in text with unique entities within a knowledge base



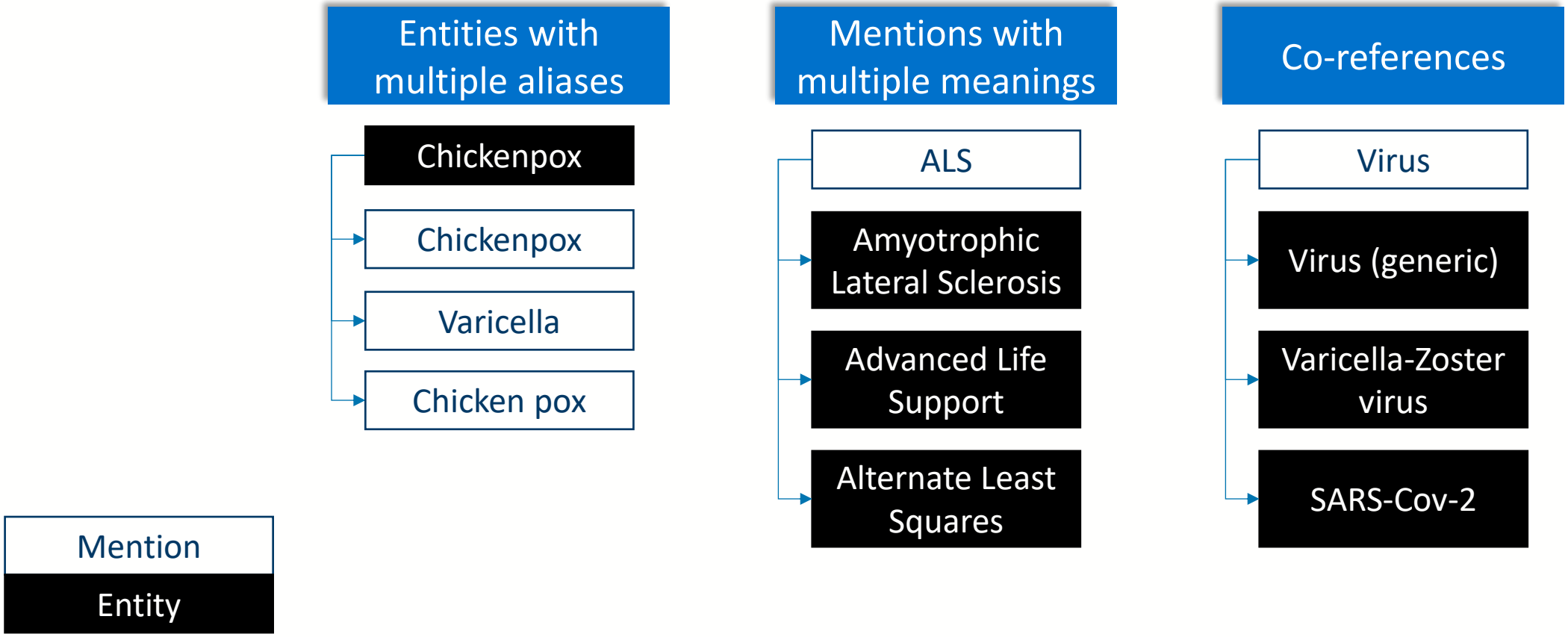
University of Glasgow

Biomedical entities are not just names

Id	C008049	**Semantic types**	Disease or Syndrome
Name	Chickenpox	**Broader concepts**	Herpesviridae infections Virus diseases Infection
Aliases	Chicken pox Chickenpox, NOS Chicken pox infection Varicella Varicella infection Varicella, NOS		
Definition (MeSH)	A highly contagious infectious disease caused by the varicella-zoster virus (HERPESVIRUS 3, HUMAN). It usually affects children, is spread by direct contact or respiratory route via droplet nuclei, and is characterized by the appearance on the skin and mucous membranes of successive crops of typical pruritic vesicular lesions that are easily broken and become scabbed. Chickenpox is relatively benign in children, but may be complicated by pneumonia and encephalitis in adults. (From Dorland, 27th ed)		

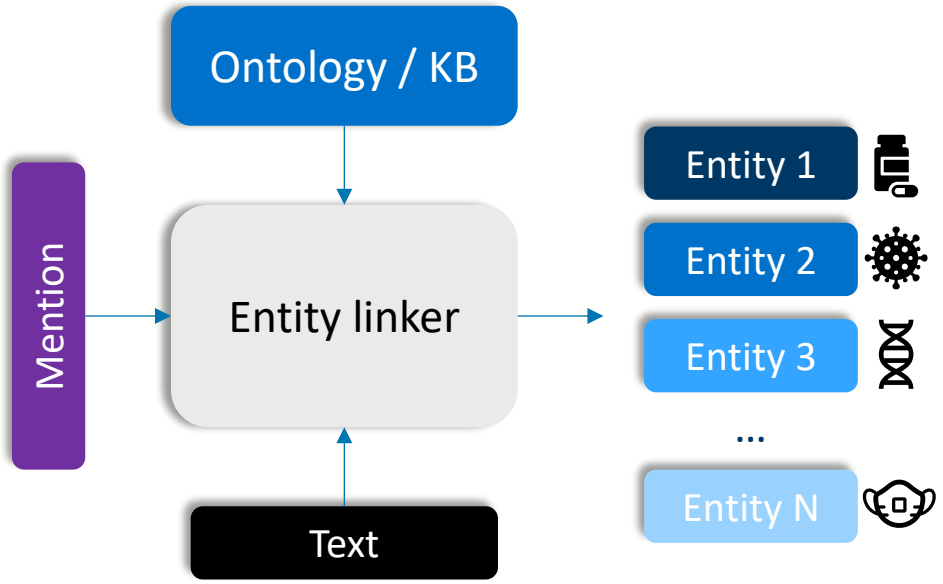
What are the challenges of biomedical entity linking?

"Varicella", differently from **"ALS"**, is a highly contagious **"viral infection".** People infected with the **"virus"** need…



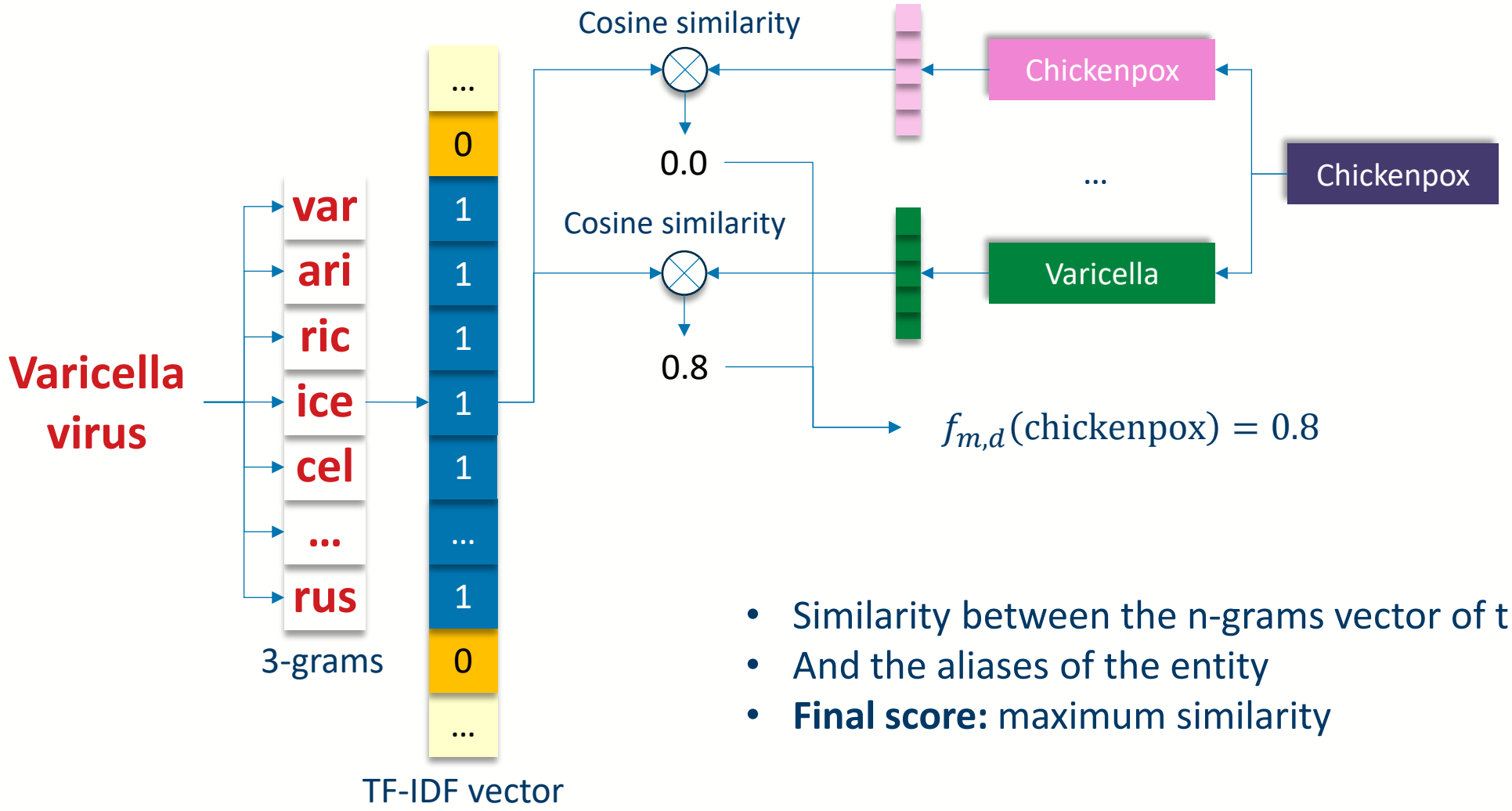
2. Entity linking strategies

One-stage entity linking



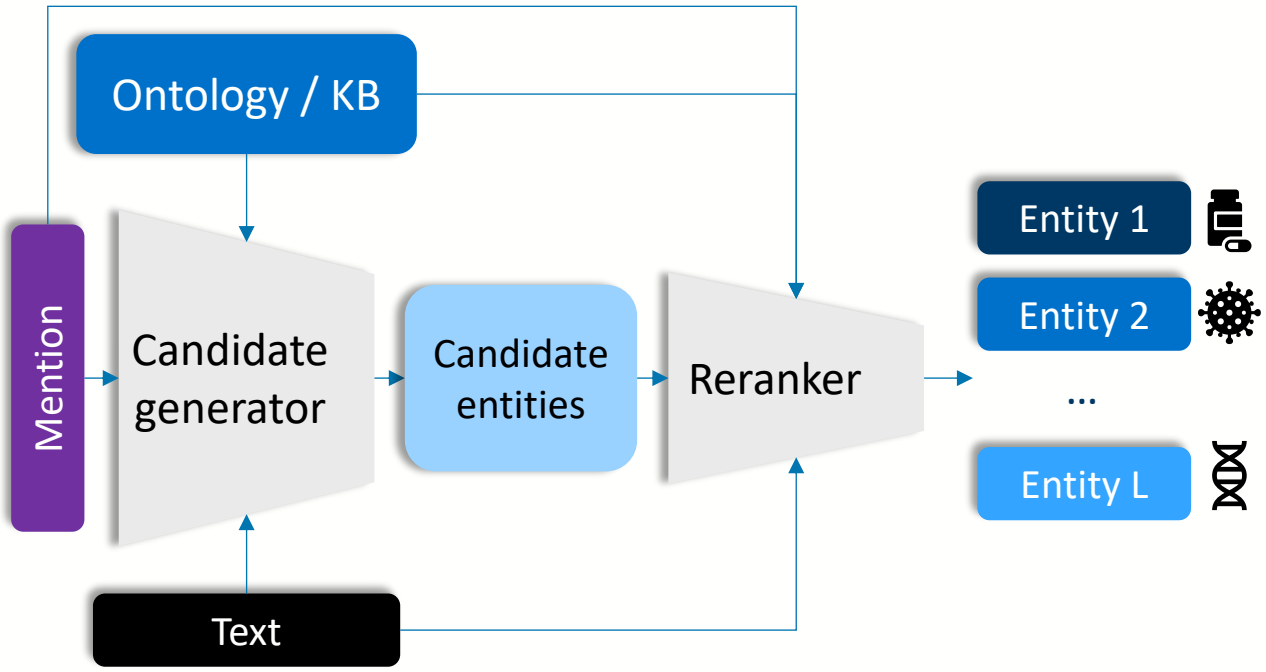
- Given a mention, rank all entities in the knowledge base
- Computationally efficient (need to rank thousands / millions of entities)
- Commonly rely on similarity between mentions and entities
- Example methods:
  - SapBERT bi-encoder (Liu et al. 2021)
  - Character n-grams (Angell et al. 2021)

