Thesis Proposal
Diving Deep into Event Semantics

Zhengzhong Liu

October 2018

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Teruko Mitamura (Chair), Carnegie Mellon University
Eduard Hovy, Carnegie Mellon University
Taylor Berg-Kirkpatrick, Carnegie Mellon University
Vicent Ng, The University of Texas at Dallas

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: event,script,event schema,coreference,semantics,indirect supervision,facets,quasi-identity

Abstract

Among various discourse elements (DEs) in natural language discourse, events are important due to their rich structure. They are often connected with other DEs, including participants, time, and location. Multiple events can form larger structures such as coreference clusters and scripts.

Prior work exploiting event structures normally treats DEs such as entities and events as holistic units: the entire denotation of a DE is assumed for producing the interpretation. However, in real-world usage, people often intend to emphasize different parts of the denotation at various times and ignore others. Even if two DEs are technically coreferent, only partial intended information may be identical. For example, "The White House's announcement" and "The president's announcement" could refer to the same event, but the two entities "The White House" and "The president" have different sets of denotation. Yet the fact that they share the "authority" aspect, allows us to establish proper event and entity coreference links.

Ignoring this decomposition of DEs risks inaccurate inference and unintended interpretations. To address the problem, we propose a linguistic framework featuring a **facet-based** representation. Facets are smaller semantic units that constitute the DEs. Our proposed solution emphasizes the **active facets** — the ones that are relevant to the communication purpose.

Based on the framework, we propose hypotheses from both static and dynamic perspectives. The static hypothesis claims that active facets of DEs can be inferred from the context and that coreference linking should be done on these active facets (e.g. the "authority" facet between "the president" and "the White House"). The dynamic hypothesis considers state changes. We treat events as functions of the active facets of the entities and hypothesize only part of the facets are required for analyzing state changes (e.g. state modeling for "burning a paper" should focus on the "flammable" related facets).

In this proposal, we first introduce our early attempts on event semantics, including an event detector and coreference engine, an event sequencing system, and an event ellipsis system. However, the datasets suffer in both scale and domain, which prevents us from exploring complex linguistic phenomenon such as facets. To this end, we attempt to obtain **indirect supervision** from the data. In Part II, we present some successful examples on this line. Finally, in Part III, we propose to validate the facet-based approach using indirect supervision and show that they can improve event semantic modeling.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern NLP systems excel particularly in recognizing and extracting information content from text documents. Yet many challenges exist in the further understanding of semantics. Examples include creating structures to connect the information pieces in discourse (e.g. Semantic Role Labeling, Discourse Parsing), and inferring implicit information from the surface (e.g. Textual entailment). Many of these inference problems involve understanding the structures among discourse elements (DEs).

One important type of DE is event. Events are important building blocks of documents and play a key role in the process of document understanding. In this thesis, we studied various aspects of event semantics and showcased several useful application of such semantic knowledge. Textual realization of events and entities are the main medium connecting to the underlying world. Accurate parsing of them is very important towards document understanding.

We will go through several important components in an event semantics pipeline. Our earlier work mainly uses supervised approaches, including Event Detection (Chapter 2), Event Coreference (Chapter 3), Event Sequencing (Chapter 4), Event Ellipsis (Chapter 5). We realize there is rich linguistic phenomenon surrounding event instances, such as various properties (e.g. Realis) attached to the mentions and the complex relations connecting them.

The amount of training data available for each task is certainly limited. For example, the datasets used generally focus on a small domain. In order to learn more general knowledge about events, we propose to incorporate indirect supervision signals from elsewhere. We describe one of our attempt at Chapter 6 that incorporates a large but noisy dataset to predict the saliency of event mentions.

1.1 Event Semantics

In an underlying world (real or imaginary), we consider the static configuration of entities and their properties as *states*. Events are the process that involving a change of states. For example, a starting state may be a vase being placed on top of a table. If the vase falls down from the table, its physical location has then changed. The change of this state can be called the "fall" event.

In text documents, events are normally realized as spans of text, which are often referred to as **Event Mentions** or **Event Nuggets**. The structures around textual event mentions are very

rich. They may be connected to a location, a time interval, and several participants. The event mentions may also interact with other ones. Analog to the entity coreference problem, the same event can also be represented by multiple text spans, one thus needs to resolve **coreference** to recover events from mentions.

On the basis of events, various other types of relations can then be established between events. For example, events may also collectively form a larger structure. A famous formulation is Schank's script theory [143], which suggests that the information is centered around the event sequences, which enables language understanding and inference. Many event structures are related to Schank's script. For example, the "subevent" relation proposed in [74] organize events into script clusters. The TempEval tasks [153] considers the temporal ordering of events. Following these directions, a new task named **Event Sequencing** (ES) is proposed for TAC KBP 2017, which aims at ordering events from text documents that belong to the same script. Both clustering events into the same script and event ordering are required in the task.

The event structures thus form the backbone of a narrative, which can be viewed as one type of representation of documents. One main goal of this thesis is to recover parse the structure from raw text documents.

