DEXTER - Data EXTraction & Entity Recognition for Low Resource Datasets

Nihal V. Nayak, Pratheek Mahishi, Sagar M. Rao
Stride.AI, Bengaluru
{nihal.nayak, pratheek, sagar}@stride.ai

Abstract

Extraction of key information such as named entities, key phrases, and numbers is critical for several banking and financial processes. Banks and Financial Institutions resort to the use of automation tools to reduce the human effort required for these processes. Training a system to extract key datapoints reliably and efficiently from text requires large labeled datasets. However, openly available datasets in the financial sector have limited labeled data. In our paper, we address the issues in developing a data extraction system for low resource datasets. We experiment with a Bi-directional long short-term memory (Bi-LSTM) model which works well on low resource datasets. We introduce a novel domain-specific Bi-LSTM layer, which allows us to add domain-specific knowledge into the neural architecture. We observed that transfer learning from out-of-domain dataset boosts the accuracy on several extraction tasks. We create three new low resource financial datasets and demonstrate that our model consistently achieves a high degree of accuracy on these datasets. Furthermore, our model outperforms the reported state of the art results on the Financial NER dataset and achieves F1 of 87.48. Our experiments consistently show that transfer learning combined with domain-specific knowledge engineering improves entity recognition in a low resource setting.

Introduction

Financial Institutions deal with a large number of documents in the form of contracts, reports, application forms etc. These documents are highly unstructured and textual in nature. Processing such documents involve the extraction of key information (entities, contract clauses, key phrases, numbers, etc.). Traditionally, companies have relied on domain experts to capture this information which is time-consuming. However, recent trends suggest that specialized tools and algorithms are being used to extract key data points from documents to augment and reduce human effort.

Building a system to extract datapoints from unstructured text documents poses several challenges, especially in the financial domain. First, the style of writing varies significantly when compared to news articles, blogs, etc. as "domain specific" lexicons and jargon are used extensively. Secondly, development of any kind of dataset for financial text requires domain experts to label the data. The process of annotation is expensive and cumbersome. Lastly, Financial Institutions are hesitant to share their data as it raises several privacy concerns. Therefore, these constraints curtail the research in the field.

The following sentence is extracted from a financial document -

> *This LOAN AGREEMENT, dated as of November 17, 2014 (this Agreement), is made by and among Auxilium Pharmaceuticals, Inc., a corporation incorporated under the laws of the State of Delaware (U.S. Borrower), Auxilium UK LTD, a private company limited by shares registered in England and Wales (UK Borrower and, collectively with the U.S. Borrower, the Borrowers) and Endo Pharmaceuticals Inc., a corporation incorporated under the laws of the State of Delaware (Lender).*[1]

From this sample, we may want to extract the date ("November 17, 2014"), type of agreement ("LOAN AGREEMENT"), names of the borrowers ("Auxilium Pharmaceuticals, Inc." and "Auxilium UK LTD") and the lender ("Endo Pharmaceuticals Inc."). In practice, there are few simple approaches for extracting the data. One of which is a combination of heuristics and out-of-the-box NER tools. We can make use of regular expressions to extract the date and the agreement name. We can use spaCy[2] or CoreNLP (Manning et al. 2014) to extract the company names. We observed that this approach is not scalable and requires enormous amount of effort to carefully craft the heuristic rules to capture all the key datapoints across different types of documents.

Therefore, our motivation is to develop a domain specific datapoint extraction and entity recognition system, even when very little labeled data is available. We treat the problem of extracting the datapoints from unstructured text as a sequence labeling problem and make use of techniques from Named Entity Recognition (NER) and sequence labeling research. Recent efforts in NER research have fo-

[1] Loan Agreement - https://goo.gl/8djHXe

[2] spaCy - https://spacy.io

cused on neural architectures (Chiu and Nichols 2016; Lample et al. 2016; Dernoncourt, Lee, and Szolovits 2017a). These neural methods require large amounts of training data. Therefore, our motivation is to develop techniques for low resource datasets.

Studies have shown that transfer learning technique improves the overall performance of the model when there is limited labeled training data. Transfer Learning is a technique where a large dataset (source dataset) is trained with a neural architecture and the learned parameters are used to initialize the weights of the target model.

In our work, we experiment with a Bi-directional Long Short-Term Memory(Bi-LSTM) architecture which works well on low resource datasets. We also develop a novel mechanism to introduce domain-specific knowledge to the neural architecture. Additionally, we show that transfer learning from a pretrained model improves the performance of the models.