The n-grams model



- Similarity between the n-grams vector of the mention
- And the aliases of the entity
- **Final score:** maximum similarity

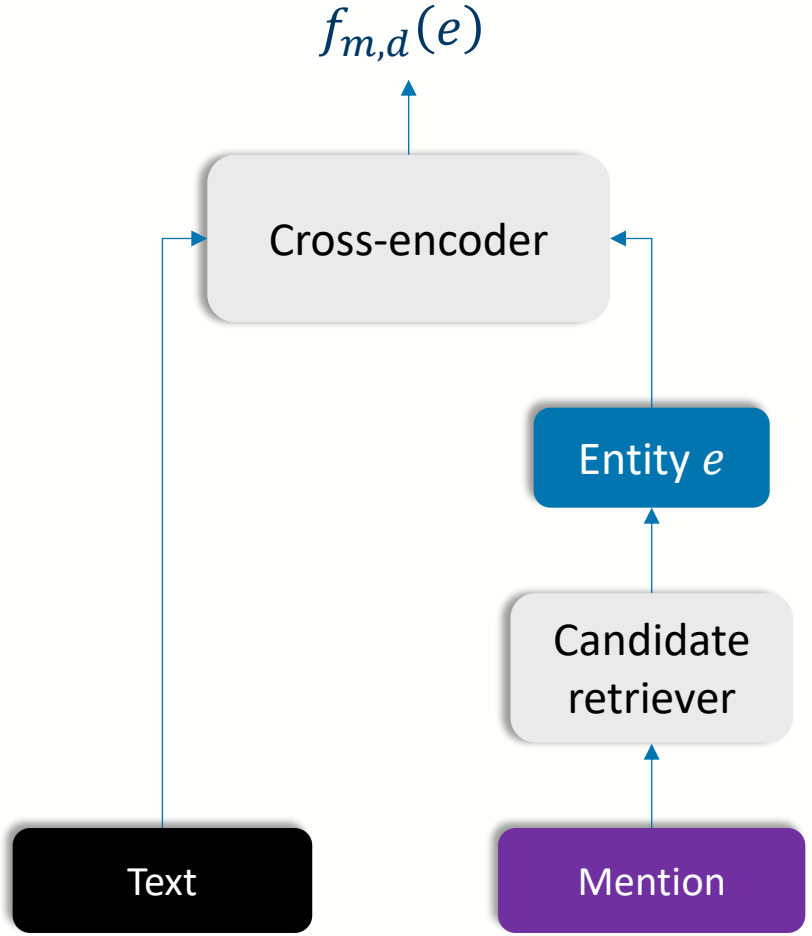
Two-stage entity linking



- Two stages
- Candidate generator
  - Inspects all entities
  - Get the top k more relevant entities $k \ll |\mathcal{E}|$
  - Computationally efficient
  - Maximize recall@k
  - Ex: single-stage entity linkers
- Reranker
  - Inspects the top candidates from first phase
  - Precise ranker
  - Maximize accuracy
  - Computationally expensive
  - Ex: cross-encoder

Cross-encoder reranker

- Transformer-based model (encoder-only model)
  - BERT
  - BiomedBERT
  - Longformer
  - ModernBERT
- Inputs:
  - Text containing a mention
  - A candidate entity
- Output:
  - $f_{m,d}(e)$: the score for the entity
- Rank entities by descending score



How does the cross-encoder work?

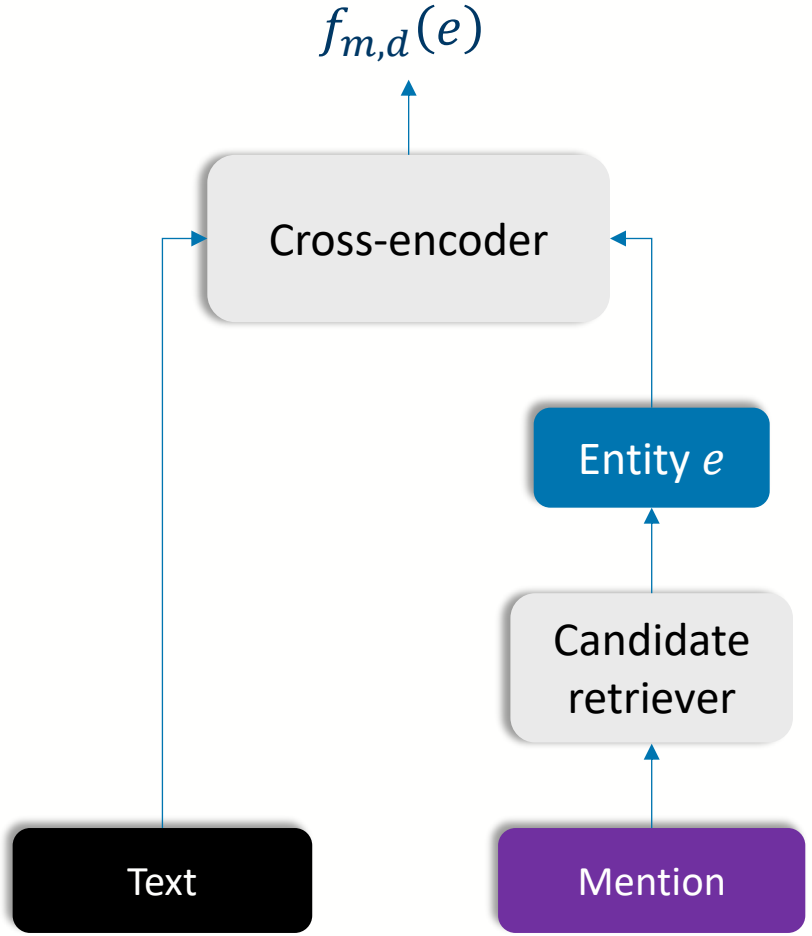
- It receives a sentence following a template

  Text [SEP] Mention [MASK] Entity name

- The mention is contained in the text
- The text here provides additional context

- The [MASK] token can take two values:
  - 1 – if the entity corresponds to the mention
  - 0 – otherwise

- Therefore, the score of the entity is the probability of the [MASK] token taking value 1



How does the cross-encoder work?

- It receives a sentence following a template

Text [SEP] Mention [MASK] Entity name

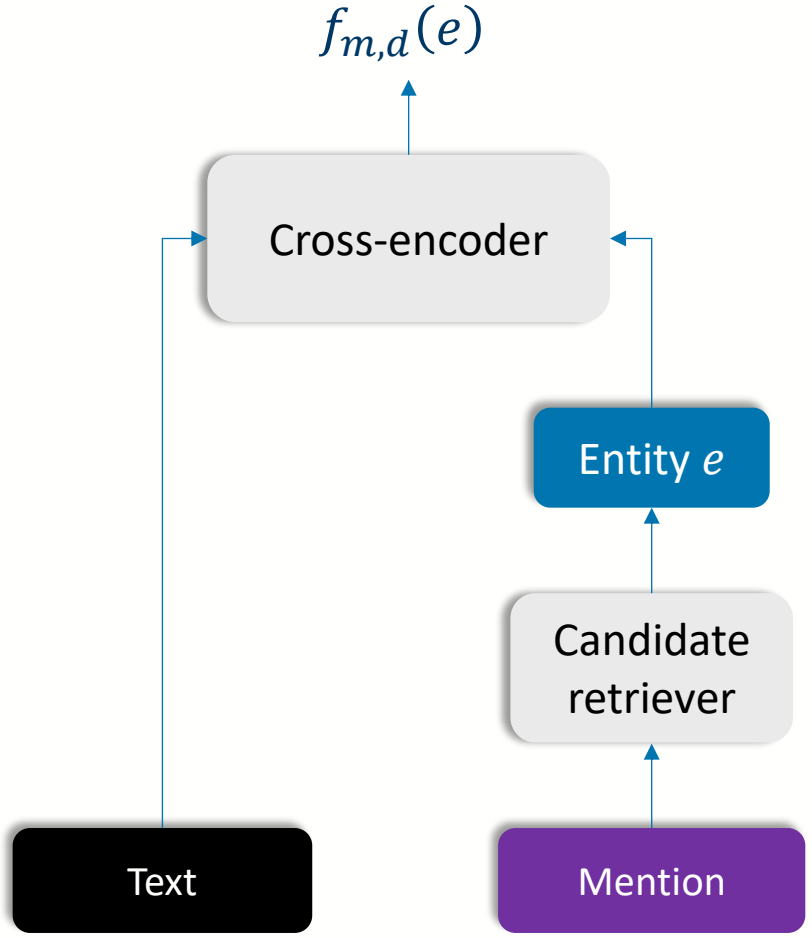
- An example

Carb is used to treat epileptic attacks [SEP] carb [MASK] carbohydrates

The entity is **wrong**, therefore **[MASK] == 0**

Carb is used to treat epileptic attacks [SEP] carb [MASK] carbamazepine

The entity is **right**, therefore **[MASK] == 1**



3. Measuring the speed of cross-encoders

University of Glasgow

Let's run an experiment

Research question

How much time do I need to train and run a cross-encoder?

