Contract Element Extraction

With Deep Learning

Contract Elements

- Contract Title
- Contracting Parties
- Start Date
- Effective Date
- Termination Date
- Contract period
- Contract value
- Governing Law
- Jurisdiction
- Legislation references

Contract elements

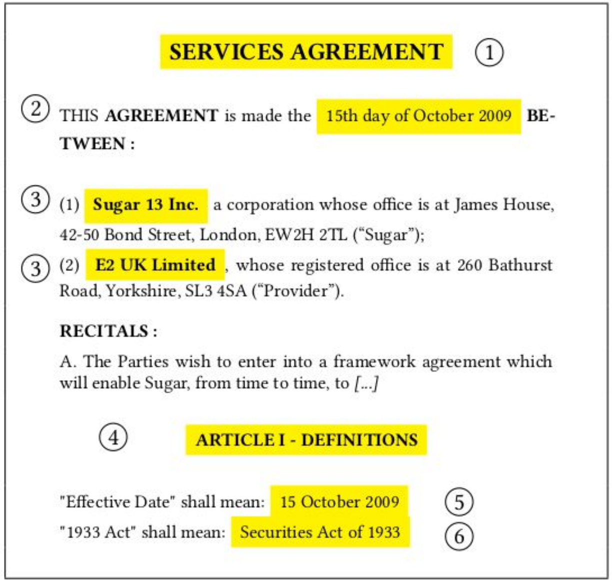
Extraction Zones (at testing)	Example Clause Heading Words	Contract Elements Typically Included
Cover page and preamble	–	Contract Title, Contracting Parties, Start Date, Effective Date
Term clause	'Term', 'Period', 'Term of Agreement'	Termination Date, Contract Period
Termination clause	'Termination', 'Termination of Agreement'	Termination Date
Governing Law clause	'Governing Law', 'Applicable Law'	Governing Law, Jurisdiction
Jurisdiction clause	'Jurisdiction', 'Jury Trial', 'Venue'	Jurisdiction
Miscellaneous clause	'Miscellaneous', 'Entire Agreement'	Governing Law, Jurisdiction
Contract Value clause	'Lump Sum', 'Salary'	Contract Value
In the text after the recitals, zones starting up to 20 tokens before and ending up to 20 tokens after each line break, not crossing other line breaks		Clause Headings
In the entire contract, zones starting up to 20 tokens before and ending up to 20 tokens after each occurrence of words like 'Act', 'Treaty' etc.		Legislation References

Table 1: Extraction zones where contract elements of different types are searched during testing.

What is contract element extraction?

- Refers to automatic extraction of important information from contracts.
- This is done through Named entity Recognition.(Sequence tagging)
- Eg-**John lives in New York and works for the European Union**
  B-PER O O B-LOC I-LOC O O O O B-ORG I-ORG

- **PER-Person**
- **LOC-location**


SERVICES AGREEMENT 1

2 THIS **AGREEMENT** is made the 15th day of October 2009 **BE-TWEEN :**

3 (1) **Sugar 13 Inc.** a corporation whose office is at James House, 42-50 Bond Street, London, EW2H 2TL ("Sugar");

3 (2) **E2 UK Limited** , whose registered office is at 260 Bathurst Road, Yorkshire, SL3 4SA ("Provider").

RECITALS :

A. The Parties wish to enter into a framework agreement which will enable Sugar, from time to time, to *[...]*

4 **ARTICLE I - DEFINITIONS**

"Effective Date" shall mean: 15 October 2009 5
"1933 Act" shall mean: Securities Act of 1933 6

Why is it important?

Banks need to monitor contracts for various tasks.Many of these tasks can be automated by extracting particular contract elements (e.g., termination dates, legislation references, contracting parties, agreed payments). Contract element extraction, however, is currently performed mostly manually, which is tedious and costly. A lot of money and human labour can be saved by automating this process.

Eg:-Contract Intelligence (CoIN)

JPMorgan Chase has developed a proprietary Machine Learning algorithm called Contract Intelligence or COiN.

It was used to analyze documentation and extract the important information from it.

Applying this tool enabled the bank to process 12,000 credit agreements in several seconds, instead of 360,000 man-hours.

