A Sequence Modeling Approach for Structured Data Extraction from Unstructured Text

Jayati Deshmukh[*]
IIIT-Bangalore
jayati.deshmukh
@iiitb.org

Annervaz KM
Accenture Technology Labs
annervaz.k.m
@accenture.com

Shubhashis Sengupta
Accenture Technology Labs
shubhashis.sengupta
@accenture.com

Abstract

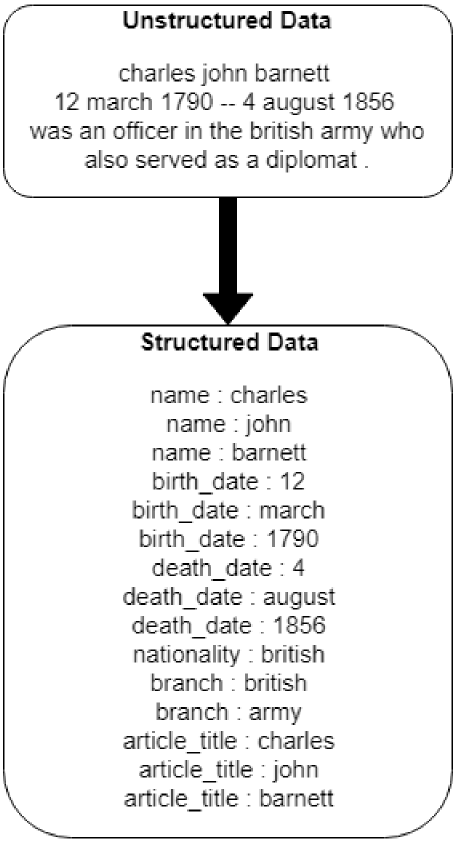
Extraction of structured information from unstructured text has always been a problem of interest for NLP community. Structured data is concise to store, search and retrieve; and it facilitates easier human & machine consumption. Traditionally, structured data extraction from text has been done by using various parsing methodologies, applying domain specific rules and heuristics. In this work, we leverage the developments in the space of sequence modeling for the problem of structured data extraction. Initially, we posed the problem as a machine translation problem and used the state-of-the-art machine translation model. Based on these initial results, we changed the approach to a sequence tagging one. We propose an extension of one of the attractive models for sequence tagging tailored and effective to our problem. This gave 4.4% improvement over the vanilla sequence tagging model. We also propose another variant of the sequence tagging model which can handle multiple labels of words. Experiments have been performed on Wikipedia Infobox Dataset of biographies and results are presented for both single and multi-label models. These models indicate an effective alternate deep learning technique based methods to extract structured data from raw text.

1 Introduction

A humongous volume of data in the form of text, images, audio and video is being generated daily. It has been reported that 90% of all the data available today has been generated in the last two years (DoMo, 2017). The pace of data generation is growing exponentially. The generation of data is not only restricted to open domains and social media; even in closed groups like private organizations and corporations, textual data is being produced in abundance. Unstructured data is present in a variety of forms like documents, reports and surveys, logs etc. Restricting this data to be captured directly in structured form prohibits the natural capturing of the data, leaving out essential pieces. But structured data presents the data in a concise and well-defined manner which is easier to understand than a corresponding document. Structured data can be transformed into tables which can be easily stored in databases. It can be indexed, queried for and searched to retrieve relevant results of a query. Thus structured representation is quintessential to facilitate machine consumption of data. Moreover in the world of data abundance, such structured representation is essential for human consumption as well.

In many business processes, like Finance and Healthcare, the transformation of the unstructured data into structured form is done manually or semi-automatically through domain specific rules and heuristics. Let's take the example of *Pharmacovigilance* (Maitra et al., 2014), where adverse effects of prescribed drugs are reported by patients or medical practitioners. This information is used to detect signals of adverse effects. Collection, analysis and reporting of these adverse effects by the drug companies is mandated by law. In most cases, it is easy for patients or medical practitioners to describe the side-effects of their drugs in a common, day to day language, in free form text. Then it has to be transformed into a structured format which is analyzed with clinical knowledge for signals of adverse effects. Currently this is done by human analysts or through very rigid text processing heuristics for certain kinds of text. Another domain is extraction and management of legal contracts in domains such as real estate. Specifically *Lease Abstraction* involves manual inspection and validation of commercial rental lease documents. It is done by offshore experts who extract relevant information

[*] Work done at Accenture Technology Labs



Unstructured Data

charles john barnett
12 march 1790 -- 4 august 1856
was an officer in the british army who
also served as a diplomat .

