An Application of Semantic Relation Extraction Models

Aleksandr Sergeev
aserge16@earlham.edu
Computer Science Department
Earlham College
Richmond, Indiana

ABSTRACT

Natural language processing (NLP) is growing to be one of the largest subfields of computer science, with application to many other fields such as linguistics or information extraction. This large growth has resulted in higher availability and a variety of resources that contributors can work with. Semantic relation datasets as a result have seen an increase in number and many today, including high profile companies at the head of NLP, are creating state-of-the-art machine learning models. These models are capable of identifying these relations from sentences by training on this selection of datasets. The number of models applied outside of their datasets, however, is still relatively low. With such a high amount of data being produced daily in our technological world, semantic relation models could potentially be used to help increase the speed at which specific information can be identified and extracted from their collections of data. This project will look at the accuracy of trained semantic models outside of their training datasets to assess the applicability of them as a tool for a general information search.

Keywords: Natural Language Processing, Machine Learning, Relation Extraction, Enitity Recognition.

1 INTRODUCTION

Information extraction is one of the most important research fields at this time because of the high volume of data society produces. With annual improvements in hardware and software, people are able to explore more and more different fields with the aid of technology, consequently creating more data. Therefore, processing data is now an important field, specifically to the topic of this paper, information extraction from unstructured texts in the field of natural language processing. This topic has the potential to greatly reduce the time needed for one to search through any given text and extract key facts and relationships that would otherwise have to be spotted by reading through the whole text. By eliminating the need to read through the supporting and/or irrelevant data, skipping straight to what information is needed, one can enable the user to concentrate on what is for them more important and therefore improve productivity. A very useful way of approaching this problem is analyzing the meaning of sentences by labelling the semantic relations they present between subjects of interest. For example, one could identify the entity of interest *"John Smith"*, put the sentences with John through a machine learning model and identify an 'Entity-Destination' sematic relation where *"John Smith is flying to New York"*. The more different and precise types of relationships datasets provide to train for, the higher the accuracy of information extraction machine learning models could apply. This labelling and grouping of sentences around certain, or perhaps all, entities of interest creates a structure to unstructured text, which gives someone the knowledge of where to look to find what they are looking for, and the lost time spent going through irrelevant text is greatly decreased. The project will present a model capable of identifying semantic relationships after training and will then work with foreign unstructured text of different types, fictional and non-fictional text to be broad. How well the sentences of interest are labelled is the subject of interest for this project as it demonstrates the potential of a new NLP tool for general users.

2 TOKENIZING

The process of tokenizing text is the first step to preparing a text for machine learning models or other approaches of information extraction. During this stage, one would invoke a current Natural Language Processing library with built in tokenizer functions to assign words to predefined categories. Tokenization of the sentence *"He ran home."* works as a simple example, where we would derive three tokens; "He": pronoun, "ran": past tense verb, "home": location. In this sentence, the two non-verb tokens are also nouns. An example of how to do this with a general text can be seen at Python Programming, demonstrated with the NLTK public library [10]. These pretrained functions work with the common vocabulary of an assortment of languages, some even able to build parse trees where some tokens have dependencies between each other, demonstrated in figure one. Sometimes, however, the function will

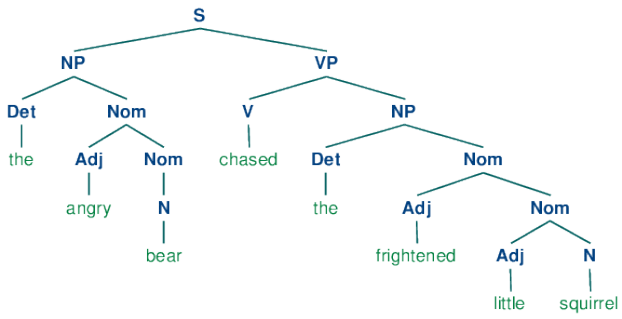

Figure 1: NLTK example of a tree depicting one of the potential sentence structures after tokenization [2]

