DEEP LEARNING MODELS FOR ELEMENT EXTRACTION IN LEGAL DOCUMENTS

Gustav Hartz, s174315@student.dtu.dk

Technical University of Denmark, DTU Compute

ABSTRACT

This paper assesses the use of deep learning models for the extraction of specific contract elements from legal documents. It does this with the aim of investigating the feasibility of an intelligent contract management/searching-systems. The primary approaches are the Bidirectional LSTM based models BiDAF and DrQA along with pretrained transformer models for reference. The results show that the models can achieve results in a range of 80-90% for F1 score and 80% for exact matching deeming it interesting to pursue the project further.

1. INTRODUCTION

Company processes across the globe often involve unstructured documents such as Email, Text files, or PDFs [1]. Recent developments within computer vision and deep learning have led to increases in automated handling of standardised elements such as forms, but there are many internal processes where this does not apply. One of these areas is work involving contracts. In an interview with Cecilia Strand, Finance Director of Scandinavia at KONE, she noted that populating their systems, such as SAP and Contract Management Software, is an entirely manual process and that tools to automate parts of this process can dramatically increase productivity.

Information retrieval in large text corpuses is an area of research that lately has received a lot of attention. The reader might have noticed how Google has started presenting answers to questions directly in the search results. This is an example of that. Intelligent information retrieval and searching are interesting in the scope of legal contracts, since well-performing systems based on this technology has the potential, as part of a larger software system, for informing decision-makers about when to renew contracts, speed up processes, and keeping relevant parties informed about supplier terms affected by new legislation[1].

The objective of this paper is to present deep learning models for extracting specific contract elements from legal documents, with the perspective of investigating the feasibility of implementing the aforementioned intelligent contract management/searching-systems based on the presented models. An outline of the full system can be seen in figure 1. What this paper assesses is the blue square - the deep learning models that power it. The contract elements investigated are a subset of the contract elements that occurs in most legal documents. This is the Document Title, Document Type, Agreement-, Renewal-, Expiration-, and Effective Dates. This project is limited to these parts to reduce the scope. Furthermore, these elements can in most cases be found on the first page of the document or narrowed down to being on a specific page using regex [2]. This drastically reduces the text spans to investigate from full contracts to smaller text sections, which is an assumption used in section 3.

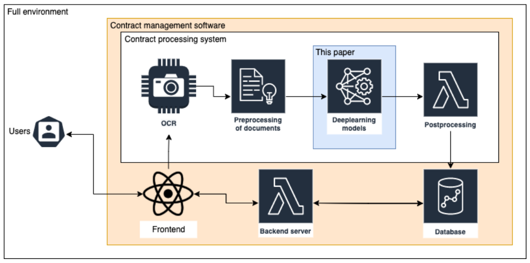

Fig. 1: Illustration of simplified full Contract Management Software system.

2. RELATED WORK

The task of machine comprehension is a key problem in Natural Language Processing and can be boiled down to reading and comprehending a text corpus to be able to answer questions about it. The first attempts at solving this problem were made in the ”Yale A.I Project” [3], but the major breakthroughs came with the publishing of large datasets (100k+ questions) such as the Stanford Question Answering Dataset [4] and the CNN/DailyMail dataset [5].

The DrQA architecture presented in [6] is an end-to-end system, both consisting of an information retrieval system for identifying correct paragraphs and a deep learning model for question answering. The deep learning model presented in the paper obtains decent results and is often used as a baseline model for Q&A[7]. The stanford attentive reader presented in [5] is very similar in architecture. Within the contract and legal domain [2] looked at extracting the contract elements outlined in this paper along with Governing Law, Jurisdic-

[1] Ideas supported by Account Director at ContractBook Christian Petersen

tion, Legislation Refs, and more. This paper uses a custom embedding from applying word2vec to an unlabelled dataset of 750.000 contracts combined with handcrafted features and POS. It uses a sliding window classifier along with manual post-processing rules. The performance on the task outlined in this paper is around 70% for the F1 score[2]. The first and second authors of [2] later published "A Deep Learning Approach to Contract Element Extraction"[8], where they revisited the problem working with a Bidirectional Long Short-Term memory approach. Some of the underlying ideas in that model are also present in the Bidirectional Attention Flow [9] model tested in this paper. On the task presented in this project, they perform a bit better with a F1 score in the range 70 to 90, but again on a dataset with features that are not publicly available. The biggest breakthrough in Q&A modelling has come with the emergence of transformer models which now outperform humans on the SQuAD dataset and dominate the leader boards[10]. Furthermore, the release of the expert annotated datasets such as Contract Understanding Atticus Dataset [11] has shown huge improvements within the contract processing field. During the research for this paper, access to the model and checkpoints used by the CUAD authors was obtained by the author of this paper. This model will be brought in for reference in section 5.