Structured Data

name : charles
name : john
name : barnett
birth_date : 12
birth_date : march
birth_date : 1790
death_date : 4
death_date : august
death_date : 1856
nationality : british
branch : british
branch : army
article_title : charles
article_title : john
article_title : barnett

Figure 1: From Unstructured to Structured Data

from the documents into a structured form. This structured information is further used for aggregate analytics and decision making by large real estate firms (Annervaz et al., 2015).

Figure 1 shows a sample unstructured text and its corresponding structured output. In case the structured data is to be stored in a database, the labels like *name*, *birth_date* etc can be the column names of the database and the corresponding values like *charles* and *12 march 1970* can be the actual values stored.

As mentioned earlier, previous work in this space involved parsing the natural language sentences, and writing rules and heuristics on the parse tree or structure to extract the information required (Culotta and Sorensen, 2004; Fundel et al., 2006; Reichartz et al., 2009). In this work, we approach the problem from sequence modeling perspective and weave together state of the art models in the space to extract structured information from raw unstructured data. The task of information extraction to build structured data can be described as generating or matching appropriate tags or labels to corresponding parts of raw data. For each token in the raw data, a corresponding tag is attached marking what kind of data it stands for. *OTHER* tag gets attached if the data in the raw text is not relevant.

We have approached the problem both from machine translation and sequence tagging perspectives. In machine translation, typically there are two sequences, one in source language (say English) and the other in target language (say French). Machine learning models try to convert the sequence of tokens in source language to sequence of tokens in target language. In case of translation problem, the core idea being expressed in the input and the output is the same, however it is in a different language. Similarly, for our problem both the input and output have same content although it is represented in an unstructured or structured format. So to start with, we approach the problem from translation perspective and treat the source text as word sequence of unstructured text and the corresponding tag sequence as target sequence. State of the art machine translation models (sequence to sequence model (Cho et al., 2014; Sutskever et al., 2014)) can then be attempted for the same. We have experimented with this approach and treat it as our baseline. We didn't find any previous work on this dataset for structured data extraction task. However we realized that this problem cannot be directly mapped to a translation problem. There is significantly more word or phrase level information in the input which cannot be appropriately represented by translation models. We realized that sequence tagging models (like for POS tagging (Huang et al., 2015)) are more suitable for this problem. We have experimented with state of the art sequence tagging model for the problem and propose some problem specific variants to improve the performance.

Main contributions of this work are as follows:

1. We approach the problem from sequence modeling perspective, which is perhaps the first attempt in literature to the best of our knowledge. Modeled this way, we can eliminate usage of traditional ways for parsing or writing domain specific rules. A parallel corpus of unstructured and structured data is sufficient to train the models.

2. We have designed a modified version of the state of the art sequence tagging model along with PoS tags and attention which further im-

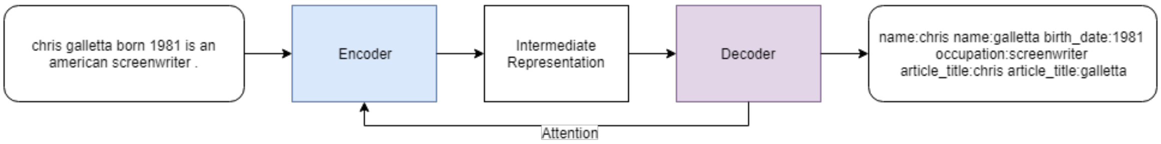

chris galletta born 1981 is an american screenwriter .
Encoder
Intermediate Representation
Decoder
name:chris name:galletta birth_date:1981 occupation:screenwriter article_title:chris article_title:galletta
Attention

Figure 2: Seq2seq Example

proves the results compared to the vanilla sequence tagging model.

3. We have also designed a multi-label sequence tagging model which can generate multiple labels of words using a customized learning loss based on Set similarity.

The paper is organized as follows: We present the details of seq2seq and sequence tagging models in Section 2 along with some interesting work which has been done previously using these models. In Section 3 we give details of our approach along with details of vanilla model and modified models for single and multi-label problems. The details of our experiments are in Section 4. We present some related work in Section 5. We conclude in Section 6 by giving some future work directions.