Project

Implementing-A Deep Learning Approach to Contract Element Extraction,

By Ilias Chalkidis, Ion Androutsopoulos ,Athens University of Economics and Business, Greece and Cognitiv+ Ltd., London, UK.

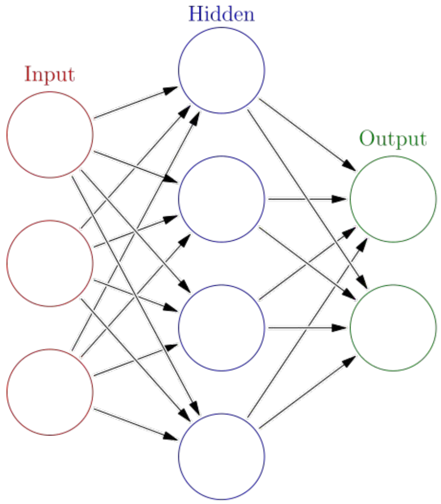
Along with the paper they provide a encoded dataset.

Deep learning

Generally refers to any learning architecture that uses neural networks of various types.

$$w \cdot X + b$$

Model - It is a set of weights and biases , and a defined architecture.



Training - The model gets trained by passing data through the model once and then backpropagating to minimise the error.

Training dataset- Data that is used to train the model.

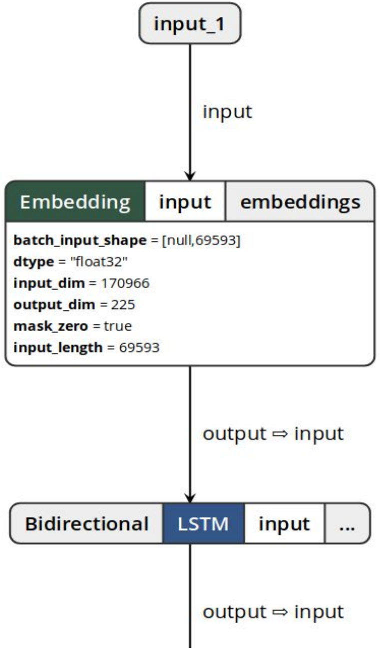
Test Dataset - Data on which the model is tested upon.

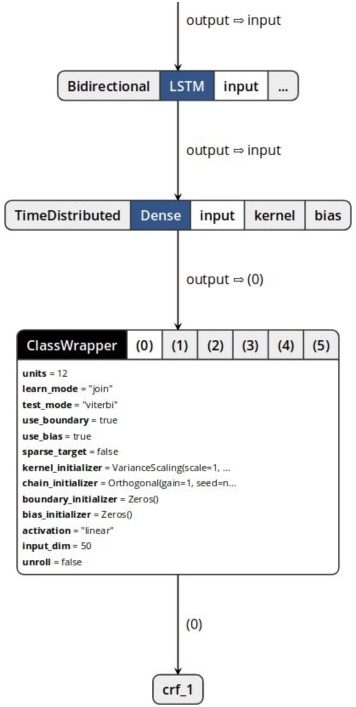
Dataset

TOKEN_2888[0] TOKEN_2889[0]
TOKEN_1490[TIT] TOKEN_6[TIT]
TOKEN_15[0] TOKEN_6[0] TOKEN_2384[0] TOKEN_263[0] TOKEN_28816[STD] TOKEN_28[STD]
TOKEN_25[STD] TOKEN_4376[STD] TOKEN_19[STD] TOKEN_1530[STD] TOKEN_31[0]
TOKEN_78167[CNP]
TOKEN_5565[CNP] TOKEN_1539[CNP] TOKEN_19[0] TOKEN_66[0] TOKEN_12279[0] TOKEN_2207[0]

- 2000 training contracts
- 350 test contracts
- 200 dimensional embeddings
- 25 dimensional POS tags