3. DATA OVERVIEW

The data used for this project consists of two different datasets. The SQuAD dataset is "...a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage ..." [10]. For the use in this modelling all questions that are unanswerable have been dropped. After processing the questions, SQuAD contains approximately 100.000 questions split into 80k for training and 20k for testing from 500+ Wikipedia articles[3]. The post-processed dataset has the structure that can be seen from appendix table 3.

The second dataset CUAD[11] is structured as folders of PDF files, text files created by OCR on the original PDFs, and two master files containing information on questions and answer-locations in the contracts. As presented in the introduction, this paper only looks at the questions related to Document Name, Agreement Date, Effective Date, Renewal Term, and Expiration Date. Prepossessing the CUAD dataset with these restrictions leaves 1764 questions[3] distributed close to evenly across the categories of the questions except for the "Renewal Term" as seen in appendix figure 6. These questions were processed into the same format as the SQuAD dataset, where the context is calculated as a 500-700 word span around the answer. The dataset has been split into 70% training and 30% validation data. More technical specifications of this preprocessing can be found in the source code of the article.

4. MODELLING

Question and answering modelling is a task of NLP that combines tasks of understanding intent, the importance of words, and being able to extract key phrases based on this understanding, all element key to the task at hand [4].

4.1. Long Short-Term Memory

Contrary to feed-forward neural networks, the Long Short-Term Memory model seen in appendix figure 7 maintains a state between inputs to support the processing of sequential data. Where the Recurrent Neural Network structure suffers from the vanishing gradient problem, due to gradients of a loss function decaying exponentially with time, the LSTM uses a more constant error flow by having a carefully regulated and gated cell state, making it a good candidate for this project.

4.2. DrQA / Stanford attentative reader

The deep learning model presented in [6] is based on a Stacked 3-layer bidirectional LSTM structure which uses the cells presented in figure7, but with the addition, that information can travel back and forth between cells. The 3 layers reefers the aforementioned BiLSTM's stacked on top of each other where the transitional hidden states are passed between layers.

The model can't interpret texts as is, thus the input sequence of context and question is transformed through an embedding layer. This embedding encodes every word into a 300-dimensional GLoVE vector. GLoVE "... is an unsupervised learning algorithm for obtaining vector representations for words" [12] and is based on a common web crawl of 840B tokens with a vocabulary of 2.2M words. Words in the input that is out of vocabulary will be represented by a zero-vector. GLoVE has the property that words that are similar in meaning have a dot-product close to one f.x. frog and toad[12].

First, the context is passed through the embedding layer. Then we perform an alignment embedding between the question and context by adding the f_align feature seen in appendix equation 2

Adding this feature encodes what segments of the context are more important with regards to the question on a token level and passes it to the model. This is utilising the similarity property of the GLoVE vector and adds "... soft alignments

[2] See section 4.5 for details on performance scores

[3] Examples can be found in appendix

[4] All model architectures visualisations can be found in the appendix of the report. The presentations of models in this section are high-level and the author refers to the source code and/or the original papers for intricate details

between similar but non-identical words ..." [6]. The original paper also includes a binary feature, POS, NER, and TF for the tokens which have been omitted from the model used in this paper. After the context and questions have passed through the word embedding and the BiLSTM they are passed through two separate bilinear attention layers, predicting the start and end logits of the answer as seen in appendix figure 3.

4.3. Bidirectional Attention Flow

The BiDAF model used in this paper is a simplified re-implementation of the one presented in [9]. This model, contrary to the DrQA model, utilises highway networks to eliminate fixed-size attention vectors. As with DrQA, the model represents the words by a GLoVE vector representation. Furthermore, it adds a character embedding to every word using a character-level Convolutional Neural Network on the characters of that word. The goal of this is to have a parameter of the model, which controls the CNN filters operating on the word vectors, where each filter should learn a specific feature, like curves and lines when applied in the computer vision domain.