2 Preliminaries

A variety of NLP problems have been solved using both seq2seq models and sequence tagging models. These models and their variants have produced state-of-the-art results and we discuss these models and some of their applications in the following subsection.

2.1 Seq2Seq Models

Seq2seq models are end to end models which transform an input sequence into an output sequence. These models basically consist of an encoder which takes the input and encodes it into an intermediate representation and a decoder which takes the intermediate representation as input and generates the output sequence, one token at a time. Encoders and Decoders structurally may be Recurrent Neural Networks like RNN, LSTM, GRU (Cho et al., 2014; Sutskever et al., 2014)) or even Convolutional Neural Networks (Gehring et al., 2017), depending on the problem it is designed to solve. It might also have different variations and additional features like attention, multiple layers etc (Bahdanau et al., 2014). These were the first citizens of *Encode, Attend, Decode* paradigm deep learning models.

Seq2seq models generate output in two steps as shown in Figure 2. Firstly x, the sequence of embeddings which is created by combining the embeddings (vectors) of words, is given as input to the encoder. The encoder transforms x into an intermediate representation z (which for example for RNN encoder may be the hidden state at the end of processing input) as follows

$$z = enc(x)$$

Next, this representation is given as input to the decoder. It generates the output Y token by token as $w_0, w_1, w_2, ..., w_l$ from z as per the following equations:

$$h_t = dec(h_{t-1}, w_t)$$
$$s_t = g(h_t)$$
$$p_t = softmax(s_t)$$

where, at $t = 1$

$$h_0 = z$$
$$w_0 = w_{sos}$$

Here, at $t = 1$, h_0 is the output of encoder z and w_0 is the embedding of *start of sentence* tag. The decoder takes the previous hidden state and current word embedding as input to generate the next hidden state. Next, function g transforms the hidden representation from hidden dimension h to the dimension of vocabulary v. Next its output is passed through a softmax function which transforms the input into probability values for each word in the vocabulary. Finally, it is passed through argmax function to fetch the index of the word of maximum probability and returns the corresponding word. This process is repeated till *end of sentence* tag is generated by the decoder.

Originally, seq2seq models were conceived for language translation task(Cho et al., 2014; Sutskever et al., 2014)), where the input text is in one language like English and the output which is its translation, is in another language like French.

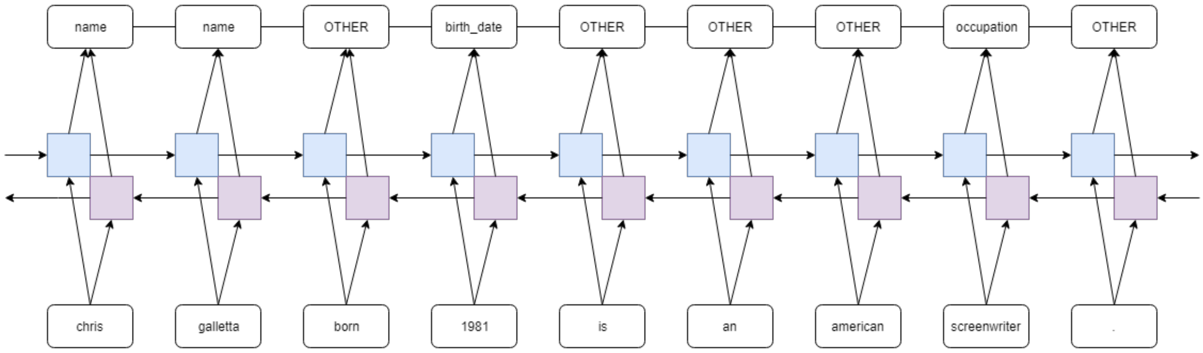


Figure 3: Sequence Tagging Example

These simple models generated encouraging results leading to production grade models which parallelize training on multiple GPUs and are used in applications such as Google Translate (Wu et al., 2016). Seq2seq models have also been used for Text Summarization. In this case, the input is a large document and output is its summary which can be used for generating news headlines or abstracts. For this purpose, an attention based encoder and a beam search decoder which generated words from the vocabulary gave the best results (Rush et al., 2015; Wu et al., 2016).

