University of Hamburg

Master Thesis

Evelink-H: Grounding Event Mentions to Wikipedia on Harder Problems

Author:
Muhammad Hashaam Ahsan
Matr.nr.:
7485720
Course of Study:
Intelligent Adaptive Systems, M.Sc.

Supervisor:
Prof. Dr. Ricardo Usbeck
Secondary Supervisor:
Junbo Huang

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Semantic Systems Research Group
Department of Informatics

April 29, 2024

Abstract

Understanding an event mentioned in an article plays a pivotal role in enhancing reader comprehension. Event Linking, linking event mentions to knowledge graphs (KGs) provides readers with additional context and background information. However, linking an event to a KG, such as Wikipedia presents several unique challenges. One challenge is the ambiguity and variability, where multiple event mentions may refer to the same event, making it difficult to determine the correct Wikipedia page. Additionally, the hierarchical structure of events poses challenges in identifying the accurate Wikipedia page for a given event mention.

Overcoming these challenges is crucial for advancing in the field of information retrieval and knowledge representation. Our research[1] contributes to solving and formulating the challenges for the event-linking task. This thesis seeks to enhance the understanding of the recently established event-linking task, as outlined by [21], with specific attention to the harder problems mentioned above in linking event mentions to Wikipedia pages.

Das Verständnis eines in einem Artikel erwähnten Ereignisses spielt eine entscheidende Rolle für das Verständnis des Lesers. Event Linking, das Verknüpfen von Ereigniserwähnungen mit externen Wissensgraphen (KGs), versorgt die Leser mit zusätzlichem Kontext und Hintergrundinformationen. Die Verknüpfung eines Ereignisses mit einem Wissensgraphen, wie z. B. Wikipedia, stellt jedoch einige besondere Herausforderungen dar. Eine Herausforderung ist die Mehrdeutigkeit und Variabilität, da sich mehrere Erwähnungen eines Ereignisses auf dasselbe Ereignis beziehen können, was es schwierig macht, die richtige Wikipedia-Seite zu bestimmen. Außerdem stellt die hierarchische Struktur von Ereignissen eine Herausforderung dar, wenn es darum geht, die richtige Wikipedia-Seite für eine bestimmte Ereigniserwähnung zu finden.

Die Bewältigung dieser Herausforderungen ist entscheidend für Fortschritte im Bereich der Informationsgewinnung und Wissensdarstellung. Unsere Forschung trägt zur Lösung und Formulierung der Herausforderungen für die Aufgabe der Ereignisverknüpfung bei. Diese Arbeit soll das Verständnis der kürzlich etablierten Aufgabe der Ereignisverknüpfung, wie sie von [21] umrissen wurde, verbessern, mit besonderem Augenmerk auf die oben erwähnten schwierigeren Probleme bei der Verknüpfung von Ereigniserwähnungen mit Wikipedia-Seiten.

[1] Data and code are available here: `https://github.com/semantic-systems/EvelinkH`

Contents

1 Introduction

Serving as a vital aspect of natural language comprehension, the task of Grounding involves disambiguation and acquiring knowledge. With the exponential growth of textual data, the need to accurately ground words to their respective reference has become increasingly popular.

Entity linking, grounding entities to a KG has already demonstrated significance in various natural language processing (NLP) tasks, including recommendation systems, question answering, dialogue generation, etc. Numerous works such as [1], [13], [17], [7], [2] and [20] have contributed to these tasks, yielding exceptional results. While the progress brought by these contributions has been significant, we argue, that grounding only entities may not provide enough background to the reader for text comprehension.

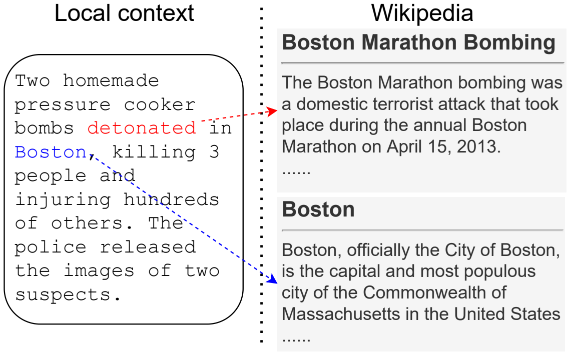
Local context
Two homemade pressure cooker bombs detonated in Boston, killing 3 people and injuring hundreds of others. The police released the images of two suspects.
Wikipedia
Boston Marathon Bombing
The Boston Marathon bombing was a domestic terrorist attack that took place during the annual Boston Marathon on April 15, 2013.
......
Boston
Boston, officially the City of Boston, is the capital and most populous city of the Commonwealth of Massachusetts in the United States
......

Figure 1: Example of Entity Linking vs Event Linking, the left side is the local context, and the right side contains Wikipedia pages. Entity linking model connects the entity “Boston” to the Wikipedia page “Boston”, while the event linking model links the event “detonated” to the Wikipedia page “Boston Marathon Bombing”, which is more relevant to the local context [21].

Take Figure 1 as an example; given the local context, an entity linking model would link the entity "Boston" to the Wikipedia page of "Boston", which provides information about the history and culture of the city. However, this information is irrelevant to the local context provided. To truly enhance the comprehension of the reader about the sentence, we need to link the event mentioned by the verb "detonated" to the Wikipedia page "Boston Marathon Bombing".

Inspired by the Oxford definition of an event, we defined an event mention as "A mention of a thing that happens or takes place, especially one of importance", and the task of linking event mention to the correct reference in KG (Wikipedia) as **Event Linking** [21]. This problem is relatively new compared to Entity linking and caught attention through some contributions such as [14],[9], and [21].

In this paper, our objective is to address and analyze the challenges in the task of linking event mentions to Wikipedia pages, highlighted by [21]. We were particularly intrigued by the more complex problems outlined in their work, such as sub-events and sub-section events, which pose significant challenges for event-linking tasks. Our work aimed to explore a deeper understanding of these challenges and tried an approach of improving candidate representations to effectively link sub-events and subsection events mentions to their corresponding Wikipedia pages.

Events exhibit a hierarchical structure, where larger events comprise various sub-events. These sub-events may either have their own Wikipedia page or be mentioned within a larger event Wikipedia page. The system should ideally link each event mention to the most suitable page. If a specific sub-event page exists, the system should link to it; otherwise, it should link to the page of the larger event. An example of a sub-event can be found in the context of "Stepping in at the 11th hour, Hillary Rodham Clinton will campaign in Florida on Saturday for her brother, Hugh Rodham, in his bid for a United States Senate seat", where the event mention "Hugh Rodham campaign" does not have its own Wikipedia page but is mentioned in the Wikipedia page of "1994 United States Senate Election in Florida".

Additionally in KG such as Wikipedia, the event might not always have its own Wikipedia page but might exist in the section of the Wikipedia page. As mentioned before, the system should link each event mention to the most suitable page. If a specific event page exists, the model should link to that page; otherwise, it should link to the page where the event is mentioned in the sub-section of the Wikipedia page. An example of this can be found in the context of "The Philippine government lifted its five-year ban on the return of Imelda Marcos today and said the widow of the late President Ferdinand Marcos was free to come home from exile in the United States.", where the event mention "return of Imelda Marcos" doesn't have its own Wikipedia page, and neither it is part of a larger event. It rather exists in the section of "Return from exile (1991-present)" on the Wikipedia page of "Imelda Marcos".

