BERT Model and Convolutional Neural Networks for Relation Extraction

Fatima Habib

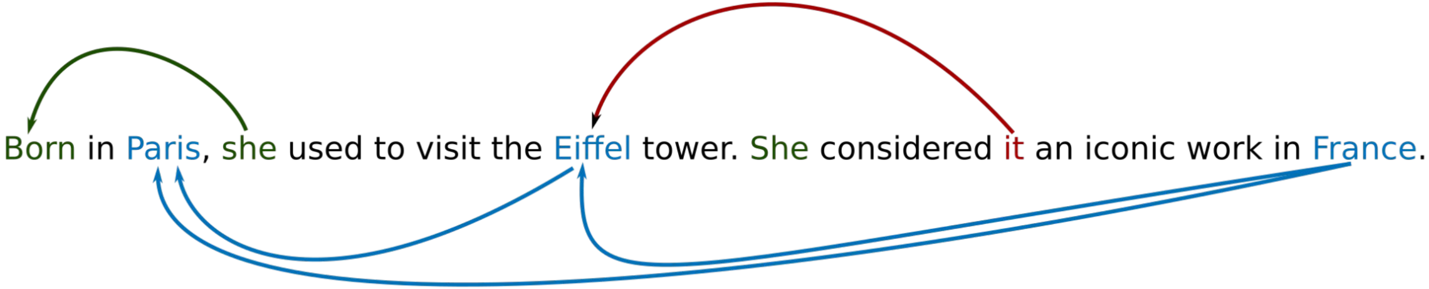
09/09/21

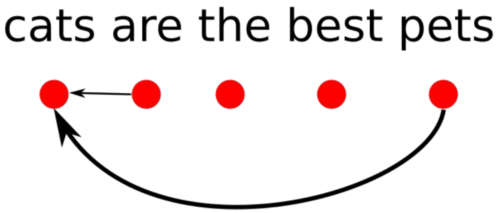
Supervisors

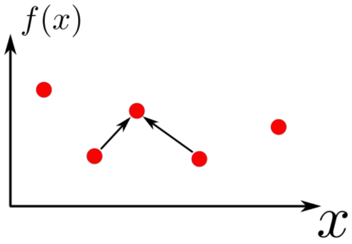
Loria Team ORPAILLEUR

Prof. Yannick Toussaint **PhD. Student** Laura Zanella

Relation Extraction







Embeddings

- Free-context embeddings .
  - The representations of words are independent in the sense that they do not contain any context content.
  - One-hot vector and term frequency-inverse document frequency (TF-IDF).

- Context aware embeddings:
  - Allows similar and related words to have similar representations.

Neural Networks: Fully Connected Neural Network

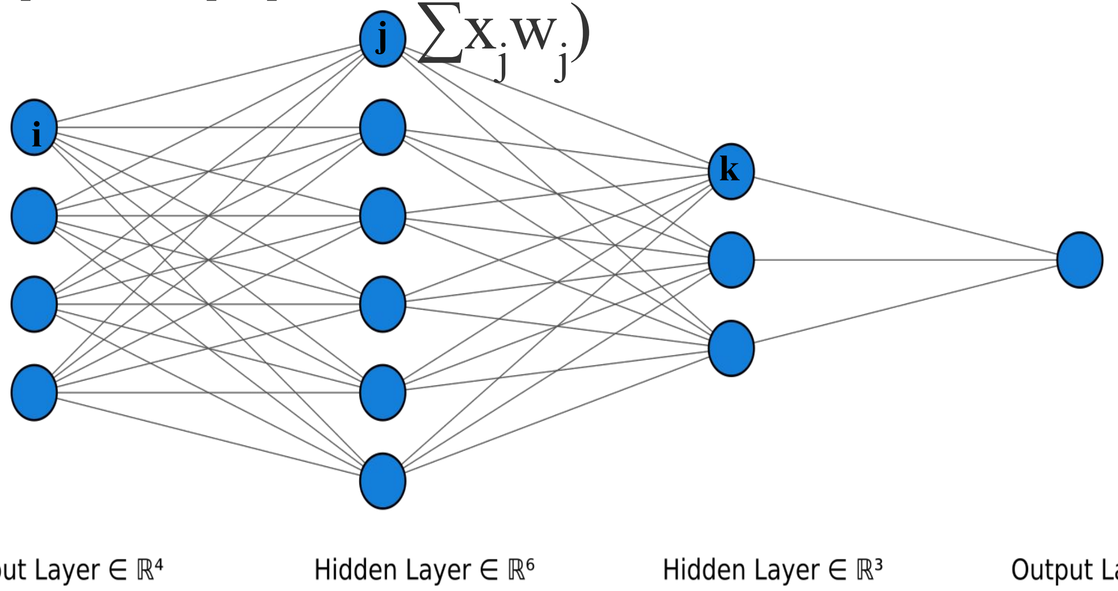
$y_i=f(\sum x_i w_i)$ $\quad$ $yj=f(\sum x_j w_j)$ $\quad$ $O=f(\sum x_k w_k)$

