AutoAyurveda: Curation of Ayurvedic Drug-Disease Pairs using Deep Learning

Devishi Kesar devishi15024@iiitd.ac.in
Y.S. Ramya yellapragada15117@iiitd.ac.in

Abstract

Ayurvedic medicines have been an essential part of medical treatment for Indians, way before the introduction of modern day 'English medicines. Ayurvedic research continuous to take place especially in India. This research often includes new drugs or treatments with the disease they address, and their function. These are often sound alternatives to modern day medicine but are lost as no methodical timely curation of this information takes place. In this paper, we propose a method to automatically extract Ayurvedic drug-disease pairs using deep learning.

1 Introduction

Medical treatment is mandated at least once in each person's life, making healthcare an important area of study. Modern medicine is synonymous with heathcare. However, treatment or cure for an ailment can be found in different ways. A study suggests that only around 20-30% of the population in the World relies on conventional sources of medicine for treatment while rest rely on unconventional sources comprising mainly of herbal sources(Wise, 2013). Indian medicine dates far back. Ayurveda has been one of the most prominent forms of treatment for Indians before the introduction of modern medicine. Ayurvedic medicine is largely based on using herbal, animal and natural components present around us ranging from gold to corals. It, in fact, is the most ancient source of treatment, yet it is not elementary with its very own strong experimental and philosophical base.

Ayurvedic research is prevalent today, mostly in India. Unlike Modern medicine, these research texts are not as well documented and curated for easy accessibility and information. More work is ongoing to extract relevant information from modern medical documents. This in itself is a challenging problem because of the difference in domain and variability in the research texts. We intent to do the same for Ayurvedic research documents. This poses additional challenges on top of the exisiting challenges. Some challenges are listed as follows:

- There is no comprehensive dataset mapping ayurvedic drugs and diseases like PubMed for modern medicine(Amin, 2016).
- The same ayurvedic drug can be expressed under different names because of their different regional identities. They can also be in different formats (compounded or otherwise, etc.)
- Additionally the regional names are not in English to apply regular nlp models, which are trained on English texts. Additionally, no one language my encompass all the names.
- No comprehensive mapping exists from even one of their indigenous names to their modern english names, for example drug/disease names in english.

Keeping in mind these challenges, we attempt to automate the extraction of drug-disease relations from Ayurvedic research documents. This would be a crucial step in making Ayurvedic treatment information more accessible.

We consider the problem of drug-disease identification from a research document as a Named Entity Recognition(NER) problem. We employ the use of a deep learning sequence tagger which included a BiLSTM and CRF using a combination of GloVe word Embeddings and Character embeddings. This model is a genarl

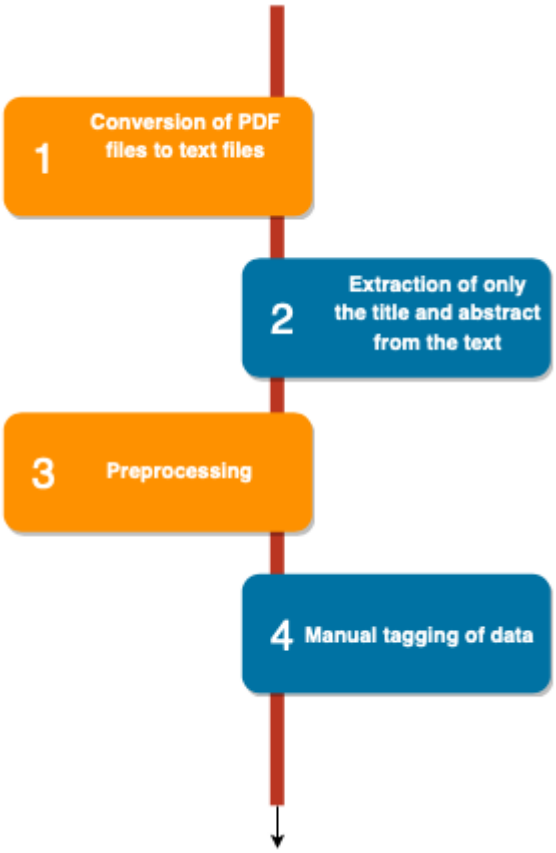

Figure 1: Processing input training data

sequence tagger proposed by (Huang et al., 2015). For our dataset, we downloaded a set of published papers from the renowned journals, pre processed it and curated a database by manually tagging it.We then evaluate our results using accuracy, f1-metric, precision and recall scores.