Based on the harder problems mentioned above, our research aims to answer the following questions.

- How can the most appropriate Wikipedia page be found in cases where the event mentioned is a sub-event of a larger event?
- How can an event mention be linked if it doesn't have its own Wikipedia page but exists in a sub-section of a Wikipedia page?

To sum up, our work contributes to challenges highlighted by [21] on harder problems such as sub-section events and sub-events. First, it includes an analysis of these challenges using the Wikipedia and NYT datasets provided by [21]. Second, it improves the model accuracy of Evelink [21] on these harder problems, indicating the importance of including candidates context. Third, it highlights the challenges with the architecture of the model Evelink [21]. Lastly, our research contributes to the extension of the Wikipedia dataset constructed by the [21].

2 Background

In this section, we discuss the important concepts needed to understand the task of event linking.

2.1 Event vs Entity Linking

Note that event linking poses different challenges as compared to entity linking. Generally, both are the process of linking a particular word (event/entity) from a given text to a knowledge base and come under the task of document linking. Also, depending on the context, some events may be referred to as entities, such as "World War II" being mentioned as a noun.

In contrast, entities usually appear as continuous text spans, while events are more structured objects. Events consist of a trigger word, usually a verb or nominal, with multiple arguments to support it. Unlike entities, events may not be fully defined by their trigger word and can involve several arguments making the linking process more challenging. In addition, events are also not extensively documented in Wikipedia [9]. As a result, the paper [21] introduced the category *Nil* for events that do not exist on Wikipedia.

2.2 Why Event Linking?

Understanding the events in a text enhances comprehension. However, often, they are not defined in detail where mentioned. For example in the sentence, "The fighting ended with the Armistice of 11 November 1918, while the subsequent Paris Peace Conference imposed various settlements on the defeated powers, notably the Treaty of Versailles.", the event "Paris Peace Conference" is not defined in detail making it harder for the reader to understand the context.

2.3 Challenges specific to Event Linking

2.3.1 Similar Titles

Events are not always unique; they can have multiple titles. For instance, titles like "Invasion of Poland" and "Occupation of Poland (1939–1945)" both refer to the same event: "the German Army invading Poland in 1939". The challenge lies in evaluating the event in a meaningful way to identify the correct title.

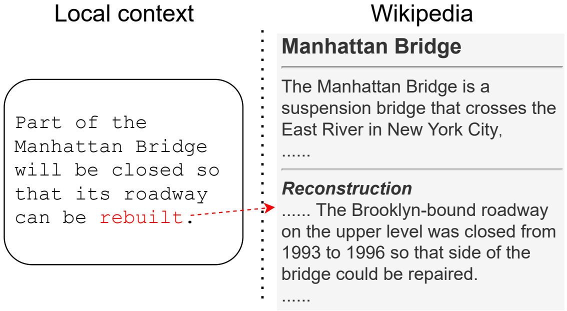
Local context
Wikipedia
Part of the Manhattan Bridge will be closed so that its roadway can be rebuilt.
Manhattan Bridge
The Manhattan Bridge is a suspension bridge that crosses the East River in New York City,
......
Reconstruction
...... The Brooklyn-bound roadway on the upper level was closed from 1993 to 1996 so that side of the bridge could be repaired.
......

Figure 2: Example of subsection event. Here it can be seen, that the event "rebuilt" mentioned in the local context (left) doesn't have its page and belongs to a subsection of the wiki page of "Manhattan Bridge" (right) [21].

2.3.2 Subsection Events

At times, events are only mentioned in subsections of a page without having their own dedicated Wikipedia page, which makes it challenging to link a specific event mention to the main page title. For instance, in Figure 2, the event "rebuilt" is referenced within a subsection "Reconstruction" of the main Wikipedia page "Manhattan Bridge", without having its own dedicated page. This example highlights the need for additional context from the Wikipedia page to facilitate the system's ability to link an event mention to the main page title.

2.3.3 Sub Events

Events often consist of sub-events, which may either possess their own Wikipedia page or be mentioned within a Wikipedia page of the bigger event. Event linking aims to link an event mention with the most appropriate Wikipedia page. In cases where a sub-event has its own Wikipedia page, the system is expected to link to its own Wikipedia page rather than to the bigger event Wikipedia page. For instance, as shown in Figure 3, the event "drafted" is linked to "Tom Brady's" Wikipedia page by the annotator. However, "Draft of Tom Brady" is a sub-event of the "2000 NFL Draft", which in turn is a sub-event of the "National Football League Draft". These event hierarchies lead to inconsistencies and challenges in linking an event mention to the correct title.

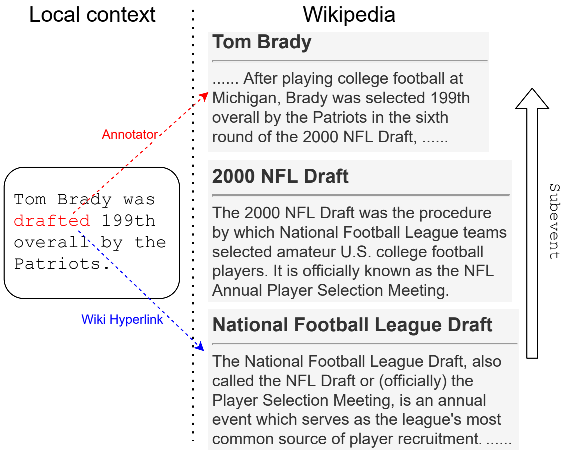
Local context
Wikipedia
Tom Brady
...... After playing college football at Michigan, Brady was selected 199th overall by the Patriots in the sixth round of the 2000 NFL Draft, ...
2000 NFL Draft
The 2000 NFL Draft was the procedure by which National Football League teams selected amateur U.S. college football players. It is officially known as the NFL Annual Player Selection Meeting.
National Football League Draft
The National Football League Draft, also called the NFL Draft or (officially) the Player Selection Meeting, is an annual event which serves as the league's most common source of player recruitment. ......
Annotator
Tom Brady was drafted 199th overall by the Patriots.
Wiki Hyperlink
Subevent

Figure 3: Example of Hierarchical Events, the event "drafted" in the local context (left) is mentioned in the "Tom Brady" wiki page but is also a sub-event of "2000 NFL Draft" which is further a sub-event of "National Football league Draft" [21].

2.3.4 Repeating Events

Certain events, like award ceremonies, sports events, or political events (such as elections), occur repeatedly at regular intervals. For instance, in the example "In 1995, his debut season, Biddiscombe made two appearances... The following year he earned a Rising Star nomination for his performance". The event AFL Rising Star occurs annually, making it more challenging to link to the correct Wikipedia page. In this context, it becomes even more challenging because no year is directly mentioned; only a hint (following year) is provided in the sentence.

3 Related Work

Event linking, introduced as a relatively new task in the paper [21], often makes researchers in this field draw inspiration from the work in Entity Linking. While entity linking has been extensively studied, with various contributions such as [1],[13],[17],[7], [2] and [20], and both tasks fall under document linking. We argue that event mentions have a more complex structure compared to entities and, therefore require more additional information.