University of Glasgow

Dataset: MedMentions (full)

- PubMed abstracts annotated with mentions of entities
- **Knowledge base:** UMLS 2017AA release
- For each mention, we provide the whole sentence as input text

	Training	Validation	Test
Documents	2,635	878	879
Sentences	25,836	8,508	8,597
Mentions	211,029	71,062	70,405

University of Glasgow

Algorithms

- **First stage candidate retrieval:** 3-grams TF-IDF
  - 3-grams TF-IDF
  - Compute 5 candidates for each mention

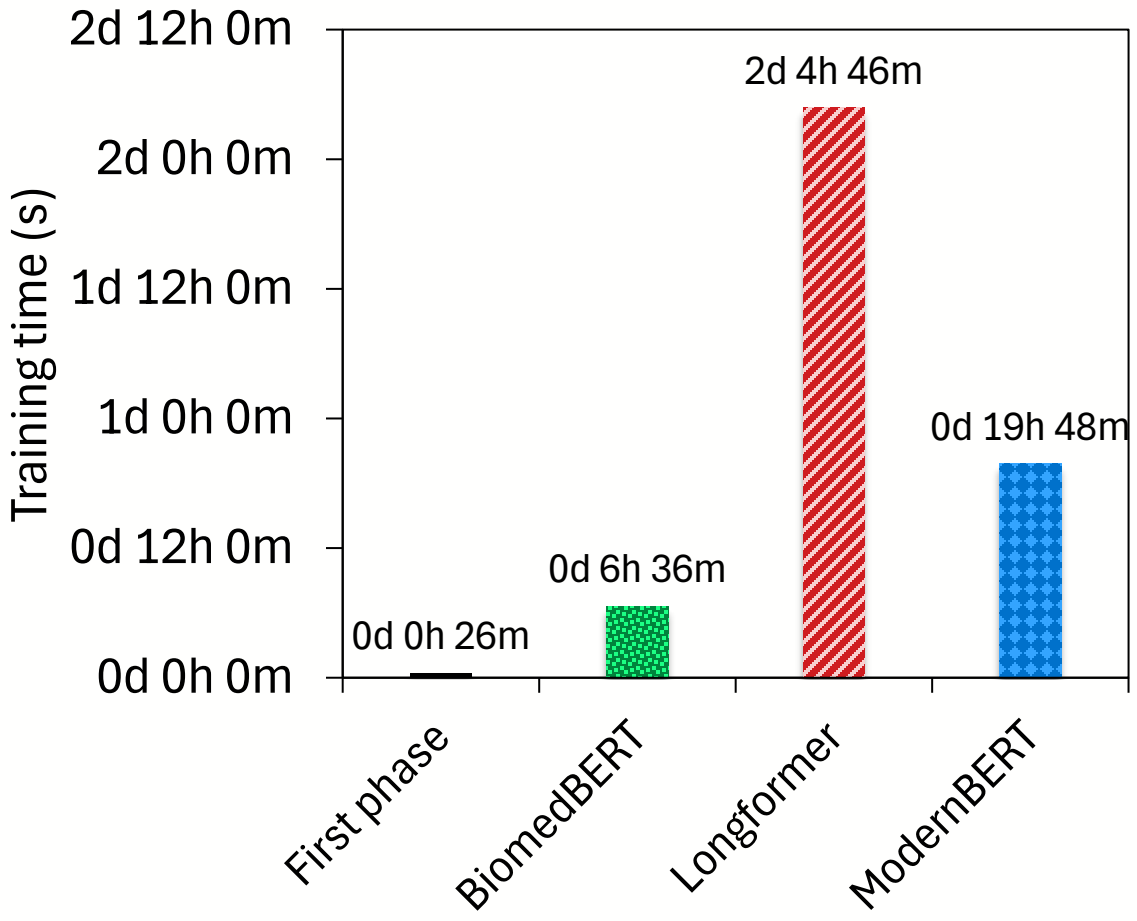
- **Second stage:** cross-encoder
  - Backbone transformer models
    1. BiomedBERT
    2. Longformer
    3. ModernBERT
  - Early stop condition: stop training if F1 on validation set does not improve after three epochs.
  - Same learning rate, different batch sizes to accommodate them in the same hardware
  - Loss function: cross-entropy loss

University of Glasgow

Hardware

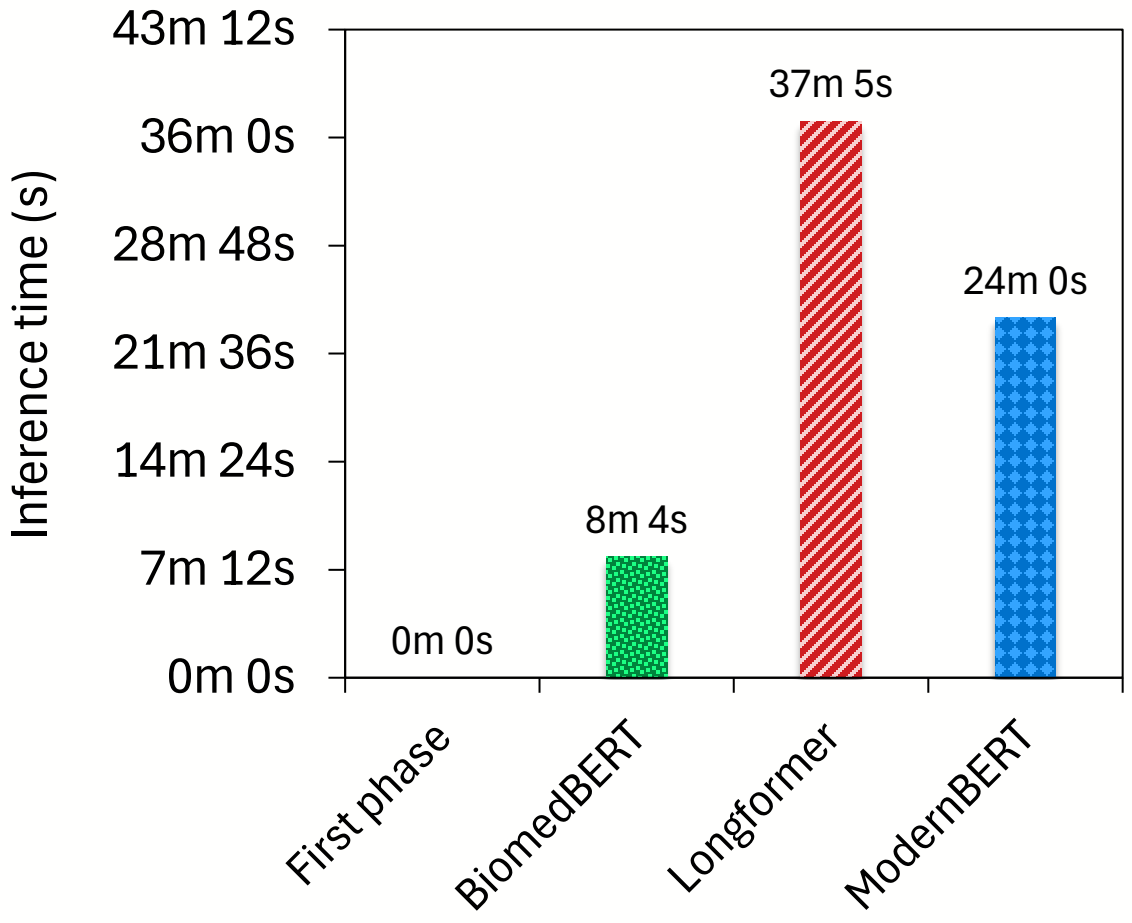
All algorithms were trained under the same hardware

Training time



- The fastest model, BiomedBERT, takes around 6 hours and 36 minutes to train the model!
- The slowest model, Longformer, takes around 8 times more: 2 days 4 hours and 46 minutes!
- Training a cross-encoder takes considerable time.
- **Note:** the comparison is not fully fair (different number of epochs and batch sizes across models). But it shows how expensive training this models is.

Inference time



- Inference is faster
- It takes between 8 to 10 minutes to compute the scores for BiomedBERT
- … but more than 30 minutes to compute the scores for Longformer

University of Glasgow

Analysis

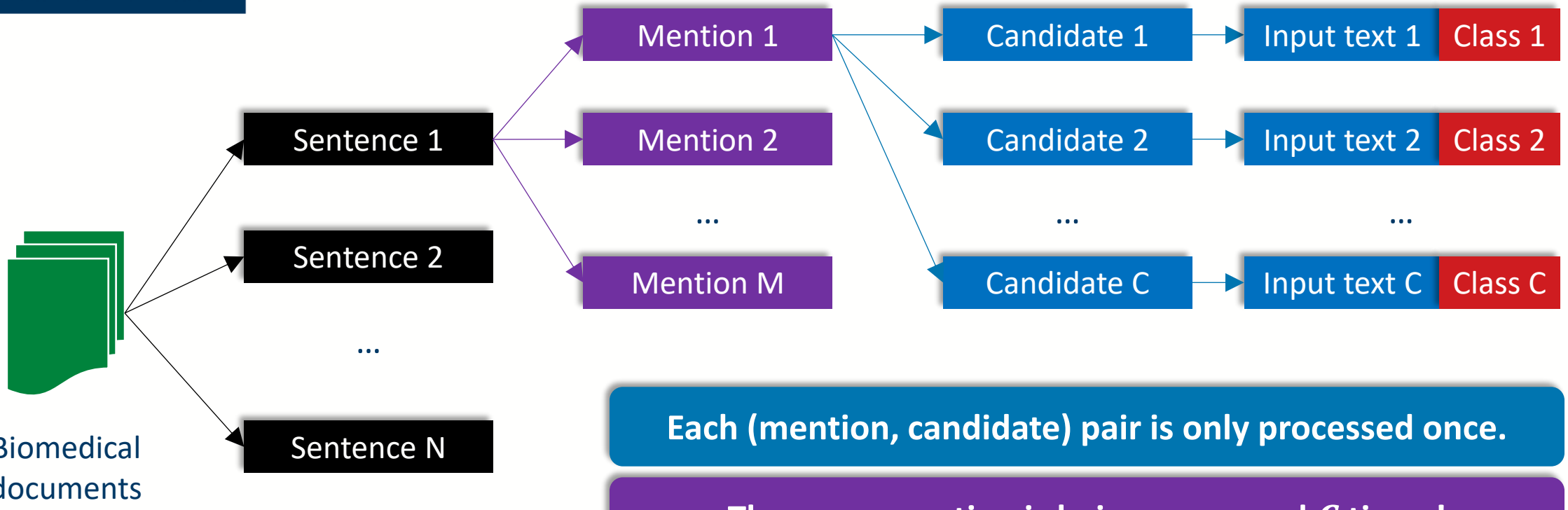
- Training a cross-encoder requires substantial resources
  - On our experiments, between 6 hours and > 2 days
  - That implies locking your resources for the same amount of time

- Inference time is faster
  - But not negligible
  - Can take more to half an hour for around 70k mentions

Can we build cross-encoders capable of improving both training and inference time while not harming the accuracy of the base cross-encoder?

4. Accelerating cross-encoders

Let's review the cross-encoder working



Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

University of Glasgow

Accelerating cross-encoders

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

Both in training and inference

Idea: Can we accelerate training / inference by showing each text less times to the cross-encoder?