Model


input_1
input
Embedding
input
embeddings
batch_input_shape = [null,69593]
dtype = "float32"
input_dim = 170966
output_dim = 225
mask_zero = true
input_length = 69593
output ⇨ input
Bidirectional
LSTM
input
...
output ⇨ input


output ⇨ input
Bidirectional
LSTM
input
...
output ⇨ input
TimeDistributed
Dense
input
kernel
bias
output ⇨ (0)
ClassWrapper
(0)
(1)
(2)
(3)
(4)
(5)
units = 12
learn_mode = "join"
test_mode = "viterbi"
use_boundary = true
use_bias = true
sparse_target = false
kernel_initializer = VarianceScaling(scale=1, ...
chain_initializer = Orthogonal(gain=1, seed=n...
boundary_initializer = Zeros()
bias_initializer = Zeros()
activation = "linear"
input_dim = 50
unroll = false
(0)
crf_1

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 69593)             0

embedding_1 (Embedding)      (None, 69593, 225)        38467350

bidirectional_1 (Bidirection (None, 69593, 100)        110400

time_distributed_1 (TimeDist (None, 69593, 50)         5050

crf_1 (ClassWrapper)         (None, 69593, 12)         780
=================================================================
Total params: 38,583,580
Trainable params: 38,583,580
Non-trainable params: 0
```

Embeddings and Input

$$f : X \hookrightarrow Y.$$

Embeddings map an input to an output. Y is called the embedding of X.

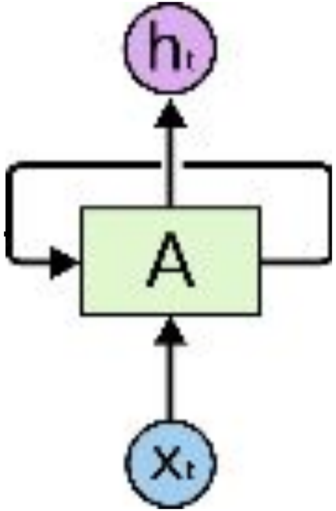
Input: ‘TOKEN_2888[0] ‘

Preprocess:2888

Output: 225 dimension vector

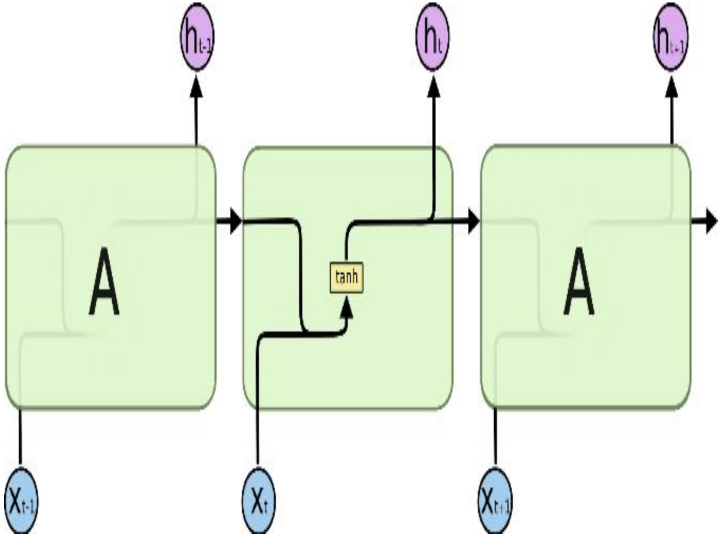
Recurrent Neural Networks