3.1 Domain Contributions

Early research, such as [5], distinguishes events from entities and defines them to be represented by their attributes and participants. Where each attribute represents the temporal, location, or purpose relation, and each participant represents the entities around an event mention. We saw the following studies, such as [14] and [9], introduce the term "Event Linking" for the first time. In paper [14], event linking corresponds to a novel approach to news event reference, associating each event with an article in the news archive. Meanwhile, the contributions in paper [9] focused on using Convolutional Neural Networks to extract intra-sentential features and using them for the co-reference decision of pairwise events.

Recent contributions include [15] which utilizes a hierarchical event grounding task to link event mentions from a document into a hierarchical structure. It integrates a hierarchy-based loss for the grounding task and conducts hierarchical relation extraction in a zero-shot setup.

While event linking is mentioned in all these studies, they mainly concentrate on co-referencing event mentions or performing hierarchical event grounding in various documents. In contrast, our work focuses on linking an event mentioned to the correct Wikipedia page.

3.2 Entity Linking

As also mentioned before researchers in this domain are often inspired by work in entity linking, considering both to be part of document linking. The [21] also follows the same trend and compares the results of Evelink with SOTA entity linking models such as BLINK [20] and GENRE [2].

BLINK introduces a two-stage approach for zero-shot linking, utilizing fine-tuned

BERT architectures. In the first stage, retrieval is conducted in a dense space defined by a bi-encoder, which embeds mention context and entity descriptions independently. Candidates are then carefully examined with a cross-encoder that concatenates mention and entity text.

GENRE short for Generative Entity Retrieval on the other hand utilizes a sequence-to-sequence architecture to generate entity names in an autoregressive manner conditioned on the context. It employs a transformer-based architecture, pre-trained with BART weights and fine-tuned to generate entity names. To ensure that generated names are valid entities, GENRE employs a constrained decoding strategy. The autoregressive formulation enables capturing relations between context and entity name efficiently.

In our research following the work of [21], we extend the BLINK [20] to the event-linking domain. Based on the dataset provided by the paper, we also employed a two-stage approach for event linking discussed in detail later in Section 5.

	Train	Dev	Test	
	Wiki	Wiki	Wiki	NYT
Verb	33,213	8,346	9,633	1,319
Seen Event	-	1,814	2,913	0
Unseen Form	-	2,585	3,828	75
Unseen Event	-	3,947	2,892	435
Nil	-	-	-	809
Nominal	33,213	8,346	9,633	443
Hard	-	4173	4817	244
Easy	-	4173	4817	15
Nil	-	-	-	184
Total	66,426	16,692	19,266	1,762

Table 1: Wikipedia and New York Times (NYT) data statistics. [21]. NYT is used only for evaluation.

4 Dataset

The paper [21] collected the training and in-domain test data from Wikipedia. Additionally, out-of-domain data was collected for evaluation from the New York Times. For Wikipedia dataset (in-domain), only event mentions of types attack, election, protest, military conflict, natural disaster, sports event, and terrorist attack were considered. This categorization was based on the FIGER dataset [11], which was used for the extraction of Event mentions. FIGER short for FIne Grained Entity Recognition is a dataset where entities are labeled using a fine-grained system of 112 tags, such as person/doctor, art/written_work, and building/hotel. Detailed statistics on the collected data are presented in Table 1. We extended the Wikipedia dataset by adding extra information about all the event mentions (sub-events) and the section titles for each candidate page (all Wikipedia pages).

4.1 Wikipedia

For the Wikipedia dataset, the paper [21] initially gathered all pairs of hyperlinks and titles. Then, each title was categorized according to FIGER types [11]. Titles labeled as "Event" in FIGER were considered event titles, and all hyperlinks linked to these titles were treated as event mentions. Since each event title on Wikipedia is typically hyperlinked only once, editors tend to hyperlink nominal mentions more often than verbs. To ensure a balanced dataset, the paper utilized the SpaCy Part-of-Speech model [2] to maintain equal sample sizes for verb and nominal events. Event mentions were further divided into easy and hard cases to prevent the model from overfitting.

4.1.1 Verbs

Verb events are event mentions represented by a verb in the sentence. For example, in the sentence " .... part of the Manhattan Bridge will be closed so that its roadway can be rebuilt .....", the "rebuilt" is a verb event. They are further categorized into easy and hard cases. If the surface form (S) of the verb exists in the training data and the event title (T) is also observed in the training data, it is considered a "Seen Event". If the surface form (S) of the verb doesn't exist in the training data but the event title (T) is seen in the training data, it is classified as an "Unseen Form". Additionally, due to the limited size of verb events on Wikipedia, titles with five or fewer verbs mentioned were labeled as "Unseen Events". In this research, "Seen Events" is considered as an easy case while both "Unseen Form" and "Unseen Events" are considered as hard cases.

4.1.2 Nominals

Nominal events are event mentions represented by the subject (noun) in the sentence. For example, in the sentence "Ibrox hosted four Scotland games in the first phase, starting with a 1994 World Cup qualifier against Portugal in October 1992", the "1994 World Cup qualifier" is a nominal event. These nominal events are further categorized into hard and easy cases. The jaccard similarity was calculated with the gold title by considering 3-gram of the surface form. If the similarity was lower than 0.1, it was classified as a hard case. Otherwise, it was considered an easy case. After classification, samples were derived using an equal number of hard and easy cases.

4.2 New York Times (NYT)

The paper [21] selected 2,500 paragraphs from the New York Times Annotated Corpus [18], which included articles from 1987 to 2006. They used an SRL model [3] to recognize the mentioned event candidates. SRL short for Semantic Role Labeling is a BERT-based model. It was trained on datasets like Nombank [12] and the

[2] https://spacy.io/usage/linguistic-features#pos-tagging

[3] https://cogcomp.seas.upenn.edu/page/demo_view/SRLEnglish

Ontonotes [16]. It was used to extract event mentions candidates which were then validated through annotation with the help of Amazon Mechanical Turk.

4.3 Domain Analysis

Linking events in the news domain presents greater challenges compared to those in the Wikipedia domain due to the following factors:

1) News articles provide more detailed descriptions of events compared to Wikipedia. For example, a news report on the Iraq War might say, "A contractor working for the American firm Kellogg Brown & Root was wounded in a mortar attack in Baghdad". This level of detail is typical in daily news coverage but not necessarily present in Wikipedia entries. In Wikipedia, the same event might be summarized more broadly, such as "When touring in Europe, the US went to war in Iraq". This difference in detail makes event linking more challenging in the news domain, where it involves extracting various sub-events, compared to the event linking in Wikipedia.

2) As discussed earlier, event linking poses challenges because some events may only exist within sub-sections of the main page, and accurate titles may vary due to event hierarchies. Firstly, these issues are more common within the news domain compared to the Wikipedia domain. This is because event mentions in Wikipedia typically lead to separate pages rather than leading to sub-sections. Secondly, within the Wikipedia domain, the designated title for the same event tends to remain consistent. For example, all mentions of the event "drafted" of football players link to the "National Football League Draft" rather than the individual player's page. However, within New York Times (NYT) domain, the annotation did not consistently align with Wikipedia. This is because, NYT annotators tend to link event mentions regarding sports player drafts to the specific player's page rather than the broader concept page, "National Football League Draft". This dissimilarity significantly complicates data annotation and model evaluation within the news domain.