University of Glasgow

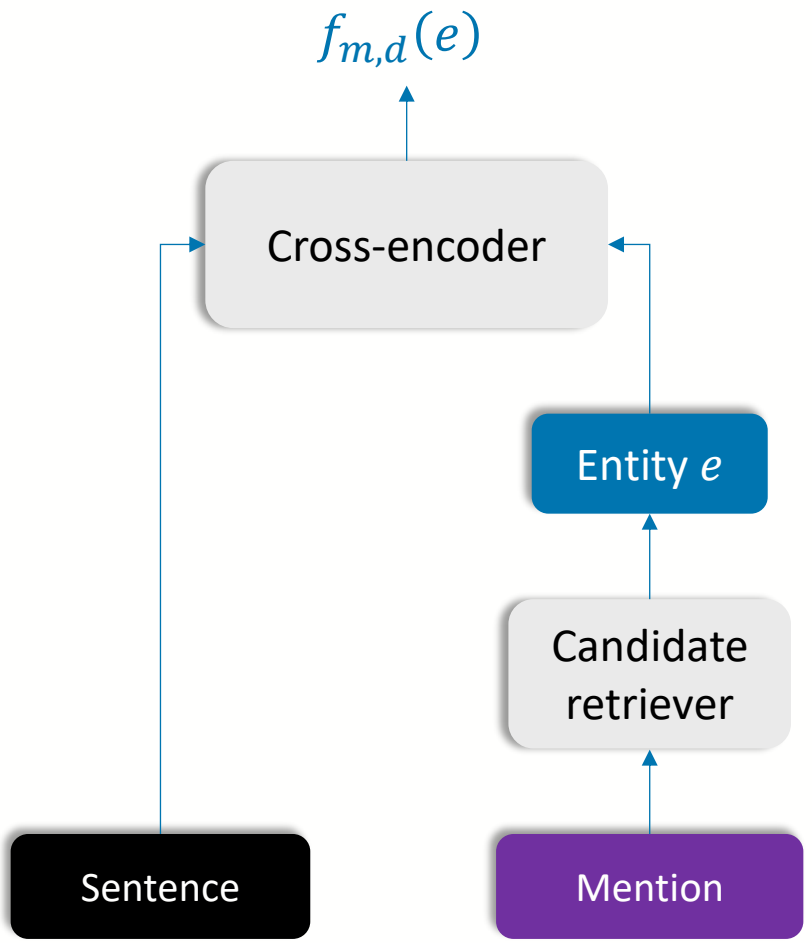
Solution 1: Parallel cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!



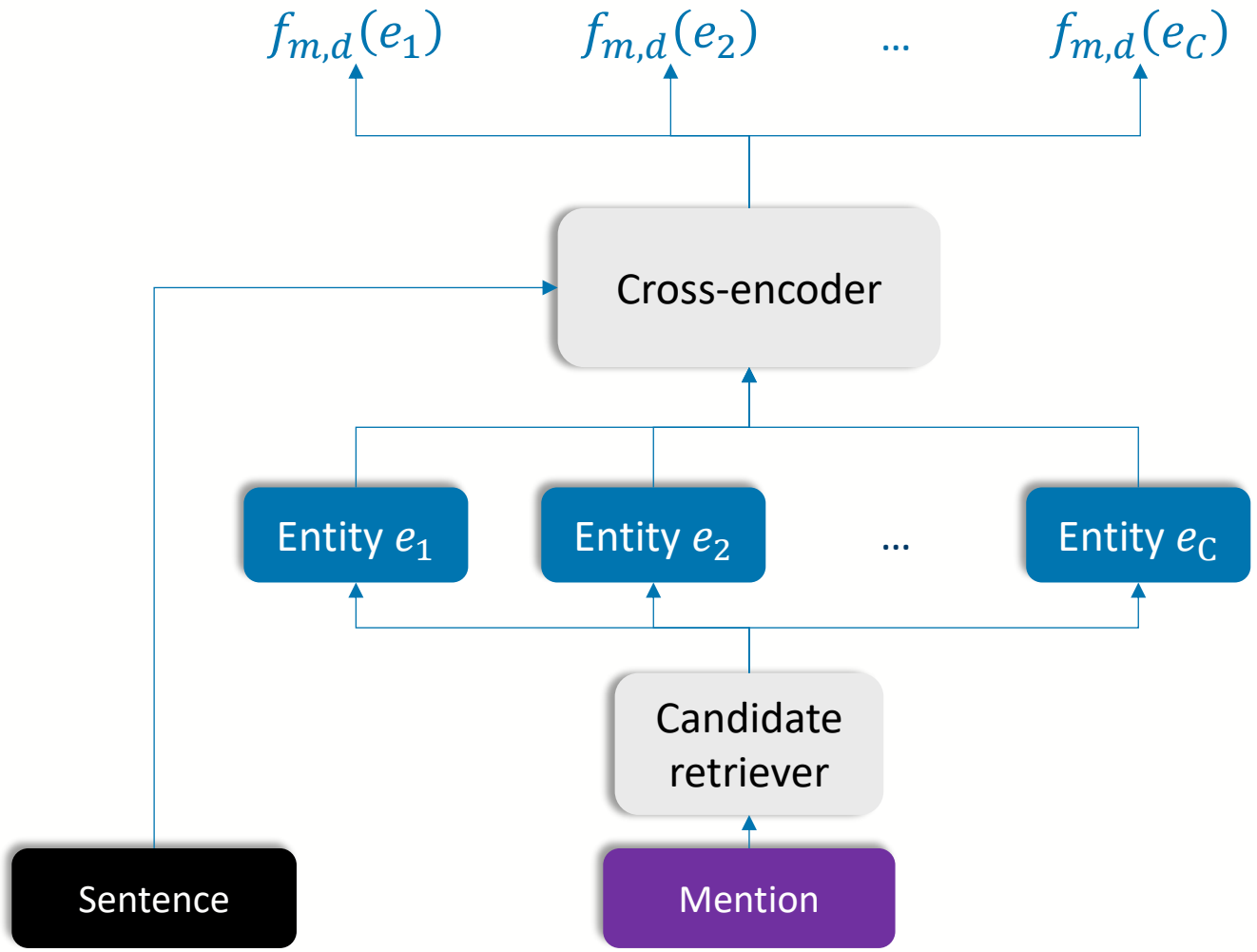
Solution 1: Parallel cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!



University of Glasgow

Solution 1: Parallel cross-encoder

- While the cross encoder uses template for each candidate:

Text [SEP] Mention [MASK] Entity e name

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

University of Glasgow

Solution 1: Parallel cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

- The parallel cross-encoder receives a sentence using the following a template for each mention:

Text [SEP] Mention [MASK] Entity e_1 name
[SEP] Mention [MASK] Entity e_2 name
…
[SEP] Mention [MASK] Entity e_C name

- Therefore, the score of the entity e_i is the probability of its [MASK] token taking value 1

University of Glasgow

Solution 1: Parallel cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is being processed C times!

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

- Example:

  Carb is used to treat epileptic attacks
  [SEP] carb [MASK] carbohydrates
  [SEP] carb [MASK] carbamazepine
  [SEP] carb [MASK] carbamazole

- Here,
  - The first [MASK] token takes value 0
  - The second [MASK] token takes value 1
  - The third [MASK] token takes value 0

University of Glasgow

Solution 1: Parallel cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is being processed once.

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

- Example:

  Carb is used to treat epileptic attacks
  [SEP] carb [MASK] carbohydrates
  [SEP] carb [MASK] carbamazepine
  [SEP] carb [MASK] carbamazole

- Here,
  - The first [MASK] token takes value 0
  - The second [MASK] token takes value 1
  - The third [MASK] token takes value 0

But, every sentence can have more than one entity!

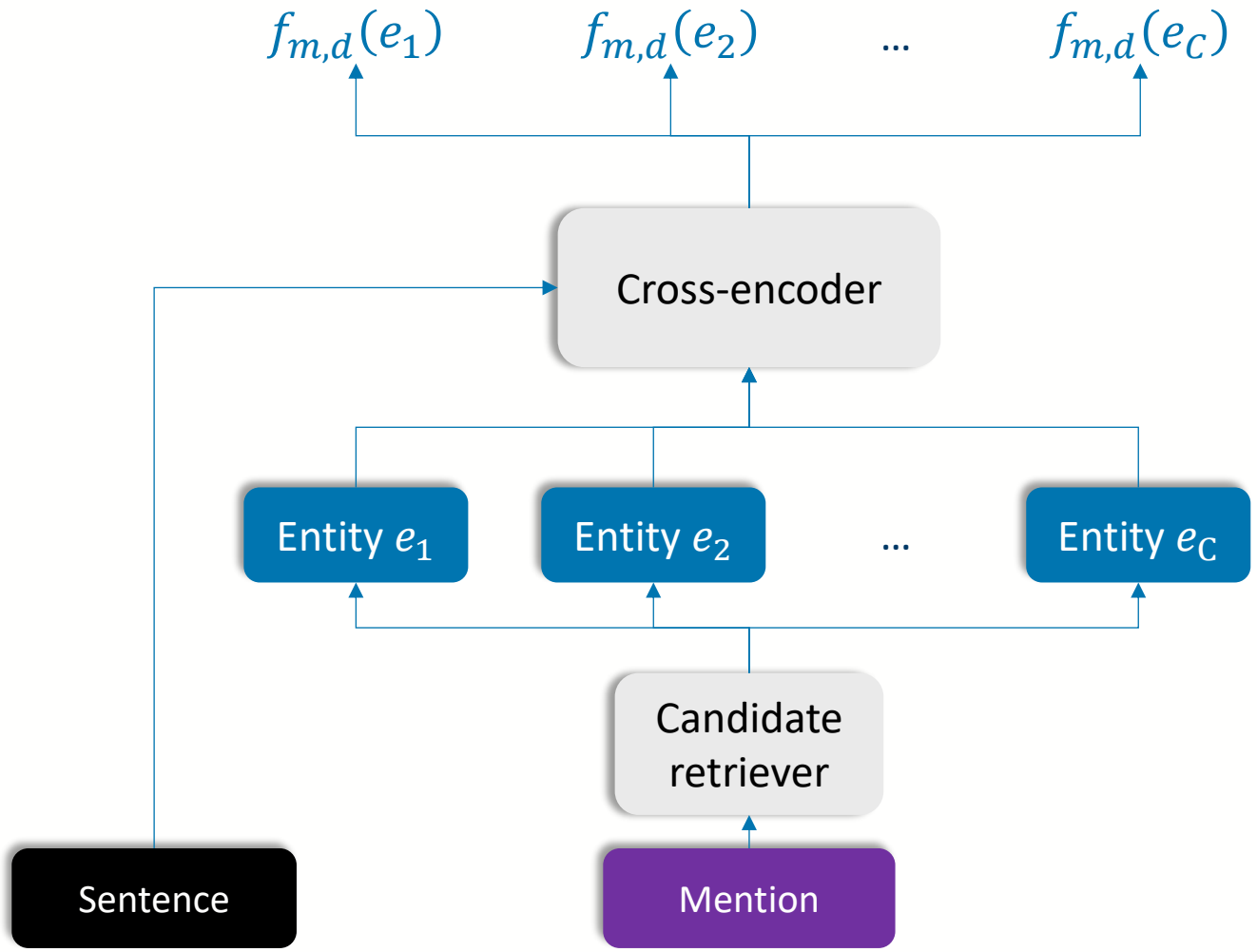
Solution 2: Multi cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!



