SANDIA REPORT
SAND2021-0000
Printed September, 2021

Sandia National Laboratories

Relationship Extraction: Automatic Information Extraction and Organization for Supporting Analysts in Threat Assessment

Katrina J Ward, Jonathan Bisila, Jamini A. Sahu

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

In order for analysts to be able to do their work, they sift through hundreds, thousands, or even millions of documents to make connections between entities of interest. This process is time consuming, tedious, and prone to potential error from missed connections or connections made that should not have been. There exist many tools in natural language processing, or NLP, to extract information from documents. However, when it comes to relationship extraction, there has been varied success. This project began with a goal to solve the relationship extraction problem which developed into a deeper understanding of the problem and the associated challenges for solving this problem on a general scale. In this report, we explain our research and approach to relationship extraction, identify other auxiliary problems in NLP that provide additional challenges to solving relationship extraction generally, explain our analysis of the current state of relationship extraction, and postulate future work to address these problems.

ACKNOWLEDGMENT

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

We want to acknowledge Dr. Damon Woodard at the University of Florida, along with his students Anushka Swarup, Avanti Bhandarkar, and Enes Grahovac for their work as a University partner. Their work contributed greatly to the curation of several relationship extraction algorithms, along with implementation, data sets, and analysis. Their help has been invaluable to this research on relationship extraction.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

This research is aimed at providing a solution to analysts to help them discover, organize, and extract critical information from structured and unstructured documents, making their jobs faster and less prone to errors due to information loss or faulty conclusions derived from the data. Our goal is to leverage natural language processing (NLP) and advance relationship extraction technology to accurately pull information from multiple documents. In our research, we have discovered multiple auxiliary problems in NLP that are critical to be solved in conjunction with the problem of relationship extraction itself. Therefore, in this report we discuss these auxiliary problems and how they relate to the problem of relationship extraction and the current state of relationship extraction techniques. We also present our initial proposed solution, how these problems impacted our approach, and how we can move forward with new research in this area.

NOMENCLATURE

Table 0-1.

Abbreviation	Definition
DOE	Department of Energy
GUI	Graphical User Interface
NLP	Natural Language Processing
UF	University of Florida
pos/POS	Part-of-Speech
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
CRNN	Convolutional Recurrent Neural Network
BLSTM	Bi-directional Long Short Term Memory
GNN	Graph Neural Network

1. INTRODUCTION

In order for analysts to gather and process information, they sift through hundreds, thousands, or even millions of documents to make connections between entities of interest. Entities are considered to be any objects of interest such as people, organizations, locations, money values, language cues, etc. These documents can be structured, which includes formal reports, news articles, or other edited content. They can also be unstructured such as social media, article comments, emails, or other free form communication. The processing of these documents is time consuming, tedious, and prone to potential error. For example, information could be missed from the many documents that might be crucial to understanding important connections between entities. Also, connections could be formed that are not necessarily true, such as people or organisations with the same name. Finally, the current approach by analysts to gather, analyze, and interpret data by hand is not scalable and is both time intensive and tedious.

There exist many tools in natural language processing, or NLP, to extract information of interest from documents. Tools like SpaCy[7] and Stanford's CoreNLP[17] are able to perform a number of NLP tasks including named entity recognition, part-of-speech tagging, entity and co-reference resolution, sentiment analysis, and dependency tagging over a number of spoken and written languages. They primarily function by engineering a data pipeline through which text is processed and information is inferred or extracted in steps. In this report, we will discuss some of these NLP tasks, the current state of tools to perform the tasks, and their challenges in regards to relationship extraction. In addition to the NLP tasks above, these tools also have some capability in relationship extraction. They are able to detect some simple and well-defined relationships, though they often fall into the misconception that if two entities exist in the same sentence, they therefore must be related.[29, 2, 1] Calculating two entities are in the same sentence is both trivial to calculate and does not necessarily indicate a relationship. Another limitation of current approaches is that they only consider each sentence individually rather than trying to infer connections across multiple documents. In addition, neither approach is scalable to handle multiple documents. [5, 7, 29, 2, 1]

The above issues are what this work was aimed to address. Our ideas were to enable relationship extraction across multiple documents with both structured and unstructured text and present it in a way, such as a graph structure, that an analyst can use to see connections between entities easily when performing their own assessments.