- Neural network in which output from the previous input is used to process the next input.
- The neural net gains memory.
- Makes learning from sequence input easier.
- But still has a problem of not being able to link at large distance.

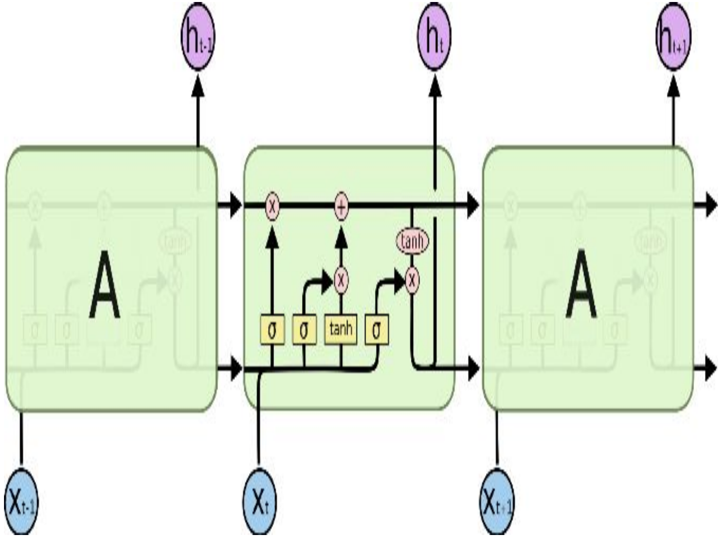


Bidirectional-Long Short Term Memory (BiLSTM)

RNN



LSTM



BiLSTMS

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

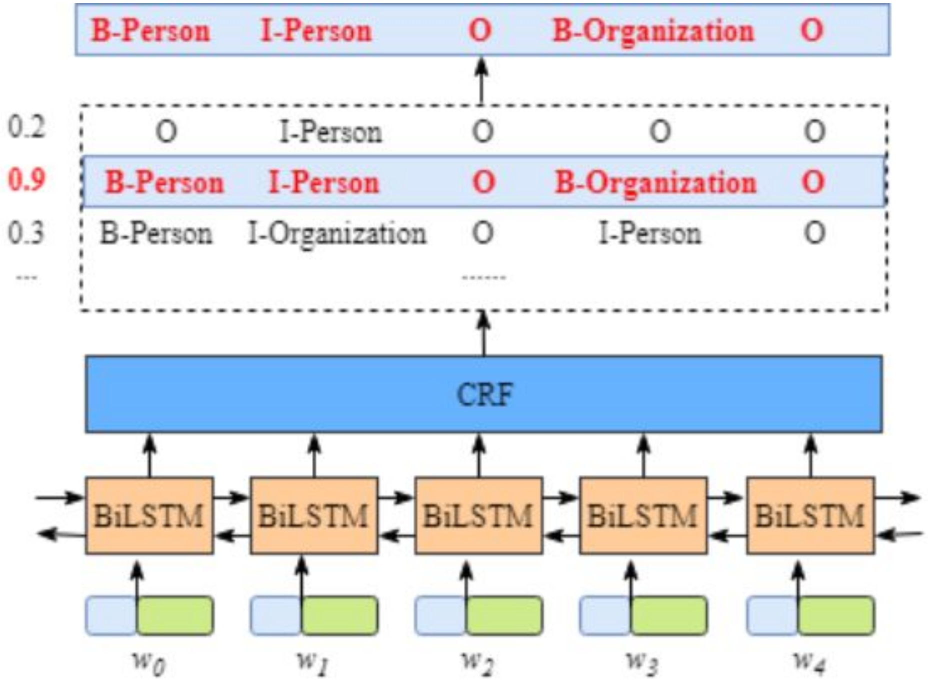
The key to LSTMs is the cell state, the horizontal line running through the top of the cell.It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.

Conditional Random Fields

CRF takes the prediction for each

Input and uses them to generate the

most likely sequence.



Metrics

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Results

	ld	f1	p	r	s	BILSTM-CRF P	BILSTM-CRF R	BILSTM-CRF F1
0	0	0.999855	0.999817	0.999894	24337213			
1	TIT	0.901983	0.873234	0.932689	2585	**0.96**	**0.95**	**0.95**
2	CNP	0.831151	0.842347	0.820249	5146	**0.98**	0.92	**0.95**
3	STD	0.724796	0.748392	0.702642	1325	0.92	**0.98**	**0.95**
4	EFD	0.120983	0.761905	0.065708	487	0.95	0.89	0.92
5	TED	0.234957	0.650794	0.143357	286	0.65	0.93	0.77
6	PER	0.425926	0.932432	0.276000	250	0.55	0.85	0.65
7	VAL	0.122667	0.696970	0.067251	342	0.72	0.60	0.66
8	GOV	0.877108	0.819484	0.943450	2122	**0.99**	0.97	**0.98**
9	JUR	0.719714	0.817497	0.642825	2195	**0.90**	0.88	0.88
10	LEG	0.830986	0.876064	0.790320	5599	0.82	0.94	0.87

Code

https://github.com/bartimaeus12/PS-project