IIT PALAKKAD

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

PALAKKAD, KERALA-678557

DEPARTMENT OF DATA SCIENCE

PHASE-2 PROJECT REPORT ON

EVENTS AND TEMPORAL EXPRESSIONS EXTRACTION

ADVISOR

DR. KONINIKA PAL MADAM

ASSISTANT PROFESSOR

SUBMITTED BY

SHRIPAD A. CHABUKSWAR

142002020

15 MAY 2023

Events and Temporal Expressions Extraction

1 Introduction

Introduction: Natural language processing has experienced significant advancements in recent years, playing a critical role in a wide range of applications. One of the key tasks in NLP is the extraction of temporal expressions and events. The identification and referencing of specific moments in time and events within a given timeframe hold immense significance across diverse domains, such as finance, healthcare, and law.

This project aims to focus on the extraction of temporal expressions and related events from the QSearch dataset[1]. To achieve this, we will utilize a rule-based model called Heideltime[2] and a handcrafted event extractor. The primary objective is to create a labeled dataset that encompasses temporal expressions and events. Subsequently, we will train a neural network model to automatically extract temporal expressions and events.

Rule-based models serve as a common approach for extracting temporal expressions and events by defining specific patterns and rules. However, these models often face challenges when dealing with complex or ambiguous expressions. To overcome these limitations, our project involves the development of a new dataset dedicated to training and evaluating temporal expression and event extraction models.

To construct this dataset, we extracted sentences from the QSearch dataset[1], which consists of research papers. We applied the Heideltime rule-based model to identify temporal expressions, while we created a custom rule-based model to extract events by identifying verbs within the context object. Finally, we utilized the dataset to train a neural network model based on Facebook BART-base architecture[3].

Our approach represents a significant improvement over traditional rule-based models for temporal expression and event extraction. By combining rule-based models with machine learning techniques, we have developed a more accurate and efficient model suitable for various NLP applications, including search engines, chatbots, and virtual assistants.

For example, consider the following sentence: "Barack Obama (born August 4, 1961) is an American politician who served as the 44th President of the United States from 2009 to 2017." Our model can accurately identify that the birthdate of Barack Obama is a situational event, falling under the event property category, while the presidency of Barack Obama is a duration event, classified as a type of temporal expression. This example underscores the importance of accurately identifying and comprehending temporal expressions and events within textual data.

In conclusion, our project aims to create a labeled dataset of temporal expressions and events by employing rule-based models and integrating them with machine learning techniques. This approach has the potential to enhance the performance of numerous NLP applications and contribute significantly to the field. By automating the extraction of temporal expressions and events, we facilitate a deeper understanding of time-related information within textual data, paving the way for improved information retrieval, question answering systems, and more sophisticated language processing applications.

1.1 Problem Statement

The project aims to use the Facebook BART-Base sequence-to-sequence model[3], for extracting temporal expressions and related events from Qsearch dataset sentences [1]. The approach involves creating a dataset that includes annotated temporal expressions using Heideltime temporal tagger [2], and handcrafted rule-based annotations for events. By training the BART model on this dataset, the objective is to automatically extract the temporal expression and related event and also improve the accuracy and effectiveness of extracting temporal expressions and events, enabling a better understanding of the temporal relationships and events.

2 Literature Review

Qsearch: Answering Quantity Queries from Text[1]: This paper discuss method for extracting quantity fact (Qfacts) from natural language text using deep learning neural networks. The method involves preprocessing the input text corpus by detecting entities and quantities, performing Named Entity Disambiguation (NED) and quantity detection using the Illinois Quantifier tool, and replacing each identified quantity with a placeholder. In the next step, the Qfacts are extracted from the preprocessed sentences using a bi-directional LSTM model for sequence labeling, which tags each token of the sentence as <E> (denoting the entity that the quantity refers to), <X> (denoting the context tokens that relate the quantity and its entity), or <O> (for all other tokens). The model employs word embeddings, the position of the pivot quantity, and entity recognition as input features, and uses a BIO tagging mechanism to tag multi-word entities. The output of the model is the probabilities of each token word being tagged with <E>, <X>, or <O>. Overall, the method is effective for extracting Qfacts from natural language text.