4.4 Dataset Extension

Based on the findings about the impact of the sub-events and the events that exist only in the sub-sections of the Wikipedia page. We collected additional data using an approach from [21] and [10]. First, for sub-events, we utilized the content of the Wikipedia page provided in the dataset by [21] as input to the NER tagger [4] (english-

[4] `https://huggingface.co/flair/ner-english-ontonotes-large`

ontonotes-large) to recognize all the event mentions on the Wikipedia page. Second, inspired by the approach from [10], we incorporated the WikiKG [5] to extract all the titles of sections and sub-sections of the Wikipedia page. We also explored other event knowledge bases, such as EventKG [6], which offers comprehensive representations of events and temporal relations. However, for our specific use case, where we aimed to focus solely on the events mentioned in section titles, we found it impractical to use EventKG. Furthermore, due to architectural constraints, we pre-processed information regarding both sub-events and sub-sections, saving them in two separate JSON files for further processing.

[5] https://www.mediawiki.org/wiki/API:Main_page

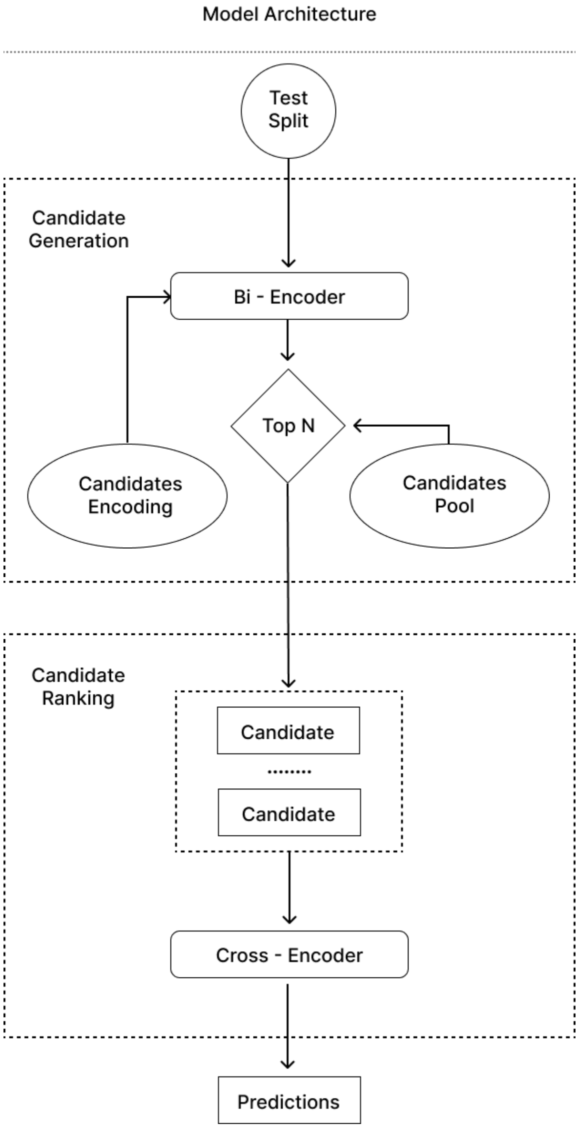

Figure 4: The model architecture comprises two major parts: candidate generation, which is handled by a Bi-Encoder, responsible for generating candidates, and candidate ranking, which is handled by a Cross-Encoder, responsible for ranking candidates.

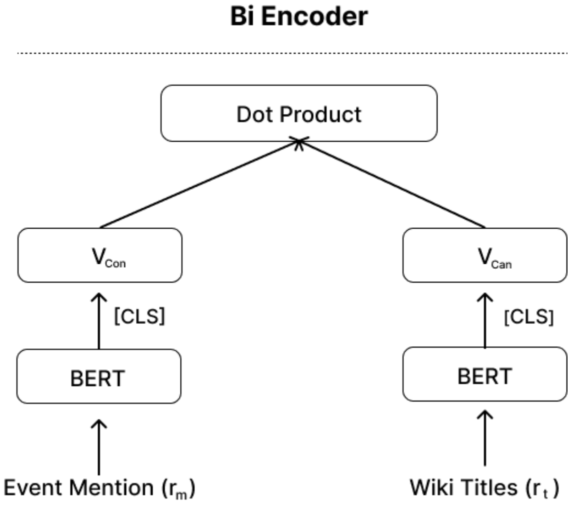

Figure 5: It encodes both the event mention and candidate title into vectors. It generates output vectors V_{con} and V_{can} from each BERT encoder. Then, it calculates the dot product between these vectors to score the candidate title for the given event mention.

5 Methods

In this section, we discuss the architecture of the model Evelink by [21] and our improvements in candidate representations for challenges such as sub-events and sub-section event mentions. Following the approach of [20], the model architecture of Evelink comprises of Bi-Encoder and Cross-Encoder components. A Bi-Encoder is utilized to generate the top-k candidates for each event mention in the dataset. Previous research [3] demonstrated that a Bi-Encoder model could encode and compute the cosine similarity for each possible pair of 10,000 sentences in just 5 seconds. In contrast, it took 65 hours for the cross-encoder to identify the most similar pair from the same collection of sentences. Despite the ability of a cross-encoder to capture more information due to its complex interaction with the input, its large computation time forces its use in the second stage for ranking the top-k candidates. The architecture can be seen in Figure 4.

5.1 Candidate Generation

The Bi-encoder is used to generate candidates as also shown in Figure 4. Its architecture consists of BERT transformers [4], with two encoders sharing the same

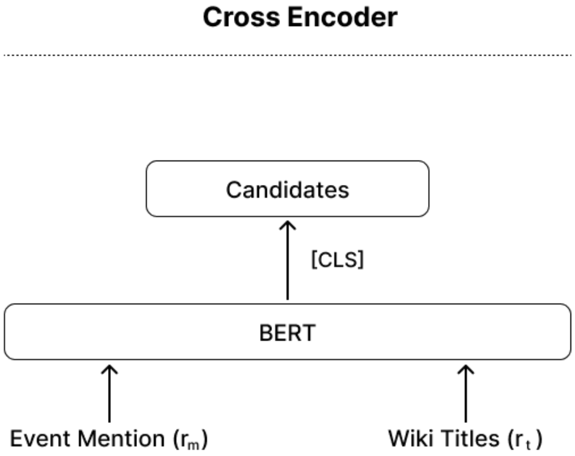

Figure 6: A concatenated representation of the event mention and candidates title is input into a single BERT encoder to score each candidate for the given event mention.

parameters to encode the representations of the event mention, r_m and candidate titles r_t as shown in Figure 5. Both encoders operate independently without interacting during the encoding stage. The resulting outputs from both encoders ([CLS] tokens) were utilized as vectors for the event mention v_m and the correct candidate title v_t. Then, a dot product was maximized between v_m and v_t, along with randomly selected negatives in a batch. During inference, the representations of all Wikipedia titles were cached as also shown in Figure 4. For each event mention, a dot product was calculated between its representation and the representations of all Wikipedia titles. Titles with higher scores were identified as candidates.

