VIVEK IYER

International Institute of Information Technology
Gachibowli, Hyderabad - 500032
+919167381316 | vivek.iyer@research.iiit.ac.in

ABOUT ME

My name is Vivek Iyer and I am an **undergrad NLP/IR Researcher** pursuing my Masters in Computer Science at IIIT, Hyderabad. I am currently working on **learning ontologies from text** as part of the SIREN project [Sanagavarapu et al.(2018)Sanagavarapu, Mathur, Agrawal, and Reddy], a **search engine** for the banking sector. I'm working under the mentorship of **Prof. Raghu Reddy** and **Prof. Vasudeva Varma**. The project is currently funded by the Department of Science and Technology (DST)

I am actively looking for a research internship in for the summer of 2020 (time period flexible), specifically on IR topics such as the enrichment/creation of ontologies and knowledge graphs for their use in search engines. In general, my research interest lies at the intersection of **Deep Learning** and **Natural Language Processing**, and their applications in **Information Extraction** tasks.

AREAS OF INTEREST

Ontologies | Ontology Learning | Ontology Evaluation | Knowledge Graphs | Text Mining | Natural Language Understanding | Semantic Web | Concept Extraction | Entity Recognition | Deep Learning | Machine Learning | Web Mining | Information Retrieval and Extraction

RESEARCH OBJECTIVE

To enrich a seed ontology using concepts and relationships extracted from an unstructured text corpus. Currently working on the implementation of an **Long Short Term Memory Network (LSTM)** trained on an Information Security corpus that extracts **concepts** and **relationships** (and later, **instances**) from text to **enrich the seed ontology**.

CURRENT PROJECT: ONTOLOGY ENRICHMENT FROM TEXT USING LSTMS

Paper: Ontology Enrichment from Text using LSTMs

Status: Working towards **SIGKDD 2020**

Objective: To train an integrated **(distributional + path-based)** LSTM on a tagged Information Security corpus, use the trained model to extract **concepts** and **relationships** (such as **hypernymy**, **hyponymy** and **synonymy**) and merge them as part of the final ontology.

Work done:

1. The concepts related (by hypernymy/hyponymy/synonymy) to concepts in the seed ontology were **mined** from online Wiki databases like **DBPedia**, **Wikidata** and **Wordnet**
2. The concepts extracted this way and the original concepts were used to build a training dataset. This training dataset was manually validated by domain experts and given the relation tag (Hypernymy/Hyponymy/Synonymy/None) only if the relation held AND they were related to the Information Security domain.
3. In addition, false negatives were added to the dataset to negatively score our models, which consisted of relations extracted from DBPedia other than the ones specified above.

4. The Information Security training corpus was constructed by mining articles from the **Wikipedia dump** which are about the concepts in the training dataset. A https://ibm.ent.box.com/s/3f160t4xpuya9an935k84ig465gvymm2 trained on the Wiki dump was used to calculate the similarity score and articles with similarity below a certain threshold were filtered out.
5. I then trained an LSTM on this corpus that uses the tagged training dataset to learn the edge weights between every pair of noun phrases in a sentence. Edges comprising one of the above mentioned relationships are given higher weights compared to other edges.
6. Our LSTM model follows an **integrated (path-based + distributional) approach** that takes the embeddings of both the concepts and the relationship edges as input. By doing so it gives higher weights only to those relations that contain a hypernymy relationship AND are related to the current domain.
7. To extract concepts containing one of the above relations from the testing corpus, I followed the procedure given below:
   (a) First, **coreference resolution** was implemented on the testing corpus.
   (b) Next, for each paragraph in the text corpus, all the noun phrases in that paragraph are extracted using the **NLTK NE Grammar parser** and a combinatorial pair-wise matching of each extracted entity is performed.
   (c) These combinations form the testing dataset and the trained LSTM model is used to classify each relation as Hypernymy/Hyponymy/Synonymy/None.

The code for the same is available on GitHub.

Results:

1. I experimented on an information security webpage and got as high as **82% precision**. This is **considerably higher than the precision of state-of-the-art** models on Ontology Learning.
2. I then experimented on a dummy webpage about pizzas to verify if the model was extracting hypernymy relationships that had nothing to do with security domain. It was found that all relations were classified as "None", as expected.

Conclusion: Our LSTM model can be easily extended to other relations as well by modifying the training dataset accordingly. It requires only one time manual intervention to obtain the trained model and performs significantly better than the state-of-the-art in this field.