Our experiments on 4 financial datasets, including three low-resource datasets - Custodian, Asset Manager, and Leverage Ratio confirm that our architecture works well for low resource conditions.

Key contributions of this paper are -

- Neural Architecture for introducing domain knowledge into the network
- Study on transfer learning for sequence labeling in a low resource scenario

Our paper is organized as follows. First, we discuss recent works in sequence labeling, low resource deep learning and finance. Second, we describe the datasets and the methodology used for creating the 3 datasets used in our experiments. We then describe the neural architecture used in our experiments. Next, we detail our experiments and results. We perform an ablation study to understand the influence of each of layer in the network with and without transfer learning. Lastly, we conclude the paper with discussion about our work and potential future work.

Related Works

Traditionally, sequence labeling problems like NER and Part of Speech Tagging have used Maximum Entropy Models and hand crafted features (Mikheev, Moens, and Grover 1999; Bender, Och, and Ney 2003). The use of neural networks for NER was popularized by (Collobert et al. 2011). Since then, there have been several improvements to the neural architecture for identifying named entities (Yadav and Bethard 2018). Most competitive NER systems use a Bi-directional Long Short Term Memory (Bi-LSTM) over the word and character embeddings, which closely resembles the architecture described in (Lample et al. 2016).

(Lample et al. 2016) concatenate word embeddings with a Bi-LSTM over the characters of a word. Then, they pass these embeddings through a sentence level Bi-LSTM and a Conditional Random Field (CRF) layer to produce the labels. (Dernoncourt, Lee, and Szolovits 2017b) implement a similar architecture in their software - NeuroNER. We draw inspiration from (Lample et al. 2016) and (Dernoncourt, Lee, and Szolovits 2017b) for our model architecture.

These networks can be trained on a large dataset and then fine-tuned for a target dataset. Recent efforts in Transfer Learning have yielded positive results in NLP Tasks (Mou et al. 2016; Young Lee, Dernoncourt, and Szolovits 2017; Newman-Griffis and Zirikly 2018).

(Mou et al. 2016) conduct a thorough study on the transferability of neural networks in NLP. Their findings indicate that word embeddings trained on a source dataset are transferable to a semantically different task.

(Young Lee, Dernoncourt, and Szolovits 2017) use transfer learning techniques for de-identification of Protected Health Information (PHI) in Electronic Health Records (EHR). They train a sequence labeling model on two datasets - i2b2 2014 and i2b2 2016. They successfully demonstrate that transferring parameters from an out-of-domain model outperforms the state of the art results. A key finding from their analysis was that transferring the parameters from the lower layers of a pretrained model was almost as efficient as transferring the parameters from the entire network.

Our work in financial data extraction closely relates to (Alvarado, Verspoor, and Baldwin 2015). In their experiments, they use a Conditional Random Field (CRF) and manually choose features. They train their model on an out-of-domain dataset (Tjong Kim Sang and De Meulder 2003) and perform domain adaptation on the target dataset. Their results indicate that training only with a small in-domain dataset is better than training with a large out-of-domain dataset and a small in-domain dataset together.

Data

We use five datasets in our experiments. For training the out-of-domain model[3], we use CoNLL 2003 English dataset (Tjong Kim Sang and De Meulder 2003). We use the following financial datasets in our experiments- (1) Financial NER Dataset (Alvarado, Verspoor, and Baldwin 2015) (2) Custodian (3) Asset Manager (4) Leverage Ratio. The Financial NER dataset is an open source named entities dataset. Custodian, Asset Manager and Leverage Ratio are internal datasets. We provide detailed descriptions about these datasets in the next section.

Financial NER Dataset

(Alvarado, Verspoor, and Baldwin 2015) create their dataset by annotating financial agreements made public by the U.S. Security and Exchange Commission (SEC) filings. They annotate a total of 8 documents for LOCATION, ORGANIZATION, PERSON and MISCELLANEOUS.