categorize incorrectly as there are some edge cases where certain preceding words may throw off libraries like NLTK, and therefore an entity, for instance, will be classified as something other than what it is. Most commonly it occurs with joint named entities, a problem discussed in detail by Sil and Yates, who find there could be multiple potential groupings of words to form entities in a sentence with the specific example of *The Rocky Mountains* [12]. Here, one can see the problems of tokenizing these words for what we know

to be the name of a single entity, so the approach of Sil and Yates is to generate a set of all potential entity names, with which they implement a ranking algorithm to decide what name is the best. This would work in symmetry with the trained tokenizer function for the highest accuracy with one pass over the text since this is usually a costly operation. When considering texts with unseen words, common in fantasy novels and other such non-fiction, a user would have to be careful with mistokenized words in the worst case that would create false dependencies or unknown tags at best when certain libraries provide such an option.

This process is the first to consider for any NLP process, separating all words and turning them into individual values with many attributes. After this, one may use these values to construct many things, for this project specifically looking at the values as sequences of data. Adding complexity to these values is also possible and is talked about later with the GloVe vectors.

3 NAMED ENTITY RECOGNITION

Regarding unstructured texts, information is most commonly sought through entities, named objects/people, and the semantic relationships these entities carry are often of interest. For example, in a news article on Global Warming one would consider "Global Warming" the entity and the facts presented in the article would be semantic relations of other entities or descriptions pertaining to Global Warming. The first task then is to recognise which words in a text are entities and which are other types like a verb or adjective. Categories of entities can be things such as time, people, locations, organizations and so on. Consequently, the text has to be broken down and *tokenized* with certain algorithms. Searching for information is most often based around some subject of interest, an entity. These are named objects or people that occur as the subjects of sentences, and the semantic relationships the sentences describe between the entities and other nouns is the sought-after knowledge.For example, in a news article on Global Warming one would consider "Global Warming" one of the main entities of interest, and the facts presented in the article would be through semantic relations between it and other entities. Named entities can fall under many categories; organizations, locations, medical terms, and so on. During tokenization of an unstructured text this is a key sub-task being executed, the location and categorization of all entities. This project is specifically interested in this process as the dataset it will be using for training a machine learning model will require knowledge of entity indexes to function properly.

4 SEMANTIC RELATIONS

Semantic relationships are the associations we form between words and phrases when we construct sentences. When we consider them, there are many factors that we can look at to determine how certain words are relating to each other and building meaning. Naturally, we are focusing on the English language for this paper, as the same rules aren't applicable when considering other languages. Distance between words is important and as the space gets further apart so does the likelihood that they are related, this brings into light the parsing trees of figure one. Vectorizing the tokens of a document will also assist in detecting patterns, where a model may understand word dependencies better by transforming words into unique vectors through either word2vec or GloVe techniques, discussed later in the building of the model. Once these relations are detected with a successful model, we may create groups of relations and filter further based on sentence subject (entities). Listing all sentences would work, however, ideas like that presented in the MING method paper for named entity recognition by Kasneci et al. create more interesting and concise ways of presenting the information.

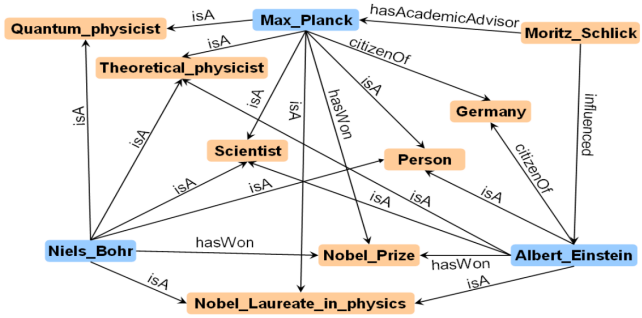

Figure 2: Graph representing relationships between 3 Nobel Prize winners from Kasneci et al. [3]

The above figure uses colorings to show different categories of entities. Blue is for entities of interest, yellow for related and the titled lines of connection describe the relation between main and secondary entities. It is an interesting take on visualizing relations in smaller space and not repeating the entirety of the sentence.