A large portion of relationship extraction research was initially performed in the early 2000's and involved rule-based approaches that were tedious and not scalable. [26] This was a result of both the amount of data needed to process and for the number of rules needed to account for every possible situation, even in structured text. Later approaches, such as CoreNLP, looked into supplementing rule-based approaches by adding in part-of-speech tagging and using parts-of-speech to understand the context of words and how they can imply relationships. Most

relationship extraction research stopped in the early 2000's due to the shift towards deep learning to solve complicated problems. While deep learning was originally developed in the 50's, its popularity and surge in use was much more pronounced around 2006.[26] It became clear that deep learning had potential to solve the relationship extraction problem, however, few tackled the work until later when there was a more developed understanding of deep learning approaches. In the process of understanding the problem, we saw this trend and decided to take a step back and try to look at the problem before the innovation and inclusion of deep learning techniques, with the intention of augmenting our work later with these techniques to create a more scalable and robust solution. However, what we found in the process of our research is a significant resurgence and prominence of recent work in relationship extraction alongside deep learning.[23, 28, 18, 40, 13, 16, 37, 21, 9, 27, 14, 24, 36, 25, 34, 15, 31, 39, 30, 19, 38] These will be further discussed in section 2 for related work.

In this work we were able to create simple relationship tuples from multiple sentences and place them into a graph, however, our results did not perform better than current approaches. What we did find was that relationship extraction is closely connected to other NLP problems that are also still open problems. We believe advancing these areas in conjunction with relationship extraction is crucial to advance the problem of relationship extraction on a larger and more general scale. Our University partners performed a survey of many relationship extraction approaches, including implementation of algorithms from recent top papers and the curation of multiple datasets for training and testing. While we did not accomplish what we initially set out to do, we have learned a tremendous amount of information about the relationship extraction problem, other NLP topics, and the most recent advances of state-of-the-art.

In this paper we will first talk about other related work towards relationship extraction in section 2, including very recent advances over the past few years. We will explain how these approaches help our goals as well as where they fall short of meeting our needs. In section 3, we will explain our initial proposal and idea for solving relationship extraction based on successes from previous techniques. We will then highlight additional natural language processing problems in section 4 and how those problems directly impact relationship extraction. In section 5 we shall present the work contributed by our University partner and finally conclude with a summary and future work in section 6.

2. RELATED WORK

In this section we will talk about the most current work in regards to relationship extraction. We will discuss the basic idea behind their work and tools existing using their work. All of the work presented here has been tested in our previous work[29] or has been implemented by our University partner, to be discussed further in section 5.

2.0.1. Current Working Tools

Many of the current working tools such as SpaCy[7], CoreNLP[17], and BERT[5] are large projects that began using simple methods such as rule-based matching and parts-of-speech parse tree creation. Later, tools like Xlnet[32] and GPT-2 depended entirely on deep learning which helped solve some problems such as entity extraction, but still failed to solve many relationship extraction tasks beyond what the earlier methods accomplished. Later, SpaCy and CoreNLP adopted a hybrid method by leveraging the power of deep learning and supporting it with various other techniques such as word vectors to indicate which words to look for and what relationship they indicate. They additionally incorporated part-of-speech parse trees to understand the context of a word and to find relationships. Our initial hypothesis was to utilize a similarly hybrid approach, using parts-of-speech and parse trees in conjunction with deep learning methods to improve and iterate upon what we found from both domains.

2.0.2. Latest Literature

In addition to tools commercially available, there have been several works in recent years addressing relationship extraction, all of which take some form of a deep learning approach. Some approaches use convolutional neural networks, CNNs, to take features of a sentence such as specific words or phrases or even a sequence of types of words, to classify parts of the sentence as a particular type of relationship.[23, 28, 13, 16, 9, 34] These works have relationship categories where there is a set of expected relationships and patterns that place sentences or sentence fragments within those relationship types. These methods often require datasets that are preprocessed to be in a clean, specific format and/or include significant metadata information. However, data is not typically available in this way and this is a time consuming process.

Other works look at different forms of CNNs such as convolutional recurrent neural networks(CRNN)[21]. The purpose of these approaches is to limit the negative effects of noise in data on the learning process by processing the data through the neural network twice. In the first pass, context is embedded into the sentences based on word embeddings and then a second pass is done to reduce weights on outlier classifications. This helps with mislabeling of relationships but

is restrained to a specific domain and set of relationships as well as still having the same issue as CNNs in that the longer a sentence is, the more likely it will be mislabeled or labeled as an outlier.