5.2 Candidate Ranking

For each event mentioned, top-k candidates generated from the candidate generation stage are selected to serve as data for the candidate ranking as shown in Figure 4. A cross-encoder model is used as the candidate ranking model. It consists of a single BERT-transformer used to encode the concatenated representation of the event mention r_m and candidates titles r_t as shown in Figure 6. The resulting output from the BERT encoder ([CLS] tokens) was utilized as vector v. Similar to the candidate generation task, the dot product was maximized between the output vector v and an additional linear layer W to determine the rankings of the generated candidates.

5.3 Creation of Representations

5.3.1 Event Mention

To create a representation of event mentions, it is important to understand the properties of an event. The context of an event is typically less diverse compared to entities. For example, when an entity such as "France" is mentioned in an article, predicting the entities or events around it becomes more challenging. However, for the event "Battle of France" represented by the trigger word "invade", it is highly probable to find entities such as "Germany", "Italy" and "France" around it. This illustrates that the context of an event is influenced by its arguments (entities), as the verb "invade" alone may not provide sufficient information to identify the event.

Based on this observation, [21] concluded that the entities mentioned in the local context of the event should coincide with those mentioned in the correct Wikipedia page. With this conclusion, surrounding entities were included in creating event representation. To explicitly embed entity information for the event representation, a method outlined in [19] suggesting concatenation of entities with special tokens [SEP] was adopted.

An off-the-shelf Named Entity Recognition Model [7] was utilized to create event-mention representations in the local context. It is trained on the 18-type OntoNotes dataset [16] to extract entities surrounding the event. The context window size is set to 500 characters around the mentioned event to enhance the efficiency of the model. After all entities e_i with their types t_i are predicted, an event mention is represented by

$$r1 = [CLS]\, ctxt_l\, [M_s]\, m\, [M_e]\, ctxt_r \tag{1}$$

$$r2 = [t_{1_s}]\, e_1\, [t_{1_e}] \ldots [t_{n_s}]\, e_n\, [t_{n_e}] \tag{2}$$

$$rm = r1\, [SEP]\, r2\, [SEP] \tag{3}$$

Where m corresponds to the token of the event mention, $ctxt_l$,$ctxt_r$ corresponds to the context on the left and right, and e_i corresponds to predicted entities. The $[M_s]$ and $[M_e]$ are special tokens to tag the start and end of the event mention. The $[t_{i_s}]$ and $[t_{i_e}\ ]$ are special tokens to tag the start and end of the entity whose type is t_i.The final representation of an event mentioned is represented by r_m.

5.3.2 Candidate Titles

To create a representation of candidate titles, it is important to understand the properties of a Wikipedia Title. Title representations typically consist of the title name and the first few sentences of the page. Based on the assumption that important mentions are hyperlinked, [21] also included the first 10 hyperlinks found on the Wikipedia page. However, we argue this representation lacks essential details. For example, on the page about World War 1, there's an event mention of war on Serbia: "Bulgaria declared war on Serbia on 14 October 1915 and joined in the attack by the .... ". Since the representation of World War 1 does not include the Serbian War, the model will not be able to link this mention to World War 1. Based on this finding we believe more information about event hierarchies, and section titles should be included for title representation. Therefore, we extended the work to improve the representation of title candidates for Evelink.

Originally, [21] took a title name, description, and the first ten hyperlinked spans as entities to represent the candidate title by:

$$\begin{aligned} r3 &= [CLS]\, title\, [TITLE]\, description && (4) \\ r4 &= h_1\, [SEP]\, h_2\, [SEP] \ldots [SEP]\, h_n && (5) \\ r_{old} &= r3\, [SEP]\, r4\, [SEP] && (6) \end{aligned}$$

Where title, h_i, and description are tokens of the title, hyperlinked spans, and the content of the Wikipedia page. The special token [TITLE] was used here to separate the title and description.

We extended [21] approach by including extra information about sub-events and sub-sections of the Wikipedia title. This choice was guided by the insights from [21], which highlighted the importance of extracting relations between sub-events in news domains. To extract the sub-events, we used the same off-the-shelf NER tagger [7] used before for recognizing entities around event mentions. All the entities of type event with a confidence value of equal or greater than 0.7 were considered sub-events. For sub-sections, we use the Wiki KG [6], to extract all the sections and subsections of a title page. Once both the sub-events and sub-sections are there for a wiki page, we combine them using the following rules.

- Extract all the common titles (event sections) that exist in extracted sub-sections and sub-events.

[6] https://www.mediawiki.org/wiki/API:Main_page

[7] https://cogcomp.seas.upenn.edu/page/demo_view/NEREnglish

- The common titles are then added to the representation, with a maximum limit of 6.
- Then we remove all the common titles (event sections) from the sub-events to avoid any duplication.
- We also make sure, to avoid any duplication within the sub-events, as the NER might extract the same event at different locations on the page. We used the distance metric Jaccard Similarity to be less than or equal to 0.75 to extract all unique sub-events.
- After we have all remaining unique sub-events, we also add them to the representation, with a maximum limit of 12 - len (common_titles).
- Including both the common titles (event sections) and sub-events, should not exceed 12 due to the limitation of the context length. This value was derived based on the average length of the old (r_{old}) candidate representation.

After we have all the required data, we incorporate them as below.

$$\begin{aligned} r5 &= c_1\,[SEP]\,c_2\,[SEP]\ldots[SEP]\,c_n && (7)\\ r6 &= s_1\,[SEP]\,s_2\,[SEP]\ldots[SEP]\,s_n && (8)\\ r_{new} &= r_{old}\,[SEP]\,r5\,[SEP]\,r6\,[SEP] && (9) \end{aligned}$$

Where c_i and s_i represent the sub-sections and sub-events of the Wikipedia page. The final representation of Wikipedia titles r_{new} was created by concatenation of r_{old}, r_5, and r_6.

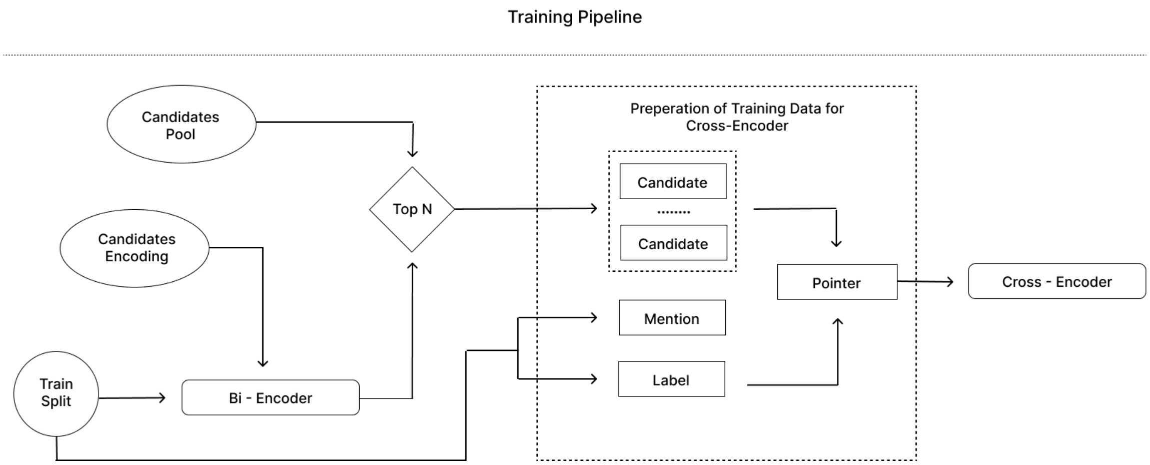

Figure 7: The training pipeline is initiated by the trained Bi-Encoder, which generates data for training the Cross-Encoder. The same train split is used as input to the trained Bi-Encoder to produce "Top N" candidates. This parameter can be controlled, and in our case, it is set to 30 as specified in [21]. A label (pointer) is calculated for the Cross-Encoder based on the gold label in the training split and the generated candidates. Later, all candidates information, event mentions information, and label pointers are combined to create training data for the Cross-Encoder.