Seq2seq models are not just restricted to textual input/outputs. There have been applications with image inputs to automatically generate captions of images using CNN encoder and RNN decoder with attention (Vinyals et al., 2015). These models have also been used on speech data to transform speech to text (Chorowski et al., 2015). They have also been used with videos for video translation, subtext generation and video generation etc (Venugopalan et al., 2014; Srivastava et al., 2015; Yao et al., 2015). Some multi-modal models which take more that one forms of inputs have also been successful (Kiros et al., 2014).

2.2 Sequence Tagging Models

Sequence tagging or labeling models tag all the tokens in the input sequence. Fundamentally, this model consists of recurrent neural network like RNN, LSTM, GRU and Convolutional Neural Network which reads input at token level and a conditional random field (CRF) (Lafferty et al., 2001) which takes as input the encoded representation and generates corresponding tags for each token. These models may also include other additional features like word and sentence features, regularization, attention etc. Originally this model was conceived for tasks like Part of Speech (PoS) tagging, chunking and Named Entity Recognition (NER) (Huang et al., 2015). A joint model for multiple tasks also seems to work well (Hashimoto et al., 2016).

A high level representation of sequence tagging model as shown is Figure 3. Here the input is passed into an bi-LSTM and the hidden vector $h(t)$ and output vector $y(t)$ are generated as follows:

$$h_f(t) = f(U_f x(t) + W_f h_f(t-1))$$

$$h_b(t) = f(U_b x(t) + W_b h_b(t-1))$$

$$h(t) = [h_f(t) : h_b(t)]$$

$$y(t) = g(Vh(t))$$

where $h_f(t)$ and $h_b(t)$ are the hidden representations of the forward and backward LSTMs respectively. These two are concatenated to generate the final hidden representation $h(t)$. U_f, W_f, U_b, W_b, V are weights computed during training. These bi-LSTM representations are combined with CRF using Viterbi Decoder (Sutton et al., 2012). It takes the hidden state and the previously generated tags in the form of sequence to generate the next tag. If the string of output tags is taken as a sequence, then we can say that the CRF generates the most likely sequence out of all possible output sequences (Huang et al., 2015; Ma and Hovy, 2016; Lample et al., 2016)

3 Approach and Models

For our problem, we started with seq2seq models. We then moved to vanilla sequence tagging models which we realized are more suitable for the task as compared to seq2seq models. We also built a variant of sequence tagging model suitable for our problem which further improves the performance.