1.2 A New View to Events and Entities

In the process of building an event semantic pipeline, we face a lot of problems in doing inference among events and entities. For example, it is possible to establish event coreference link between "a man is `arrested` in `Brazil`" and "the suspect was `arrested` Thursday in the `city of Curitiba in southern Brazil`". However, most entity annotation corpus will not consider `Brazil` to be coreferent with `city of Curitiba`. Similarly, "`The White House`'s announcement" and "`The president`'s announcement" could refer to the same event, but it is seems inappropriate to treat "The White House" and "The president" to be the same. There is an inference gap between event identity and entity identity.

We consider that there are two problems that result in this problem. Firstly, most existing approaches consider the denotation of all these discourse elements as holistic units. However, natural language is not precise and people are sometimes only uses a mention to emphasize part of the mention information. Hence although the full denotation of `the White House` is not the same as `the President`, the emphasized facet are both "the government authority".

Secondly, the identity between entities is context dependent. It depends on the purpose of the entity mentions. In the `Brazil` and `city of Curitiba` example, the purpose served by both entity mentions are the same, though the levels of granularity are not the same.

These problems are related to the phenomenon known as quasi-identity, which is theoretically analyzed by Recasens et al. [134]. In this thesis, we attempt to bring in the relevant theory to events and to close the gap in event-entity inference. In Chapter 7 we will describe our newly proposed facet based representation for entities and events, targeting at providing a more feasible and explainable computational approach.

1.3 Indirect Supervision for Semantics

Learning complex semantics is difficult in many perspectives. On one hand, the semantic tasks are difficult to be annotated even for a human at the first place, which will result in low quality and small-scale dataset. On the other hand, a large amount of training data is often required for learning based systems to successfully capture these knowledge. The result of limited training data will result in a system that is hard to generalize.

Furthermore, many semantic phenomena is better modeled as latent representations, which are difficult to be explicitly annotated from the first place. Examples include the facets of events and entities. Thus a more practical approach is to conduct semi-supervised with available supervision signals. Hence another main goal of this thesis is to study methods to utilize external knowledge and indirect supervision signals for learning event semantics.[1]

[1]This approach is also called "incidental supervision" by Dan Roth.

Part I

Learning with Supervised Approaches

Chapter 2

Event Detection

2.1 Introduction

Detecting the event mention instances is the first and most fundamental step for event semantic modeling. There are different ways to mark an event mention instance in text. One can mark the event predicate span, or the whole context of the event mention, including the arguments. In this chapter, we describe our approach in detecting the predicate span, a task which is often referred to as Event Nugget Detection or Event Trigger Detection. In this chapter, we present our system on event detection on two languages: English and Chinese. Our featured based models are ranked as one the top systems in the TAC-KBP evaluations [103, 104].

2.1.1 Event Mention Detection Task

An event nugget detection task normally requires a system to identify spans of text the refers to an event mention, and then classify them as one of the predefined event types. We follow the task definition in TAC-KBP event track evaluations [103, 104, 105], where each mention is additionally assigned with a realis attribute of ACTUAL (events that actually occurred), GENERIC (mentions of the event in general) or OTHER (other variations). An example annotation is shown in Figure 2.1.

Limited by resources, event detection datasets are normally limited to a small scope. For example, in all the RichERE annotations, there are only 38 type-subtypes, as listed in Table 2.1.

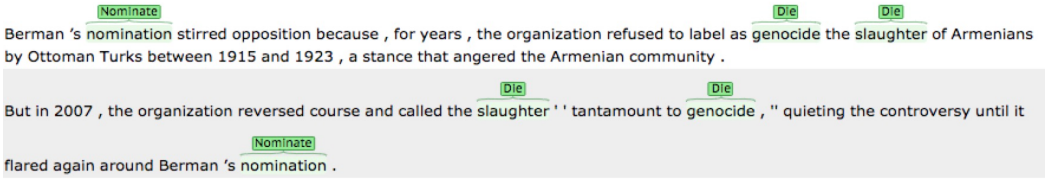


Figure 2.1: An example event nugget annotation

Type	Subtype	Type	Subtype	Type	Subtype
Business	Start-Org	Life	Divorce	Justice	Release-Parole
Business	End-Org	Life	Injure	Justice	Trial-Hearing
Business	Declare-Bankruptcy	Life	Die	Justice	Sentence
Business	Merge-Org	Transaction	Transfer Ownership	Justice	Fine
Conflict	Attack	Transaction	Transfer Money	Justice	Charge-Indict
Conflict	Demonstrate	Transaction	Transaction	Justice	Sue
Contact	Meet	Movement	Transport-Person	Justice	Extradite
Contact	Correspondence	Movement	Transport-Artifact	Justice	Acquit
Contact	Broadcast	Personnel	Start-Position	Justice	Convict
Contact	Contact	Personnel	End-Position	Justice	Appeal
Manufacture	Artifact	Personnel	Nominate	Justice	Execute
Life	Be-Born	Personnel	Elect	Justice	Pardon
Life	Marry	Justice	Arrest-Jail		

Table 2.1: List of Event Types and Subtypes in the RichERE annotations

Other event related triggers in a text documents are ignored. The task of event detection is essentially searching for domain-specific triggers.

2.2 Model for Type Classification

We consider event nugget detection as a sequence labeling task and deploy a Conditional Random Field (CRF) model to detect mention span and event type. The CRF model is trained with the structured perceptron [35], which is outlined in Algorithm 1. The decoding step is done using standard Viterbi algorithm. After training, we obtain the model using the average weight variation as described in Collins [35].

