Cognitive Disease Hunter: Developing Automated Pathogen Feature Extraction from Scientific Literature

Timothy NeCamp[1,2], Prasanna Sattigeri[2], Dennis Wei[2], Emily Ray[2], Youssef Drissi[2], Ananya Poddar[2], Diwakar Mahajan[2], Sarah Bowden[3], Barbara A. Han[3], Aleksandra Mojsilovic[2], Kush R. Varshney[2]

[1]University of Michigan, [2]IBM, [3]Cary Institute

***Abstract*— Emerging infectious diseases are an increasing global problem. Developing the ability to predict new or unexpected outbreaks remains one of the most difficult problems of modern times. The majority of infectious diseases emerging in humans are zoonotic (i.e., have animal origins), but these diseases are the exception rather than the norm - the vast majority of microbes do not cross the species barrier to cause disease in humans. What distinguishes zoonotic from non-zoonotic microbes is therefore a fundamental outstanding question that underlies our ability to mitigate the risk of future outbreaks. Since many distinguishing pathogen features are measurable and observable, it should be possible to learn intrinsic features that differentiate zoonotic pathogens from other microbes, and to predict microbes that may have an unrealized zoonotic capacity. The first step towards learning these relevant features and making predictions is to create a database of pathogen features. Unfortunately, that database does not exist and these features are locked away in academic papers. In this paper, we discuss ongoing work on building and training a machine learning annotation model which ingests biology publications and extracts pathogen features. We describe our developed type system, a framework which defines the classes of words/phrases we wish to extract (entities) and how they relate to one another (relations). We also explain how we used domain expertise to combine our type system with an annotator protocol to identify all relevant information within our corpus. Lastly, we discuss results of our initial machine learning annotation model and how we plan to obtain more training data to improve model performance.**

1 Introduction

Zoonotic pathogens are human pathogens which originate in animals. Many major epidemics have been caused by zoonotic pathogens, such as bird flu, Ebola, influenza (including avian and swine), Lyme disease, rabies, West Nile fever, and Zika fever (Han, Kramer, & Drake, 2016; Kilpatrick, Randolph, & Drivers 2012). These epidemics cause substantial human losses, but arise from a small fraction of the total microbial diversity on earth. Most microbes that originate in animals are unable to cross the species barrier to successfully infect a human host.

The expert consensus from previous work is that the fraction of microbes that do infect and cause disease in humans share key features (Woolhouse, Haydon, & Antia, 2005). For example, there are more outbreaks of zoonotic viral and bacterial diseases than those caused by worms, protozoa or fungi (Woolhouse & Gowtage-Sequeria, 2005). Among viruses, those containing RNA are more likely to be zoonotic than those with DNA (Geoghegan, Duchêne, & Holmes, 2017). In addition to pathogen type and genetic material, there are many more microbial traits routinely described in primary literature that underscore an intrinsic biological capacity to cross the species barrier.

Given the global importance of zoonoses, we aim to learn these intrinsic features associated with zoonoses and predict future zoonotic threats. To learn these characteristics and make predictions, we first need an encompassing database of pathogens and their features. Currently, that database does not exist. The pathogen feature information is known; however, it is contained within academic papers. It is an inhibitively intensive task for humans to read through these papers and manually create the desired database.

For our work, we developed an automated way to extract relevant pathogen features from unstructured academic texts. Our work relies on creating an ontology, a formal specification of all relevant information within the texts. We then build a machine learning annotation model which learns how to find this information automatically. Below, we describe how we developed both our ontology and machine learning model. We also discuss initial performance of our approach. Lastly, we discuss the next steps for improving our annotation algorithm and other future work.