University of Glasgow

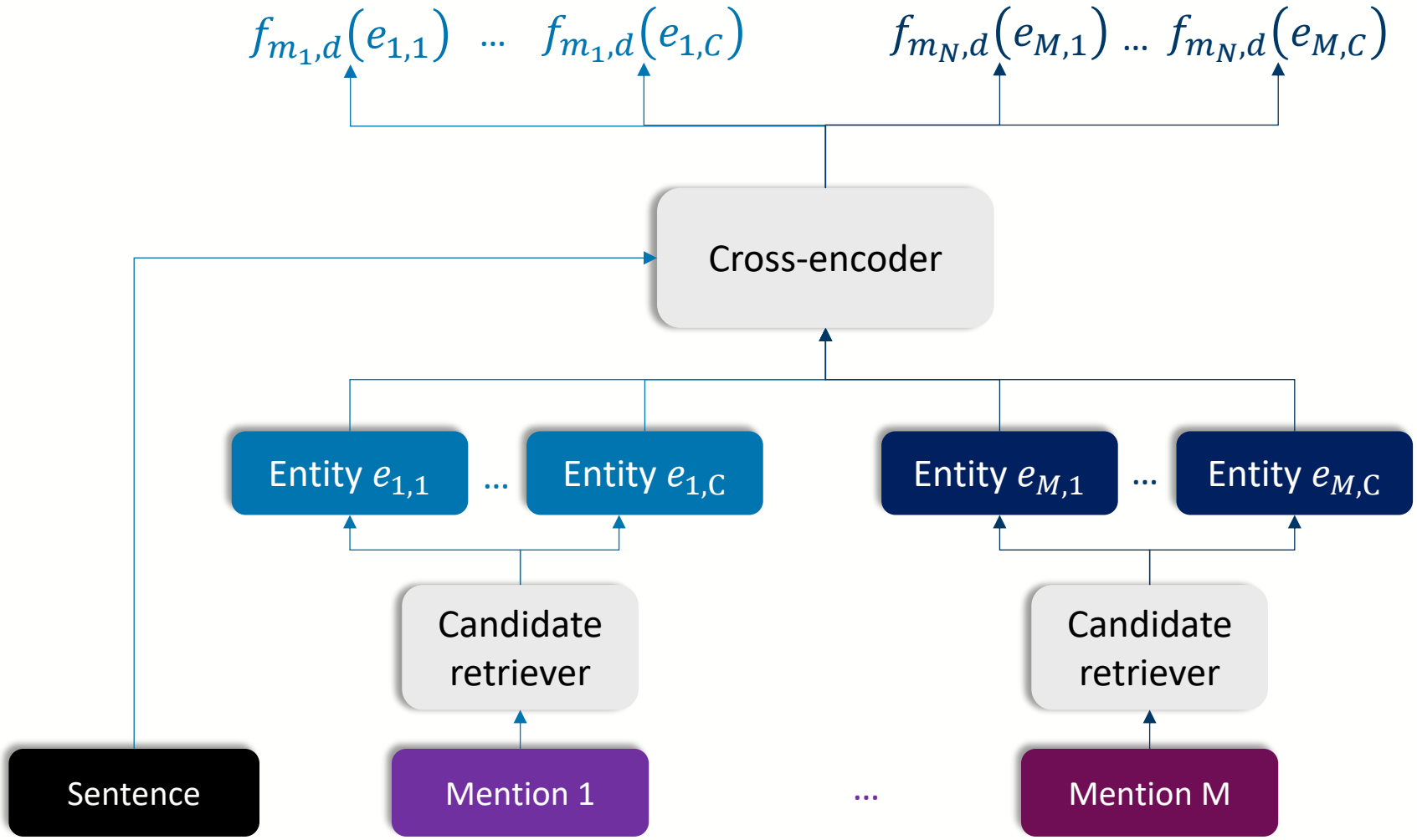
Solution 2: Multi cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!



University of Glasgow

Solution 2: Multi cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

- Use a similar trick to the parallel cross-encoder
- The new template is:

Text [SEP] Mention 1 [MASK] Entity $e_{1,1}$ name
[SEP] Mention 1 [MASK] Entity $e_{1,2}$ name
…
[SEP] Mention 1 [MASK] Entity $e_{1,C}$ name
…
[SEP] Mention M [MASK] Entity $e_{M,1}$ name
[SEP] Mention M [MASK] Entity $e_{M,2}$ name
…
[SEP] Mention M [MASK] Entity $e_{M,C}$ name

- And, again, the score for each entity and mention is the probability of the [MASK] token being one

University of Glasgow

Solution 2: Multi cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is being processed $C \times M$ times!

And the document has N different sentences!

- Example:

Carb is used to treat epileptic attacks
[SEP] carb [MASK] carbohydrates
[SEP] carb [MASK] carbamazepine
[SEP] carb [MASK] carbamazole
[SEP] epileptic attacks [MASK] epilepsy
[SEP] epileptic attacks [MASK] epigenetics
[SEP] epileptic attacks [MASK] epidural

- [MASK] tokens should be classified as [0,1,0,1,0,0]

Solution 2: Multi cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is only processed once

And the document has N different sentences!

- Example:

Carb is used to treat epileptic attacks
[SEP] carb [MASK] carbohydrates
[SEP] carb [MASK] carbamazepine
[SEP] carb [MASK] carbamazole
[SEP] epileptic attacks [MASK] epilepsy
[SEP] epileptic attacks [MASK] epigenetics
[SEP] epileptic attacks [MASK] epidural

- [MASK] tokens should be classified as [0,1,0,1,0,0]

Solution 3: Document cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is only processed once

And the document has N different sentences!

- The previous trick can be further applied
- Instead of processing one sentence, we can process multiple at the same time.
- How? Concatenating the templates for a sentence using a [SEP] token
- We call this document cross-encoder
- **Note:** if each document is divided in passages, we can have an intermediate cross-encoder. We denote this as passage cross-encoder

Solution 3: Document cross-encoder

Each (mention, candidate) pair is only processed once.

The same mention is only processed once

The same sentence text is only processed once

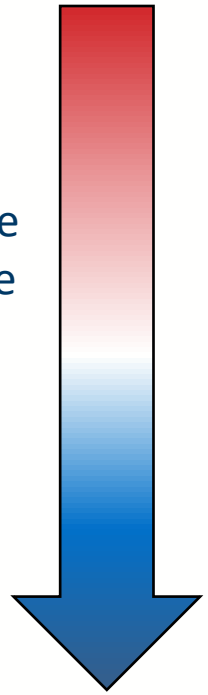
The document is only processed once

- The previous trick can be further applied
- Instead of processing one sentence, we can process multiple at the same time.
- How? Concatenating the templates for a sentence using a [SEP] token
- We call this document cross-encoder
- **Note:** if each document is divided in passages, we can have an intermediate cross-encoder. We denote this as passage cross-encoder

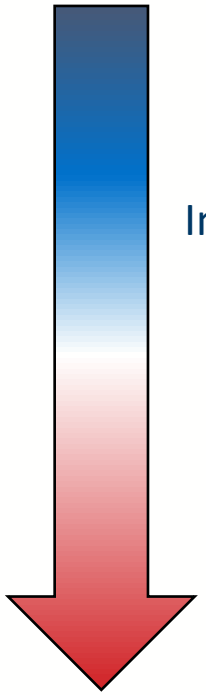
Expectations for our solutions

Task speed

Increases as we process more text within a single call to the cross-encoder



- Base cross-encoder
- Parallel cross-encoder
- Multi cross-encoder
- Document cross-encoder



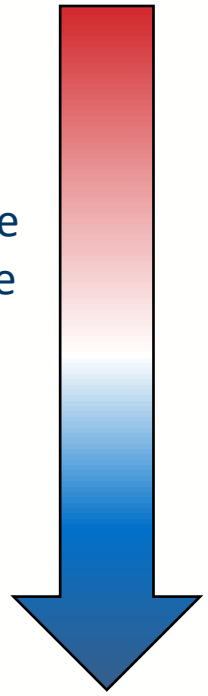
Task complexity

Increases as we need to classify more tokens within a single input

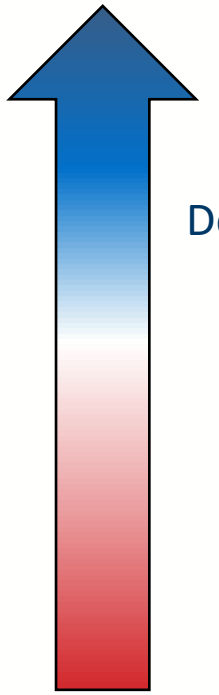
Expectations for our solutions

Task speed

Increases as we process more text within a single call to the cross-encoder



- Base cross-encoder
- Parallel cross-encoder
- Multi cross-encoder
- Document cross-encoder



Task accuracy

Decreases as we need to classify more tokens within a single input

5. Experiments and evaluation

University of Glasgow

Research questions

Research question 1

How does the parallelism of the cross-encoder affect the effectiveness of the model?

Research question 2

How does the parallelism of the cross-encoder affect the training and inference speeds?

University of Glasgow

Experimental setup

- We test our models on four biomedical datasets:
  - **MedMentions:** PubMed abstracts annotated with entities in UMLS 2017AA
  - **NCBI Disease:** PubMed abstract annotated with disease mentions of entities in the MEDIC ontology
  - **NLM Chem:** Full-text PubMed Central articles, with annotated mentions of chemical entities in MeSH 2021
  - **BC5CDR:** PubMed abstracts with chemical and disease annotations. Linked with MeSH 2015.

University of Glasgow

Algorithms

- **First stage candidate retrieval:** n-grams TF-IDF
  - 3-grams for MedMentions, 2-grams for the other datasets
  - Compute 5 candidates for each mention

- **Second stage:**
  - Baseline: base cross-encoder
  - Parallel cross-encoder
  - Multi cross-encoder
  - Passage cross-encoder (NLM Chem and BC5CDR only)
  - Document cross-encoder

University of Glasgow

Cross-encoder configurations

- **Backbone LMs:** We use models with different context-window size
  - BiomedBERT: 512
  - Longformer: 4096
  - ModernBERT: 8192

- **Early stopping:** if F1 is not improved on the validation set after three epochs
- **Learning rate:** all cross-encoders use the same one (1e-6)
- **Batch size:** depends on backbone model (fit on a single 4090)
- **Loss function:** cross-entropy loss

- **Hardware:** 2 CPU, 16 GB RAM, 1 4090 for every cross-encoder

University of Glasgow

Metrics

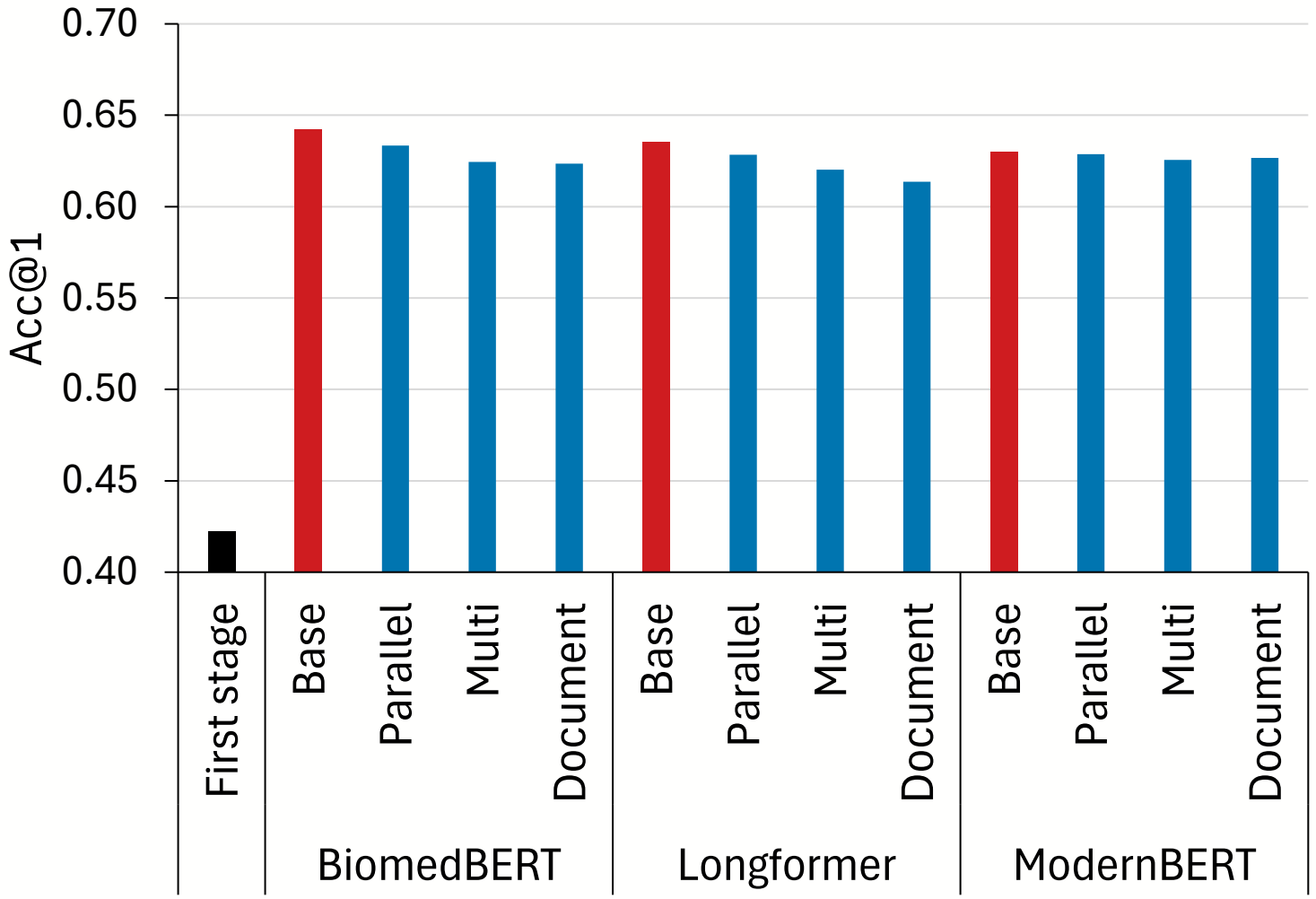
- **Acc@1:** is the top-ranked entity correct?