A number of "multi-tagged" mentions are annotated in the corpus, in which a mention might have one or more event types. For instance, an event nugget "KILL" is often associated with "Life-Die" and "Conflict-attack". To deal with them, We simply combine multiple labels for each mention into a single label[1].

[1]Though this can be treated as a multi-label classification problem, however, simple concatenation only result in a label set of 56 types, which can be easily handled

Algorithm 1 Structured perceptron.

Input: training examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
Input: number of iterations T
Output: weight vector $\mathbf{w}$

$\mathbf{w} \leftarrow \mathbf{0}$ ▷ Initialization.
for $t \leftarrow 1..T$ **do**
  for $i \leftarrow 1..N$ **do**
    $\hat{y}^{(i)} = \arg\max_{y \in \mathcal{Y}(x^{(i)})} \mathbf{w} \cdot \Phi(x^{(i)}, y)$
    if $\hat{y}^{(i)} \neq y^{(i)}$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \Phi(x^{(i)}, y^{(i)}) - \Phi(x^{(i)}, \hat{y}^{(i)})$
return $\mathbf{w}$

An event mention is normally composed by its mention trigger and the arguments. To get a list of arguments for the event mention. We run two Semantic Role Labeling system, the PropBank style Fanse Parser [42] and the FrameNet style Semafor Parser [150]. In addition, to reduce sparsity, we further incorporate a few external data resources, including WordNet [54] and a set of Brown Clustering labels trained on newswire data [149]. Lemma, part-of-speech, NER and parsing information are all obtained through the Stanford CoreNLP system [98].

Using these resources, we employ regular linguistic features for mention type detection, which are summarized as followed:

1. Part-of-Speech, lemma, named entity tag of words in the 2-word window of the trigger (both side), the trigger word itself and the direct dependent words of the trigger.
2. Brown clusters, WordNet Synonym and derivative forms of the trigger.
3. Whether the words in the 5 word window match some selected WordNet senses, including "Leader", "Worker", "Body Part", "Monetary System", "Possession", "Government", "Crime" and "Pathological State".
4. Closest named entity type.
5. Dependency features, including lemma, dependency type and part-of-speech of the child dependencies and head dependencies.
6. Semantic role related features includes the frame name and the argument role, named entity tag, argument head word lemma and WordNet sense (selected from the above list as well) of the arguments.

The WordNet related features are selected following the intuition that certain category of words are likely to imply the existence of certain events. For example, "Leader" are normally associated with "Personnel" type. The model generalize better by selecting appropriate levels of word sense.

2.3 Realis

We train a separate Realis detection model using Supper Vector Machine in LIBLINEAR[2]. We reuse many features from the mention detection to capture the context of these mentions. However, we exclude most of the lexicalized features because they tend to be overfitting in our prior experiments. We design a specific feature to capture whether the phrase containing the event mention is quoted (if the whole sentence is quoted, we do not fire this feature).

2.4 Adapting to a Chinese Event Detection system

To extend our system to handle Chinese documents, we develop similar features for Chinese. Most of the features can be reused without changes in the Chinese system, which includes: window based features[3], syntactic based features, entity features, head word features and SRL features. We also use the Brown clustering features with clusters induced form Chinese Gigaword 3[4].

Due to the nature of Chinese language, the Chinese tokens normally contain more internal structure and each single character in the token may convey useful semantic information. The way how the individual characters combine will affect the semantic of the event word. This is previously discussed in Li et al. [82] as verb structures. In other words, the position of a character in the verb matters. For example, the character 解 means "unbind" in the word 解雇 (fire), but means "console" in the word 劝解 (console). Following these intuition, we further add the following character related features:

1. Whether the token contains a character.
2. The contained character and its character level Part-of-Speech.
3. The first character of the token.
4. The last character of the token.
5. Base verb structure feature as described in [82]: we use a feature to represent one of the base verb structure. In addition to the 6 main structures proposed by Li et al. [82], we added 3 structures for completeness: 1. No verb character found 2. The verb character is found after 2 characters and 3. Other: any cases that are not defined above.

2.4.1 Improving Recall on Chinese

During the system development, we observe that our Chinese system suffers from serious low recall despite all the features we added in. By following the training procedures, we hypothesize that the annotated Chinese data is not complete (see § 2.6.2 for more discussion). As a result, our learning algorithm will be biased by the missed events and learn incorrect negative signals. The final model thus will be very conservative in making predictions, leading to a low recall.

We mitigate the problem by ignoring all training sentences that do not contain an event mention, which reduce the probability of missed annotations. On our development experiments,

[2]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

[3]However since the discussion forum training data are quite noisy, we restrict the POS window to 1 instead.

[4]`http://www.cs.brandeis.edu/~clp/conll15st/dataset.html`

we found that this simple trick can directly raise our nugget detection performance by about 3%. The performance improvement also support our hypothesis that the Chinese dataset is indeed not fully annotated.