The final group of CNNs explore is Graph Neural Networks(GNN).[39, 36, 27] The goal of these works is to extract both entities and their relations into a graph structure to understand how the entities are connected. This approach is most similar to the approach we intended to take in that it places entities as nodes and relations between them as the edges. In addition, [36] went further to prune the graph like a dependency tree to obtain the shortest path between nodes. This however, caused many relationships to be lost. Like other CNNs, these models focus on a set of relations and other relations not in the set are lost. In our work, we focused on maintaining as much information as possible, since losing information could be a critical failure point for an analyst.

Some approaches took a different deep learning approach and use Long Short-Term Memory, or LSTM, architectures. [37, 15, 31] In these approaches, data is defined to specific domains and structured to certain types of sentences to ensure they have better supervised data. They are also designed to get around the need for other NLP processing such as parts-of-speech tagging and entity extraction. The training data is then turned into feature vectors where entity positions are marked and the neural network discovers sequences of entities, the entity types, and the words connecting them to create a language model that can be used to predict sequences needed to detect specific relationship types. Like entity extraction, the entities first need to be known and annotated, which makes this solution very close to those using an entity extraction tool. Additionally, like all the other approaches, this solution begins to fail when the sentences are longer and contain more than one clause within them.

Finally, there are methods that use the transformer and encoder/decoder architectures to solve the problem.[30, 19, 38] Like the LSTM approaches, the goal is to identify entities while in the process of performing other NLP tasks instead of before. They work by encoding sequences and using those sequences to identify further representations of those sequences for labeling. Transformer and encoder/decoder models perform well in entity context awareness, semantic analysis, test prediction, and other common NLP tasks. However, all of them including Bert and GPT3 have stated that relationship extraction falls far short of expectations[3]. While this approach seems to be the standard for NLP tasks right now, they do not solve this problem.

3. INITIAL PROPOSAL

Natural language processing, particularly entity-relationship extraction, is not a new topic. Entity extraction refers to pulling out subjects of interest from text such as people, places, dates, etc. Several tools have established algorithms for identifying entities and their categories. SpaCy[7], CoreNLP[17], BERT[5], and XLNet[32] are some of the more recent and most successful approaches using word dictionaries, word vectors, databases, parse trees, deep learning, or a combination of all of the above. Entity linking is closely related and also supported by these tools. Our original proposal was based on the understanding that these tools were the current state of the art and given their severe limitations, there existed no real capability to solving relationship extraction. This is explained in further detail in a previous tech report[29]. In this section, we will discuss our original idea and justifications for our approach.

Our proposed solution took a different approach than the current state of the art by focusing on the parts-of-speech and applying relationship meaning to them. We will leverage successes from CoreNLP and expand upon them using a new algorithm.

Consider the following excerpt of text as an example:

"Viruses mutate all the time, and most mutations have no significance even if they spread," said Adriana Heguy, director of the Genome Technology Center at New York University, who was not involved with the research.

Assume that an analyst is looking into Adriana Heguy as part of a threat assessment of a new pathogen. In our approach, we will start by leveraging what current tools do well already and parse individual sentences to identify small, easy facts. We replace pronouns with the exact entities they reference. For example, in the first sentence of our example, “they” would be replaced with “viruses”. We would also identify the entities and the labels they belong to, including custom labels for topics of interest. This is done so that we know what we are finding relationship information for. In this case, we would have the following:

Viruses, biological
Mutations, biological
Adriana Heguy, person
Director, person
Genome Technology Center, organization
New York, Location
New York University, organization

As mentioned before, identifying entities is well-researched, therefore we will use the best tools identified in our previous report[29]. As we discovered in that report, if training to recognize

many categories of entities, training multiple small models is more efficient and accurate than one large model for identifying different entity labels. This is due to the fact that multiple entity types can share common patterns. For example, if we had the "John Doe" and "American Airlines" as entities, we intuitively know one is a person and one is an organization. However, to a model, they share the same pattern and have a greater probability of being labeled the same. We found that having multiple smaller models with fewer entity types gave us higher accuracy in correct entity labeling. Therefore we planned to spend some time training multiple accurate models. However, we learned that we were able to reduce some of the work on this by simply using the parts-of-speech tagging capability and find nouns for a more general solution.

Once we have the information we want to learn, we organized the sentences into parts-of-speech parse trees, similar to the CoreNLP approach. Instead of connecting only a single sentence into the tree, we inserted larger chunks of text into a single tree. While we recognized a potential scalability issue, we found that processing a single tree for each document to be connected later, keeping a dictionary of unique entities and maintaining an index of both would address this issue. In the tree, entities are connected by the relationships between them.