Approaching the detection of these relationships in chunks of words (sentences) and predicting them can be done in many ways. Emily Jamison provides us with an interesting take on the matter, considering certain forms of semantic relationships to have grammar rules [7]. This approach looks to convert a sentence into a tree-like structure of dependencies and then making calculations between nodes of interest, which more often are the named entities we have previously discussed. This approach is vaguely similar to parse trees, looking at sentences with structure and doing math on features of the structure. One may also consider a sentence as a sequence of words, each individual word corresponding to some value. With this approach and tagged points of interest, one may begin to look for sets of patterns that indicate what kind of relationship the sentence may hold. Unlike the mathematical structure approach, the whole sequence must be considered simultaneously, as one word may influence the semantic relation of the sentence or that context changes as one progresses along the sequence before switching back to the original topic. Cao et al. discuss this complexity of nested relationships, and how two subject entities of a sentence may have their relation nested within another [15]. Their approach to extract relationships relies heavily on the LSTM machine learning model due to its excellent performance in this form of pattern recognition.

4.1 The Dataset

The SemEval 2010 conference had multiple tasks presented with research, specifically task 8 which came with a new semantic relation dataset. Hendrickx et al. wrote a paper discussing in detail their dataset, which totaled 10717 sentences categorized into nine mutually exclusive categories of semantic relations [6]. This dataset, in comparison to others, is relatively small in terms of instances count, however, it's size has allowed others to work easier with it and create great models from it and nine total types of relationships also works in the favor of those wishing to dabble with semantic relation extraction. Each instance has a predetermined label as well as indications as to the position of both entities, noting here that the dataset only deals with two entities at a time. A full summary of the dataset can be found on GitHub by user *teffland* [13].

4.2 Bidirectional LSTM Model

An LSTM (Long Short-term Memory) model is a form of recurrent neural networks used in artificial intelligence, focused on processing sequences of data rather than single data points. When analysing the sentiment of text, it is important to consider the whole sequence of words as linked points of data, as words bring meaning to each other rather than as single instances, which is why an LSTM network is first chosen. Here, the model can remember previous context which allows it to retain some information over time which assists in its predictions. The flow of information into each trainable parameter (cell) is regulated via three gates; input, output, forget. The forget gate is to regulate when certain values are 'forgotten' (overridden), so the model can retain certain values of terms that are not immediately neighbouring for longer. This memory is what makes the model so applicable to sequence classification. In a paper about a Bi-LSTM combined with other layers to improve semantic attribute prediction of fashion images, Shen et al. summarize well the purpose of an LSTM model; "LSTM architecture is good at processing long range context" [9]. However, there is only a forward flow of data and with our dataset the subject entity may exist at the end of the sentence which some of the model's nodes would be unaware of, affecting the outcome of the prediction. Ping et al. discuss the application towards text classification, in which they demonstrate improved results for this task applying the LSTM model with a word2vec text representation matrix [11]. Similar to their project, this one applies a GloVe embedding for word representation, provided by the Standford NLP team. GloVe is an unsupervised algorithm capable of obtaining multi-dimensional vector representations of words, and pre-trained files are available which raise the accuracy of identifying semantic relations between words [8]. To further improve the model, a second LSTM layer is introduced to the model, which allows for a bidirectional flow of data, hence the Bi-LSTM model. Then, our contextual data has both forward and backward flow along the sequence and positions of the sentence subjects are not necessarily required at the front of the sequence for accurate predictions, and we can observe the relationships of long-distance dependencies. The result is that the whole context of the sentence is considered when predicting the semantic relation.

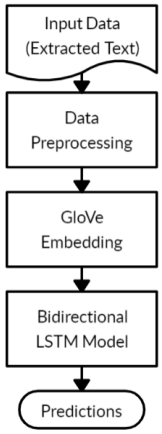

Figure 3: Visual representation of Bi-LSTM model architecture

The Bi-LSTM model was compiled with the python3.5 Keras library, a high-level API that runs over the TensorFlow library [1]. Loss functions are an important choice and essential to compile a Keras model. Due to the multi-class classification of the dataset used for this project, the chosen loss function was *categorical crossentropy*, to optimize for our total ten classes. Finally, since both the GloVe and Bi-LSTM layers use dimensions of over fifty, we require our output to be of ten dimensions representing our total pick of classes. For this, two dense layers are used, which make transformations on the results to lower the dimensions. Two are used so that there is not one single transformation to ensure that data dropout doesn't occur during the compression. Finally, this project makes use of the *softmax* activation function for the final dense layer, as it aims to create a probability distribution of the input vector, which is what we want to predict the most likely relation class for the input sentence. Figure three is a visual presentation of the overall model this project has built and applies to obtain its predictions.