2.5 Experiments

We have participated in TAC-KBP Event Nugget track 2015 and 2016 [91, 93]. We thus follow the official evaluation setup. The TAC-KBP 2015 track provides a training corpus of 158 documents and an evaluation set of 202 documents. There are no new training data provided in TAC-KBP 2016, thus we train our system using both the training and testing data from TAC-KBP 2015. The evaluation set of TAC-KBP 2016 contains 169 documents. The datasets are all coming from two different genre: newswire or discussion forum. One major change in the TAC-KBP 2016 evaluation is that the types required for evaluation is reduced to 18 types, which is a subset of the previous 38 types.

2.5.1 Evaluation Results

Here we directly report our official system performance in the evaluation workshop [103, 104]. In TAC-KBP 2015, our official event nugget performance is summarized in Table 2.4a. Our system ranks the third place in the Span and Realis sub-score, and second in the All attribute sub-score. In TAC-KBP 2016, our official english nugget score in Table 2.4b. Our system ranks 2nd among all the participants in terms of the final event type F1 score (Table 2.2, our system name is LTI-CMU1). The other sub-scores are also competitive, which ranks at the top few systems.

	Prec.	Recall	F1
UTD1	47.66	46.35	46.99
LTI-CMU1	61.69	34.94	44.61
wip1	51.76	38.98	44.47
NYU3	41.88	47.21	44.38
SoochowNLP3	49.92	38.81	43.67
RPI-BLENDER2	44.51	39.87	42.07
SYDNEY1	46.48	30.33	36.70
Washington1	42.15	29.41	34.65
aipheshd-t161	36.83	29.28	32.62
UMBC2	37.36	27.33	31.57
HITS3	41.79	25.30	31.52
CMUML3	60.44	15.58	24.77
UI-CCG2	25.81	18.53	21.57
IHMC20161	0.69	0.52	0.59

Table 2.2: English Nugget Type System Ranking of TAC-KBP 2016

We are a little surprise to see our English nugget detection performance drops about 13% (span and type) comparing to from KBP 2015 to KBP 2016. However, our relative ranking is almost unchanged. Our analysis [104] has shown that this is actually because the change in the type set: systems on both years perform equally well (or even better in 2016) on the selected types. In fact, the average score on these types are actually lower than the full set.

Our Chinese event detection systems also produce competitive results. The Chinese system is the first place on all the sub-score in TAC-KBP 2016 (Table 2.3).

		Prec.	Recall	F1
Span	LTI-CMU1	56.46	39.55	46.52
	UTD1	47.23	43.16	45.1
	LTI-CMU3	56.19	35.35	43.4
	UI-CCG1	28.34	39.61	33.04
	RPI-BLENDER1	62.46	18.48	28.52
Type	LTI-CMU1	50.72	35.53	41.79
	UTD1	41.9	38.29	40.01
	LTI-CMU3	49.7	31.26	38.38
	UI-CCG1	24.01	33.55	27.99
	RPI-BLENDER2	59.87	17.5	27.08
Realis	LTI-CMU1	42.7	29.92	35.18
	UTD1	35.27	32.23	33.68
	LTI-CMU3	43.11	27.12	33.29
	RPI-BLENDER2	48.46	14.16	21.92
	UI-CCG1	9.65	13.49	11.25
All	LTI-CMU1	38.91	27.26	32.06
	UTD1	31.76	29.02	30.33
	LTI-CMU3	38.54	24.25	29.77
	RPI-BLENDER2	46.69	13.65	21.12
	UI-CCG1	8.31	11.62	9.69

Table 2.3: Official Chinese Nugget Performance Ranking at TAC-KBP 2016.

In order to test the effect of our realis model, we evaluate its performance given gold standard mention span and types in the training data. We report the 5-fold validation result in Table 2.5. Note that the precision and recall are the same because the gold spans are given.

2.6 Discussions

2.6.1 Multi-Type Events and Facets

There is a small number of event types in the evaluations. Further, the possible types of double tagging are limited by in the dataset. This is because the current annotation scheme only considers

	Precision	Recall	F1
Span	82.46	50.30	62.49
Type	73.68	44.94	55.83
Realis	62.09	37.87	47.05
All	55.12	33.62	41.77

(a) Official English Event Nugget Performance at TAC-KBP 2015.

	Prec.	Recall	F1
Span	69.82	39.54	50.49
Type	61.69	34.94	44.61
Realis	45.78	25.93	33.11
All	40.19	22.76	29.06

(b) Official English Event Nugget Performance at TAC-KBP 2016.

Fold	Precision	Recall	F1
1	71.68	71.63	71.66
2	64.06	64.06	64.06
3	62.07	62.07	62.07
4	72.66	72.66	72.66
5	62.21	62.21	62.21

Table 2.5: 5-fold validation for Realis detection on training data with gold span and types

a limited pool of event types. Our current solution is simply treating the double-tagged types as a new class in classification. However, we realize there are rich phenomenon behind this. In fact, we find the event arguments to be closely related to these different event types. For example, a predicate "kill" may have types "Conflict.Attack" and "Life.Death": the former one is more related to the state of the "Attacker" while the latter one relates to the state of the "Victim". This is actually an example of event facet, which will discuss this more in Chapter 7. The facet based representation can be a more principled solution to this case.