2 Methodology

To automatically extract pathogen information from unstructured academic text, we take an ontological learning approach (Bodenreider & Stevens, 2006; MacLean & Heer 2013). Ontological learning is a method in which the desired information within the text is categorized and then text patterns are used to learn how to find and label (i.e., annotate) it automatically. It entails 3 steps described in the following subsections: (1) specifying what information one aims to extract (the type system), (2) obtaining examples of annotated documents (getting a training corpus), and (3) using these examples to learn how to automatically extract information (developing an automated annotator).

2.1 Creating a type system

Before we can extract relevant information, we must first specify what type of information is of interest. A type system does exactly this. In our setting, the type system defines two quantities, entities and relations.

Entities are classes of words or short-phrases we want to extract from text. Table 1 contains a few examples of various entities for the project.

Entity	**Mentions**
Disease Infectious Organism	“zika virus”, “yellow fever virus”, “ebola”
Percent	“5 %”, “40%”, “10 percent”
Capsid Structure	“helical”, “icosahedral”, “tubular”

Table 1: Examples of entities and text that would be labeled as that entity

Relations specify how two entities can relate to one another. Table 2 contains a couple examples for the project.

Relation	**Entity 1**	**Entity 2**
has feature	Disease Infectious Organism	Capsid Structure
has case fatality rate	Disease Infectious Organism	Percent

Table 2: Examples of relations and the two entities it connects

With our type system one can now take unstructured text and annotate and extract the desired information.

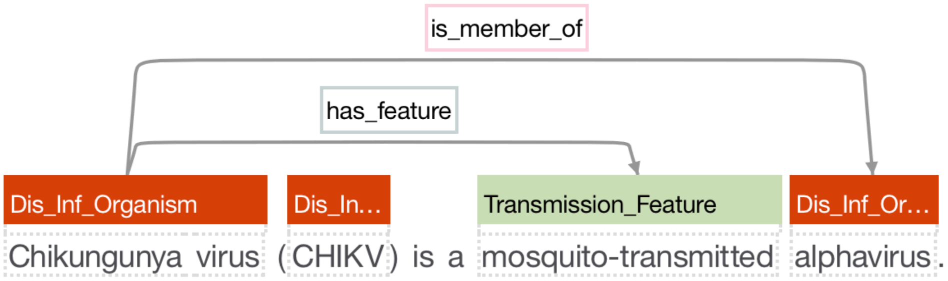

Figure 1: Example of annotated text for a given type system.

There are many important considerations when developing a type system. The type system defines everything that can and cannot be extracted from the text; if an entity or relation is not defined, then that information cannot be extracted. It is important to ensure the type system is both relevant to domain experts and feasible for a machine learning model to learn the extraction.

2.2 Getting a training corpus

Once a type system is developed, we must obtain a set of example documents for which these entities and relations have been extracted. These examples comprise our training corpus. This training corpus will be used to build machine learning model by understanding the patterns within the text that relate to different entities and relations.

The primary source for the training corpus is human annotators who have relevant domain expertise. Two of the issues raised using human annotators are 1) they may not be consistent with one another and 2) their time and effort should be reduced when possible. These issues are addressed below.

2.2.1 Annotator Guidelines

Even after a type system has been created, there is still ambiguity in how entities and relations should be identified from text. For example, the type system may contain entities for *Units*, describing the units of measure for numbers, and *Animal*. If a document contained the sentence "Zika was found in 25 primates in Brazil", should "primates" be labeled as an *Animal* or *Units* of 25? Both possibilities are reasonable and may lead to inconsistencies in the annotations. These inconsistencies are problematic as inter-annotator disagreement convolutes the training data. To avoid disagreement and ambiguity, we developed annotator guidelines to describe how entities and relations should be identified and address any possible ambiguities.