1. Architecture.

2. Activation functions.

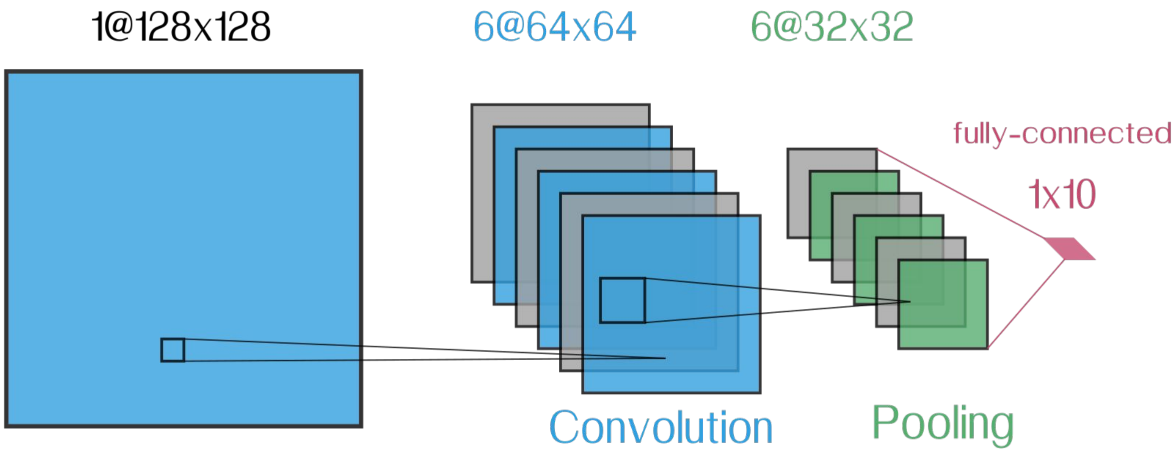
3. Cost function: it is a non-negative function measuring the accuracy of the outcomes of the neural network.

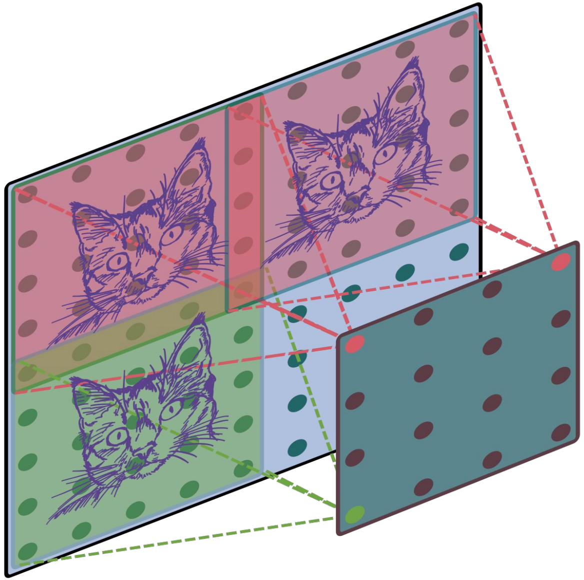
4. Learning algorithm (optimization algorithm): minimize the cost function.