2.6.2 Chinese Data Annotation

We hypothesize that the Chinese datasets are not fully annotated. We take a closer look in the data and found a number of missed event nuggets. Here we list a couple examples:

(2.1) 支持香港同胞争取[Personnel.Elect 选举]与被[Personnel.Elect 选举]权！

(2.2) 司务长都是骑着二八去[TransferOwnership 买]菜去。

(2.3) 海豹行动是绝密，塔利班竟然可以预先得知？用个火箭就可以[Conflict.Attack打]下来，这个难度也实在是太高了吧。

In the above examples, we show several event nuggets. However, mentions annotated in red are not actually annotated in the Rich ERE datasets. Especially, in example 2.1, the first 选举 is annotated but the second one is not. Such inconsistencies happen a lot across the dataset. When training with such data, the classifier will likely to be quite conservative on making event nugget predictions. We conduct a very simple quantitative analysis by comparing the ACE 2005 Chinese

annotation against the Rich ERE Chinese annotation. Table 2.6a and Table 2.6b summarize the top 5 double-character tokens annotated in ACE and RichERE. For the most popular event mentions, Rich ERE annotated only a smaller percentage comparing to ACE.

In addition, we find that the most popular event nuggets are mostly single character in the Rich ERE datasets, such as 打(170), 说(148), 死(131), 杀(118) . In fact, in top 20 most popular event nuggets of Rich ERE, there are 17 single-character nuggets, this number is only 6 in ACE. These single character tokens are more ambiguous comparing to a double character mention (for example, 打 can represent the action of "calling someone" or "attacking someone", which corresponds to very different event type. This is because language in discuss forum posts are normally not formal. This actually challenges our event nugget systems to deal with deeper semantic problems.

Token	Annotated	Total	%
冲突	100	119	84.03%
访问	64	90	71.11%
受伤	53	59	89.83%
死亡	46	50	92.00%
前往	44	52	84.62%

(a) Top 5 double character mentions in ACE 2005 Dataset

Token	Annotated	Total	%
战争	96	223	43.05%
死亡	24	33	72.73%
暗杀	22	40	55.00%
入侵	18	22	81.82%
自杀	17	33	51.52%

(b) Top 5 double character mentions in TAC-KBP 2016 Dataset

Chapter 3

Pairwise Event Coreference

3.1 Introduction

Coreference resolution, the task of linking surface mentions to their underlying discourse entities, is an important task in natural language processing. Most of the early work focused on coreference of entity mentions. Recently, event coreference has attracted attention on both theoretical and computational aspects. However, most event coreference work is preliminary and applied in quite different circumstances, making comparisons difficult or impossible.

In this chapter, we first provide an overview of many prior work around event coreference and highlights the differences in settings. The comparisons to related work in prior papers are not really appropriate due to these differences. We then present a supervised approach to event coreference, and describe a method for propagating information between events and their arguments that can improve results. In our method, different argument types support different methods of propagation. For these experiments, we annotate and use a corpus of 65 documents in the Intelligence Community (IC) domain that contains a rich set of within-document coreference links [74].

3.2 Related Work

Table 3.1 summarizes recent work on event coreference resolution. For the reasons below, only one supervised system [1] and two unsupervised [10, 38] on within-document event coreference are suitable as a basis for ongoing comparison.

3.2.1 Problem Definition

Different approaches use different definitions of the problem (see Compatible Definition column). However, as discussed in recent linguistic studies [74, 134], the existence of different types and degrees of coreference makes it necessary to agree on the definition of coreference before performance can be compared. The lack of clarity about what coreference should encompass rules out several systems for comparison. OntoNotes created restricted event coreference [128], linking only some nominalized events and some verbs, without reporting event-specific results. Both

	Gold standard used	Cross/within Document		Compatible definition	Corpus
		Within	Cross		
Lee et al. [80]			✓	✓	ECB
Sangeetha and Arock [142]	ACE mention, arguments and attributes	✓		✓	ACE
Cybulska and Vossen [38]	Mention head word	✓		✓	IC
McConky et al. [100]	ACE mention, arguments and attributes	✓		✓	ACE
Li et al. [83]	Human entity and event mention detection	✓		✓	Unavailable
Bejan and Harabagiu [10]		✓(ACE, ECB)	✓(ECB)	✓	ECB, ACE
Chen and Ji [26]	ACE mention, arguments and attributes	✓		✓	ACE
Elkhlifi and Faiz [51]		✓			Unavailable
Naughton [109]		✓			IBC, ACE
Pradhan et al. [128]		✓			OntoNotes
Ahn [1]	ACE mention, arguments and attributes	✓		✓	ACE
Bagga and Baldwin [5]			✓		Unavailable

Table 3.1: A range of event coreference resolution work with different settings.

Naughton [109] and Elkhlifi and Faiz [51] worked on sentence-level coreference, which is closer to the definition of Danlos [40]. However it is unclear when one sentence contains multiple event mentions, and hence these are not comparable to systems that process more specific coreference units.

3.2.2 Dataset and Settings

Early work by Bagga and Baldwin [5] conduct experiments only on cross-document coreference. Recent advanced work on event coreference is by Bejan and Harabagiu [10] and Lee et al. [80] use the ECB corpus[1] (or a refined version[2]) to evaluate performance, which is annotated mainly for cross-document coreference. In this corpus, within-document coreference is only very partially annotated; most difficult coreference instances are not marked.