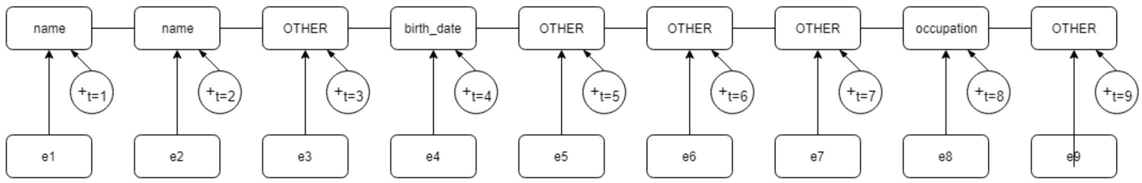


Figure 4: Sequence Tagging with Attention

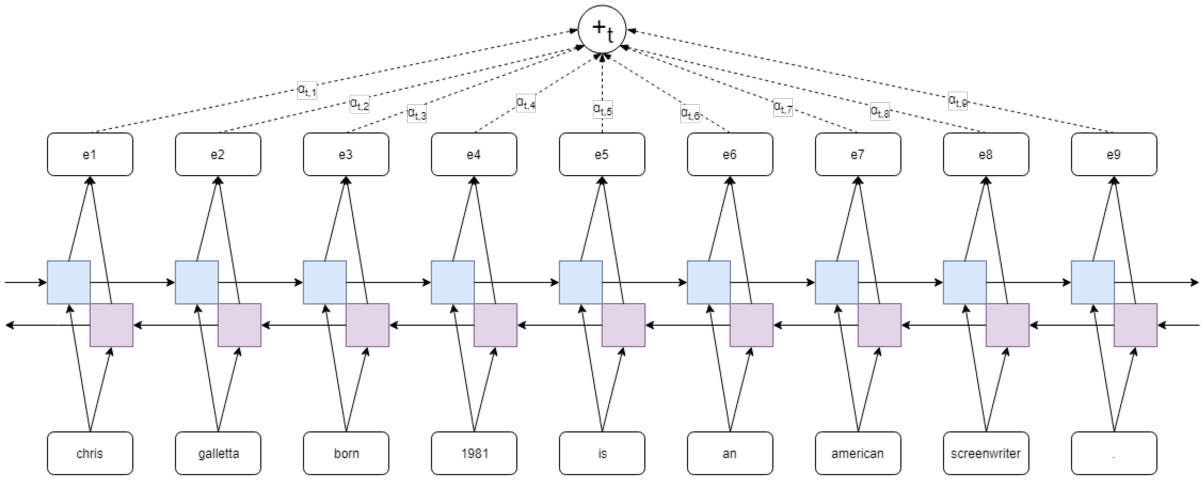


Figure 5: Computing Attention

Finally, a sequence tagging model which can generate multiple labels for each token has also been designed. Further details of the models are presented in the following subsections.

3.1 Baseline Approach : Seq2seq model

For seq2seq model, the input is the sentence or set of sentences and the structured output is transformed to a string which is a series of key-value pairs corresponding to the word-label pairs of the sentence. Here we assume that the tags are at a word level. This model can learn multiple labels of the same word for example *chris* is *name* as well as *article_title* in Figure 2. This model by its design also learns the sequence of label-word pairs. Experiments as detailed in Section 4 have been performed on different combinations of RNN and CNN encoders and decoders.

3.2 Vanilla Sequence Tagging model

Sequence tagging model reads the input word by word and simultaneously generates the corresponding label for the word as shown in Figure 3. Here, blue cells represent the forward LSTM and the violet cells represent the backward LSTM. The line between the output labels represents the CRF. For this model, the data is transformed such that the sentence is split in words by spaces and then each word is tagged to a corresponding label. Only the first label of a word is considered. If a word does not have any label then it is manually labeled as 'OTHER'. Structurally this biLSTM-CRF model can comparatively work better even in case of longer inputs. However, this model cannot learn multiple labels of a word. The model generates labels at a word level and thus it does not have an ordering as in case of seq2seq models.

3.3 Modified Sequence Tagging model

We have modified the vanilla sequence tagging model to incorporate following variations which models our problem better and was found to generate improved results. Part of Speech (PoS) tags of words carry rich information and are connected to the corresponding labels of each word. To utilize this, we used the word itself and the PoS tag of each word as input. We randomly initialized the PoS tag embeddings. Word embeddings and PoS tag embeddings were concatenated and passed as input to the bi-LSTM. We believe that while generating label for the current word, not all the words of the input are equally important. Words nearby to the current word are contextually more important compared to words farther away. Thus, every word has different importance or weight while generating the label of cur-

Table 1: Sample Results

Table 2: Single Label Results

Sentence 1	Label 1	Sentence 2	Label 2
w.	name	renan	name
lamont	name	luce	name
was	OTHER	born	OTHER
a	OTHER	5	birth_date
scottish	OTHER	march	birth_date
footballer	OTHER	1980	birth_date
who	OTHER	,	birth_place
played	OTHER	paris	birth_place
as	OTHER	is	OTHER
a	OTHER	a	OTHER
right	position	french	OTHER
winger	position	singer	occupation
.	OTHER	and	OTHER
		songwriter	occupation
		.	OTHER

Table 3: Multi-Label Results

Word	Labels
begziin	article_title name
yavuukhulan	article_title image name
,	OTHER
1929-1982	OTHER
was	OTHER
a	OTHER
mongolian	nationality language
poet	occupation
of	OTHER
the	OTHER
communist	OTHER
era	OTHER
that	OTHER
wrote	OTHER
in	caption
mongolian	nationality language
and	OTHER
russian	language
.	OTHER

rent word. To incorporate this in the model, we used self-attention (Vaswani et al., 2017; Tan et al., 2018) as depicted in Figure 4 and 5.