The first three layers of the model add features to the context and question (handled separately). Afterward, they are passed through an attention layer where context-to-question attention is calculated "... (C2Q) attention signifies which query words are most relevant to each context word" [9] and question-to-context attention which "... signifies which context words have the closest similarity to one of the query words and are hence critical for answering ..." [9].

The output of the attention layer is then combined with the features calculated earlier and passed through BiLSTM. Finally, the information reaches the output layer which predicts the start and end logits of the answer constituting the answer span.

4.4. Transformers

The Transformer models included in this paper use encoder-decoder architecture[13] with a different attention mechanism compared to the previously presented models. The transformer models use a mechanism called self-attention, which allows them to learn connections such as how a question is phrased. The encoder part of the model uses multi-headed attention with 3 distinct fully-connected layers to create query, key, and value vectors as seen in appendix figure 4. The dot-product of these is the score matrix that determines the focus a given word should have[14]. This encoding and embedding will help the decoder part focus on the appropriate words in the decoding process. The core elements of the decoder are, as with the encoder, self-attention, attention on encodings, and a feed-forward neural network. Transformers do not process data sequentially, but take all the data in at once, thus requiring the positional argument to be embedded in the input sequence.

BERT
The Bidirectional Encoder Representations from Transformers model[14] uses a bidirectional training, which has shown better performance and understanding of language context, than models only trained on sequences from left to right or opposite. BERT is only a language model, constituting the encoder part of our model. To make predictions with the model a question answering head has been added on top of the hidden-states output, which computes the span of the answer. The model implemented in this paper is based on the Huggingface pre-trained model on extractive question-answering like in the SQuAD dataset and the model checkpoint from[11].

GPT-3
The GPT-3 model is the third generation of the GPT-n models, a series of autoregressive deep learning language models. It's trained on over 500 billion tokens and is created by researchers at the company OpenAI and is considered state-of-the-art within transformers. The interaction with this model is through an academic license to their rest API.

4.5. Performance Measures

The evaluation of performance on both the SQUAD dataset and the CUAD dataset is done using: Exact match and F1 score seen in appendix equation 1. Exact match is a binary value indicating if the prediction of the model spans the exact answer. In the F1 score, precision is the ratio between the number of shared words for the prediction and the answer, to the total number of words in the prediction. The recall is the ratio between shared words and the total number of words in the ground truth. This measure allows the model to be positively evaluated on close to prediction answers.

5. RESULTS AND DISCUSSION

The models' performance on the SQuAD validation dataset can be seen from table 1[5]. From the results, we see that our re-implementations perform reasonably well getting within a few percentage points of the original authors' performance. As the SQuAD leaderboards also have shown, transformer models perform very well [10]. As we are only investigating feasibility, the training has not included a lot of hyperparameter tuning, except getting the models to converge. Fairly large performance increases can often be obtained through proper tuning[7]. The results from GPT-3 are fairly poor compared to the other models, but the model is not trained on SQuAD and has not had any hyperparameters tuned. The model is included due to the controversy about it and for reference.

On the CUAD dataset, the DrQA model again has lower performance2 than BiDAF, but performs relatively better on this limited task compared to its performance on SQuAD.

[5] Experiment setup can be seen in appendix section

Name	# Param	EM	F1	Optimizer
DrQA	37M	55.8	67.6	Adamax
BiDAF	11.6 M	61.1	70.3	Adadelta
BERT [15]	109M	80.9	88.2	.
GPT-3 Curie	6.7B*	25.1	34.5	.

Table 1: Performance of models on SQuAD Dev dataset

Name	# Param	EM	F1	Optimizer
DrQA	37M	62.3	71.2	Adamax
BiDAF	11.6 M	70.8	80.5	Adadelta
BERT[15]	109M	2.9	16.3	.
RoBERTa-base[11]	100M	83.5	91.6	.
GPT-3 Curie	6.7B*	14.9	34.4	.

Table 2: Performance of models on the CUAD dataset. DrQA and BiDAF trained for 15 epochs with early stopping

The BERT model and GPT3 perform much worse with below 17 and 35 % respectively on the F1 score. The BiDAF model reaches a decent performance of 80.5% on the F1 score. The model created by the authors of the CUAD dataset[11] RoBERTa-base, has been added for reference and performs very well.

Experiments with pretraining the BiDAF models on the SQuAD dataset yielded marginal better performance and reduced convergence from approximately 13 epochs to 5. The trade-off of this approach was a larger space consumption due to the increased vocabulary size. This was tested to see if better performance could be achieved, using the same principles as transfer learning within image classification tasks seen with imagenet.