Custodian, Asset Manager and Leverage Ratio

To test our model in the wild, we collected mutual fund prospectus documents which are publicly available on the internet. These documents are fairly large in size (varies from 80 to 300 pages) and have no discernible patterns which can be used by a heuristic system. The documents were collected from the websites of individual fund houses

[3]This model will be referred as out-of-domain model and pretrained model interchangeably

Dataset	Train		Validation		Test		Entities
	Tokens	Sentences	Tokens	Sentences	Tokens	Sentences	
CoNLL 2003	203621	14041	51362	3250	46435	3453	23499
Financial NER	41015	1164	-	-	13249	303	1164
Custodian	16201	574	1726	57	2248	58	166
Asset Manager	22833	672	2407	71	2835	73	165
Leverage Ratio	4414	140	-	-	1551	47	125

Table 1: Description of the datasets. Table indicates number of tokens and sentences used for training, validation and test sets in each of the datasets. The column Entities indicates the number of entities present in the train set.

(Ex. BlackRock[4]) or investment research services (Ex. Morningstar[5]). From these documents we identify a few key datapoints like Custodian, Asset Manager, Leverage Ratio, etc. which are relevant to organizations dealing with such documents. Our task was to extract the correct entities for each of these datapoints from candidate sentences retrieved from the source document.

In order to create the dataset for Custodian, Asset Manager and Leverage Ratio, we use a proprietary tool to identify parts of the PDF such as table of contents, section headings, keywords, etc. and localize to the approximate region of interest, where the datapoint could be present. Then, the domain experts manually annotate all candidate sentences identifying the correct datapoints.

In Table 1, we describe all the datasets used in our paper.

Model Architecture

Our proposed model uses two Bi-LSTM layers - character and word and a domain specific Bi-LSTM layer. First, we have the character embedding layer which passes through a character Bi-LSTM layer. Then, the output of the character Bi-LSTM layer is concatenated with the word embeddings. We also concatenate the output of the domain-specific layer to the word embedding. We use GloVe word embeddings (Pennington, Socher, and Manning 2014). The concatenated word embedding is passed through a word Bi-LSTM layer. The output of this layer is passed to the projection layer and followed by a Conditional Random Field (CRF) layer to generate the output. Our model is shown in Figure 1.

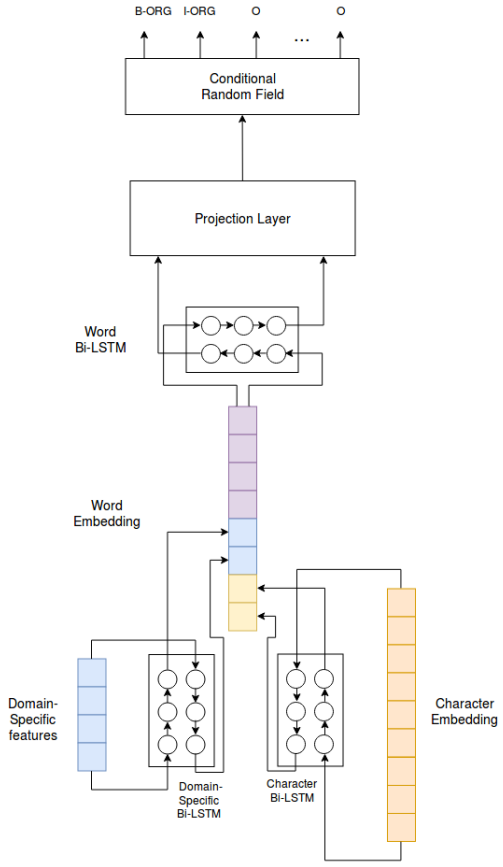

Figure 1: Architecture of our model

Domain Specific Knowledge Engineering

We observed that the correct named entities are often accompanied by dataset specific keywords. Consider the following example from the Asset Manager dataset -

> *Since January 1, 2002, the Fund is managed by Fideuram Gestions S.A. (the Management Company), a Luxembourg company, controlled by Banca Fideuram S.p.A. (Intesa Sanpaolo Group).* [6]

From the above sentence, we observe that the correct named entity is 'Fideuram Gestions S.A.' and is accompanied by the keyword 'Management Company', which is a known synonym for the Asset Manager. The datapoint Asset Manager has several other keywords such as Investment Advisor, Investment Manager, etc. These keywords are different for Custodian, Leverage Ratio and Financial NER.

In order to introduce this domain knowledge into our neural network, we encode this information as embeddings and pass it to a Bi-LSTM layer. The output of the Bi-LSTM network is concatenated with the word embedding.