- **Training speed:**
  - How many training examples (mention, candidate) pairs can we process per second?
  - Ensures fair comparison, as different models might run for different epochs.

- **Inference speed:**
  - How many inference examples (mention, candidate) pairs can we process per second?

University of Glasgow

RQ1: Effectiveness (MedMentions)



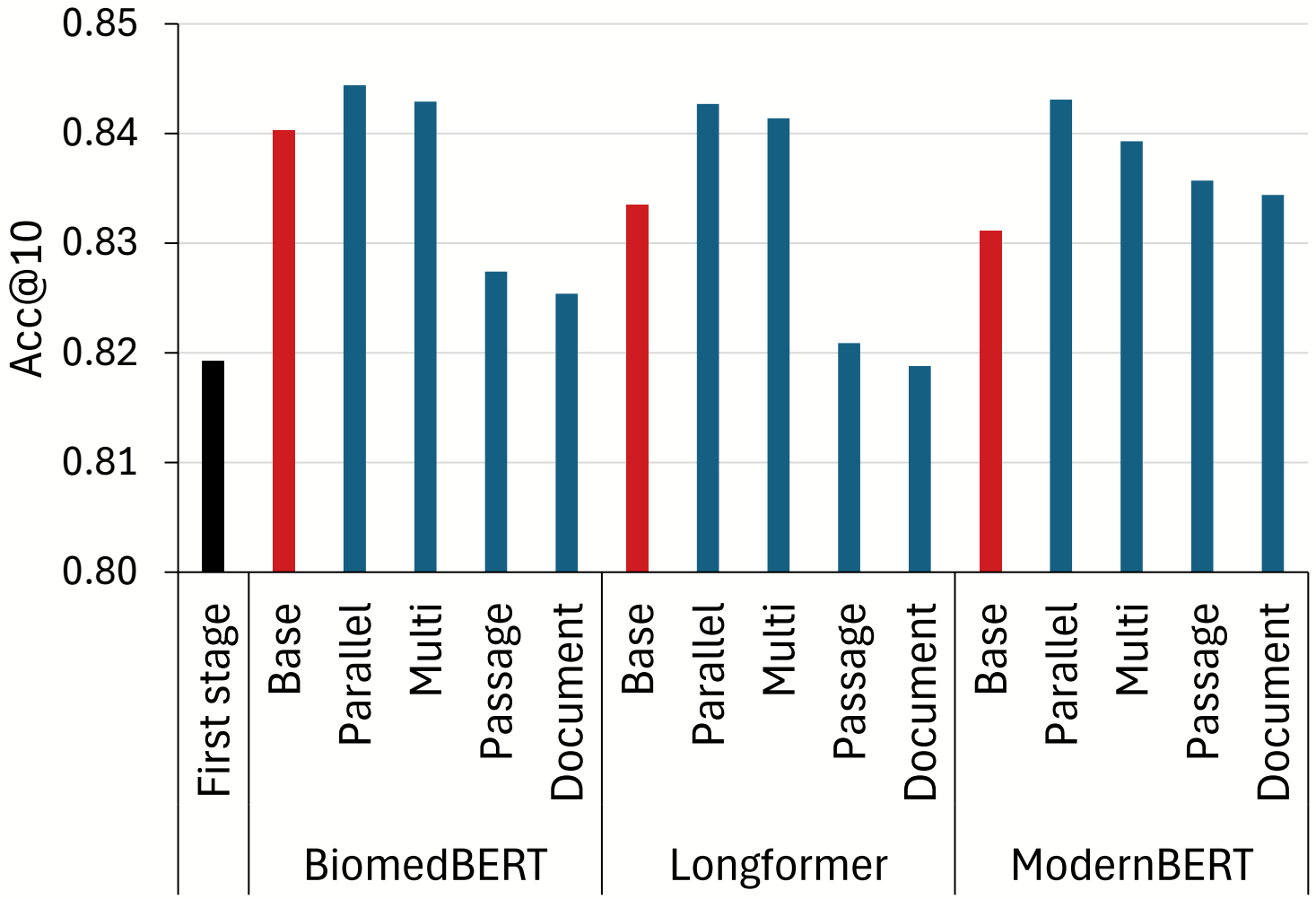
Cross-encoders improve effectiveness of the first stage model

Adding more information reduces Acc@1 on MedMentions

But difference is small (between 0.54% and 3.42% loss)

University of Glasgow

RQ1: Effectiveness (BC5CDR)



- Cross-encoders improve effectiveness of the first stage model
- Parallel and multi-cross encoders achieve some advantage
- But difference is small (<1% difference)
- Passage and document cross-encoders lose accuracy (up to 1.77% loss)

Similar behaviours in the other two datasets

University of Glasgow

RQ1: Effectiveness

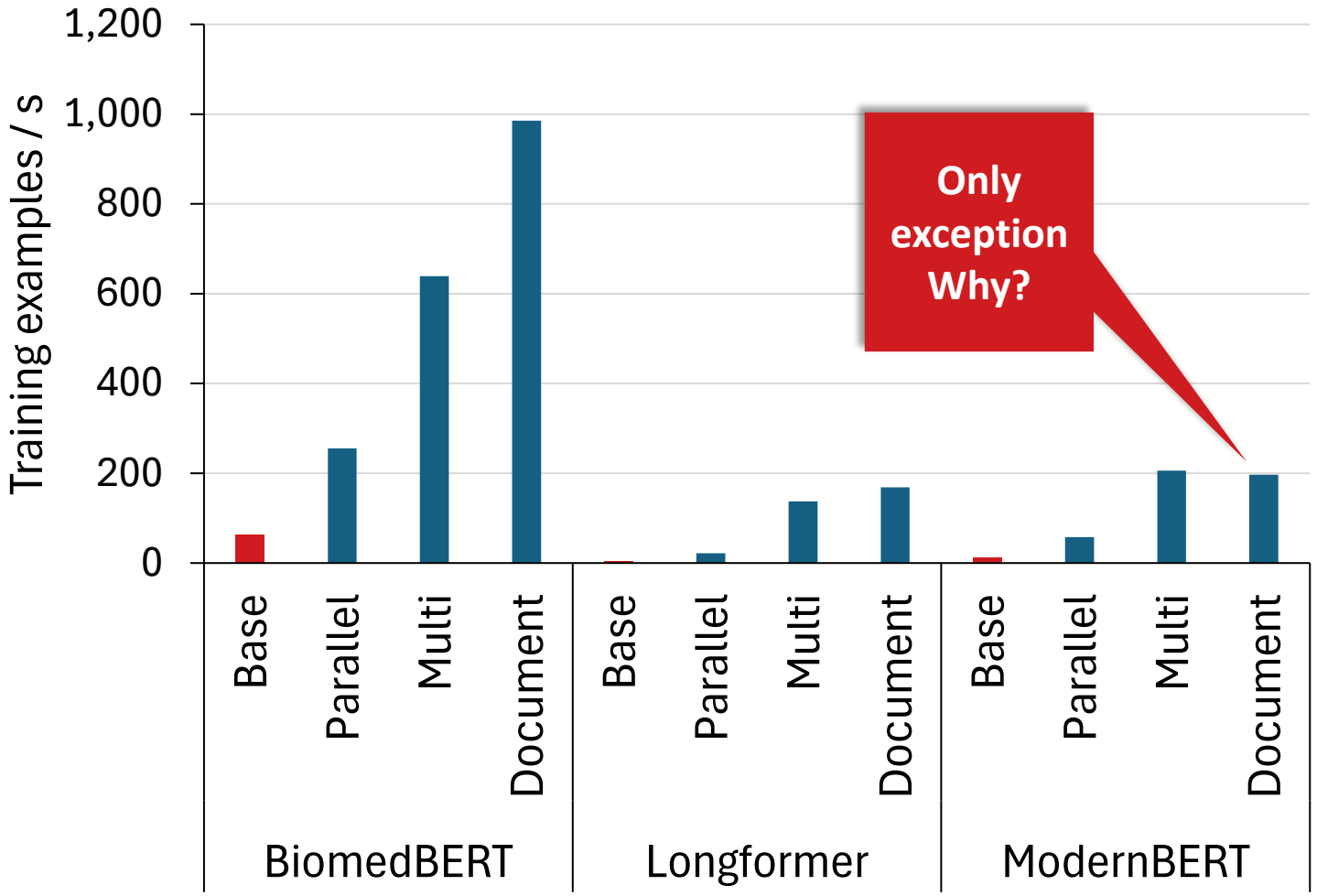
Adding more (mention, entity) pairs to the cross-encoder has limited impact on accuracy.

Different datasets can react differently to the parallelism of the cross-encoders.