The training aims to learn the **weights** that reduce the **cost function.**

Convolutional Neural Networks (CNNs)



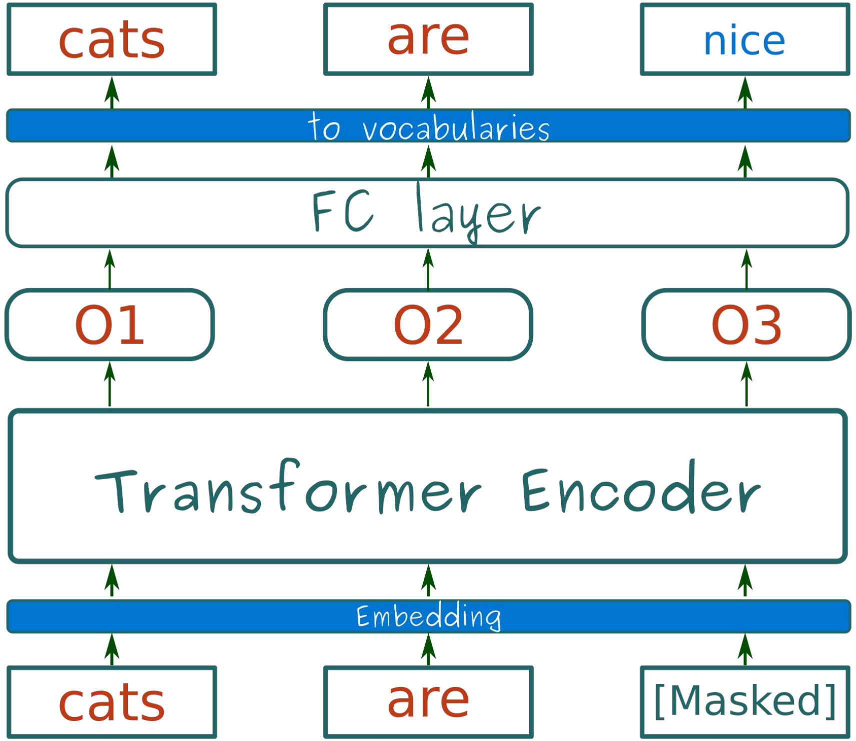


Translation invariant

BERT Model - 1

- BERT model elements Devlin et al[2018]:
  - Embedding layer: the input sentence is converted it is numerical representation.
  - Transformer encoder: a layer improving the representation of each word by including more context.
  - Fully connected classification layer trained on two strategies:
    - Masked language model (MLM).
    - Next sentence prediction (NSP).

Remark: the main strategy used in other methods is the next word prediction contrary to BERT that uses the above strategies.



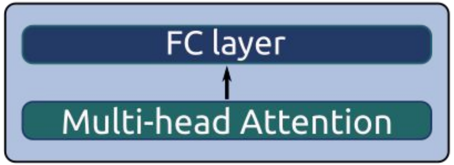
BERT Model - 2: Transformer Encoder

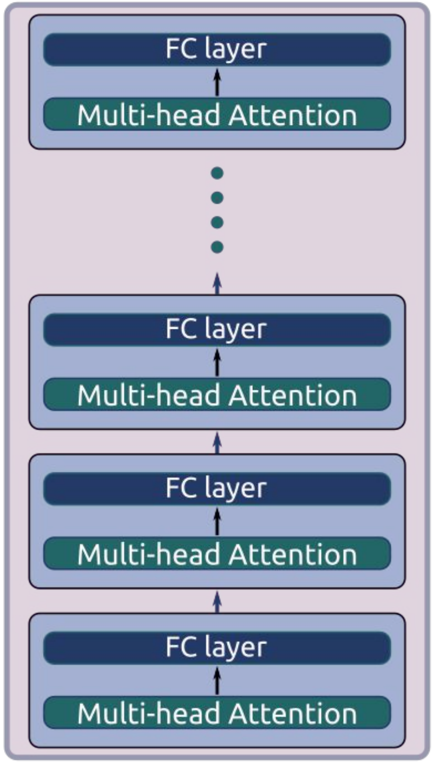
- Transformer encoder Vaswani et al[2017] is a stack of encoders trained to obtain a more context aware representation of the word.