In CoreNLP, the words themselves were the relationships. For example, using our first sentence above, CoreNLP would recognize that “Viruses verb mutate”, “mutations negation verb spread”, “Adriana Heguy of noun Genome Technology”, Center at noun New York University”, etc. While this is not entirely unhelpful, it can be difficult to decipher and understand. We believe that we can organize parts-of-speech into relationship groups, like clusters, to apply better meaning. In our case, we would look at the following:

- Injunctions are removed as they do not supply useful information
- Nouns are topics of interest and are assumed to be entities extracted in the first step.
- Pronouns are also entities, however can indicate possession such as his, hers.
- Verbs indicate action or a state of being. Put into terms of relationships, it is something done to something.
- Adjectives apply description to a single thing or collection of things as a whole. These words connect to an entity to provide more details.
- Adverbs provide little useful information at this time. They can indicate intensity of an action, however at this early stage we are concerned with only detecting the verb. The exception is words indicating negation.
- Prepositions can indicate compound relationships that extend beyond two nouns in a sentence.
- Conjunctions are similar to prepositions in that we can learn if multiple (3+) entities are related or not by considering logical words such as “and” and “or”, or understanding that multiple relationships in a sentence are connected with words like “because”.

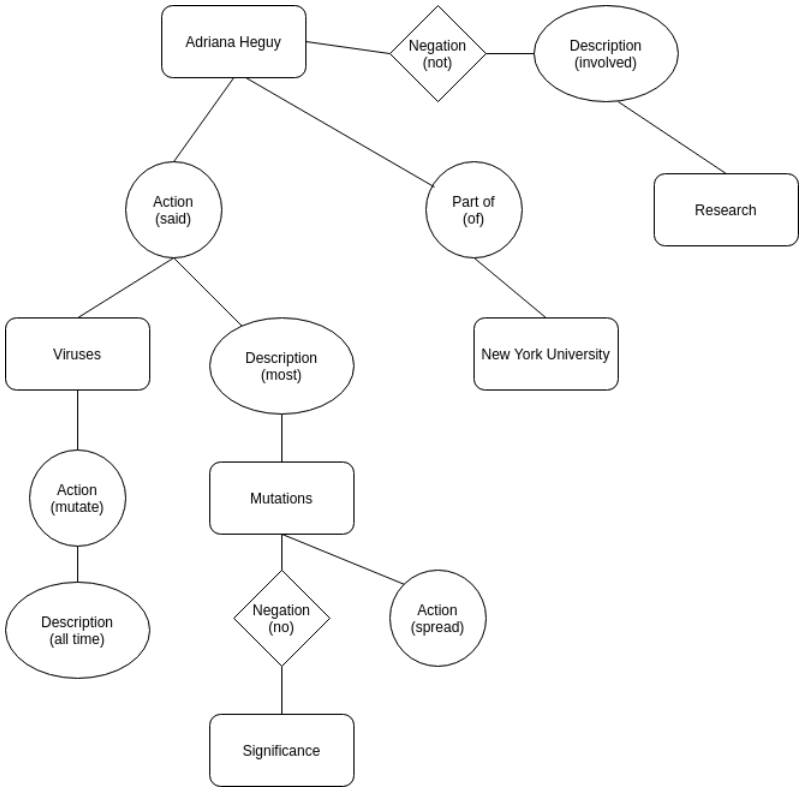

Figure 3-1. Example parse tree using parts-of-speech

If we parse the sentence above using our approach so far, we would have something similar to figure 3-1. In the figure, we have mapped the relationship groups that are built from particular parts-of-speech to the entities of interest. If we were to continue, later paragraphs would be connected to the same tree, providing a larger picture of the relationships in the text. Creating a new parse tree is not enough, however. To help with efficiency and scalability, we borrow a concept from database design where the tree is optimized through a series of steps to remove redundant information and make finding information faster while giving the best results[6]. In figure 3-1 we can see that not every word in the sentence is included. Irrelevant or redundant words are pruned from the tree to reduce the data size. Redundant branches where the same relationship is represented would also be pruned such that only one instance of it existed to further reduce the data size. Using this tree, we can then infer both explicit relationships by direct connections in the tree(Adriana Heguy said most mutations spread), as well as implicit relationships from indirect connections(Adriana was not involved in research about virus spread).