6 Experiments

In this section, we discuss experiments that were conducted on the model Evelink with the extended dataset mentioned in Section 5. Additionally, experiments were performed with the original dataset provided by [21].

6.1 Original Experimentation Setup

The paper [21] evaluated the Evelink model on both in-domain Wikipedia and out-of-domain NYT datasets. The NYT dataset due to its different levels of granularity and not being utilized during the model's training phase, presents greater challenges. The paper originally mentioned using 4 Nvidia RTX A6000 48GB GPUs for the training and evaluation of the model. As there was no existing event linking systems, the paper compared the results with the SOTA entity models such as [20] and [2]. For a fair comparison, the paper follows a two-stage solution.

Models	Verb				Nominal			Verb + Nominal
	Seen	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
Prior	62.21	2.38	1.24	38.81	34.65	85.99	61.65	54.79
BLINK-Entity	64.13	48.56	45.92	52.48	46.79	88.27	67.53	60.00
BLINK-Event	77.72	69.78	62.72	70.06	62.59	82.29	72.44	71.25
GENRE-Entity	75.04	57.00	44.85	58.81	65.29	**90.91**	78.10	68.45
GENRE-Event	**95.50**	73.80	45.16	71.76	72.60	88.04	80.32	76.04
Evelink BERT-Large	91.21	**80.30**	**78.08**	**82.93**	**75.90**	89.70	**82.80**	**82.87**
Evelink BERT-Base	83.37	76.34	75.14	77.81	70.10	87.43	78.77	78.44

Table 2: Accuracy on Wikipedia Test [21]. The Verb+Nominal accuracy corresponds to accuracy for the top first candidate mentioned in [21].

Models	Verb			Nominal			Verb + Nominal
	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
Prior	0.00	0.00	0.00	0.00	6.25	0.39	0.13
BLINK-Entity	1.33	2.76	2.55	4.92	33.33	6.56	3.90
BLINK-Event	17.57	5.28	7.06	11.11	37.50	12.74	8.97
GENRE-Entity	8.11	5.73	6.08	3.29	31.25	5.02	5.72
GENRE-Event	**39.19**	8.03	12.55	7.82	31.25	9.27	11.44
Evelink BERT-Large	28.37	13.07	15.29	**14.81**	43.75	**16.60**	15.73
Evelink BERT-Base	27.48	**13.09**	**15.75**	12.90	**62.5**	15.96	**15.82**

Table 3: Accuracy on New York Times data without Nil [21]. Only event mentions that exist in Wikipedia are given. The Verb+Nominal accuracy corresponds to accuracy for the top first candidate mentioned in [21].

6.1.1 BLINK/GENRE - Entity

The paper compared the results of Evelink with models BLINK [20] and GENRE [2], originally trained on entity datasets. This comparison was based on the observation that most event mentions are nominal and thus, can be considered as entities. It would be interesting to analyze how Evelink performs in comparison to models trained specifically on entity datasets. It is important to note that the entity-based training dataset consisted of 9 million data points, whereas the event-based training dataset only consisted of about 66,000 data points.

6.1.2 BLINK/GENRE - Event

Both model BLINK [20] and GENRE [2] were also retrained by the paper [21] on the newly created event-specific dataset, for a fair evaluation of their accuracy with the Evelink. According to the paper, both BLINK-Event and Evelink generated the top 100 candidates for further candidate ranking. For the GENRE, being a generation model, the paper followed the original setting of [2] to use the beam search with 5 beams.

6.2 Recreation of Results

For the experiments, we tried to replicate the results by running inference. The paper originally used a bert-large-uncased as the pre-trained model. Unfortunately, the paper only provided the checkpoints for the bert-large-uncased. For this reason, we had to retrain the model again with a smaller version bert-base-uncased due to the limited resources of our university (2 Nvidia RTX A6000 64 GB GPUs).

The code was also made more efficient, by fixing some code in the data generation stage. The dataset originally existed in different parts, a script was also provided to compile the dataset together into train, test, and validation splits. The retraining of the bi-encoder and cross-encoder models in total took approximately 68 hours for 10 epochs each. The recreated results can be seen in Tables 2 and 3.

6.3 Ablation Study

In this section, we devise the four use cases to assess the impact of additional information on both the candidate ranking (Bi-Encoder) and candidate generation (Cross-Encoder) sides. The motivation for these use cases stemmed from our investigation into the effects of extra information on candidate titles at both the candidate generation and candidate ranking stages. The extra information was integrated as presented in Equation 9.

6.3.1 No Extra Information on Bi-Encoder and Cross Encoder

In this use case, no extra information was added to the Bi-Encoder and Cross-Encoder. Instead, both models were re-trained using BERT-base-uncased, and the original pre-computed candidate representation and candidate pool were utilized.

6.3.2 Extra Information only Bi-Encoder

In this use case, extra information was added only to the Bi-Encoder. This was accomplished by retraining the Bi-Encoder model with the new representations and generating new candidates encoding for accurate prediction. Additionally, new candidates encoding was used for the inference stage.

6.3.3 Extra Information only Cross-Encoder

In this scenario, extra information was added only to the Cross-Encoder. This was achieved by retraining the Cross-Encoder model with a new representation and ensuring the use of the new candidates pool with the updated representation during training. Once the original Bi-Encoder generates the top-k candidates, extra information is incorporated with the new candidates pool during inference.

6.3.4 Extra Information on both Bi-Encoder and Cross-Encoder

In this scenario, extra information was added to both the Bi-Encoder and Cross-Encoder models during training and inference. Additionally, new candidates encoding and candidates pools were used for the inference stage.

6.4 Challenges during Ablation Study

In this section, we will discuss the model's training pipeline, also shown in Figure 7, and the challenges encountered during the ablation study mentioned earlier. The paper [21] originally used the work from [20] and extended the methods for the problem of event linking. The architecture consists of 3 stages as follows.

6.4.1 Bi-Encoder Training

In our research, the Bi-Encoder is trained using the new train and valid splits as mentioned earlier in Section 4. Each data point from the train split and validation split consisted of two pieces of information: event mention and candidate title (label). The candidate representation was updated based on the Equation 9. Following the approach used by [20], both encoders with pre-trained Bert-Base (uncased) were fine-tuned against each mention representation and candidate representation to maximize the dot product between them. A learning rate of "1e-5" was used for both encoders. For a fair comparison, we used all the same hyper-parameters provided by the paper [21]. All models were implemented with PyTorch[8] and optimized by Adam [8].