5 APPLICATION OF THE BI-DIRECTIONAL LSTM

Like many other similar projects, there is now an existing and trained model able to classify sequences into multiple classes of semantic relationships (defined by the SemEval dataset). However, what this project aims to accomplish is to allow easy application of this model to data outside of the dataset and observe the results, potentially assisting a user in their search for knowledge through unstructured data. Figure four demonstrates the user flow of the software and the major steps required to produce results.

5.1 Pre-Processing of Data

The dataset provided the sentences with two entities tagged as *<e1>entity</e1>*. To ensure that the model would be able to understand where an entity was, this format was split into three tokens; *<e1>*, *entity*, *</e1>*. This was to simplify the process of feeding data into the model by avoiding generating another stream of data, which would have been in this case indexes of entity tokens for each sentence. All text was tokenized and labelled using the natural

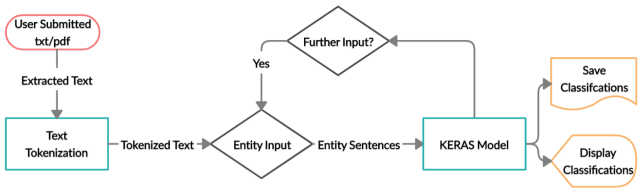

Figure 4: User flow for predictions with data outside of the SemEval dataset

language processing python library spaCy, a very powerful tool with pre-trained entity recognition algorithms. It is claimed to be the integrating tool that seamlessly links pre-processing tasks like tokenization with python machine learning libraries. Once pre-processed, the total 10717 sentences of the dataset were used for training and validation of the model, with each step using a batch of 40 for the model to work with.

Foreign data would be presented in .pdf or .txt format, where all text would be extracted for prepossessing. With spaCy, we were able to find all instances of entities during the tokenization process. From there, permutations of all entities paired with the requested entity generated tagged sentences of the dataset format, which the model could accept and classify. To later present sentences, a small cleanup function removes entity tags so the user is able to see the original sentence.

5.2 User Interaction

The python programming language comes with the Tkinter library, which is the standard for building a python GUI. For this reason, the project will be using this library to build a basic GUI for user interaction with predicting classes. It will allow a user to browse through a specified directory where they may select the file to extract text from. Then the data will be processed and allow for entity input and finally the user will be able to save all predictions from one resource to a text file for further analysis.

6 COMPARISONS & RESULTS

Validating the model on the total SemEval dataset, the Bi-LSTM model achieved 96.2% for validation, but this was on sentences already seen by the model. Figure five is the outputted confusion matrix from that validation result, which shows the distribution of predictions, both accurate and inaccurate. Herein, the potential label names are also displayed.

From figure five, it is clear that the dataset does not have an even distribution of examples. The 'other' label, or sentences unable to be classified, has roughly double the examples of some of the other labels and certainly the most. Some labels, like instrument-agency, have an extremely low amount in comparison. This imbalance perhaps did not allow the model to understand enough the patterns of some labels well enough to predict accurately. This is reinforced by the work of Mozetic et al. in their paper on multilingual twitter classification [5]. Herein, they discuss how classification model accuracy is much more dependent on the quality and size of the dataset it uses. With datasets of hundreds of thousands of sentences in comparison to ours, this may have hindered. For testing, a variety