3.4 Multi-label Sequence Tagging model

As shown in Figure 1 a word can have multiple associated tags / labels. Vanilla sequence tagging models are designed to predict only a single tag for each word. Thus a lot of information might be lost by using these models. The following modified model can give multiple possible labels of words. At the output layer, instead of using softmax which was used in single label prediction case, we use sigmoid which normalizes each of the label prediction scores between 0 and 1 independently. We used hamming loss, which is the most common metric used in case of multi-label classification problems (Tsoumakas and Vlahavas, 2007; Elisseeff and Weston, 2002). Hamming loss is defined as the fraction of wrong labels to total number of labels. It takes into account both correct and incorrect labels. Let y_t be the vector of true labels and y_p be the vector of independent probabilities of predicted labels. Then Hamming Loss (HL) is computed as follows:

$$HL = y_t \; XOR \; y_p$$

Here, XOR is non-differentiable and cannot be used to train the multi-label sequence tagging model. To overcome this problem, the HL equation is transformed as below:

$$HL_{diff} = average(y_t * (1 - y_p) + (1 - y_t) * y_p)$$

For example, let a word have true labels as $[1, 0, 0, 1]$ and the model predicts the labels $[0.9, 0.1, 0.2, 0.9]$, then hamming loss in this case is computed as $avg([1, 0, 0, 1] * [0.1, 0.9, 0.8, 0.1] + [0, 1, 1, 0] * [0.9, 0.1, 0.2, 0.9])$ or $avg(0.1 + 0.1 + 0.1 + 0.2)$ or 0.125.

4 Experiments & Results

We have used the Wikipedia Infobox dataset created by (Lebret et al., 2016) which is available in the public domain[1]. It consists of total 728, 321 biographies, each having the first Wikipedia paragraph and the corresponding infobox, both of which have been tokenized. Originally this dataset

[1] https://github.com/DavidGrangier/wikipedia-biography-dataset

Table 4: Baseline Results - Seq2Seq Model

Model	**Accuracy %**	**Perplexity**
CNN Encoder Decoder	63.34	5.78
LSTM Encoder Decoder	68.42	3.95
LSTM Encoder Decoder with PoS	69.60	3.45

Table 5: Sequence Tagging Results

Model	**Accuracy %**	**F1 Score %**
biLSTM-CRF	79.34	65.00
biLSTM-CRF with PoS & Attention	82.82	62.32

was created to build models to generate text based on the the infobox. In our case, the problem is reversed. Given a paragraph of unstructured data, we try to generate the corresponding infobox or structured data. In the dataset, some information might be present in the paragraph but not in infobox and vice versa. We have pruned the infoboxes so that it contains only that information which is present in the paragraph. The information which is not present in the paragraph cannot be generated by any model by itself without external knowledge.

We have split the dataset into three parts in the ratio 8:1:1 for train, validation and test. We have done basic pre-processing on both paragraphs and infoboxes. Extra information and labels tagged as none have been removed from infoboxes. The words have been initialized to GloVe (Pennington et al., 2014) embeddings and character embeddings (Santos and Zadrozny, 2014) have been randomly initialized. Words are 300 dimensional and characters are 100 dimensional. The models have been trained for 15 epochs or until it showed no improvement. Single label model has been trained using Adam Optimizer (Kingma and Ba, 2014) and multi-label model using Adagrad Optimizer (Zou and Shen, 2018). Adaptive learning rate has been used. Dropouts (Guo et al., 2016) have been used as regularizer. Table 1 shows some sample results of single and multi-label sequence tagging models.

Table 4 shows the Accuracy and Perplexity scores of the baseline approach using seq2seq model. Here, accuracy is calculated as total number of correctly predicted words by total number of words. Perplexity metric is from NLP models and it represents probability distribution of a language model over the text[2]. Lower perplexity represents better generalization and thus better performance. We observed that LSTM Encoder-Decoder performs better than CNN Encoder-Decoder as it is able to take the temporal order or words into account and also because it handles short / medium length text well. We also gave sequence of words and corresponding PoS tags as input and the results of this were the best among all the seq2seq models. Despite these enhancements, this model does not perform well and has a low accuracy.

Table 5 shows the Sequence Tagging results on the same data using vanilla model and other model variants described earlier. In this case, accuracy metric is computed as number of labels correctly predicted by total number of words and F1 score is calculated as usual as the harmonic mean of precision and recall. We present the results of vanilla model and sequence tagging model with improvements like PoS tags and attention. We notice that the results of sequence tagging models are significantly better than the seq2seq models. In multi-label sequence tagging model, the hamming loss on the test dataset was 0.1927.