Once these relationships are extracted, we wanted to bring the information important to an analyst to the top by introducing a rank ordering algorithm. The idea is to calculate a score for each relationship statement by considering the following factors:

1. Is the relationship statement explicit or implied?

2. Does the statement mention the subjects the analyst is looking for?

3. Is the relationship a direct relation or through other entities?
4. Does the relationship mention sensitive topics?
5. How many connections does the entity have?

Based on these criteria, we would then create a list of relationships of interest to the analyst.

With a rank order list of facts extracted, we can insert them into an adjacency list where each entry contains two entities known to be connected by some relationship and document what type of relationship was found connecting them. This adjacency list would be entered into a graphing tool where high-ranking facts are highlighted and where an analyst can see a clear and organized view of entities and their relations as well as their connections.

The proposed solution was successful in some ways, but failed in others. While we were able to extract simple relations from sentences, we learned that in building the parse trees we were being pulled back to rule based matching, which is the early stage naive solution. Even while using parts-of-speech in sentences, many complex sentences required additional help and rules in order to piece the trees together. In addition, many of the NLP components such as parts-of-speech tagging and co-reference resolution to resolve pronouns turned out to not be as much of a solved problem as we believed. In the cases we found in our preliminary research where tools had solved the problem, it was in fact for very specific datasets tailored to those tools and solutions. In section 4 we will illuminate additional NLP problems and how they affected our work.

4. NATURAL LANGUAGE PROCESSING AUXILIARY PROBLEMS

Natural language processing encompasses many objects and problems needing to be solved. When we think of NLP, we tend to focus on the tasks we want to use it for such as semantic analysis, text prediction, and answering questions about text. However, underneath the applications of NLP are the core problems that when solved facilitate and enhance these capabilities. This section will talk about these core problems, their strengths and limitations, and how they apply to relationship extraction. As stated in our previous work, while relationship extraction depends on entity extraction, we have learned it further depends on many of these other problems being improved in conjunction.

4.0.1. Entity Extraction

Entity extraction is the process of finding things or topics of interest from text. For example, if we have the sentence:

"Apple is looking at buying U.K. startup for $1 billion."

We could detect:

Apple -> Organization
U.K. -> Geopolitical Entity
$1 billion -> Money

An example is shown in Figure 4-1 using SpaCy's visualization capability and in Figure 4-2 for the text representation.

Some tools already do this well and have assembled and published pre-trained models for use to find general entity types or to be trained further to be more accurate. Alternatively, these tools provide you with the framework such that you can train your own models. By default, they work by finding patterns in words or phrases that indicate a particular entity type. Money values can be detected by a money marking character like "$" or by common money indicators such as "USD".

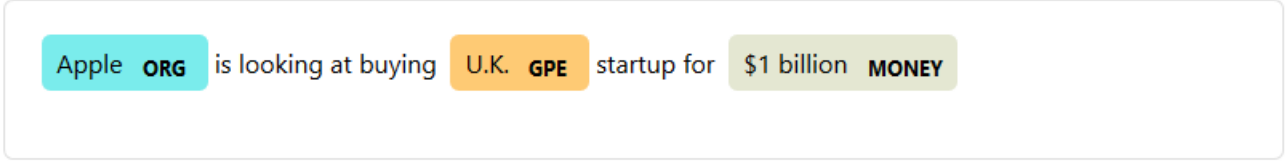

Figure 4-1. Visualization of entity extraction using SpaCy's visualizer. [7]

TEXT	START	END	LABEL	DESCRIPTION
Apple	0	5	ORG	Companies, agencies, institutions.
U.K.	27	31	GPE	Geopolitical entity, i.e. countries, cities, states.
$1 billion	44	54	MONEY	Monetary values, including unit.

Figure 4-2. Entity extraction example using SpaCy. [7]

Things like phone numbers, names, and other entities also have distinct patterns. In many cases, the model can be trained to identify uncommon patterns such as IP addresses, online tags for people, or topic trends marked with hash tags. Developing custom models can be improved by applying word vectors to give the models examples of which words or word types are of interest.

Entity extraction is critical for relationship extraction as mentioned above. Relationships are often represented as relationship triplets which comprise of two entities and a relation. Many approaches start with entity extraction to identify entities and their positions, while later coming back to see which words connect them. Other approaches such as the LSTM architectures try to identify the entities at the same time as finding the relationships in order to understand context. Regardless, identifying entities is a key component to most relationship extraction techniques.