	Cause-Effect	Component-Whole	Content-Container	Entity-Destination	Entity-Origin	Instrument-Agency	Member-Collection	Message-Topic	Other	Product-Producer
Cause-Effect	1294	0	0	0	6	1	0	4	6	4
Component-Whole	0	1190	4	1	3	8	6	13	20	1
Content-Container	0	3	721	1	0	0	0	0	3	0
Entity-Destination	0	0	21	1093	0	0	0	1	10	0
Entity-Origin	5	1	0	1	941	2	0	1	9	1
Instrument-Agency	0	3	1	0	0	623	1	0	17	7
Member-Collection	0	3	0	1	0	1	889	1	16	1
Message-Topic	0	1	0	0	2	0	0	872	13	0
Other	16	23	11	24	28	10	19	15	1675	23
Product-Producer	4	2	0	1	7	3	0	1	12	909

Figure 5: Confusion matrix of validation results on the total dataset

of different material was used; two news articles on cryptography and the Higgs-Boson particle, The Great Gatsby novel, and the first Harry potter novel. This selection was to present the model with very different writing styles and perhaps unseen vocabulary that the GloVe file would not have entries for. From testing on these materials with all possible combinations of valid sentences (two recognized entities), similar distributions of label predictions were seen among all of these materials. Figure six shows the distribution of the Harry Potter and Higgs-Boson predictions below and is a good representation of the same patterns with other foreign material.

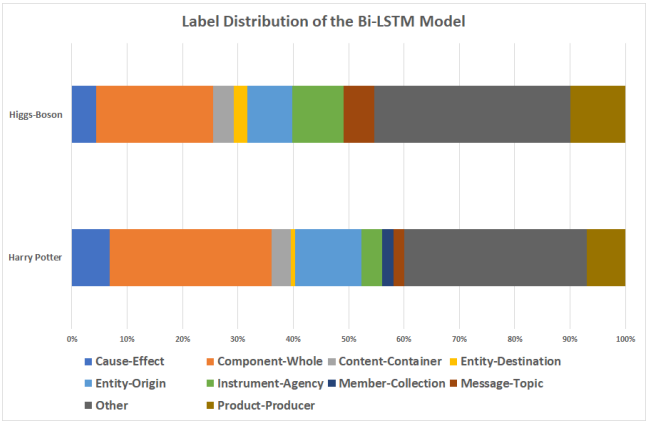

Figure 6: Label distribution of all valid sentences for Harry Potter 1 & the Higgs Boson article

All the foreign material was very different in vocabulary and purpose; novels or educational for example. We, therefore, believe that these patterns between the foreign data should not be existing and this shows inaccuracy outside of the dataset. Perhaps though this is again due to the fact that this model is not well refined and the data set too small to allow the model accurate predictions on new data because it does not give enough instances of each label. These patterns show perhaps that the model had to assign certain percentages of sentences to certain labels in order to continue the pattern it had found with the original dataset and is not very flexible.

Optimally, when results were varied and indicating a more correct

set of predictions instead of pattern outcomes, a group of participants would be needed. Human validation on sentence labelling still remains the most accurate, above machine learning models so the model's predictions would need to be sampled and verified to confirm its accuracy.

7 CONCLUSION

This project looks to see if current machine learning models trained to identify semantic relationships can be applied outside of their training datasets accurately. This would then speed up the acquisition of information from unstructured texts when a user is able to identify what entity they are searching for information around. Building a simple Keras Bi-LSTM model, sentences from the SemEval dataset and foreign sentences have been turned into sequences of vectors and classified into nine specific labels and a tenth non-conforming label. The model performed well within the dataset, scoring high on the validation. The foreign data presented to the model went through the classification, but patterns in the label distributions were present. These patterns indicate that the model was not making accurate predictions but rather ensuring certain percentages of predictions were made.

7.1 Future Work

Those that look to build the next best semantic relation models with new and more rich datasets should look to leave an opening at the very least where one can develop prediction functions on top of their model. With the current rate of progression in this field, these models may soon have the potential to be very applicable and available to the general user in assisting with the search for knowledge through unstructured text. More and better categorized labels would improve the quality of information extracted, which is dependent on the dataset. Many datasets today have hundreds of thousands of sentences, which allow a model to better learn the patterns in sequences and more accurately classify with fewer epochs. For such variety in the presentation of information, only ten different labels also seems very low and constricting. FewRel, by Han et al, is an example of a more modern dataset, released in 2018 with over 700,000 sentences and over one hundred types of labels [14]. Similar work to this project should aim to work with more modern and rich datasets like FewRel.