(3.1) 1. Indian naval forces `came to the rescue (E1)` of a merchant vessel under `attack (E3)` by pirates in the Gulf of Ade on Saturday, `capturing (E2)` 23 of the raiders, India `said (E4)`.

2. Indian commandos boarded the larger pirate boat, **seizing** 12 Somali and 11 Yemeni nationals as well as arms and equipment, the statement `said`.

(3.2) 1. The Indian navy `captured (E2)` 23 piracy suspects who `tried (E5)` to `take over (E3)` a merchant vessel in the Gulf of Aden, between the Horn of Africa and the Arabian Peninsula, Indian officials `said (E4)`.

2. In addition to the 12 Somali and 11 Yemeni suspects, the Indian navy `seized` two small boats and "a substantial cache of arms and equipment", the military `said` in a statement.

The examples sentences are extracted from two documents from the ECB. In both documents, event mentions appear in the first sentence are annotated once, but not in the subsequent sentences. In example 3.1, we find in one of the subsequent sentences the event mention "`seizing`" which should actually marked as coreferent with "`capturing (E2)`". In example 2, we find a more tricky case: the mention "`seized`", which has semantics similar to "`captured`" but this pair is not marked as coreference due to different patients. In cross-document settings, we also find discrepancies between the definitions. In ECB, "`attack (E3)`" in example 3.1 is annotated as coreferent with "`take over (E3)`" in example 1, which we believe is wrong: at best, the attack is only a part of the attempt to take over the merchant vessel.

Goyal et al. [63] use a distributional semantic approach on event coreference. However, they adopt a unconventional evaluation setting. They draw from the IC corpus an equal number of positive and negative testing examples, which is different from the natural data distribution.

3.2.3 Gold Standard Annotations

Recent work using the ACE 2005 corpus[3] Ahn [1], Chen and Ji [26], Chen et al. [27], McConky et al. [100], Sangeetha and Arock [142] agrees with our definition of coreference. However, the

[1] http://adi.bejan.ro/data/ECB1.0.tar.gz
[2] http://nlp.stanford.edu/pubs/jcoref-corpus.zip
[3] http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/

	Total	Avg.
Event Mention	2678	41.2
Non-elliptical Domain Event Mention	1998	30.7
Reporting Event Mention	669	10.29
Full coreference relations	1253	21.6
Subevent relations (parent-child)	455	8
Membership relations (parent-child)	161	2.9

Table 3.2: Corpus Statistics

ACE corpus annotations, in addition to event mentions, also include argument structures, entity ids, and time-stamps. Most coreference systems on the ACE corpus make use of this additional information. This makes them impossible to compare to systems that do not make this simplifying assumption. It also makes results achieved on ACE hard to compare to results on corpora without this additional information. Among these work, only Ahn [1] reported some results using system generating arguments, we compare our system against it.

There are also other problems that make the comparison difficult. Li et al. [83] use a hand-annotated web corpus, which is not publicly available for comparison. In summary, anyone wanting to work on within-document event coreference has to obtain a corpus that is fully annotated, that does not include additional facilitating information, whose definition of coreference respects the theoretical considerations of partial coreference, and that has other systems freely available for comparison. Meeting these criteria is not easy. The closest work we find is by Cybulska and Vossen [38] and Bejan and Harabagiu [10], both adopt unsupervised methods for event coreference. Ahn [1] also reported results on ACE by swapping gold standard annotations with system results.We compare our system to their results on their corresponding corpus.

3.3 Corpus

Our system is trained and evaluated on the IC domain corpus, which annotates several different event relations. Table 3.2 summarizes the corpus level statistics and the average over documents. In this work, we focus on full coreference relations. The inter-annotator agreement among 3 annotators for full coreference is 0.614 in terms of Fleiss' kappa [59]. For detailed definition for the corpus, we refer readers to Hovy et al. [74]. To facilitate future research, We also report our system results on the ACE 2005 training dataset, which contains 599 documents.

3.4 System description

Our system is almost end-to-end, except that we start with a minimal gold standard head word annotations in order to focus on the core coreference problem. This approach is the same as Cybulska and Vossen [38] and Bejan and Harabagiu [10][4].

3.4.1 Procedure

Similar to Chen et al. [27], we approach the problem first with a conventional pairwise model:

1. Supervised classification that determines the probability whether two mentions co-refer. The classifier used in the experiment is Random Forest [13], implemented in Weka [66].
2. Clustering that processes all the pairwise scores to output the final clusters of pairs.
3. In addition, we added a third step after clustering, information is propagated between event mentions to enrich the original feature set.

The last step tries to enrich the event representations during clustering. Typically, the information carried from one event to its coreferent mention is about the participants (agent, patient, etc.). When an event has been enriched by receiving information from another, it may in turn now be linkable to a third event. The system repeats this process until no more information can be propagated. Currently, the propagation includes two parts: 1. if one mention has missing arguments, they will be copied over from the co-referred counterpart; 2. if both arguments are present, information not presented in one will be copied from another.

Similarly, Lee et al. [80] show that jointly modeling references to events and entities can boost the performance on both. We hold a similar assumption. But by focusing on events and their arguments, we can perform propagation specific to each type of argument, for instance, geographical reasoning as described below.