As on SQuAD, only a few experiments with hyperparameter tuning and regularization such as gradient clipping, optimizers, etc. have been done for DrQA and BiDAF, which strongly suggest that the performance of the models can be increased[7]. Due to the nature of the scope ”assessing feasibility” this was not pursued. It’s is not completely clear from the RoBERTa-base model what training/validation split has been used in the model, thus performance on the dataset might appear higher than what it is, due to the validation data being of training data for that model. That being said, the transformer models are currently state of the art[7] and they can reasonably be expected to perform better than models like BiDAF [7]. This brings the expected performance of a properly trained transformer into the same range as seen in table 2.

Looking at the results from a product standpoint, keeping in mind the potential for performance increases, the F1 performance of 90%, which was seen from the RoBERTa-base model, is evidence to suggest that a full contract system as presented in figure 1, can be built that could work well on the limited task outlined in this paper. If the system was built in a way where the user’s input could be used as a feedback mechanism, a much larger dataset could be obtained which has shown to be a major factor in NLP performance[7].

For the contract element extraction problem presented in this paper one could argue for the use of regex due to the relatively simple task. A full-scale system should be able to expand into the prediction of laws and clauses and here earlier research[2] has shown regex does not obtain good results. Furthermore, the modelling choice as a single model for a more general question answering is chosen so a potential application will be more versatile to the changes in needs of its users and because they have already shown good results being implemented within the Google Knowledge Graph, IBM Watson, etc. In an application, it would likely be interesting to produce highly specialised models for reoccurring tasks such as identifying the document type, but relating this to the introduction, one of the potential implementations of this could be an intelligent search engine for legal documents. A new verdict in a case can suddenly require a review of specific clauses. A general Q&A model is much more interesting in this case, as they presumably will be more likely to work on this new task.

6. CONCLUSION AND FUTURE WORK

This paper has presented methods for formulating the contract element extraction problem as a Question and Answering problem and presented deep learning models in form of the BiDAF and DrQA, along with state-of-the-art transformers, for solving it. It has successfully shown that simple contract element extraction tasks can achieve a high accuracy rate of 80-90% with room for improvement. The presented results have also shown that a further investigation of the full contract management or searching system is reasonable.

The next steps in a further investigation of this could be building the full pipeline of the contract processing system presented in figure 1. That is a pipeline that can take a legal document as input, process the document and extract these elements and convey them in a meaningful way. Looking at this from the perspective of an intelligent search engine, an obvious question to ask is how general can the questions be phrased? These models have been trained on 5 different questions. What if the question in appendix section 8.2 was phrased as ”What type of contract is this?”.

The performance measure of the models could also be rewritten to better reflect a real use case. If implemented in an application, the model should likely suggest multiple answers when unsure and let the user pick the correct one, thus a performance using top x predictions weighted with their confidence could be an idea. Furthermore, outputting actual text from the models and accounting for variances f.x. in dates 01/12/2021, 1-12-21, 1 December after document renewal date might also make the model able to model perform better along with working for yes and no answers, which is currently not feasible.

7. REFERENCES

[1] "Unstructured data," https://www.mongodb.com/unstructured-data, Accessed: 2021-11-24.

[2] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos, "Extracting contract elements," in *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, New York, NY, USA, 2017, ICAIL '17, p. 19–28, Association for Computing Machinery.

[3] S. Slade, "The yale artificial intelligence project: A brief history," *AI Mag.*, vol. 8, pp. 67–76, 1987.

[4] ," .

[5] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching machines to read and comprehend," *CoRR*, vol. abs/1506.03340, 2015.

[6] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes, "Reading wikipedia to answer open-domain questions," *CoRR*, vol. abs/1704.00051, 2017.

[7] Christopher Manning, "Stanford university, cs224 - nlp with deep learning," https://www.youtube.com/watch?v=yIdF-17HwSk&ab_channel=stanfordonline, Accessed: 2021-10-03.

[8] Ilias Chalkidis and Ion Androutsopoulos, "A deep learning approach to contract element extraction," in *JURIX*, 2017.

[9] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi, "Bidirectional attention flow for machine comprehension," *CoRR*, vol. abs/1611.01603, 2016.

[10] "Squad leaderboard," https://rajpurkar.github.io/SQuAD-explorer/, Accessed: 2021-11-27.