2.2.2 Speeding up annotations

Manually annotating a document from scratch is cumbersome. To minimize annotation time, we aim to create a pre-annotator that automates identification of some entities prior to the manual annotation. Currently, for certain entities, we use 2 types of pre-annotators:

- Dictionary-based pre-annotator- A dictionary based pre-annotator creates a list of possible mentions a specific entity can be. For example, we know the entity *Capsid Structure* might only have mentions "helical", "tubular", "icosahedral", and "ovoid". This type of pre-annotator is useful when the number of possible mentions is small.
- Existing annotators – Automatic annotators have been developed for other applications. We borrow labels from annotators with similar entities to our type system. For example, the WatsonX API (IBM, 2017, May 22), which is used to extract information from medical text, can identify *Disease* and *Animal*, which are also entities in our type system. We can first run a document through an existing annotator and use its annotations for the entities of interest.

2.3 Developing an automated annotator

With a large enough corpus, we will train a model that identifies desired entities and relations automatically. Specifically, we plan to fit a Statistical Information and Relation Extraction (SIRE) model (IBM, 2015). A SIRE model uses Maximum Entropy techniques to identify patterns in the text that lead to the labeling of specific entities and relations. This model first identifies all entities within a text based off sentence characters and structure. Once the entities are labeled, the model then identifies the relations between those entities. Both tasks can be understood as complicated classification problems with text-based features.

2.4 Watson Knowledge Studio

Watson Knowledge Studio (WKS) provides an intuitive user interface for completing all the tasks necessary to develop an automated annotator (IBM, 2017, About Knowledge Studio). Specifically, within WKS, we can: create a type system, import annotator guidelines, import training documents to annotate, annotate documents manually, apply pre-annotators to documents, fit a SIRE model to a training corpus, evaluate a SIRE model on a test set, and export a SIRE model to run on new documents.

3 Results

In this section, we describe some preliminary results of our automatic annotator.

3.1 Training Data

To obtain well-performing model, we need a large, comprehensive manually annotated training set. Below are some statistics of our current training set and end goals.

	Current Training Set:	**Training Set Goal:**
Number of documents	16	150-300
Total number of words	20,813	300,000
# of Annotators	3	5+
Most mentioned entity	*Disease Infectious Organism*	--
# of mentions	843	50-100
Least mentioned entity	*Cell Exit Method*	--
# of mention	0	50-100

Table 3: Current training data statistics and training data goals

3.2 Initial Model Results

To evaluate the model, we use three valuable statistics which summarize the model's performance. The precision is the number of correctly identified entities/relations (true positives) over the total number of entities/relations labeled (true positives + false positives). The recall is the number of correctly identified entities/relations (true positives) over the total number of entities/relations in the documents (true positives + false negatives). The F1 score is the harmonic mean of the precision and recall and provides an overall summary of the performance.

We perform a leave-one-out-cross-validation to evaluate the model. We train the model on 15 documents and evaluate the trained model on the 1 document not included in the training set. We repeat this 10 times, randomly choosing a document to leave out each time, and get the average performance of the model over these 10 iterations. The results are in Tables 4 and 5.

	Entity			Relations		
	F1	Precision	Recall	F1	Precision	Recall
Overall Mean (Standard Error)	.49 (.02)	.64 (.02)	.41 (.02)	.19 (.03)	.31 (.05)	.14 (.02)

Table 4: Mean overall F1, precision and recall scores for entities and relations

Due to the small number of training documents, the model's performance is expected to be mediocre. In fact, Table 4 indicates the model is performing better than expected. Table 4 provides preliminary evidence that the type system and annotator guidelines can eventually be learned by the machine learning annotator.

Table 5 provides some key insights. The performance of the model tends to increase as the number of mentions in the document set increase. This is expected. However, there are exceptions. *Mammal* is well-identified with a small number of mentions. *Disease Infectious Organism Feature* is poorly identified with a larger number of mentions. This is due to the complexity of the entity itself. *Mammal* is a well-defined entity while *Disease Infectious Organism Feature* is more ambiguous. The performance indicates that it will be critical to find training documents with many mentions of entities and relations that are not explicit.