Should there be a large number of valid sentences labelled, finding the more useful sentences the user is looking for may be the next step by filtering those that seem less 'important'. Duk Kim et al. have written about text summarization for the specific task of mining for opinion and employ heuristics to find the explanatoriness of a sentence that they will choose to add to their summarization. Using mathematical rankings they came up with, they try to rank sentences based on three heuristics that contribute to a rich opinion-based sentence; length, popularity, and discriminativeness, which nicely lends itself to unsupervised summarization [4]. Therefore, one could after labelling all sentences find similarities between them and pick summarizing sentences, should facts reoccur.

REFERENCES

[1] Multiple contributors. [n.d.]. Keras: The Python Deep Learning library. https://keras.io/ Last accessed 15 March 2020.

[2] NLTK 3.5 Documentation. [n.d.]. Chapter 8. Analyzing Sentence Structure. https://www.nltk.org/book/ch08.html

[3] Shady Elbassuoni Gjergji Kasneci and Gerhard Weikum. 2009. MING: Mining Informative Entity Relationship Subgraphs. *18th ACM Conference on Information and Knowledge Management* (2009), 1653–1656. https://doi.org/10.1145/1645953.1646196

[4] Meichun Hsu ChengXiang Zhai Umeshwar Dayal Hyun Duk Kim, Malu G. Castellanos and Riddhiman Ghosh. 2013. Ranking explanatory sentences for opinion summarization. *36th international ACM SIGIR conference on Research and development in information retrieval* (2013), 1069. https://doi.org/10.1145/2484028.2484172

[5] Jasmina Smailović Igor Mozetič, Miha Grčar. 2016. Multilingual Twitter Sentiment Classification. *PLoS ONE* (2016). https://doi.org/10.1371/journal.pone.0155036

[6] Zornitsa Kozareva Preslav Nakov Diarmuid O S'eaghdha Sebastian Pado Marco Pennacchiotti Lorenza Romano†† Stan Szpakowicz Iris Hendrickx, Su Nam Kim. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. *Proceedings of the 5th International Workshop on Semantic Evaluation* (2010), 33–38. https://www.aclweb.org/anthology/S10-1006

[7] Emily Jamison. 2011. Using grammar rule clusters for semantic relation classification. *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics* (2011), 46–53. https://doi.org/10.1145/1835804.1835902

[8] Christopher D. Manning Jeffrey Pennington, Richard Socher. [n.d.]. GloVe: Global Vectors for Word Representations. https://nlp.stanford.edu/projects/glove/ Last accessed 17 March 202.

[9] Xueming Li Mengling Shen, Xianlin Zhang. 2019. Attentional Bi-directional LSTM for Semantic Attribute Prediction. *Proceedings of the 3rd International Conference on Video and Image Processing* (2019), 217–221. https://doi.org/10.1145/3376067.3376074

[10] Python Programming. 2019. Tokenizing Words and Sentences with NLTK. https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/ Last accessed 1 November 2019.

[11] Chaoyi Huang Qiancheng Liang, Ping Wu. 2019. An Efficient Method for Text Classification Task. *Proceedings of the 2019 International Conference on Big Data Engineering* (2019), 92–97. https://doi.org/10.1145/3341620.3341631

[12] Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. *22nd ACM international conference on Conference on information & knowledge management* (2013), 2369–2374. https://doi.org/10.1145/2505515.2505601

[13] teffland. 2010. SemEval2010 Task 8. https://github.com/teffland/Relation-Extraction/tree/master/SemEval2010_task8_all_data/SemEval2010_task8_testing Last accessed 25 February 2020.

[14] Pengfei Yu Ziyun Wang Yuan Yao Zhiyuan Liu Maosong Sun Xu Han, Hao Zhu. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018), 4803–4809. https://doi.org/10.18653/v1/D18-1514

[15] Hongwei Li Ping Luo Yixuan Cao, Dian Chen. 2019. Nested Relation Extraction with Iterative Neural Network. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), 1001–1010. https://doi.org/10.1145/3357384.3358003