6.4.2 Pre-computation for Cross-Encoder Training

Following the approach of [20], we trained the Bi-Encoder with the updated candidate representations and then utilized it to generate training data for the Cross-Encoder. As discussed in Section 5, the Bi-Encoder relies on pre-computed representations for all candidate titles (approximately 6 million) during inference. Additionally, a pre-computed candidates pool for all candidate titles is also required to generate training data for Cross-Encoder.

This pre-computation turned out to be a significant challenge in our research, shedding light on the limitations of this architecture. We aimed to enhance candidate representation with additional information about sub-events and sub-sections. But this requires computing extra information for all possible candidate titles (approximately 6 million).

Efficiency was a concern as the extra information from the NER tagger and Wiki API [9] resulted in a computing time of around 3 iterations per second, which amounted to approximately 555 hours for 6 million data points. Despite attempts to incorporate additional information only during the training step for both the Bi-Encoder and Cross-Encoder, efficiency remained an issue, especially during inference. For example, with a test set comprising 19,000 data points, the Bi-Encoder generated 100 possible candidates for each, resulting in a need to add extra information for 1.9 million data points. This volume makes the real-time addition of extra information unfeasible, demanding pre-computation of candidates embedding and candidates pool with extra information.

Based on this finding, we decided to pre-compute the extra information of sub-events and sub-sections. A new script was created for the parallel computation of all Wikipedia titles, resulting in a total computation time of 240 hours. We considered this extra-information data as an extension to the original dataset provided by the paper [21]. Furthermore, we used the new pre-computed data to create the new candidates encoding and candidates pool for further processing.

Once we had the updated candidates pool and encodings, we used them to create the

[8] https://pytorch.org/docs/stable/index.html

[9] https://www.mediawiki.org/wiki/API:Main_page

dataset for the cross-encoder. We selected the top 30 candidate representations from the candidates pool generated by the Bi-Encoder. Then, we combined them with the label (pointer) for the cross-encoder, which is simply an integer indicating the correct candidate index among those 30 candidates. This allowed the Cross-Encoder to learn how to re-rank the candidates efficiently. All the data was combined and saved in tensor format for input to the cross-encoder.

6.4.3 Cross Encoder Training

Once the dataset is generated using the newly trained Bi-Encoder, the training of the Cross-Encoder is initiated. Due to the encoder comprising of only one BERT, the data from the Bi-Encoder needs to be modified. This involves acquiring a mention representation along with the top 30 candidate representations. It is then combined to create 30 vector representations for a particular event mention along with its corresponding label. The [CLS] token from each candidate representation is removed to enable the Cross-Encoder to learn efficiently with the self-attention mechanism. After obtaining representations for each event mentioned in the training set, the modified data is used to train the cross-encoder.

Models	Verb				Nominal			Verb + Nominal
	Seen	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
EVELINK BERT-Base	83.37	76.34	**75.14**	77.81	70.10	87.43	78.77	78.44
EVELINK (info Bi/Cross)	84.75	74.57	73.86	77.84	69.77	86.96	78.37	78.10
EVELINK (info Cross)	80.60	74.74	74.03	76.30	68.84	86.29	77.57	76.93
EVELINK (info BI)	**85.86**	**77.02**	75.06	**79.10**	**70.46**	**87.72**	**79.10**	**79.10**

Table 4: Accuracy on the Wikipedia test increased across all categories except Verb Unseen, with the most significant increase observed in Verb Seen.

Models	Verb			Nominal			Verb + Nominal
	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
EVELINK BERT-Base	27.48	13.09	15.75	12.90	62.5	15.96	15.82
EVELINK (info Bi/Cross)	21.62	12.61	13.92	12.76	56.25	15.44	14.43
EVELINK (info Cross)	21.62	11.47	12.94	13.58	50.0	15.83	13.91
EVELINK (info BI)	**30.18**	**13.91**	**16.27**	**13.01**	**64.58**	**16.23**	**16.36**

Table 5: An increase in accuracy across all categories in the New York Times data was observed excluding Nil mentions, with the most significant increase seen in verb mentions.

7 Evaluation

In this section, we discuss the results of the newly trained models with and without extra information on both the in-domain (Wikipedia) and out-of-domain (NYT) datasets. The results were evaluated with the four use cases defined earlier. The recreation of original results with a Bert-Base is shown in Tables 2 and 3. Additionally, the results with extra information for both the Wikipedia and NYT datasets are shown in detail in Tables 4 and 5, respectively. The results presented in the mentioned tables were calculated by averaging the outcomes obtained from three different seed values 52313, 92917, and 74574 for the bi-encoder only. The decision was influenced by a fair comparison on the bi-encoder side, as no significant improvements were seen for the other use cases.

Models	Verb				Nominal			Verb + Nominal
	Seen	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
EVELINK BERT-Base	84.28	89.47	**92.70**	88.87	84.39	92.46	88.42	88.65
EVELINK (info BI)	**88.26**	**90.57**	92.19	**90.36**	**85.68**	**93.09**	**89.38**	**89.87**

Table 6: Wikipedia Recall.

Models	Verb			Nominal			Verb + Nominal
	Unseen Form	Unseen	Overall	Hard	Easy	Overall	
EVELINK BERT-Base	51.35	50.0	50.20	43.62	75.0	45.56	48.63
EVELINK (info BI)	**55.40**	**52.06**	**52.55**	**46.91**	**87.5**	**49.42**	**51.50**

Table 7: NYT Recall.

7.1 Result Analysis

First, the results from Evelink with the BERT-base model show a relatively consistent trend. Due to the smaller size of the BERT-base, there's a decrease in accuracy across all categories for the Wikipedia dataset. Interestingly, the smaller model was able to have slightly higher accuracy for the out-of-domain NYT dataset. However, Evelink's BERT-Base still outperforms the SOTA models for both Wikipedia and NYT datasets in most categories, particularly in the verb event mentions, Evelink's BERT-Base surpasses all other models. Detailed results are provided in Tables 2 and 3.

Second, after the addition of extra information, the most effective model we trained was the one where additional data was only added to the Bi-Encoder. In this use case, across all data categories except Verb Unseen, we observed a consistent improvement. This observation was further confirmed by analyzing the Bi-Encoder's recall for candidates generation on both the Wikipedia and NYT datasets, as depicted in Tables 6 and 7. We noticed a significant increase in the Bi-Encoder recall, validating our previous observation that extra information has a greater impact on the Bi-Encoder compared to the Cross-Encoder. We believe, this is due to the complexity of representations of candidate titles, which may confuse the Cross-Encoder resulting in inaccurate predictions.

Our research concludes that the Bi-Encoder plays a crucial role in the architecture pipeline proposed by [20]. If it can generate candidates for more challenging problems, the Cross-Encoder can effectively rank them with sufficient representation. Additionally, findings such as a small difference in accuracy and recall for Verb Seen suggest that the Bi-Encoder might create a bottleneck for predictions due to low recall. This highlights the importance of effectively generating candidates to enable accurate ranking by the Cross-Encoder.