Entity	F1	Precision	Recall	Number of mentions in document set
Disease Infectious Organism	.51 (.04)	.65 (.03)	.44 (.04)	843
Disease Infectious Organism Feature	.19 (.04)	.35 (.08)	.13 (.03)	210
Mammal	.84 (.04)	.87 (.04)	.82 (.05)	176
Transmission Feature	.27 (.04)	.44 (.08)	.20 (.03)	138
Disease Symptom	.21 (.06)	.33 (.08)	.19 (.06)	80

Table 5: Mean (and standard error) F1, precision and recall scores for particular entities

4 Conclusions and Future Work

Currently, our machine learning annotator needs to achieve better performance before it can be applied to a larger corpus. Immediate steps can be taken to improve performance. First, we need a much larger training set. To obtain more training data, we need to increase the number of annotators. Unfortunately, the initial burden to train new annotators is quite large; our type system is intensive and the annotator guidelines are very specific. As opposed to having new annotators just read all the documentation, we are developing a training protocol to teach annotators in a dynamic and less-burdensome setting.

Once our training corpus is larger, there are smaller fixes that can be made to improve the model. Minor changes to the type system and annotator guidelines can help reduce certain errors. For example, if one entity is rarely found within the corpus, identification of that entity will always be difficult. Grouping entities together may eliminate this issue.

Once we have a well-performing model, we can apply it to a large corpus of academic papers and extract all pathogen features. The features extracted for zoonotic pathogens can be used to understand the characteristics of zoonoses and develop a prediction model. We can then use extracted features of other pathogens as inputs into the prediction model and find pathogens with unrealized zoonotic capacity.

In addition to being useful for the long-term goal of predicting zoonoses, the methods used here are applicable in a wide range of domains. These methods can be used for any problem in which information needs to quickly be extracted from a large corpus of text.

Though our automated annotator greatly improves the efficiency of extracting information from unstructured text, there is still room for future improvements. Finding ways to automatically build a type system based off the text (i.e., unsupervised learning of entities and relations) reduces time spent defining a type system. Also, finding ways to better map information from other automatic annotators would reduce the annotation timeline. Lastly, using active learning techniques to better select documents for the training corpus would also decrease annotation time.

References

Bodenreider, O., & Stevens, R. (2006). Bio-ontologies: current trends and future directions. *Briefings in Bioinformatics*, *7*(3), 256–274

Geoghegan, J. L., Duchêne, S. & Holmes, E. C. (2017). Comparative analysis estimates the relative frequencies of co-divergence and cross-species transmission within viral families. *PLoS Pathog*. 13(2), e1006215

Han, B. A., Kramer, A. M. & Drake, J. M. (2016). Global Patterns of Zoonotic Disease in Mammals. *Trends in Parasitology* 32, 565–577

IBM. (2015, March 10). Statistical Information and Relation Extraction (SIRE). Retrieved from: http://researcher.ibm.com/researcher/view_group.php?id=2223

IBM. (2017). About Knowledge Studio Retrieved from: https://www.ibm.com/watson/developercloud/doc/wks/index.html

IBM. (2017, May 22). Watson for Patient Record Analytics (aka Watson EMRA). Retrieved from: http://researcher.ibm.com/researcher/view_group.php?id=7664

Kilpatrick, A. M. & Randolph, S. E. (2012). Drivers, dynamics, and control of emerging vector-borne zoonotic diseases. *Lancet* 380, 1946–1955

MacLean D. L., Heer J. (2013). Identifying medical terms in patient-authored text: a crowdsourcing-based approach. *J Am Med Inform Assoc*, 20 (6): 1120-1127

Woolhouse, M. E. J., Haydon, D. T. & Antia, R. (2005). Emerging pathogens: the epidemiology and evolution of species jumps. *Trends Ecol. Evol.* 20, 238–244

Woolhouse, M. E. J. & Gowtage-Sequeria, S. (2005). Host Range and Emerging and Reemerging Pathogens. *Emerg. Infect. Dis.* 11, 1842–1847