5 Related Work

Traditionally relationships have been extracted from raw text using dependency parse tree based methods (Culotta and Sorensen, 2004; Reichartz et al., 2009). Dependency parse tree shows the grammatical dependency among the words or phrases of the input sentence. To extract relation among words from a dependency parse tree, classifiers are trained to classify the relation. Sometimes rules are applied on on dependency parse

[2] http://www.cs.virginia.edu/~kc2wc/teaching/NLP16/slides/04-LMeval.pdf

trees to further improve the results (Fundel et al., 2006; Atzmueller et al., 2008). These rule based models have shown improved results and have been used in medical domain. It might also fare well in closed domain areas where there is less variation in text. Even at a Web scale, there have been efforts to extract information specifically in the form of named entities and relationships using DBpedia spotlight (Mendes et al., 2011) and OpenIE (Pasca et al., 2006). A joint entity and relation extraction model (Miwa and Bansal, 2016) is primarily built using LSTMs. It comprises two LSTM models - word sequence LSTM predicts the entities and dependency tree LSTM predicts the relationships among the entities. They also use additional features like PoS tags, dependency types etc as input. However in our models, we label the words of raw text, these labels are not categorized into entities and relationships. The datasets on which they have performed the experiments contain very few (< 10) entities and relations as compared to our labels (1000). Attention based encoder-decoder model (Dong and Lapata, 2016) has been used to convert raw text to logical format. The output is not entity or relationship but a logical string corresponding to the input. They show that this model gives consistent results across different domains and logical formats. The seq2seq model which we used as a baseline is similar to this model.

6 Conclusion & Future Work

We proposed a deep learning based approach for the age old NLP problem of information extraction. We have used multiple variants of deep learning based sequence tagging models to extract structured data from unstructured data. Large publicly available dataset of Wikipedia Biographies has been used in experiments to prove the efficacy. Sequence tagging models further improved with additional features like PoS tags and attention mechanism. Multi-label sequence tagging model gave more complete results from practical perspective. Unlike the traditional methods, our models are generic and not dependent on the structure of Wikipedia Infobox dataset. Similarly, it is also not dependent on English language specifically. Ideally, it should work well for other similar languages or datasets. A parallel corpus of unstructured data and its corresponding structured data is all that is required to train these models. The actual performance might be affected by language specific issues like word order, double negation or other grammatical issues. And there might be minor modifications needed specific for different datasets or languages.

To the best of our knowledge, this is the first attempt in using sequence models for structured data extraction. Being an initial work, there are plethora of possible future work extensions. In the practical setting, the information to be extracted tends to be hierarchical. So the tags have a hierarchical structure to it. Current model proposed, handles only flat tag structure. Alterations to incorporate and handle hierarchical tag structure is one direction of work we are considering. In the Wikipedia Infobox dataset the text from where the structured information is extracted is already identified or don't have large span. In practice, this is not the case. The text usually have larger span, this makes the problem tougher. We have to devise models first to prioritize the text snippets from where the information has to be extracted, such an end-to-end trainable model is another direction of work. Similarly there are lot of options for future work, we hope our initial work and results will inspire the community to work in these directions.

References

K. M. Annervaz, Jovin George, and Shubhashis Sengupta. 2015. A generic platform to automate legal knowledge work process using machine learning. In *14th IEEE International Conference on Machine Learning and Applications, ICMLA 2015*, pages 396–401.

Martin Atzmueller, Peter Kluegl, and Frank Puppe. 2008. Rule-based information extraction for structured data acquisition using textmarker. In *LWA*, pages 1–7.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 423. Association for Computational Linguistics.

DoMo. 2017. Data Never Sleeps 5.0.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*.

André Elisseeff and Jason Weston. 2002. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. Relexrelation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. 2016. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Anutosh Maitra, K. M. Annervaz, Tom Geo Jain, Madhura Shivaram, and Shubhashis Sengupta. 2014. A novel text analysis platform for pharmacovigilance of clinical drugs. In *Proceedings of the Complex Adaptive Systems 2014 Conference - Conquering Complexity: Challenges and Opportunities*, pages 322–327.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2009. Dependency tree kernels for relation extraction from natural language text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 270–285. Springer.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818–1826.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *European conference on machine learning*, pages 406–417. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515.

Fangyu Zou and Li Shen. 2018. On the convergence of adagrad with momentum for training deep neural networks. *arXiv preprint arXiv:1808.03408*.