As of now, this is the among the simplest tasks of NLP tools and we have found it to be a well-researched topic. However, that does not mean it is without challenges. Scalability remains an issue. Fortunately, most current tools can be run in a parallel way. One additional obstacle comes in that some languages or particular datasets are more challenging and complex than others and as such entities that share a similar pattern create many errors. Still, compared to other NLP problems, entity extraction is one of the most developed auxiliary problems.

4.0.2. Entity and Coreference Resolution

Entity and coreference resolution is the task of identifying and resolving multiple representations of the same entity. For example, let us consider the following sentence:

"John Doe went to the site to complete his project. Jane said her husband works long hours."

In the above sentence, "John Doe" and "his" [project] would be recognized as the same person. However in the second sentence, "her husband" is ambiguous, and could refer to John Doe, or another mentioned entity previous to or after this particular sentence. This illustrates the difficulty of coreference resolution in text.

When it comes to relationship extraction, coreference resolution is another critical component. When making connections between multiple entities, recognizing multiple representations of an entity reduces the complexity of the final output data, reduces mislabeling of entity-relation triplets, and makes the results much more clear and less prone to error in the form of marking two entities as separate when they are in fact the same. In addition, when we look at the first step of

entity extraction, pronouns are often overlooked and not extracted as entities, whereas specific names are. Detecting and applying specific entity labels based on whom the pronouns are referencing gives us more data for understanding context and relationships.

In our work we saw that tools capable of coreference resolution worked on simple sentences like those above. However, when it came to more complex sentences containing multiple clauses, the tools failed. Incorrect entities were applied to pronouns and sometimes entities were linked to some other entirely different entity. We believe this is because the algorithms within the tools depend on word sequence and parts-of-speech to gain some context to decide which words are actually referencing the same thing. However, in more complex sentences, the word sequences are not so clear and this ambiguity can cause error. In our work, we turned to sentence simplification to get around this problem and were met with some success. However, sentence simplification has its own challenges as mentioned below. For tools such as SpaCy[7], coreference resolution initially appeared to be solved, though we found it to only work well on their own specific dataset and not in general practice. In reality, entity and coreference resolution is in the similar stage of research as relationship extraction.

4.0.3. Parts-of-Speech and Dependency Parsing

Parts-of-speech (POS) tagging is a well-documented task that lays the foundation for many other NLP tasks. It involves labelling the words in a sentence with their parts-of-speech. For example, consider the following sentence:

"Apple is looking at buying U.K. startup."

Figure 4-3 shows how a POS tagger would label the words in the above sentence.

In the beginning of our research, we performed a complexity and time analysis of different POS implementations. We included four of the most popular options: Stanza (Stanford CoreNLP's python library), SpaCy, NLTK, and TextBlob. We found that, in small document sets (100 sentences), all performed relatively similarly. As the number of sentences increased, the Stanza implementation fell behind the others in run time. Despite that, the Stanza implementation remained the most accurate.

Dependency parsing is the task of analyzing a sentence's grammatical structure using dependencies between the words in that sentence. This builds directly off of POS tagging, using those speech tags to determine the relationships between words. SpaCy has a fairly robust dependency parser implementation, which creates a tree structure of a sentence's dependencies, as well as a dependency visualizer called displaCy. In Figure 4-4, we see what the dependencies would appear as for the sentence above using SpaCy's visualizer. Our goal was to utilize dependency parsing primarily for use in sentence grammatical simplification. We believed this would assist in detecting the relationships located in sub-clauses of a sentence that relationship extraction algorithms in practice have a harder time finding. Finally, we hoped that it would help improve the results in coreference resolution techniques, as the simplified structure would make it less ambiguous who a pronoun was referring to. However, as we discuss in the next section, this was met with limited success.

TEXT	LEMMA	POS
Apple	apple	PROPN
is	be	AUX
looking	look	VERB
at	at	ADP
buying	buy	VERB
U.K.	u.k.	PROPN
startup	startup	NOUN

Figure 4-3. An example of a parts-of-speech(POS) tagger result.

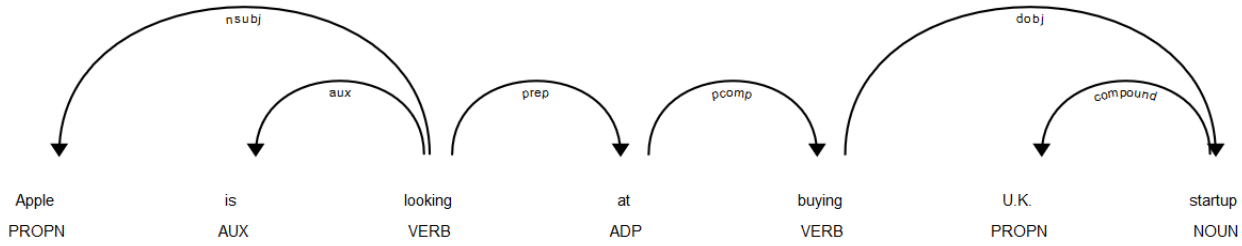

Figure 4-4. An example of a dependency parse tree example result.

