CLUSTERING EVENTS BASED ON COMMON SUBEVENTS

An Undergraduate Research Scholars Thesis

by

MAITREYI RAMASWAMY

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Ruihong Huang

May 2020

Major: Computer Science

TABLE OF CONTENTS

ABSTRACT

Clustering Events Based on Common Subevents

Maitreyi Ramaswamy
Department of Computer Science
Texas A&M University

Research Advisor: Dr. Ruihong Huang
Department of Computer Science
Texas A&M University

The objective of this project is to create a methodology to find communities of similar events based on their context, which is represented by their subevents. The similarity is measured through a new metric we propose, which takes into account the similar subevents between events. The motivation behind clustering these events into larger labeled groups is to enrich the overall understanding of each individual event. The event and subevent relationships have been extracted using a weakly supervised event acquisition method and have been stored in a knowledge base. Using these pairs and the idea of hierarchical event representation, we cluster the events, which will provide insights on the similarities and differences between events in context.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Ruihong Huang and my mentor Wenlin Yao for their guidance and support throughout the course of this research.

CHAPTER I

INTRODUCTION

Communication is built on conveying ideas. The way we understand these ideas, in this case events, is by way we characterized them. Subevents characterize the individual idea/event and learning how the multitude of ideas being conveyed in a message relate to one another facilitates understanding.

As humans, when reading text mentioning an event, let us say a protest, we might describe it with conflict and possible activities such as chanting, marching, burning flags, or riots. We are able to enrich our understanding of an event with other events, or subevents, that describe the original event. The relationship between events and subevents characterize the event, as knowing the typical subevents of an event aid in gaining information about that event. This information could include the sentiment of the event or provide insights on what the event is about such as when it happened, where it happened, or even unforeseen details that will characterize the event in a different way.

The motivation behind the idea of clustering such existing events stems from the similar notion of general understanding and association of ideas. Clustering events based on their subevents will provide insights on the similarities and differences between events in context. After clustering we hope to find relationships between events as expected, but also find some strong relationships between events we did not expect.

Word2vec or other similar non-contextual embeddings try to create an all- encompassing representation for events [1]. When the embeddings are created, the position of the event in the corpus, thereby the context of the event, is not accounted for. However, extracting hierarchical

event and subevent relationships is based on the position of the events and subevents in text, which creates an enriched event representation. For instance, the event “fire” has different meanings when attached to arguments in the events “to start a fire”, and “to fire a person”. In the former event, the subevents might “to burn wood” or “to light a match”, but in the latter, a subevent might be “person looks for job”. The example demonstrates the importance of relative location of an event in its representation as it encapsulates its context based on the subevents.

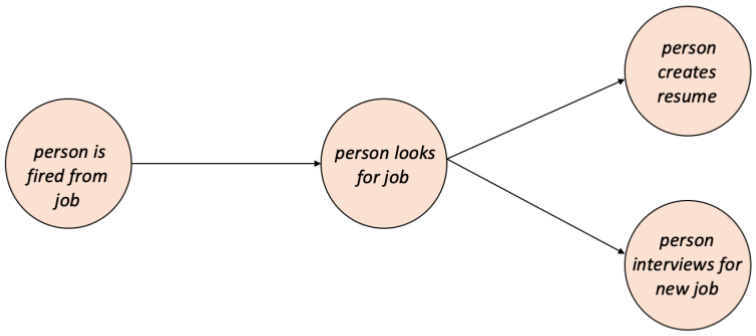

Figure 1: Example of a multilevel hierarchical event/subevent relationship

Events have relationships with subevents that help enrich their understanding; however, subevents also act as events in that they have relationships with their own subevents to help characterize them. In Figure 1 we show an example of a multilevel hierarchical relationship between events and subevents.

We present the idea of clustering events which can be represented in a hierarchical fashion and use a bipartite graph to represent a hierarchy. We can introduce a new metric to measure similarity between every pair of events at a given level. The metric is computed based on the number of shared subevents between the two events, and the weights of the corresponding edges. This metric can then be utilized to characterize the events at each level into clusters,

which gives us insights into the intra-level relationships. The measure of similarity represents how closely two events within the same level are related based on each of subevents, and the generated clusters may show events of common sentiment, events that have common occurrences, or an unprecedented grouping of events.

CHAPTER II

METHODS

We can represent the relationship between the event and subevent as shown in Figure 2. Because there is a hierarchical relationship between them, we are able to represent them in a graphical form, using a directed edge to connect the parent, the event, and child, the subevent. We encapsulate the arguments of the child, and the weight of the edge in the actual edge itself, creating a structure that can be visualized in Figure 2.