3.4.2 Features

In addition to typical lexical and discourse features, we also model an event mention with its surface form and its arguments, including agent, patient[5], and location. We use a rich set of 105 semantic features, described in table 3.3.

Agent, patient extraction and propagation

We use the semantic parser *Fanse* [150] to annotate the predicate arguments defined in PropBank. For nominal events, we extract agent and patient using heuristics such as finding the token attached to the event mention with specific words (such as "by") and modifiers as agent (e.g., HAMAS in HAMAS's attack). During the propagation step, information not present in one entity can be copied from another.

[4] Although Bejan and Harabagiu [10] use automatic mention detection to extend the mention set for training, they only use true mentions of the ACE dataset at evaluation time.

[5] Specifically, these are defined as ARG0, ARG1 in PropBank. They could be more-specific variants roles such as *experiencer*, but we prefer a smaller set for simplicity.

Location extraction and propagation

In contrast to agent and patient, the propagation of location information employs external information to gain additional power. We use the *Stanford Entity Recognition* [58] engine to identify location mentions. *DBpedia Spotlight* [101] is run to disambiguate location entities. DBpedia [3] information, such as cities, country, and alternative names, are then injected. When the location is not found in DBpedia, we search the mention string in *GeoNames*[6] and use the first result with highest Dice coefficient with the surface string. This world knowledge enriches annotation. For example, we can now match the mention "Istanbul" with the country name "Turkey".

3.4.3 Clustering

We conduct experiments with two simple clustering methods. The first is a pure transitive closure that links all pairs mentions that the classification engine judges as positive. The second is the Best-Link algorithm of Ng and Cardie [110], which links each mention to its antecedent with the highest likelihood when the classifier judges as positive.

3.5 Evaluation

3.5.1 Evaluation Metrics

Coreference evaluation metrics have been discussed by the community for years. To enable comparison, we report most metrics used by the CoNLL 2012 shared task [127], including MUC [30], B-Cubed [4], entity-based CEAF [97], and BLANC [132]. Pairwise scores are used to provide a direct view on performance.

3.5.2 Experiments and Results

We split the documents in IC corpus randomly into 40 documents for training and development, and 25 for testing. Parameters such as the probability threshold to determine coreference are tuned on the 40 documents using five-fold cross validation. Optimization is not done separately for each metric. We simply use a universal classifier threshold optimized for pairwise case. During experiment, the propagation step is actually performed for only one iteration, since no further information is propagated. On the ACE corpus, we simply apply the best model configuration from IC corpus and train on 90% of the documents (539) for training and 10% for testing (60).

Table 3.3 summarizes the overall average results obtained by BestLink on both ACE and IC corpus (BestLink consistently outperforms naively full transitive closure). We also attach three other reported results at the end. Note that these results are not directly comparable: Cybulska and Vossen [38] and Bejan and Harabagiu [10] use unsupervised methods, thus their reported results are evaluated on the whole corpus; Ahn [1] also use a 9:1 train-test split, but the split might be different with ours. A simple comparison shows that our results outperform these systems in all

[6] http://www.geonames.org/

metrics, which is notable because all these metrics are designed to capture the performance from different aspects.

To interpret the results, it should also be noted that because of the existence of large number of singleton clusters, some measures such as B^3 seem to be high even using the most naive feature set. By looking at the pairwise performance, however, we see that current best F-score is only about 50%. There are still many challenges in event coreference.

	Pairwise			MUC			B^3			CEAF-e			BLANC		
IC corpus	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Discourse + Lexical	32.69	25.11	28.40	41.7	33.58	37.2	79.46	74.06	76.67	66.89	73.95	70.24	59.77	61.2	60.43
+ Syntactic	47.12	35.15	40.26	52.6	47.63	50.0	82.24	81.46	81.85	76.91	80.21	78.53	64.76	68.59	66.42
+ Semantic (no arguments)	51.15	42.22	46.26	54.5	49.1	51.68	82.12	82.08	82.1	74.93	78.31	76.58	65.41	69.98	67.35
+ Arguments	55.96	47.86	51.60	56.87	**55.81**	56.33	83.38	**85.58**	**84.46**	**88.13**	80.73	**80.43**	68.77	**75.21**	71.46
+ Propagation	**59.04**	**48.27**	**53.11**	**68.72**	55.5	**61.44**	**89.28**	79.89	84.33	75.14	**82.9**	78.83	**82.28**	70.77	**75.06**
Cybulska and Vossen [38]	—	—	—	—	—	—	81.0	71.0	76.0	—	—	—	—	—	—
ACE corpus															
This work	55.86	40.52	**46.97**	53.42	48.75	50.98	89.9	88.86	**89.38**	85.54	87.42	**86.47**	70.88	70.01	70.43
Bejan and Harabagiu [10]	43.3	47.1	45.1	—	—	—	83.4	84.2	83.8	76.9	76.5	76.7	—	—	—
Ahn [1]	—	—	43.3	—	—	—	—	—	—	—	—	—	—	—	—

Table 3.3: Evaluation results and comparisons

3.6 Discussion

The evaluation results show that almost all types of features help to improve the performance over all metrics rather consistently. However, preliminary error analysis shows that some events are still clustered incorrectly even when arguments match. We argue that limitations in argument extraction and entity coreference prevent these features from contributing directly to correct coreference decisions. On the other hand, the results of propagation show that new information helps to find more links but inevitably comes with a drop in precision. We consider that modeling event and arguments holistically like Lee et al. [80] would help guide the propagation. By inspecting the data, we hypothesize that the main benefits brought by the propagation scheme is to match arguments of two coreferent events. If the arguments are nominal events, they will be then marked as coreferent due to the feature "Event as Entity" (See Semantic features in table 3.4). In the following example, if the two event mentions "planning" are marked as coreference, then the corresponding argument "attack" will be also marked as coreference.