Extracting Events and Temporal Expressions A Literature Survey [4]: This survey provide a overview of various approaches for extracting temporal information from text, including rule-based, machine learning, and hybrid methods. The paper also discusses different types of temporal expressions and their features.

BART: Denoising Sequence-to-Sequence Model[3] This paper introduces BART, a state-of-the-art pre-training method for natural language processing tasks. BART uses a denoising autoencoder to train a sequence-to-sequence model on a large corpus of text. The resulting model can be fine-tuned for various downstream tasks, including machine translation and summarization.

Heideltime Rule base model[2]: Heideltime is a rulebased temporal tagger that extracts temporal expressions from text and normalizes them into standardized formats. The tool is designed to work with multiple languages and can handle a variety of temporal expressions, including dates, times, durations, and frequencies.

To extract temporal information, HeidelTime first pre-processes the input text to identify linguistic features such as part-of-speech tags and named entities. It then applies a set of flexible rules, based on regular expressions and language-specific patterns, to identify temporal expressions in the text. Once identified, each temporal expression is classified into a specific type, such as DATE, TIME, or DURATION, which enables HeidelTime to apply appropriate normalization strategies.

HeidelTime normalizes temporal expressions to standardized formats, such as the ISO-8601 format for dates and times. For example, "January 1st, 2022" might be normalized to "2022-01-01". HeidelTime can also handle more complex temporal expressions, such as durations and sets. Overall, HeidelTime is a powerful tool for extracting and normalizing temporal information from text. It has a flexible rule-based approach and can be used for a variety of applications, including information retrieval, text mining, and natural language processing.

3 Dataset Creation

The success of the project relies on the quality and accuracy of the dataset use to train the machine learning model. The dataset consist of group of instances that have similar characteristics and serve different purpose in the overall system. To train the facebook bart model, we use the training dataset, and then we use the validation dataset to verify the models understanding of the data. By using subsequent datasets, we can further shape the model, and the more data we provide,the model can learn and improve.

We use a dataset from the Qsearch dataset that have already extracted the quantity, entity, and context of the sentence. We create a new dataset that contains sentence with temporal expression and related events. we took the qsearch dataset because temporal expression are in the form of quantity string. To extract the temporal expression, we use temporal tagger which generates output in the TIMEX3 standard annotation format. We only consider sentences that include a single temporal expression that matches the quantity string or any single token of the context to extract the event from a phrase.

When labeling the events in sentence, we select the entity string, the verbs from the context, and the quantity string tokens as event tokens if the temporal expression matches one of them. We then use these tokens to form an event substring, While training, we consider around 5,000 sentences. We select only those sentence extract the verb from the context of the Qsearch dataset and take it as an event because verbs mostly show action events. For example, in the provided example, we extracted the temporal expression and related event from a sentence, We identified the temporal expression as "18 June 2018" and the event as "go90-announced-eight-episodes," which contains the extracted quantity, entity, and context tokens.

`Example from Datset`

{'sentence': 'On 18 June 2018 , American video streaming service go90 announced eight episodes of " We Are CVNT5 " , which will expand on the original CVNT5 mockumentary .','quantity$'_1$ $: \{'quantity' :' (8.000; episodes; =)',' entityStr' :' go90', 'context' : ['18June2018',' American', 'video',' streaming',' service',' announced' ,' CVNT5'],' quantityStr' :' eightepisodes'\}, 'heideltime\_tag' : ['< TIMEX3tid = ''t3'' type = ''DATE''value = ''2018-06-18'' > 18June2018 < /TIMEX3 >'],' TE_quantity' : [1],$
$'Event' : ['go90 - announced - eight - episodes']\}$

3.1 Statistics Of Dataset:

Dataset: Here we taking of around 1.20 lakh sentences, out of a total of 1.6 million unique sentences from the Qfact dataset. Temporal expressions: we identified using Heideltime's temporal tagger, approximately 52,000 temporal expressions in the sentences. We Only considered those sentences with a single temporal expression for training the model, filtering out sentences with more than two temporal expressions to examine the model's behavior for a single temporal expression sentence before training with more complex sentences.