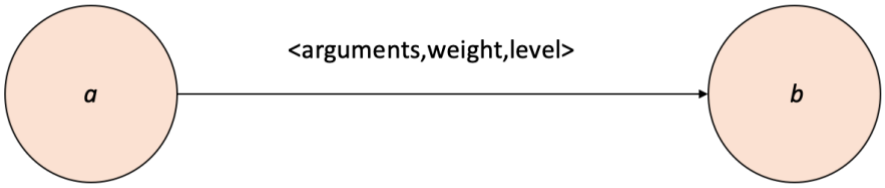

Figure 2: This figure shows the visualization of the edge encapsulation

We add the concept of levels as an attribute to the edge representation; a level represents the event's position in the hierarchy. Due to the nature of the hierarchical relationship between events, we can assume that some events will not be children of other events, making them at the top of the hierarchy, assigning their level attribute to be one. Similarly, those events that are children of events in level one will be in the second level, those that are children of events in the second level will belong to the third level, and so on. In our implementation, we uniquely identify an event based on arguments, weight, and its level. Two events with the same arguments and weight, but in different levels will be considered two different events, ensuring that an event will not occur across multiple levels.

By ensuring that the same event will not occur in multiple levels, we can represent such event relationships using a bipartite graph structure. The level of the events will correspond to the layer they belong to in the bipartite graph; The events in level one can be visualized in the first layer of the bipartite graph, the events in the second level can be visualized in the second layer, and so forth. Once the bipartite graph is created, we compute the similarity between each pair of events at a given level using the metric. The metric represents similarity between two events, but also distance; the smaller the value between two events in the same level, the more similar they are.

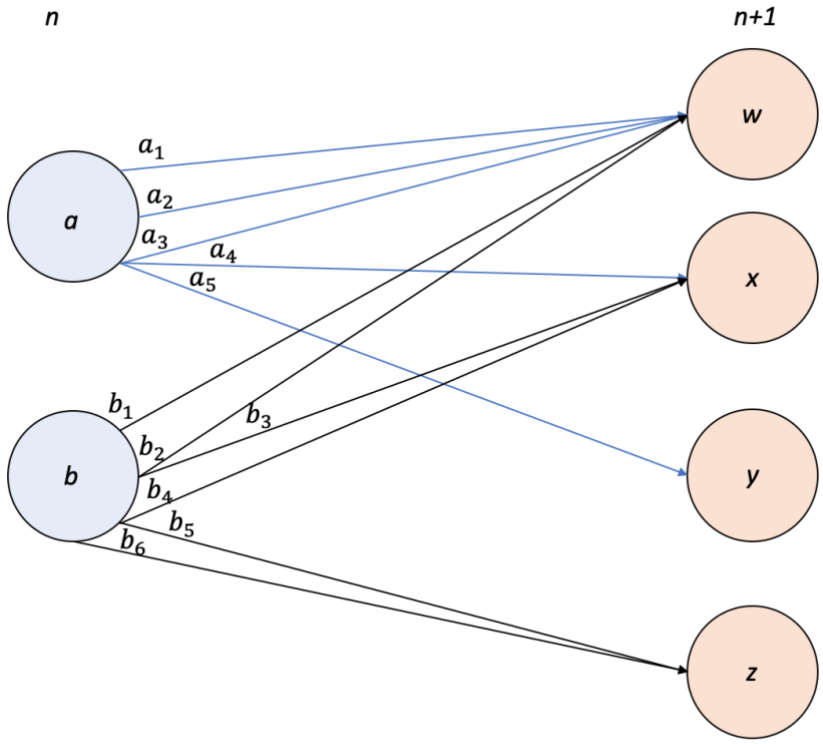

Figure 3: The figure shows the representation between two levels. Events *a* and *b* are in level *n*, and events *w*, *x*, *y*, and *z* are in level *n*+1

Let's say we have two events, *a* and *b*, in a level *n*. In order to compute the similarity score, we first find all of the common subevents between *a* and *b* in layer *n+1*. For each common

subevent, we want to find how closely they are related to *a* and how closely they are related to *b*. We do so by finding the ratio between the sum of the weights of the edges between *a* and the subevent, and the sum of the weights of the total outgoing edges of *a*, and we do the same for *b*. We use a ratio to find the importance the relationship between a parent and a subevent. The higher the ratio, the higher the importance of that subevent in computing the similarity. Once we have the ratios for *a* and *b*, we find the absolute difference. The lower the difference, the higher importance in computing the overall similarity between *a* and *b*. Once we have computed this difference value for each subevent, we find the product of each value.