[11] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball, "CUAD: an expert-annotated NLP dataset for legal contract review," *CoRR*, vol. abs/2103.06268, 2021.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

The source code can be found at https://github.com/gustavhartz/legal-contract-elements.

8. APPENDIX

8.1. Figures

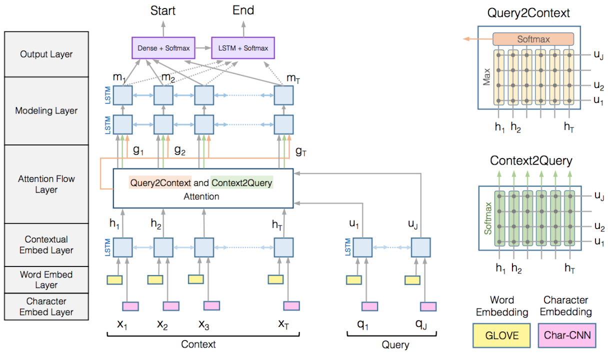

Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

Fig. 2: Bidaf model architecture as presented in the original paper [9]

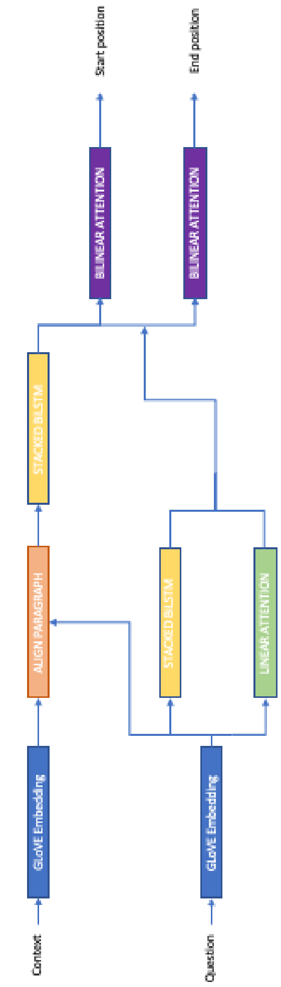

Fig. 3: Simplified DrQA model architecture

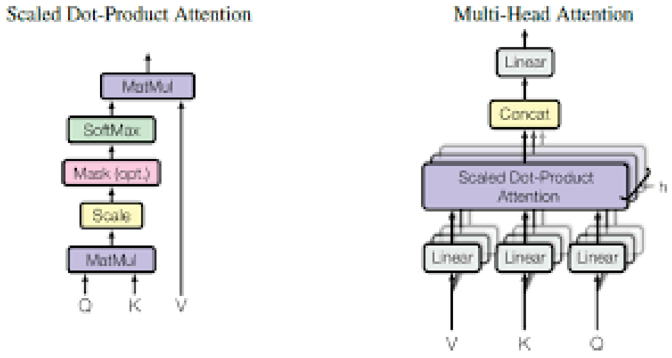

Fig. 4: Transformer attention architecture as presented in [13]

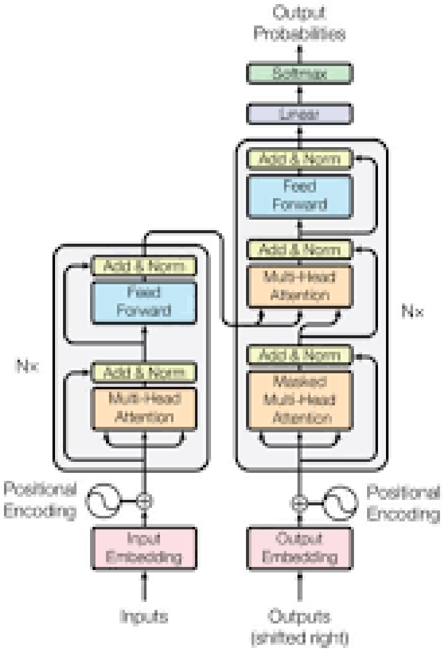

Fig. 5: Transformer architecture as presented in [13]