Transfer Learning

Our transfer learning approach is similar to the methods followed by (Young Lee, Dernoncourt, and Szolovits 2017), where we transfer the parameters of different layers from

[4]BlackRock - https://goo.gl/bs3vU3
[5]Morningstar - https://www.morningstar.com/
[6]Fideuram Fund - https://goo.gl/UDQqiA

Architecture Type	Custodian		Asset Manager		Financial NER
	Validation	Test	Validation	Test	Test
Baseline	85.11	77.55	75.86	66.67	84.14
Domain_θ	86.96	80.77	77.78	75.00	84.73
Word_θ	87.50	88.89	80.70	58.62	85.48
Character_θ	86.96	85.11	80.00	67.86	84.36
Projection_θ	88.89	77.78	75.86	62.96	83.33
Word_θ + Character_θ	86.96	**91.67**	**81.97**	73.68	**87.48**
Word_θ + Character_θ + Domain_θ	**89.36**	85.71	71.88	**77.19**	85.35
Word_θ + Character_θ + Domain_θ + Projection_θ	86.96	89.36	78.69	74.07	82.96

Table 2: Results on the custodian, asset manager, and Financial NER dataset for various architectures. The columns indicate the F1 scores for all the architectures.

Architecture Type	F1
Baseline	90.11
Domain_θ	**95.65**

Table 3: Results on the leverage ratio dataset for various architectures.

the pretrained model to the target model. We transfer the parameters of the character embeddings and word embeddings. In case we do not perform transfer learning, we randomly initialize the character embeddings and domain-specific embeddings and use GloVe embeddings for the words.

Experimental Setup

In our study, we experiment by transferring parameters at various layers from an out-of-domain model. The Baseline model is trained only on the in-domain dataset (only Custodian or Asset Manager or Leverage Ratio or Financial NER dataset). We train the model with the same architecture described in 1 without the domain-specific features.

For the pretrained model, we train a Baseline Model on the CoNLL 2003 English dataset(Tjong Kim Sang and De Meulder 2003). We achieve F1 of 89.30 on the CoNLL 2003 Test Set. All the results in our experiments are obtained by transferring the parameters from this pretrained model.

In our experiments, we transfer the following layers - (1) Word Embeddings (Word_θ) (2) Character Embeddings (Character_θ) (3) Projection Layer (Projection_θ). We additionally activate the Domain-Specific Features in our network. (Domain_θ).

Results

We describe our results on the Custodian, Asset Manager and Financial NER dataset in Table 2. It can be observed that the best performing models have transferred parameters from word and character embeddings and along with the domain-specific features for the Custodian and Asset Manager dataset. From Table 2, it is evident that our neural architecture without transfer learning, outperforms the reported state of the art results on the Financial NER dataset[7]. Our best performing model achieves F1 of 87.48 on the Financial NER dataset which makes use of transferred word and character embeddings. Results in Table 3 suggests that domain-specific layer enhances the model's performance.

We observe that in all the datasets, the domain-specific features improve over the baseline F1. However, in the case of the Financial NER dataset we note that the best performing system is when word and character embedding layer is transferred. This observation is consistent with the findings mentioned in (Young Lee, Dernoncourt, and Szolovits 2017), where most of the lower layers contribute to the greatest improvement of the model. But, we find that the including the final layer or the task dependent layer decreases the performance.

Conclusion

For our future work, we would like to combine our word embeddings with ELMo Embeddings (Peters et al. 2018) and BERT Embeddings (Devlin et al. 2018). We intend to introduce document level meta data like PDF layout and local meta information such as bold, underline and italics in to the domain specific layer.

Our work can be extended to clinical texts, where annotating data is very expensive. Our work closely relates to Multi-Task Learning (MTL). Recent works have shown promise in Multi-Task Learning for Sequence Labeling Problems in a low resource scenarios (Peng and Dredze 2017; Lin et al. 2018).

In conclusion, we demonstrate a Bi-LSTM architecture for low resource datasets. Our experiments consistently show that transfer learning combined with domain-specific knowledge engineering improves entity recognition in a low resource setting.

Acknowledgements

We would like to thank our anonymous reviewers for their helpful feedback in improving our work. We wish to thank Arjun Rao for internally reviewing the paper. Lastly, we thank the Stride.AI team for their valuable inputs in the research.

[7](Alvarado, Verspoor, and Baldwin 2015) report F1 of 82.7

Appendices

Examples In this section, we show a few sample examples from our datasets. Refer to Table 4 5 and 6

References

[Alvarado, Verspoor, and Baldwin 2015] Alvarado, J. C. S.; Verspoor, K.; and Baldwin, T. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, 84–90.