- The encoder is composed of two layers:
    - Attention layer: is used to learn more contextualized representation of each word.
    - A fully connected layer.

- The output of the transformer layer is the language model of the sentence.

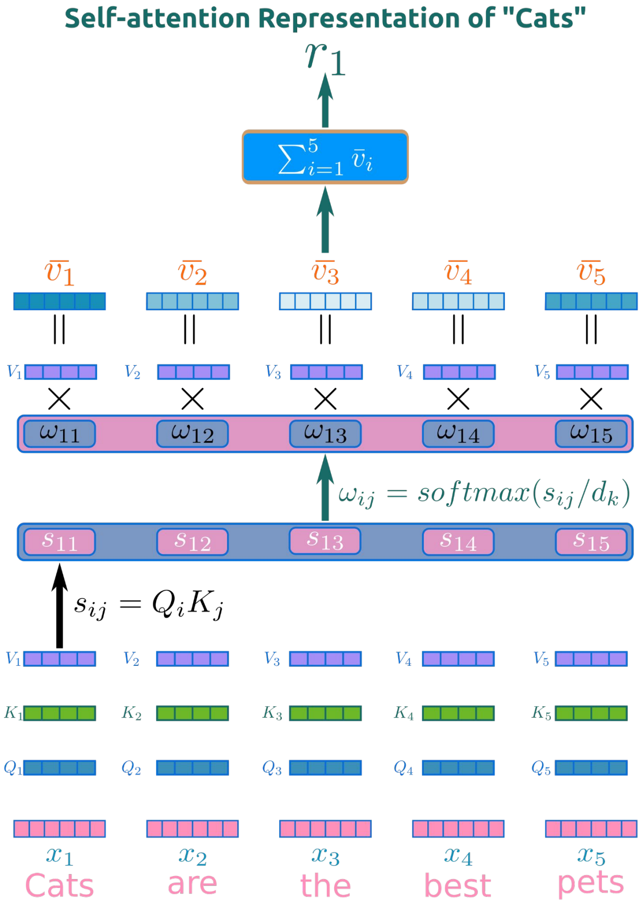
An encoder





BERT Model - 3: Attention Mechanism

- Each word is associated with Q, K, and V vectors.
- The score vector (s) of each word is obtained.
- We transform s into weight vector ω to avoid numerical instability issues (softmax).
- We multiply the weight vector of each word with the old value vectors to obtain the new value vectors.
- We finally add the updated value vectors together to obtain the new representation of each word.



Experiments

We mainly used two neural network architectures:

1. CNNs based architectures with BioWordVec as an embedding layer.
   - BioWordVec is a pre-trained embeddings for biomedical words.
2. SciBERT based architectures.
   - SciBERT: is a variation of BERT trained on scientific texts (82% from biomedical field).

BioCreative IIV

- We participate in the Track - 1 Text mining drug and chemical-protein interactions[1].

- We use the **DrugProt** corpus a manually annotated corpus with:
  - All chemical and gene mentions.
  - All binary relationships between them.

- The goal is to predict one of 13 relations (chemical interactions) between the annotated entities in a sentence.

1. M. Krallinger et al. “Overview of the biocreative vi chemical-protein interaction track,” 2017.

DrugProt

- It is built using PubMed[1] abstracts:

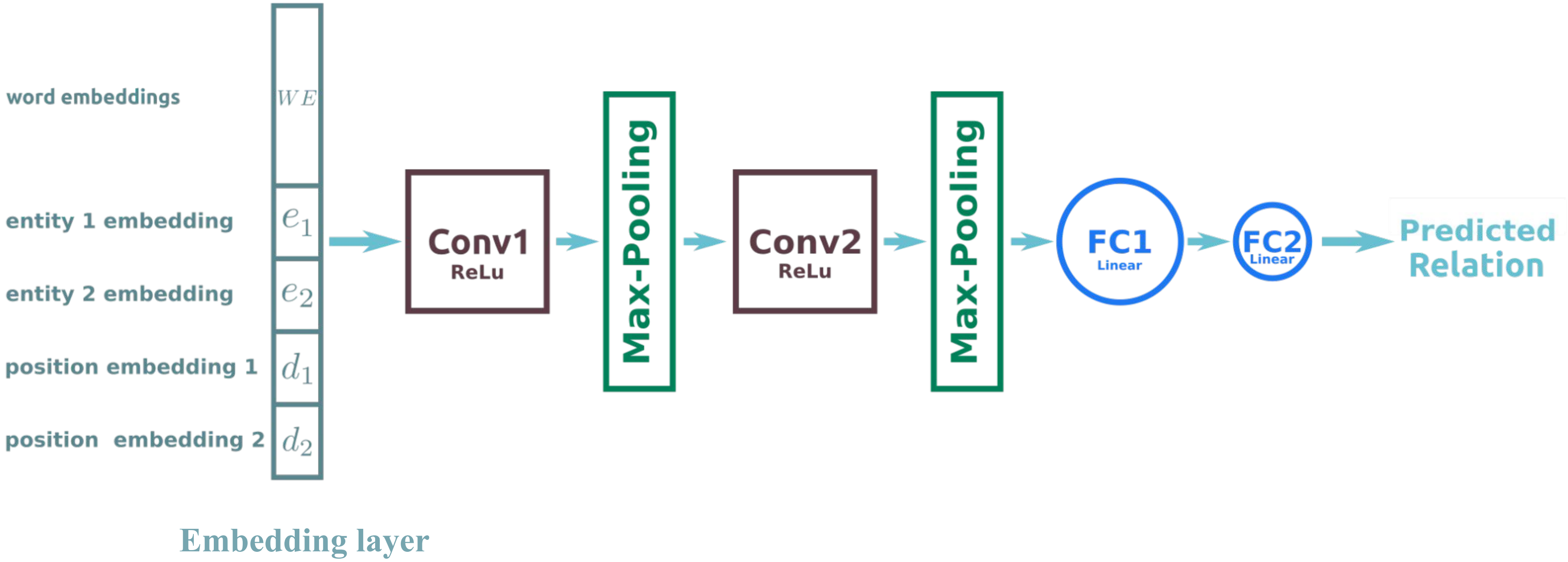
	Training	Development	Testing
Documents	3500	750	10750
Annotated entities (CHEMICAL and GENES)	89529	18858	310805
Annotated Relations	17288	3765	To be predicted

1. Comprises more than 32 million citations for biomedical literature from MEDLINE, life science journals, and online books https://pubmed.ncbi.nlm.nih.gov/.

Relations Types Distribution

Relations	**Total**	Training	Testing
ACTIVATOR	1423	1149(8.15%)	274
AGONIST	658	524(3.72%)	134
AGONIST-ACTIVATOR	29	26(0.19%)	3
AGONIST-INHIBITOR	13	12(0.08%)	1
ANTAGONIST	970	767(5.44%)	203
DIRECT-REGULATOR	2240	1785(12.67%)	455
INDIRECT-DOWNREGULATOR	1328	1073(7.61%)	255
INDIRECT-UPREGULATOR	1376	1078(7.65%)	298
INHIBITOR	5377	**4307(30.57%)**	1070
PART-OF	882	729(5.17%)	153
PRODUCT-OF	916	735(5.23%)	181
SUBSTRATE	2002	1591(11.29%)	411
SUBSTRATE_PRODUCT-OF	24	14(0.099%)	10
no_relation	44932	300(2.129%)	300

CNN Based Architecture



Embedding Layer

Positional Embedding:

- The positional vector d_n of entity e_n has a size equals to the number of the tokens in the sequence.

- The component of d_n at entity position is considered to be the origin.

- Each other component of d_n represents the distance of the corresponding token from the entity.