The extension of this model to **extract instances from a text** corpus will be done as **future work**.

CURRENT PROJECT: ONTOLOGY VISUALIZATION AND VALIDATION

Paper: Ontology visualization and validation using **crowdsourcing quality evaluation metrics**

Status: Ongoing

Objective: To build an ontology visualization and validation tool that allows users to login and accept/reject newly extracted concepts and relationships, and then use crowdsourcing quality evaluation mechanisms from **social media profiles** to estimate **user credibility** to take the final decision

Work done:

1. First, an ontology visualization and validation tool was built on top of the WebVOWL source code.
2. The feature to accept and reject decisions was then added to this tool and a flask framework was used to write the changes from the WebVOWL tool to the database.
3. An option is provided for the user to login using their Twitter account.

4. I then trained a **Multinomial Naive Bayes classifier** on a manually constructed dataset of top Information Security accounts that uses that account's **Twitter posts and followers** to learn a **credibility score**.
5. This credibility score is used as a weighting parameter while making the final accept/reject decision.

The code for the same is available on GitHub.

The results are yet to be concluded as the project is still ongoing.

CURRENT PROJECT: ONTOLOGY EVALUATION AND RANKING

Paper: Developing a tool for ontology Quality Evaluation using Ranking System

Status: Submitted to **ICSE SRC 2020**

Objective: To quantitatively evaluate the quality of an ontology and score them based on the number and severity of pitfalls in the ontology

Work done:

We propose OntoEvaluator, an ontology evaluation tool that uses the **OOPS! pitfall scanner**[Poveda-Villalón et al.(2014)Poveda-Villalón, Gómez-Pérez, and Suárez-Figueroa] to recognise pitfalls in the ontology and then uses a **Machine Learning** approach, in particular, **Support Vector Machines**, to score and rank ontologies based on these recognised pitfalls. I describe my approach in detail as follows:

1. First, OntoEvaluator takes a list of pitfalls (from the OOPS! catalogue) to evaluate an ontology on.

2. It then makes a POST request to OOPS! (the state-of-the-art tool in ontology evaluation) REST API, and OOPS! scans the ontology for the above-mentioned pitfalls and returns the count and severity of each pitfall that is present in the ontology.

3. I construct my training dataset of ontologies this way by manually introducing violations (of the pitfalls mentioned above) in an original seed ontology.

4. The ontologies are then represented in terms of the count of each pitfall, which form the **feature vectors**.

5. The **pitfall score** of each ontology becomes the **label**. This pitfall score can either be manually assigned based on preferred ontology quality order or it can be semi-automatically determined based on the importance of the pitfalls present.

6. Finally, I train an **Epsilon-Support Vector Regression** (SVR) model to learn the weights for each pitfall using the pitfalls as feature vectors and scores as labels as described above.

7. This trained model is used to generate a pitfall score for each ontology in the testing dataset and rank them accordingly.

Results:

1. We experimented on 3 ontologies containing pitfalls of increasing severity, namely, minor, important and critical pitfalls.

2. As expected, we got the **highest score** for the ontology with **critical** pitfalls, followed by **important** and then **minor** pitfalls, as shown here

3. We conclude that OntoEvaluator is able to learn weights for pitfalls appropriately and rank ontologies based on their importance.

Future work would involve construction of larger training datasets, testing on more ontologies and developing an interactive application for ontologists too, so that they don't need to meddle with the codebase.

The code for the same is available on GitHub.

PREVIOUS PROJECT: SURVEY ON ONTOLOGY LEARNING

Paper: A Survey on Ontology learning from Text

Status: Published as part of **ICON 2019**

Objective: To conduct a literature survey in the field of Ontology learning from Text by analysing, implementing and identifying the gaps proposed in the algorithms from this domain

Work done:

1. As part of this literature survey, a total of 188 papers were extracted from IEEE, ACM and Springer.
2. From this list, 23 relevant papers were narrowed using a funneling approach. These papers were then categorized into four categories (Similarity-based **Clustering** algorithms, **Set Theoretic** algorithms, **Web Corpus** based algorithms and **Learning-based** algorithms).
3. One paper from each category was selected, implemented and the performance of each of these algorithms was compared. The papers selected were:
   (a) **Guided Agglomerative Clustering Algorithm** [Cimiano and Staab(2005)]
   (b) **C-Pankow Algorithm** [Cimiano et al.(2005b)Cimiano, Ladwig, and Staab]
   (c) **DYNAMO-MAS** [Sellami et al.(2013)Sellami, Camps, and Aussenac-Gilles]
   (d) **Formal Concept Analysis** [Cimiano et al.(2005a)Cimiano, Hotho, and Staab]
   (e) **Word2Vec** [Wohlgenannt and Minic(2016)]

All the algorithm implementations are available on GitHub.