Single TE sentences: we obtain 37,000 sentences with a single temporal expression. More than two TE sentences: Around 15,000 sentences obtained with more than two temporal expressions.

BART model training: Approximately 5,000 sentences considered, with events extracted by selecting the verb from the context of the Qsearch dataset as verbs mostly indicate action events.

4 Model

Facebook/BART-base model:
The Facebook/BART-base model uses an encoder-decoder architecture to extract temporal expressions and related events from text. The architecture involves several mathematical operations, including the use of attention mechanisms and a

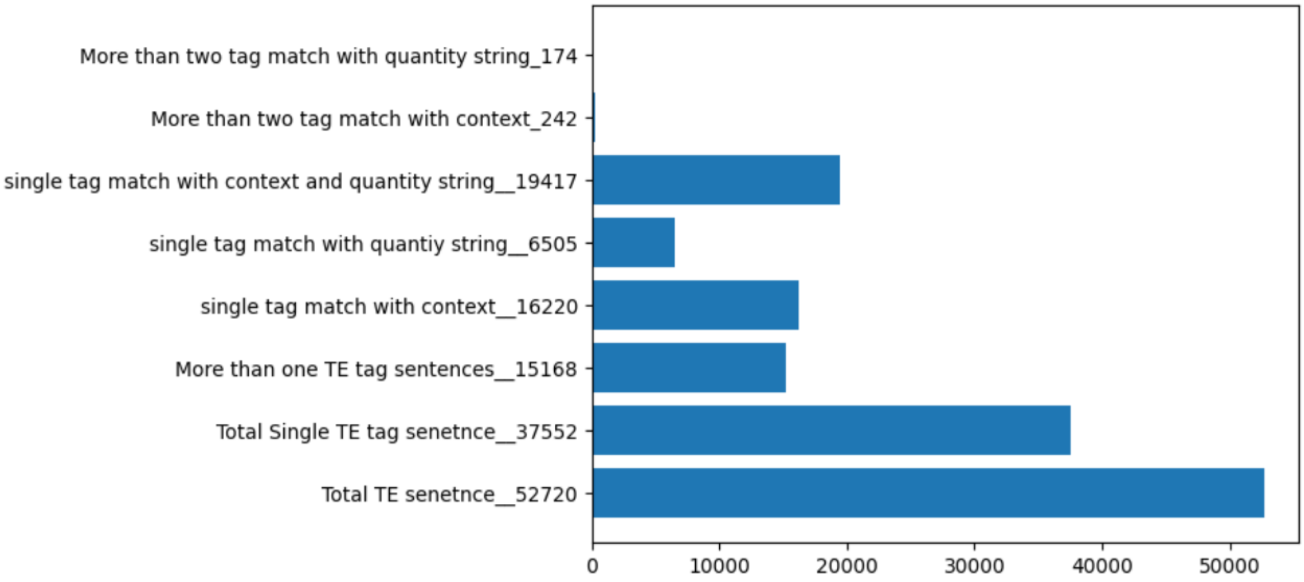

Figure 1. Statistics Of Dataset

transformer network. The input text is first processed by the encoder, which converts the input sequence of words into a sequence of continuous representations. The encoder uses a transformer network, which allows for parallel computation and reduces the computational complexity of the model. The transformer network uses self-attention mechanisms to capture the dependencies between different parts of the input sequence.

The decoder then uses the encoded sequence to generate a sequence of outputs. The decoder also uses attention mechanisms, which allow the model to focus on specific parts of the encoded sequence when generating each output. The attention mechanisms are based on a soft alignment between the decoder state and the encoder outputs.

The output sequence includes predicted temporal expressions and related events, which are generated using a softmax function to produce a probability distribution over a predefined set of labels. The model is trained using a cross-entropy loss function, which penalizes the model for making incorrect predictions.