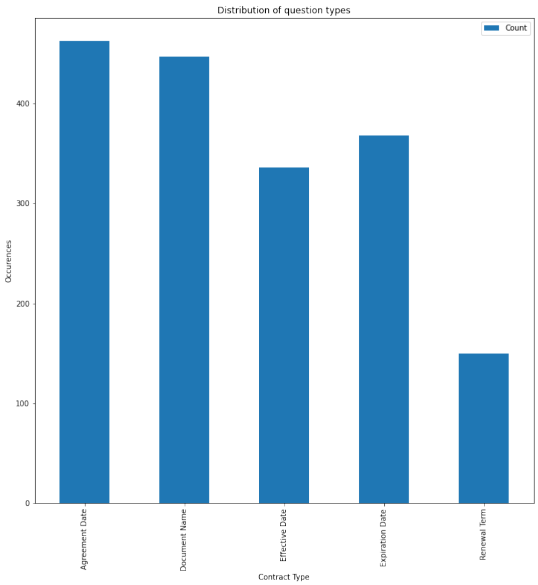

Fig. 6: Distribution of contract questions

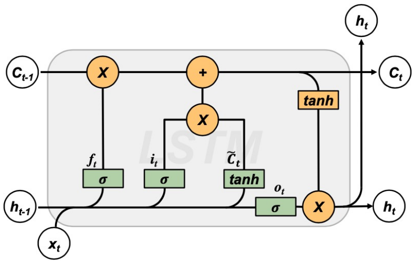

Fig. 7: LSTM cell. Square = layer, white circle = state, orange circle = arithmetic operation

8.2. CUAD example

8.2.1. Example 1

Question: Highlight the parts (if any) of this contract related to ”Document Name” that should be reviewed by a lawyer. Details: The name of the contract?

Context: XHIBIT 10.6DISTRIBUTOR AGREEMENT-THIS DISTRIBUTOR AGREEMENT (the ”Agreement”) is made by and between Electric City Corp., a Delaware corporation (”Company”) and Electric City of Illinois LLC (”Distributor”) this 7th day of September, 1999.RECITALSA. The Companyś Business. The Company is presently engaged in the business of selling an energy efficiency device, which is referred to as an ”Energy Saver” which may be improved or otherwise changed f

Answer: DISTRIBUTOR AGREEMENT

8.2.2. Example 2

Question: Highlight the parts (if any) of this contract related to ”Effective Date” that should be reviewed by a lawyer. Details: The date when the contract is effective

Context: receipt and sufficiency of which are hereby acknowledged, the parties agree as follows:1. Definitions. As used herein, the following terms shall be defined as set forth below:a. ”Contract Period” shall mean that period of time from February 21, 2011 through December 31, 2012.b. ”Contract Year” shall mean the specific period of time during the Contract Period as more specifically set forth below: · Contract Year 2011 (2/21/11 - 12/31/11) · Contract Year 2012 (1/1/12- 12/31/12)c. ”Contract Territory” shall mean the world.d. ”Northś Likeness” shall mean and include Northś name, image, photograph, voice, initials, signature, biographical information, and persona.f. ”Northś Endorsement” shall mean and include Northś public statements and commen

Answer: ”Contract Period” shall mean that period of time from February 21, 2011 through December 31, 2012

8.3. SQuAD example

8.3.1. Example 1

Question: In what country is Normandy located?

Context: The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse (”Norman” comes from ”Norseman”) raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.
Answer: France

8.4. Experiment Setup

All deep learning experiments have been performed in Azure ML Studio with a Standard_NC6s_v3 instance 6 cores, 112 GB RAM, 336 GB disk, and a single NVIDIA Tesla V100 GPU.

8.5. Data

`id`	string	ID of datapoint
`padded_Context`	List[int]	Question word IDs
`padded_Question`	List[int]	Context word IDs
`context_Mask`	List[bin]	Padding location
`question_Mask`	List[bin]	Padding location
`label`	(int,int)	Location of ans in context
`context_Char`	List[int]	Character level IDs
`question_Char`	List[int]	Character level IDs

Table 3: Dataset format. The context and questions words and characters have been converted to integers keeping the mapping in dictionaries. Both context and question are padded for keeping fixed size in a give batch

8.6. Equations

$$em(q,p) = \begin{cases} 1, & \text{if } q = p \\ 0, & \text{otherwise} \end{cases}, F1 = 2 \cdot \frac{(\text{prec} \cdot \text{rec})}{\text{prec} + \text{rec}} \qquad (1)$$

$$f_{align} = \sum_{j} a_{i,j} E(q_j), \text{ where, } E = \text{embedding vector and}$$

$$a_{i,j} = \frac{exp(\alpha(E(p_i)) \cdot \alpha(E(q_j)))}{\sum_{j'} exp(\alpha(E(p_i)) \cdot \alpha(E(q_{j'})))} \tag{2}$$