4.0.4. Sentence Simplification

Sentence simplification is the task of modifying text so that it is easier to read, but overall maintains its original meaning. For example, consider the following sentence:

"Owls are the order Strigiformes, comprising 200 bird of prey species."

One possible simplification of this sentence might be:

"An owl is a bird. There are about 200 kinds of owls."

In this simplification, there are a couple important changes. The language itself is simpler and the complex sentence structure was broken down into two single-clause sentences.

Our goal in exploring the sentence simplification task was to reduce the complexity of the sentences in our data so that coreference resolution would perform better on them. To achieve

this, we tried using dependency parse tree representation of the sentences to break them into clauses. Both NLTK and Stanford CoreNLP offer implementations to derive a parse tree from a sentence. In some cases, these were successful in separating clauses using parts-of-speech to determine when new clauses began. NLTK's implementation used noun subjects and direct objects to make this determination, and Stanford CoreNLP used subordinate clause labels to do so. Unfortunately, this approach still worked better on already simple sentences and performed less effectively with sentences of more than two clauses.

Other implementations we explored for the task only replaced complex words with simpler synonyms, a task also known as lexical simplification. This was was insufficient for our goal, as the sentences in our data were most often too long and grammatically complex rather than being too complex in terminology. We found that this limitation of most algorithms performing lexical simplification, as opposed to grammatical simplification, to be an obstacle for using this on a larger, more general scale. Additionally, one other challenge arrives in that lexical simplification can often change the meaning of the sentence, as the connotation between two synonyms or the usage of a particular term may be different between the original and the simplified form. This loss of information due to the simplification, especially when a part of an automated text analysis pipeline, can introduce error into the data without an analyst ever being aware.

4.0.5. Validation

A especially poignant challenge for us, though not specifically an NLP domain problem, is validation. Nearly all of the tools and the research validate their approach using precision, recall, and F-score. These metrics are excellent in determining how much of the data the models are getting correct answers for and how consistent the answers are. Furthermore, they are great for comparing accuracy to other approaches. However, they fail to give intuition as to why those scores are what they are and what sorts of failures may have occurred. For example, there is a possibility that certain types of sentences are the most difficult for a particular approach while others are always correctly marked. In our work, we discovered almost no documented context for where the methods failed and no developed methodology to really understand why certain types of approaches or architectures failed or succeeded in the ways they did. A means for fuller analysis and validation needs to be developed to understand current approaches in order to create a more robust understanding of the failure and success modes of these various architectures and approaches. This is something we wish to address in our future work as we continue to better understand relationship extraction approaches.

5. UNIVERSITY OF FLORIDA PARTNERSHIP

In this work we partnered with Dr. Damon Woodard and his students: Anushka Swarup, Avanti Bhandarkar, and Enes Grahovac at the University of Florida to leverage their knowledge and experience in NLP algorithms and tools. During their work, they implemented 21 relationship extraction algorithms.
[23, 28, 18, 40, 13, 16, 37, 21, 9, 27, 14, 24, 36, 25, 34, 15, 31, 39, 30, 19, 38] In addition, they curated nine (9) datasets [11, 4, 10, 33, 20, 8, 35, 22, 12] used in several NLP validation tasks and converted each to work with as many of the 21 algorithms as possible. With the ability to test the most recent work in relationship extraction, we are able to see what the current state-of-the-art algorithms are able to accomplish while also seeing how robust they are to general datasets or if they are specially tailored to specific ones. The results of this survey are pending publication in ACM Computing Surveys and this report will be revised when publication is confirmed.

In addition to the publication, we want to extend our work with Dr. Woodard to look at some questions developed during this research. While none of the papers presenting the algorithms above discuss in detail how they failed or why, we wish to look deeper into what types of sentences do fail. Some questions we want to address are:

- Do certain types of algorithms(CNN, LSTM, etc) fail on specific types of sentences?
- Are there sentences with specific patterns that many of the algorithms fail on?
- Can we map the progression of other NLP components critical to relationship extraction to the success and/or failure of the latest relationship extraction algorithms?
- What correlation is there between how effectively one of the auxiliary problems cleans a sentence and the subsequent effectiveness of a relationship extraction algorithm on that processed sentence?
- Are there combinations of tools and approaches that complement one another better for improved results?