$$S_a = all\ direct\ subevents\ of\ a$$
$$S_b = all\ direct\ subevents\ of\ b$$
$$E_{a,i} = all\ edges\ from\ a\ to\ i$$
$$E_{b,i} = all\ edges\ from\ b\ to\ i$$
$$A = \bigcup_{\forall i \in s_a} E_{a,i} \qquad B = \bigcup_{\forall i \in s_b} E_{b,i}$$
$$p(a,b) = \prod_{\forall i \in s_a \cap s_b} \left| \frac{\sum_{\forall j \in E_{a,i}} j}{\sum_{\forall k \in A} k} - \frac{\sum_{\forall j \in E_{b,i}} j}{\sum_{\forall k \in B} k} \right|$$

Figure 4: Generalized metric computation where a and b are events in the same level

We perform this computation for every pair within a level in the bipartite graph. For example, in Figure 5, we have five events in the shown level. Figure 6 is a matrix that represents the similarity between each pair of elements. The similarity between an event and itself is zero,

and metric is symmetrical. Once we have the similarity between each pair of events at every level, we can use hierarchical clustering methods, specifically agglomerative clustering, to perform intra-level clustering.

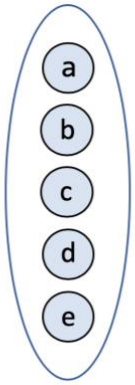

Figure 5: Level *n*

	a	b	c	d	e
a	0	*p(b,a)*	*p(c,a)*	*p(d,a)*	*p(e,a)*
b	*p(a,b)*	0	*p(c,b)*	*p(d,b)*	*p(e,b)*
c	*p(a,c)*	*p(b,c)*	0	*p(d,c)*	*p(e,c)*
d	*p(a,d)*	*p(b,d)*	*p(c,d)*	0	*p(e,d)*
e	*p(a,e)*	*p(b,e)*	*p(c,e)*	*p(d,e)*	0

Figure 6: Distance Matrix for Figure 5. The metric is symmetrical, $p(a,b) = p(b,a)$

We have a knowledge base that contains extracted event and subevent pairs from English Gigaword, a corpus is made up of 10 million news articles [2]. The extraction is done using a weakly supervised event acquisition method, and the weights between the edges of the event and subevent are assigned during extraction using word embeddings.

In order to validate the metric, we randomly sample half of the clusters in each level, and manually analyze the subevents of each event in the cluster. Looking at the cluster, if there are any events that seem dissimilar, we compare the subevents of each event to those of the other events in the cluster and ensure that there is overlap amongst the subevents.

CHAPTER III

RESULTS

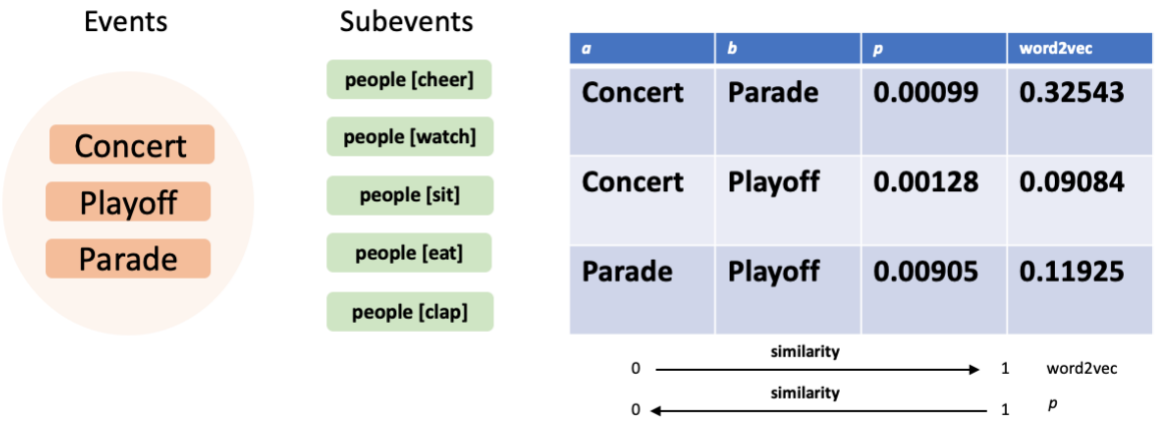

Figure 7: Comparison between word2vec and *p*

Figure 7 compares the pairwise similarity computed using our metric and word2Vec. The events “Concert”, “Playoff”, and “Parade” are clustered together using our proposed methodology. We see that the word2vec solution does not evaluate these words as similar, however our metric, does. This is because the actual definitions of the words are not similar, but according to our knowledge base, they have similar subevents. When analyzing the subevents, we see that the subevents common to these events are those such as “people [cheer]”, “people [watch]”, “people [eat]”, “people [sit]”, and “people [clap]. From the knowledge base, we can assume that news articles may have reported on these large events where people congregate as audiences to watch a performance and engage in the activities described by the subevents, which is why our metric evaluates these three seemingly unrelated words as similar.