All the proposed cross-encoders are reasonable entity linking rerankers

University of Glasgow

RQ2: Training speed (MedMentions)



Parallel cross-encoders accelerate the training between 3.12 and 3.9 times

Multi cross-encoders accelerate the training between 9.3 and 29.93 times

Document cross-encoders accelerate the training between 14.88 and 36.97 times

Similar patterns are observed on other datasets

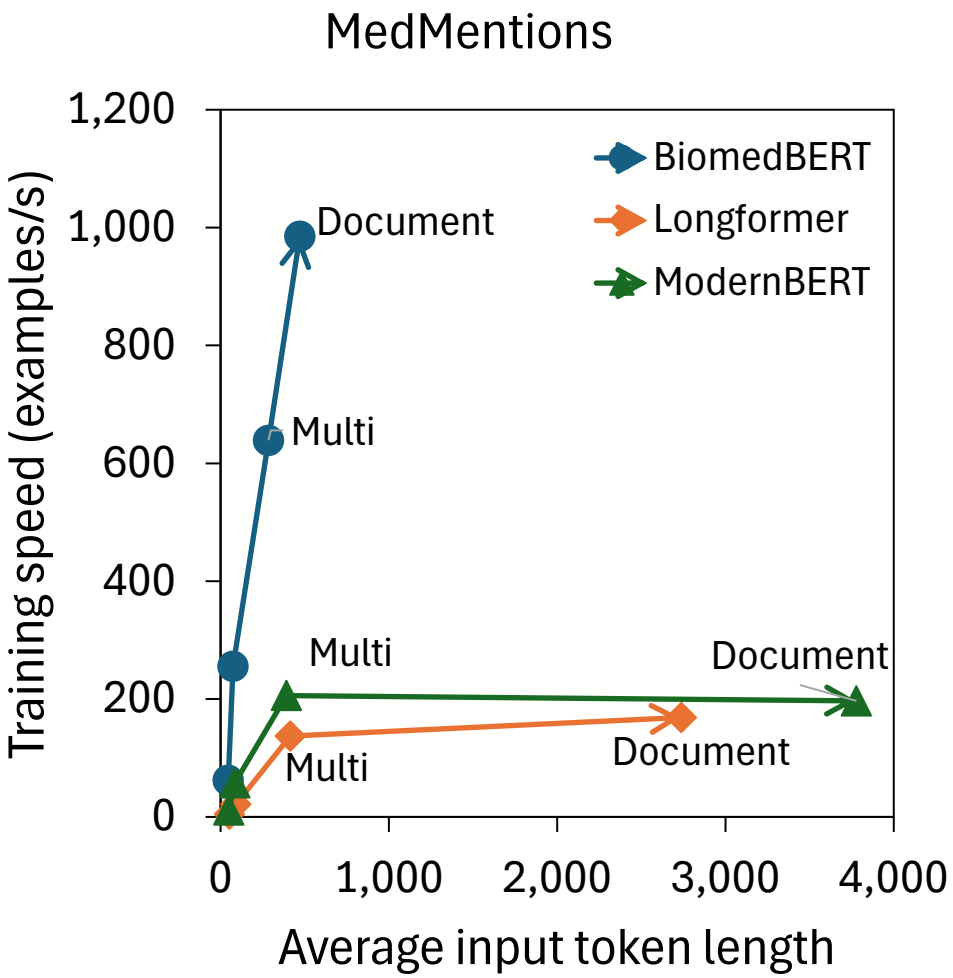
University of Glasgow

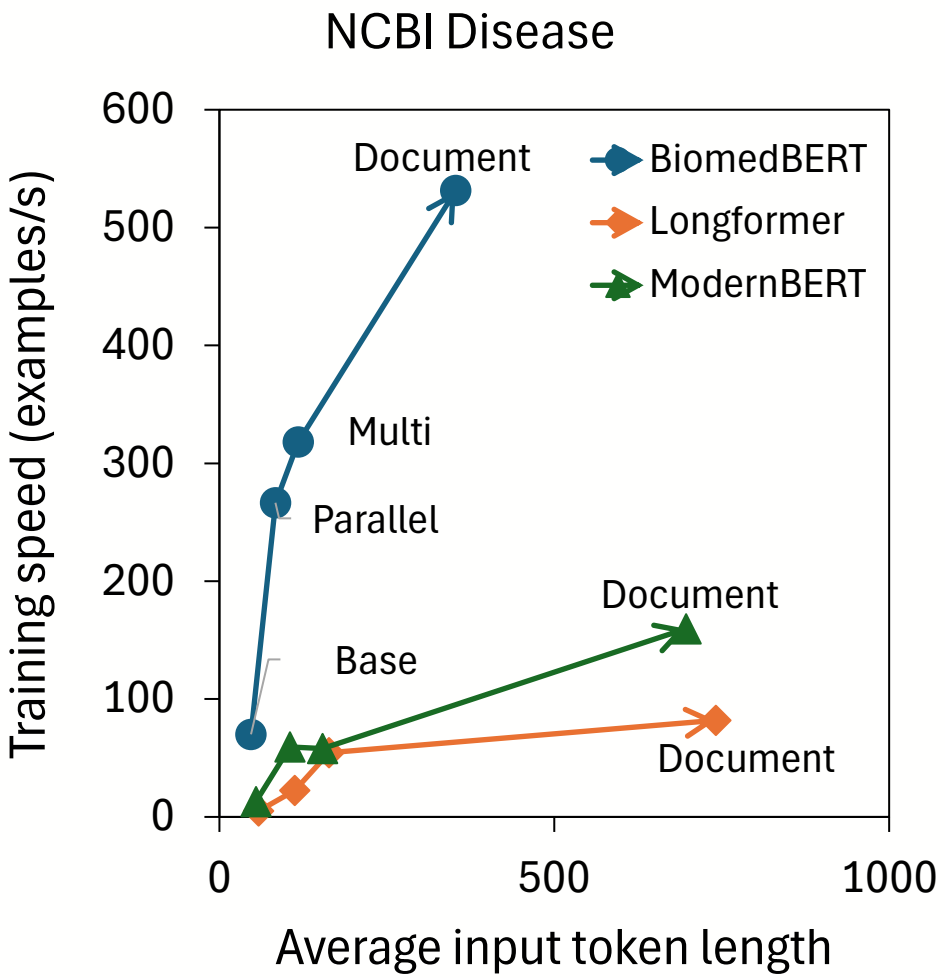
RQ2: Limitations on speed improvements

- Different backbone models have different context windows (maximum amount of tokens they can process at once)
- We limit the number of (mention, entity) pairs to those we can fit into the context window.
- Therefore, for different models we have
  - Different number of input sentences.
  - Each input sentence might have a different number of (mention, entity) pairs.
  - On document-cross encoders, different number of sentences, even.

University of Glasgow

RQ2: Limitations on speed improvements





University of Glasgow

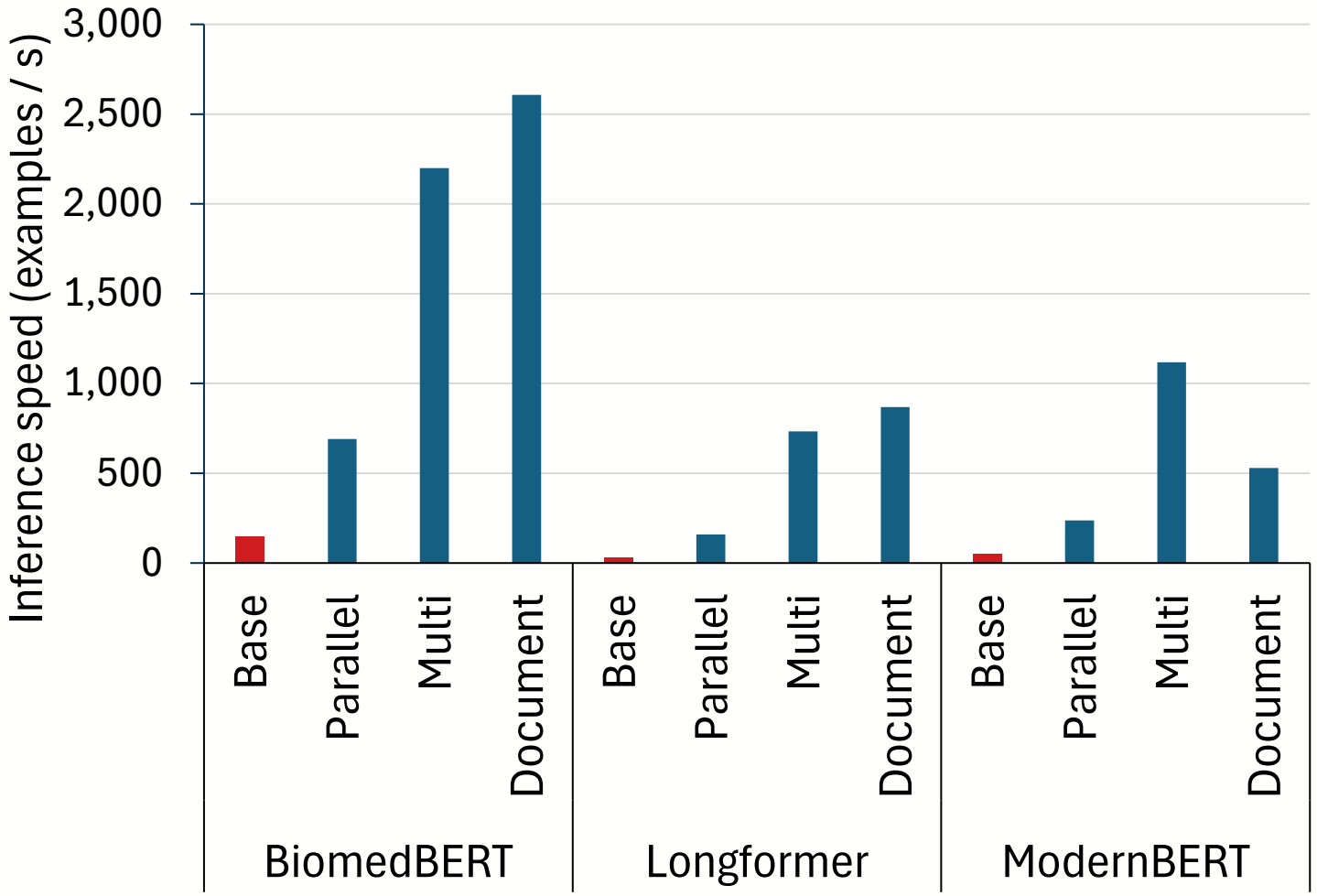
RQ2: Limitations on speed improvements

- ModernBERT allows for longer sentences than Longformer
- However, it struggles with sentences longer than 4096 tokens

- Speed diminishes when there are many examples requiring more than 4096 tokens (still faster than base cross-encoder)
  - MedMentions
  - NLM Chem

- Speed keeps increasing when there are not
  - NCBI Disease
  - BC5CDR

RQ2: Inference speed (MedMentions)



Parallel cross-encoders accelerate the inference between 3.75 and 4 times

Multi cross-encoders accelerate the inference between 15 and 22.18 times

Document cross-encoders accelerate the inference between 9.83 and 26.47 times

Similar patterns are observed on other datasets

University of Glasgow

RQ2: Efficiency

Adding more (mention, entity) pairs to the cross-encoder greatly increases training speed.

Adding more (mention, entity) pairs to the cross-encoder greatly increases inference speed.

Very lengthy input sentences can hinder the efficiency of the models.