2 Data set

2.1 Curating Data

For our **training data**,a set of 120 recently published research papers in Ayurveda were downloaded from the Ayurpharm online archive. For our **test data** , a set of 20 recently published research papers in Ayurveda were downloaded from the International Journal of Ayurveda and Pharma Research(IJAPR).

2.2 Pre-Processing

Each of the files downloaded for training purposes went through the following process:

Entity	Tag
medhava	B-Dg
kriti	I-Dg
oggugulu	E-Dg
is	O
a	O
cure	O
for	O
arthiritis	S-Ds
and	O
a	O
therapeutic	B-Fn
agent	E-Fn

Table 1: Entity Tagging example

2.3 Tagging Data Set

We tagged the data in IOBES format.

- I is a token inside a chunk,
- O token outside a chunk
- B is the beginning of chunk immediately following another chunk of the same Named Entity.
- E tag is used to mark the last token of a chunk immediately preceding another chunk of the same named entity.
- S is used to represent a chunk containing a single token. Chunks of length greater than or equal to two always start with the B tag and end with the E tag.

The tags we named are Drug as Dg, Disease as Ds and Function as Fn. For example, in Table 1, a word is followed by space followed by IOBES tag.

3 Algorithm

3.1 Building Vocab

Once our dataset is curated, we create a vocabulary of words in the documents and tags used. The size of the vocabulary for our database is 3789 and the vocabulary size for tags is 13.

3.2 Generating Word Embeddings

The word vocabulary is now used to generate word level embeddings for use in the BiLSTM model.

Figure 2: Flow Chart for using Deep Learning for name entity recognition

We fine tune the Glove 840B word vector representation over our training and test dataset for word representation.

Glove Character level embeddings are used as they are.

3.3 Sequence Tagging

3.3.1 Bi-LSTM

We apply a Bi-LSTM model using the character level and word level embeddings to create word representations. A fixed sized vector is obtained as a result of each of the words capturing morphology. To get the final word encoding, we concatenate the word embedding from GloVe for the word with the above obtained word embedding after Bi-LSTM. An example formulation is shown in Figure 3.

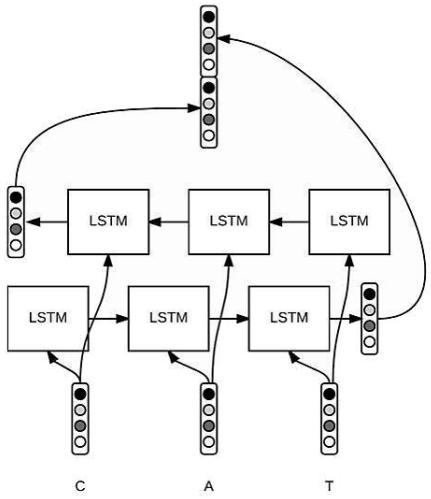

Figure 3: Character level encoding using Bi-LSTM

After obtaining the word representation from the above LSTM model we apply LSTM once again over the sequence of word vectors to get another sequence of vectors.

3.3.2 Prediction of tags

We use a linear chain CRF(conditional random fields) to predict the best possible sequence of tags. We have applied dynamic programming to fasten the results and finally applied a softmax to the scores of all possible sequences to get the probability of a particular sequence. We have used cross entropy loss as our objective function. The code for generic sequence tagging is available on https://github.com/guillaumegenthial/tf_ner

4 Results

We see that the accuracy of the test set comes out to be 0.90 while the F1 score is low at 0.45. On testing for different data sizes we observe that the F1 score improves. Hence, there is a requirement for a larger data set for obtaining better accuracies.

Metric	Result
Accuracy	0.9012245
Precision	0.4681648
Recall	0.43554008
F1 score	0.45126355

Table 2: Results for Test Data Set

Train Data Size	F1 score
50	0.110045
93	0.451264

Table 3: The F1 score obtained improves with an increase in train data set size

5 Conclusion

The results obtained from the deep learning network we have applied are very good with an accuracy of 0.90. We can observe from our experiments that the performance metrics increase with an increase in data size as shown in Table 2. Future work will require additional curation to see an improvement in other precision metrics like F1 score and precision.

Acknowledgments

We would like to acknowledge the help provided by our course instructor Dr. Tanmoy Chakraborty

for motivating us to do this project as well as the course teaching assistants for providing their valuable help in understanding certain concepts.

References

Hetal Amin. 2016. How data mining is useful in ayurveda. *Journal of Ayurved and Herbal Medicine* 2:61–62.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991. http://arxiv.org/abs/1508.01991.

Jacqui Wise. 2013. Herbal products are often contaminated, study finds. *BMJ* 347. https://doi.org/10.1136/bmj.f6138.