Sentence Padding (SP):

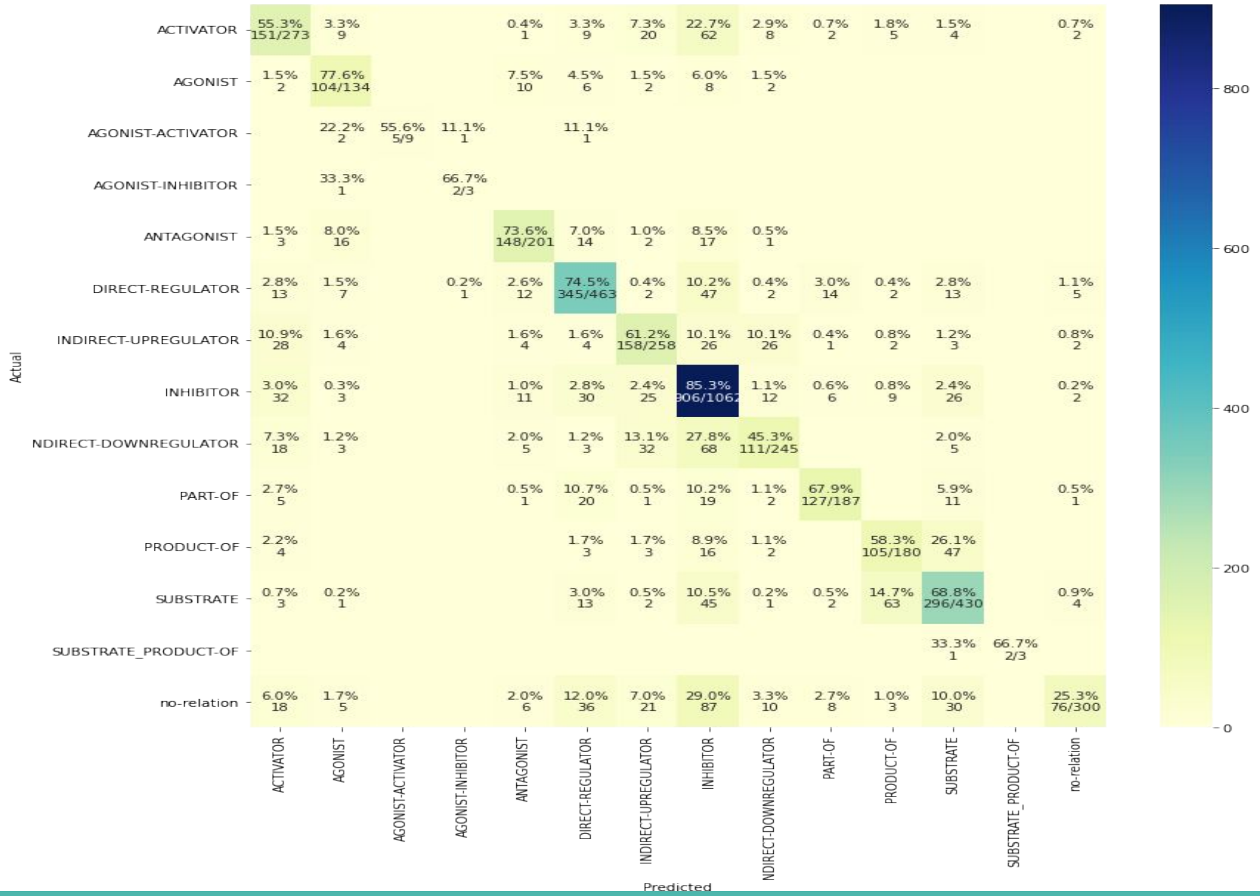
- Each sentence is represented by m*n matrix (m is the number of words in the sentence while n is the size of the word embedding).
- As m changes from sentence to another, SP includes unifying m for all sentences in the corpus.

		d_1	d_2
	Ornithine	-17	-3
	decarboxylase	-16	-2
	(	-15	-1
e_2 :	ODC	-14	0
	)	-13	1
	catalyses	-12	2
	the	-11	3
	first	-10	4
	step	-9	5
	in	-8	6
	the	-7	7
	synthesis	-6	8
	of	-5	9
	the	-4	10
	polyamines	-3	11
	putrescine	-2	12
	,	-1	13
e_1 :	spermidine	0	14
	and	1	15
	spermine	2	16
	.	3	17