Overall, the Facebook/BART-base model is a powerful tool for extracting temporal expressions and related events from text. Its use of attention mechanisms and transformer networks allows it to capture complex dependencies between different parts of the input sequence, making it well-suited for a wide range of natural language processing tasks.

4.1 Experiment and Result

In the experiment conducted, we aimed to extract temporal expressions and events related to them using a BART-based sequence -to-sequence model. We started with a dataset from the Qsearch dataset, which had already extracted the quantity, entity, and context of the sentences. We created a new dataset that only contained sentences with temporal expressions and related events. To extract temporal expressions, we used the Heideltime rule-based tagger and considered sentences with a single temporal expression that matched the quantity string or any single token of the context. We then extracted the event from the phrase using the entity string, verbs from the context, and the quantity string tokens as event tokens.

We trained the BART model using approximately 5,000 sentences where we only extracted the verb from the context of the Qsearch dataset and took it as an event because verbs mostly indicate action events. The dataset was split into a training set of 4,000 sentences and a validation set of 821 sentences. The Seq2Seq model was trained using the Hugging Face implementation of BART architecture with hyperparameters such as a learning rate of 1e-5, batch size of 32, weight decay of 0.001, and 20 epochs.

The evaluation of the trained model was performed on a test set, which consisted of 100 input output pairs not used in the training or validation process. The precision, recall, and accuracy values for the generated output by the BART model were calculated for temporal expression and also for event, indicating that the model has a reasonably good performance in generating output sequences that match the ground truth. However, there is still room for improvement, and increasing the size of the training data, fine-tuning hyperparameters, and trying different models or architectures can lead to better results. Overall, the experiment demonstrates that the BART model can successfully extract temporal expressions and related events from text to a significant extent.

5 Future Plan

Increase the size of the training data: By adding more sentences with multiple temporal expressions and events to the dataset, the model can learn from a larger and more diverse set of examples, which may improve its ability to extract temporal expressions and related events.

Experiment with different models or architectures: It may be useful to try other models or architectures, such as transformers or LSTM-based models, to see if they can achieve better performance than the BART-based Seq2Seq model.

Incorporate a more advanced temporal event tagger tagger: While the Heideltime and event rule-based tagger was effective, there are more advanced temporal and event taggers available that may improve the accuracy of the model's temporal expression extraction.

6 Evaluation

To make sure our results are accurate, we checked each of our models carefully. First, we looked at 100 random sentences from the Heideltime tagger model that had time expressions in them. We compared these sentences to the real answer and found the Heideltime model was right 78 percentage.

Next, we checked our rule-based model for finding events. We couldn't check it with real answers, so we looked at 100 sentences that had both time words and events in them. We found the rule-based model was right 72 percentage.

Finally, we checked precision and recall for Facebook/BART-base model that can change sentences to other sentences. We tested it on 100 sentences seprately for temporal expression and for event, the evaluation score is shown belows table$_1$.

Label	Precision	Recall
Temporal Exp	0.89	0.92
Event	0.829	0.854

Table 1. Evaluation Table

7 Conclusion

In conclusion, the experiment showed that the BART model is a highly effective tool for extracting temporal expressions and related events from text. By training the model on a relatively small set of data and using evaluation metrics like precision, recall, and F1-score, the BART-based sequence-to-sequence model is well-suited for a variety of natural language processing tasks. This finding is particularly relevant given the growing importance of processing temporal information in many fields, including finance, healthcare, and law.

References

[1] H. Vinh Thinh, Y. Ibrahim, K. Pal, K. Berberich, and G. Weikum, *Qsearch: Answering Quantity Queries from Text*, 10 2019, pp. 237–257.

[2] J. Strötgen and M. Gertz, "Heideltime: High quality rule-based extraction and normalization of temporal expressions," pp. 321–324, 08 2010.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[4] N. K. Gupta, "Temporal information extraction extracting events and temporal expressions a literature survey," 2015.