Conclusions

University of Glasgow

Conclusions

- **We can accelerate cross-encoders by allowing them to classify multiple (mention, entity) pairs at once**
  - As we add more information, training / inference speeds improve
  - Training speed: between 2.68 and 36.97 times faster
  - Inference speed: between 3.8 and 26.47 times faster
- **However, this is limited by backbone LM capacity:**
  - ModernBERT has difficulties processing texts longer than 4000 characters.
- **Adding more information produces small effects on performance**
  - Usually, parallel cross-encoders achieve slightly better performance
  - Document cross-encoders worsen base performance
  - Differences in a -3.42% to 2.76% differences
- **We can have a major training/inference speed improvement at a small accuracy cost!**

University of Glasgow

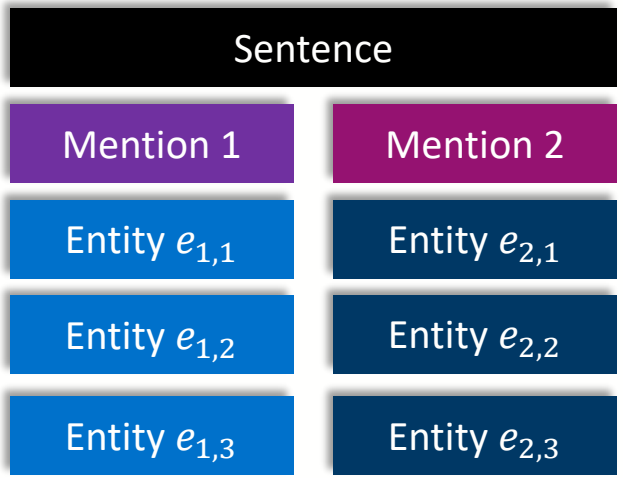
Future work

- **Apply this to cross-encoder with pair-wise or list-wise losses**
- **Accelerate other transformer architectures (bi-encoders, poly-encoders)**
  - Can we, for instance, accelerate the training of SapBERT?
- **Is this consistent with entity linking in other domains beyond biomedical?**

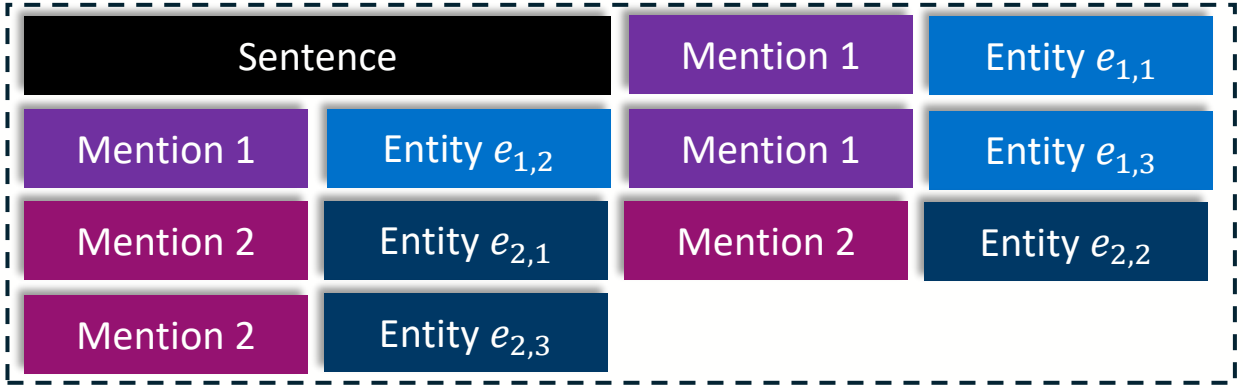
Questions?

Dr. Javier Sanz-Cruzado
AI4BioMed Group, University of Glasgow
javier.sanz-cruzadopuig@glasgow.ac.uk
JavierSanzCruza
Javiersanzcruza.bsky.social
https://www.linkedin.com/in/javier-sanz-cruzado-puig/

Dealing with multiple context windows

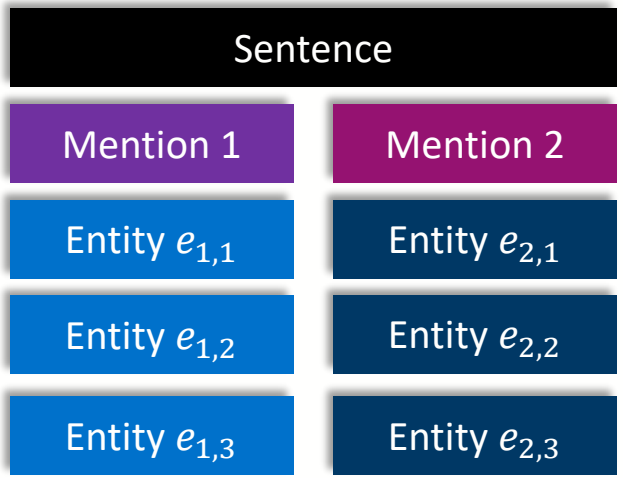


Long context window

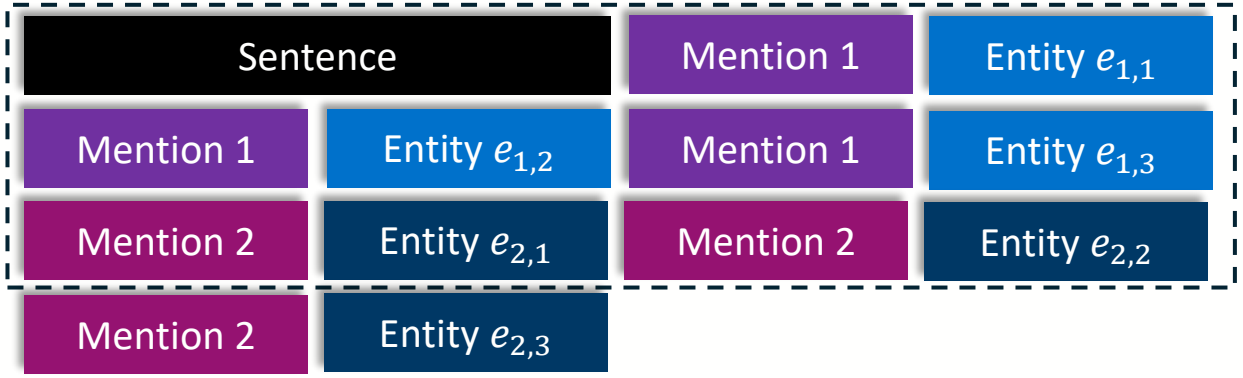


We can fit the whole set of (mention, entity) pairs in one input sentence

Dealing with multiple context windows

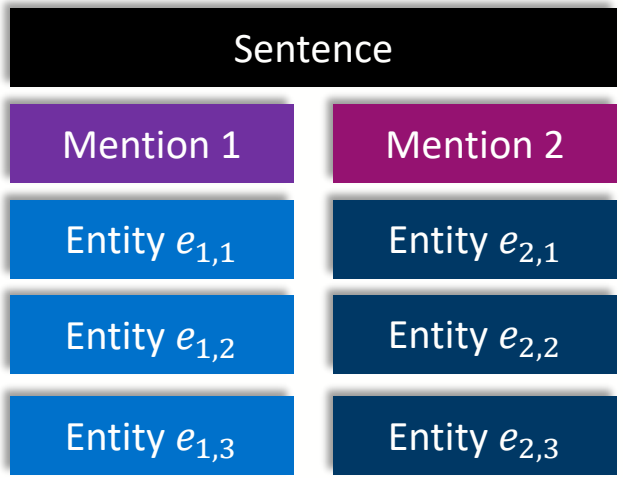


Shorter context window

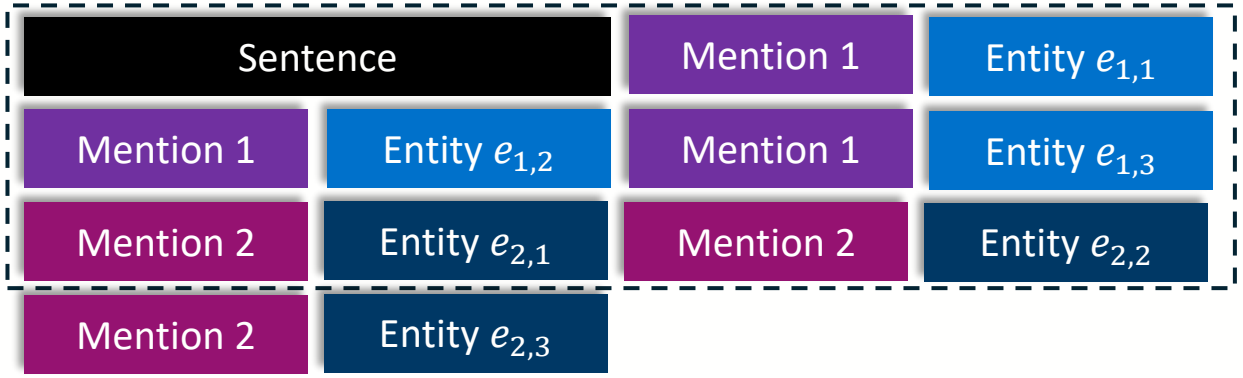


The last mention escapes the context window!

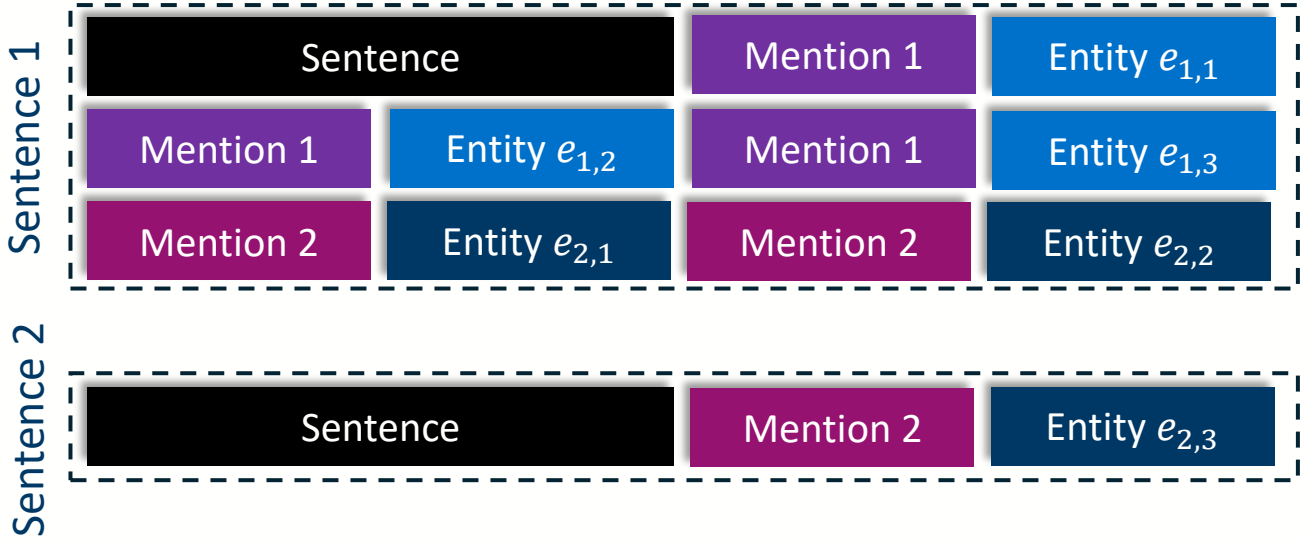
Dealing with multiple context windows



Shorter context window



Divide the input sentence into two



University of Glasgow

Dataset statistics

	MedMentions	NCBI Disease	NLM Chem	BC5CDR
Ontology	UMLS	Medic	MeSH 2021	MeSH 2015
Documents (train)	2,635	593	80	500
Documents (val)	878	100	20	500
Documents (test)	879	100	50	500
Passages (train)	-	-	5,555	1,000
Passages (val)	-	-	1,285	1,000
Passages (test)	-	-	3,470	1,000
Sentences (train)	25,836	5,173	20,126	4,242
Sentences (val)	8,508	888	4,855	4,299
Sentences (test)	8,597	901	12,031	4,524
Annotations (train)	211,029	4,836	19,361	9,323
Annotations (val)	71,062	711	4,927	9,570
Annotations (test)	70,405	896	11,164	9,725