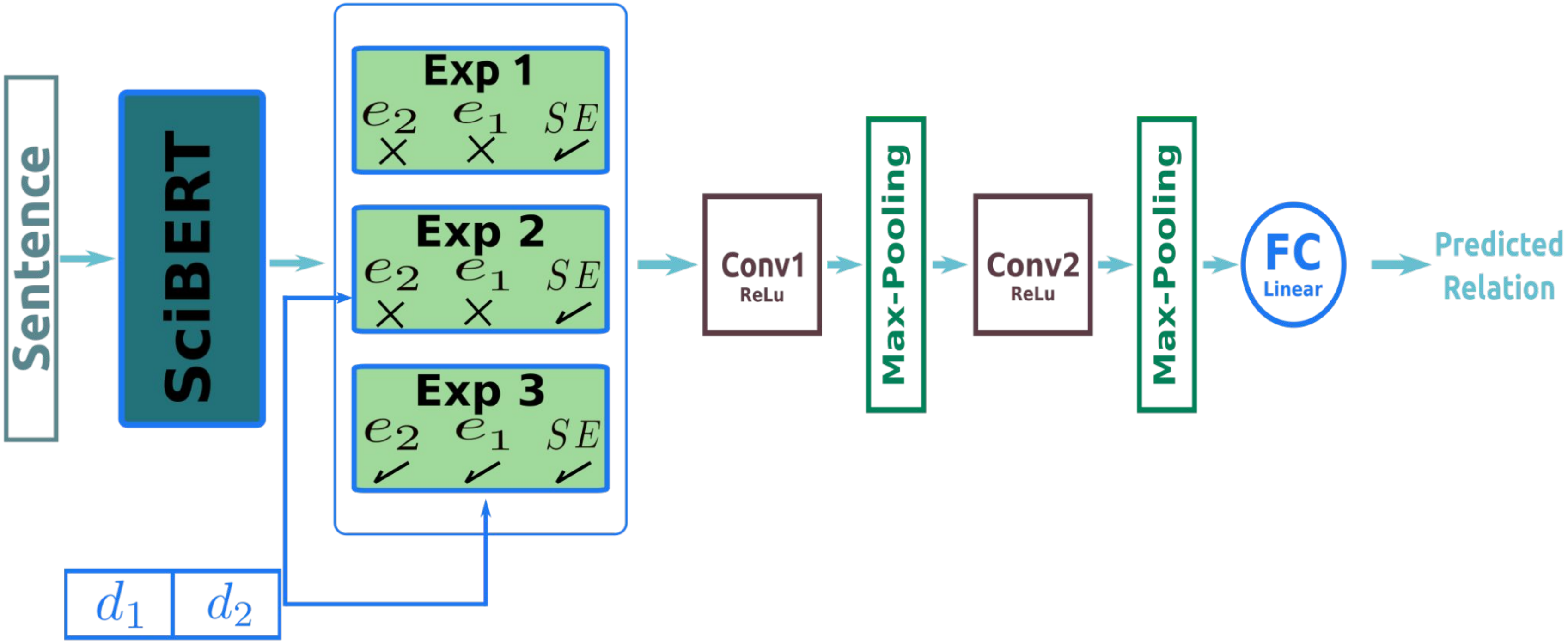
CNN Based Model Experiments

Sentence padding	**Accuracy**	**Precision**	**Recall**	**F1 score**
High number	66.765 %	0.699 %	0.664	0.67
High number	66.969 %	0.678 %	0.5960	0.611
Constant number	66.061 %	0.678 %	0.650	0.644
Constant number	**67.666 %**	**0.710 %**	0.630	0.649

CNN Based Model Experiments



SciBERT Based Model Architecture



SciBERT Based Model Experiments

Position embedding	Entity embedding	Accuracy	Precision	Recall	F1 score
-	-	70 %	0.63	0.63	0.60
+	-	76 %	0.66	0.59	0.61
+	+	77 %	0.60	0.57	0.59

SciBERT Based Model Experiments

Conclusions

- The positional embeddings play a key role in enhancing the performance of both models (CNN and SciBERT based models).

- SciBERT based models outperform CNNs based models due to the rich contextualized representations given by BERT.

- Unbalanced data influence the results in way that it slow down the performance.

Future Work

- Apply oversampling techniques like the random oversampling and use external resources like datasets that contains same relations.

- Use different architectures: LSTM.

- Use additional features as inputs: the shortest dependency path between the entities.

Thank you