Results: After implementing each of the above mentioned algorithms, I arrived at the following results:

1. **Guided Clustering** was found to generate a lot of noise, so precision was very less (13%). Also it failed to take context (and thus, semantics) into account and relied on naive pattern matching
2. Though **C-PANKOW** was able to improve on precision, efficiency and also partially address the issue of context relevance (using a mixture of frequency-based mapping and document similarity scores), it uses naive syntactic pattern matching which results in selecting irrelevant terms and dropping relevant ones.
3. The **DYNAMO-MAS algorithm**, despite solving disambiguation and having better precision, has serious limitations like data sparsity and unscalability.
4. **FCA**, which uses **pseudo-syntactic dependencies**, was found to have better precision than both Clustering and C-PANKOW. But construction of a concept hierarchy is very time inefficient as compared to C-PANKOW.
5. The **Word2Vec algorithm** was able to improve the problems of efficiency, precision and data sparsity by using **word embeddings** and **skip-grams**, and was found to outperform previously mentioned algorithms. However, this algorithm also suffers from some shortcomings like the inability to handle previously unencountered words, selecting of too similar terms, scalability issues due to manual intervention etc.

Conclusion: It was thus concluded that the proposed algorithms in the field of Ontology learning from Text suffer from efficiency, precision, data sparsity, lack of contextual relevance etc. As an alternative to pattern-matching algorithms, there was a need for a model that keeps track of the current concept

being talked about across long durations and forgets irrelevant concepts. I then proposed an LSTM-based deep learning model that uses considers both semantics and syntactics to identify and extract concepts embeddeded in words, sentences and paragraphs.

PROJECTS

Here's a list of all the projects I've worked on, inside and outside my research area:

An **FAQ chatbot** that uses an **Encoder-Decoder with Luong attention** based model. Originally made as part of a hackathon conducted by Walmart, for which we won first prize.

Question Generation system built using syntactic dependency parsers

A basic **search engine** that indexes Wikipedia

Machine Translation systems using **Sequence to Sequence Neural networks**, **Attention** and **Encoder-Decoder models**

A **Resume parser** that uses **Machine Learning** to learn field-wise information from a resume

References

[Cimiano et al.(2005a)Cimiano, Hotho, and Staab] Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005a. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

[Cimiano et al.(2005b)Cimiano, Ladwig, and Staab] Philipp Cimiano, Günter Ladwig, and Steffen Staab. 2005b. Gimme'the Context: Context-driven Automatic Semantic Annotation with C-PANKOW. In *Proceedings of the 14th international conference on World Wide Web*, pages 332–341. ACM.

[Cimiano and Staab(2005)] Philipp Cimiano and Steffen Staab. 2005. Learning Concept Hierarchies from Text with a Guided Agglomerative clustering Algorithm. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*.

[Poveda-Villalón et al.(2014)Poveda-Villalón, Gómez-Pérez, and Suárez-Figueroa] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34.

[Sanagavarapu et al.(2018)Sanagavarapu, Mathur, Agrawal, and Reddy] Lalit Mohan Sanagavarapu, Neeraj Mathur, Shriyansh Agrawal, and Y Raghu Reddy. 2018. Siren-security information retrieval and extraction engine. In *European Conference on Information Retrieval*, pages 811–814. Springer.

[Sellami et al.(2013)Sellami, Camps, and Aussenac-Gilles] Zied Sellami, Valérie Camps, and Nathalie Aussenac-Gilles. 2013. DYNAMO-MAS: a Multi-agent System for Ontology Evolution from Text. *Journal on Data Semantics*, 2(2-3):145–161.

[Wohlgenannt and Minic(2016)] Gerhard Wohlgenannt and Filip Minic. 2016. Using word2vec to Build a Simple Ontology Learning System. In *International Semantic Web Conference (Posters & Demos)*.