The results of these questions in combination with the algorithm analysis is planned for a potential journal publication with Dr. Woodard and his students. This report will be updated with the results of such work.

6. CONCLUSION

In this paper we presented the problem in natural language processing called relationship extraction and how its application could assist analysts in their jobs to collect information and efficiently organize it to be least prone to error. We discussed our proposed solution to the problem, and while there were some successes in our approach, there were also some challenges that we did not predict. However, despite not solving the problem as we intended, we investigated the most current state-of-the-art algorithms and their approaches to gain a better understanding of the problem and why an ideal, mission-space-ready solution still does not exist. We also learned about auxiliary problems in NLP and how some affected our approach. We were able to present this information for use in future work. While we recognize relationship extraction is still not a solved problem, we have identified topics in NLP that need to be solved in conjunction with the problem of relationship extraction to improve generalizability and scalability. In addition, we worked closely with our University partners to get a set of 21 algorithms and nine datasets in order to conduct tests for a publication and to thoroughly analyze for future work.

REFERENCES

[1] Bernt Andrassy, Pankaj Gupta, Subburam Rajaram, and Thomas Runkler. Neural relation extraction within and across sentence boundaries, January 21 2021. US Patent App. 16/517,161.

[2] Nguyen Bach and Sameer Badaskar. A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15, 2007.

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[4] Linguistic Data Consortium and New York Times Company. *The New York Times Annotated Corpus*. LDC corpora. Linguistic Data Consortium, 2008.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Ramez Elmasri and Shamkant Navathe. *Fundamentals of Database Systems*. Pearson, 7th Edition, 2015.

[7] Explosion. spacy: Industrial-strength natural language processing. `https://spacy.io/`, 2021. Accessed September 2021.

[8] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[9] Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245, 2018.

[10] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[11] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of

nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[12] Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. Improving distantly supervised relation extraction using word and entity based attention, 2018.

[13] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, 2016.

[14] Sunny Lai, Kwong Sak Leung, and Yee Leung. Sunnynlp at semeval-2018 task 10: A support-vector-machine-based method for detecting semantic difference using taxonomy and word embedding features. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 741–746, 2018.

[15] Joohong Lee, Sangwoo Seo, and Yong Suk Choi. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing, 2019.

[16] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, 2016.

[17] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[18] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

[19] Tapas Nayak and Hwee Tou Ng. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8528–8535, 2020.

[20] David Orr. 50,000 lessons on how to read: a relation extraction corpus. `https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html`, 2013. Last Accessed - September 2021.

[21] Desh Raj, Sunil Sahu, and Ashish Anand. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)*, pages 311–321, 2017.

[22] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[23] Yatian Shen and Xuan-Jing Huang. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536, 2016.

[24] Robyn Speer and Joanna Lowry-Duda. Luminoso at semeval-2018 task 10: Distinguishing attributes using text corpora and relational knowledge. *arXiv preprint arXiv:1806.01733*, 2018.

[25] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7072–7079, 2019.

[26] Mike Thomas. The history of deep learning: Top moments that shaped the technology. `https://builtin.com/artificial-intelligence/deep-learning-history`, 2020. Accessed in September 2021.

[27] Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*, 2018.

[28] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, 2016.

[29] Katrina Ward, Jonathan Bisila, and Kelsey Cairns. Survey of current state of the art entity-relation tools. Technical Report SAND2020-9355, Sandia National Laboratories, Albuquerque, NM, September 2020. USDOE National Nuclear Security Administration (NNSA).

[30] Shanchan Wu and Yifan He. Enriching pre-trained language model with entity information for relation classification, 2019.

[31] Peng Xu and Denilson Barbosa. Connecting language and knowledge with heterogeneous representations for neural relation extraction, 2019.

[32] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[33] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*, 2019.

[34] Zhi-Xiu Ye and Zhen-Hua Ling. Distant supervision relation extraction with intra-bag and inter-bag attentions. *arXiv preprint arXiv:1904.00143*, 2019.

[35] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network, 2015.

[36] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

[37] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017.

[38] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

[39] Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 219:106888, 2021.

[40] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212, 2016.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	L. Martin, LDRD Office	1910	0359

Email—Internal [redacted]

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.