[Bender, Och, and Ney 2003] Bender, O.; Och, F. J.; and Ney, H. 2003. Maximum entropy models for named entity recognition. In Daelemans, W., and Osborne, M., eds., *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 148–151.

[Chiu and Nichols 2016] Chiu, J., and Nichols, E. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics* 4:357–370.

[Collobert et al. 2011] Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

[Dernoncourt, Lee, and Szolovits 2017a] Dernoncourt, F.; Lee, J. Y.; and Szolovits, P. 2017a. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

[Dernoncourt, Lee, and Szolovits 2017b] Dernoncourt, F.; Lee, J. Y.; and Szolovits, P. 2017b. Neuroner: an easy-to-use program for named-entity recognition based on neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 97–102. Association for Computational Linguistics.

[Devlin et al. 2018] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Lample et al. 2016] Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–270. San Diego, California: Association for Computational Linguistics.

[Lin et al. 2018] Lin, Y.; Yang, S.; Stoyanov, V.; and Ji, H. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of The 56th Annual Meeting of the Association for Computational Linguistics (ACL2018)*.

[Manning et al. 2014] Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 55–60.

[Mikheev, Moens, and Grover 1999] Mikheev, A.; Moens, M.; and Grover, C. 1999. Named entity recognition without gazetteers. In *EACL*.

[Mou et al. 2016] Mou, L.; Meng, Z.; Yan, R.; Li, G.; Xu, Y.; Zhang, L.; and Jin, Z. 2016. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 479–489. Austin, Texas: Association for Computational Linguistics.

[Newman-Griffis and Zirikly 2018] Newman-Griffis, D., and Zirikly, A. 2018. Embedding transfer for low-resource medical named entity recognition: A case study on patient mobility. In *Proceedings of the BioNLP 2018 workshop*, 1–11. Melbourne, Australia: Association for Computational Linguistics.

[Peng and Dredze 2017] Peng, N., and Dredze, M. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 91–100. Vancouver, Canada: Association for Computational Linguistics.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Association for Computational Linguistics.

[Peters et al. 2018] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[Tjong Kim Sang and De Meulder 2003] Tjong Kim Sang, E. F., and De Meulder, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W., and Osborne, M., eds., *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147.

[Yadav and Bethard 2018] Yadav, V., and Bethard, S. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2145–2158. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

[Young Lee, Dernoncourt, and Szolovits 2017] Young Lee, J.; Dernoncourt, F.; and Szolovits, P. 2017. Transfer learning for named-entity recognition with neural networks.

Example	Entity	Explanation
The ICAV has appointed RBC Investor Services Bank S.A to act as Depositary for the safekeeping of all the investments, cash and other assets of the ICAV and to ensure that the issue and repurchase of Shares by the ICAV and the calculation of the Net Asset Value and Net Asset Value per Share is carried out and that all income received and investments made are in accordance with the Instrument of Incorporation and the UCITS Regulations.	RBC Investor Services Bank S.A	The custodian is RBC Investor Services Bank S.A which is referred to as Depositary in the sentence. Although ICAV and UCITS are Organizations, they are not the Custodian.

Table 4: Example from Custodian Dataset.

Example	Entity	Explanation
Prior to joining Deutsche Bank, Barbara was a Fund Tax Project Manager at Dexia-BIL, Dexia Fund Services in Luxembourg for two (2) years, and a Senior Fund Manager for DWS Investment S.A. (now the Management Company) in Luxembourg for ten (10) years.	DWS Investment S.A.	DWS Investment S.A. is the management company or the asset manager because of the phrase "now the Management Company". The reason Deutsche Bank is not the Asset Manager is because the sentence does not mention if it is the Asset Manager.

Table 5: Example from Asset Manager Dataset.

Example	Entity	Explanation
Under normal market conditions the level of leverage is expected to be between 200% and 800% of the Net Asset Value of the Fund where leverage is calculated using the sum of the absolute value of the notional amounts of the FDI positions in accordance with the "gross method" as set out in the Commission Delegated Regulation.	200%, 800%	The example indicates that the expected leverage or the leverage ratio is between 200% and 800%. The system should pick both "200%" and "800%".

Table 6: Example from Leverage Ratio Dataset.