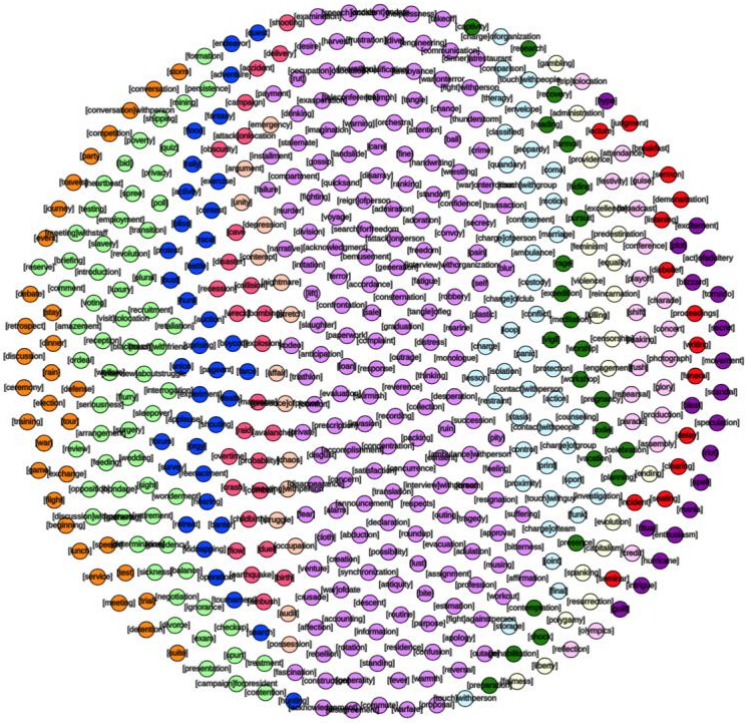

Figure 8: A level of the knowledge base clustered in which every color represents a different cluster

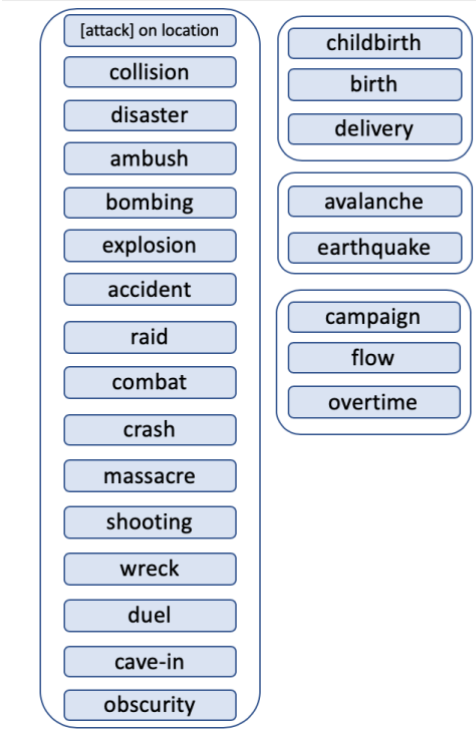

Figure 9: Events of a cluster in the level represented in Figure 8

All the words in Figure 9 are the same level and cluster. With an understanding of all these events, we manually grouped the ones that we thought were related. we created four groups; war and battle, childbirth, natural disaster, and politics. While initially these groups were seemingly unrelated, we remember that the corpus used to generate the knowledge base was made up of news articles. Often times, the news reports tragedy. On further examination, we noticed that all of the events had subevents similar to “people being[killed]”, “person [dying]”, “[loose] person”, and “fight[person]”. These subevents, while unfortunate, accurately described the original events in the context that they were present in.

CHAPTER IV

CONCLUSION

We have presented a method for clustering events based on their subevents within knowledge base. It would be interesting to see how relationships between events and subevents differ based on the context they were extracted from. For example, how the same event extracted from a corpus made up of novels is differently characterized from the same event extracted from news articles. We would also explore the limitations of our metric and see if there are any improvements to be made given new findings across knowledge bases. In addition, we would like to explore the idea of assigning events to multiple clusters in order to see the relationship between clusters within a level. Insights into such relationships can help in applications of knowledge of events in overlapping or completely distinct clusters.

REFERENCES

[1] Mikolov, Tomas, et al. Distributed Representations of Words and Phrases and Their Compositionality. 2013, papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[2] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, pages 95–100. Association for Computational Linguistics.