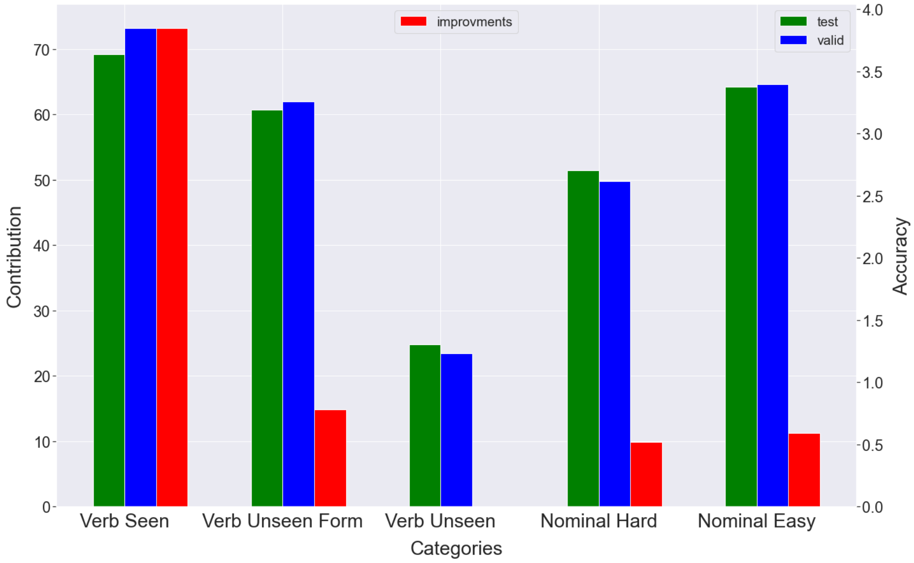

Figure 8: The graph illustrates how much improvement was made in each category due to the extra information for the Wikipedia dataset. The left y-axis indicates the contribution percentage relative to the total number of data points for the test and valid splits. Meanwhile, the right y-axis shows an accuracy improvement relative to the original accuracy. The x-axis shows the different categories in the dataset by paper [21].

7.2 Improvement Analysis

Based on our observation in Tables 4 and 5, we were interested in understanding how different categories of test data showed varying levels of improvement. To explore this further, we analyzed the dataset based on the amount of extra information it contained. Our goal was to identify areas where the extra information made a significant contribution. We also examined the distribution in the validation dataset, which is crucial for training the model. If there are insufficient data points with extra information, the model may not effectively learn from it. The distribution is visualized in Figure 8.

An interesting trend emerged when comparing the contribution of the data to the observed improvement. Most of the improvement was observed for the Verb Seen category, likely because a substantial amount of extra information was added for this category. Similarly, other categories, particularly those involving verb mentions, followed a similar trend. However, the Verb Unseen category did not show any improvement. This lack of improvement can be attributed to the limited number of data points with additional information, resulting in minimal impact on the model's performance. These findings suggest that the observed improvements may be more attributable to the presence of extra information rather than any inherent characteristics of the data.

8 Future Work

In our research, we added extra information about the sub-events and sub-sections to the representation to address the error patterns identified in the paper [21]. However, we believe there is room for further improvement. Currently, we have not incorporated any hierarchical structure for the sub-events on the main event page. We hypothesize that utilizing an Event KG such as [6] to find more relevant events could enhance accuracy for candidate generation. Additionally, our work focused solely on two error patterns related to sub-events and sub-sections, leaving improvements for other error patterns for future research.

9 Conclusion

In this research, the event-linking formulation was explored in comparison to entity-linking. Challenges specific to event linking were discussed, with an emphasis on the importance of the candidate's representation for harder problems. We also discussed the impact of candidate generation and candidate ranking on the overall architecture. Additionally, the dataset was extended with extra information, and further improvements were left for future work.

References

[1] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics, 2006. URL https://aclanthology.org/E06-1002/.

[2] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

[3] Justin Chiu and Keiji Shinzato. Cross-encoder data annotation for bi-encoder based product matching. In Yunyao Li and Angeliki Lazaridou, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022*, pages 161–168. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-INDUSTRY.16. URL https://doi.org/10.18653/v1/2022.emnlp-industry.16.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

[5] George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association, 2004. URL http://www.lrec-conf.org/proceedings/lrec2004/summaries/5.htm.

[6] Simon Gottschalk and Elena Demidova. Eventkg: A multilingual event-

centric temporal knowledge graph. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 272–287. Springer, 2018. doi: 10.1007/978-3-319-93417-4_18. URL `https://doi.org/10.1007/978-3-319-93417-4_18`.

[7] Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2681–2690. Association for Computational Linguistics, 2017. doi: 10.18653/V1/D17-1284. URL `https://doi.org/10.18653/v1/d17-1284`.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

[9] Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. Event linking with sentential features from convolutional neural networks. In Yoav Goldberg and Stefan Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 239–249. ACL, 2016. doi: 10.18653/V1/K16-1024. URL `https://doi.org/10.18653/v1/k16-1024`.

[10] Thomas Lin, Mausam, and Oren Etzioni. Entity linking at web scale. In James Fan, Raphael Hoffman, Aditya Kalyanpur, Sebastian Riedel, Fabian M. Suchanek, and Partha Pratim Talukdar, editors, *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX@NAACL-HLT 2012, Montrèal, Canada, June 7-8, 2012*, pages 84–88. Association for Computational Linguistics, 2012. URL `https://aclanthology.org/W12-3016/`.

[11] Xiao Ling and Daniel S. Weld. Fine-grained entity recognition. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, pages 94–100. AAAI Press, 2012. doi: 10.1609/AAAI.V26I1.8122. URL `https://doi.org/10.1609/aaai.v26i1.8122`.

[12] Adam L. Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The nombank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation@HLT-NAACL 2004, Boston, MA, USA, May 6, 2004*, 2004. URL `https://aclanthology.org/W04-2705/`.

[13] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão, editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 233–242. ACM, 2007. doi: 10.1145/1321440.1321475. URL `https://doi.org/10.1145/1321440.1321475`.

[14] Joel Nothman, Matthew Honnibal, Ben Hachey, and James R. Curran. Event linking: Grounding event reference in a news archive. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 228–232. The Association for Computer Linguistics, 2012. URL `https://aclanthology.org/P12-2045/`.

[15] Jiefu Ou, Adithya Pratapa, Rishubh Gupta, and Teruko Mitamura. Hierarchical event grounding. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13437–13445. AAAI Press, 2023. doi: 10.1609/AAAI.V37I11.26576. URL `https://doi.org/10.1609/aaai.v37i11.26576`.

[16] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards robust linguistic analysis using ontonotes. In Julia Hockenmaier and Sebastian Riedel, editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 143–152. ACL, 2013. URL `https://aclanthology.org/W13-3516/`.

[17] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Pro-*

ceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, pages 1375–1384. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1138/.

[18] Evan Sandhaus. The new york times annotated corpus. 2008. URL https://catalog.ldc.upenn.edu/LDC2008T19.

[19] Yogarshi Vyas and Miguel Ballesteros. Linking entities to unseen knowledge bases with arbitrary schemas. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 834–844. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.65. URL https://doi.org/10.18653/v1/2021.naacl-main.65.

[20] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.519. URL https://doi.org/10.18653/v1/2020.emnlp-main.519.

[21] Xiaodong Yu, Wenpeng Yin, Nitish Gupta, and Dan Roth. Event linking: Grounding event mentions to wikipedia. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2671–2680. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.196. URL https://doi.org/10.18653/v1/2023.eacl-main.196.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Unterschrift ____________________ Ort, Datum Hamburg, 29/04/2024