(3.3) A member of the Islamic militant movement HAMAS suspected of **planning** a suicide **attack** against Israel surrendered to Palestinian police here after a six-hour shootout on Friday. HAMAS's military wing, was on the run from both Palestinian and Israeli police for **planning** anti-Israeli **attacks**.

This hypothesis is also in line with our observation that propagation can only be performed for one round, because the nominal event themselves are unlikely to have other nominal events as arguments. Such interactions between event mentions also remind us that conference can be possibly improved by other types of event relations, such as subevent relations.

Furthermore, the system tends to merge clusters where the event mention head words are the same because the head word feature receives a high weight in the model, even when this is not appropriate. More work should be performed on disambiguating such difficult cases.

We show that rich linguistic features, especially event arguments, can improve event coreference performance. Argument specific information propagation further help finding new relations. However, our proposed model is based on a simple pairwise event coreference model, which we haven't incorporate the structural information of the document. In the next chapter, we will propose a structure aware model for event coreference.

Type (counts)	Feature Name	Description
Discourse (5)	Sentence Distance	Number of sentences between two events mentions.
	Event Distance	Number of event mentions between two event mentions.
	Position	One event is in the title, or the first sentence.
Lexical (12)	Surface Similarity	Several string similarity measures computed between event mention headwords: Dice coefficient, edit distance, Jaro coefficient, lemma match and exact phrase match.
	Modifier Similarity	Dice coefficient similarity of the modifiers of the event mentions.
Syntactic (38)	Part Of Speech	Binary features for plurality, tense, noun or verb for the event mention head words.
	Dependency	The dependency label connecting the two event mentions.
	Negation	Whether two mention head words are both negated.
	Determiner	Whether the event mentions are modified by determiners
Semantic (16)	Coreference	Whether the predicates are in the same entity coreference cluster (only for nominal events).
	WordNet Similarity	Wu-Palmer similarity [117] of the headword pair.
	Senna Embeddings	Cosine similarity of event mention head word embeddings (Senna embeddings [36]).
	Distributional	Distributional similarity between predicates in Goyal et al. [63].
	Verb Ocean	Predicate word relations in "Verb Ocean" [31].
	Semantic Frame	Whether two predicates trigger the same semantic frame (extracted by Semafor [41]).
	Mention Type	Predicate word type (generated by IBM Sire [60]) match.
Arguments (34)	Surface	Dice coefficient and Wu-Palmer similarity between argument pairs;
	Coreference	Entity coreference between argument pairs; Numeric word match between the argument pairs.
	Existence	Whether each argument slot is instantiated.
	Location	Containment and alternatives name match between the location arguments based on geographical resources such as DBpedia [3] and GeoNames.

Table 3.4: List of features (with counts) in the pairwise model. Entity coreference are from the Stanford Entity Coreference Engine [79].

Chapter 4

Graph Based Event Coreference and Sequencing

4.1 Introduction

The rich relations among the textual event mentions help connect them. The mentions then collectively convey the meaning of the narrative. In the previous chapter, we introduce a pairwise approach for event coreference. However, the simple pairwise model does not consider the structural constraints. In this chapter, we propose a graph based approach, and apply them to two different types of relation: **Event Hopper Coreference (EH)** and **Event Sequencing (ES)**.

Event Hopper Coreference: In this chapter, we use the dataset of the TAC-KBP dataset, that defines a new **Event Hopper** Coreference task [105]: Two event mentions are considered coreferent if they refer to the conceptually same underlying event, even if their arguments are not strictly identical. For example, mentions that share similar temporal and location scope, though not necessarily the same expression, are considered to be coreferent (*Attack in Baghdad on Thursday* vs. *Bombing in the Green Zone last week*). This means that the event arguments of coreferential events mentions can be non-coreferential (18 killed vs. dozens killed), as long as they refer to the same event, judging from the available evidence.

Event Sequencing: The coreference relations build up events from scattered mentions. On the basis of events, various other types of relations can then be established between them. The Event Sequencing task studies one such relation. The task is motivated by Schank's *scripts* [143], which suggests that human organize information through procedural data structures, reassembling sequences of events. For example, the list of verbs *order, eat, pay, leave* may trigger the restaurant script. A human can conduct reasoning with a typical ordering of these events based on common sense (e.g., *order* should be the first event, *leave* should be the last event).

The ES task studies how to group and order events from text documents belonging to the same script. Figure 4.1 shows some annotation examples. Conceptually, event sequencing relations hold between the events, while coreference relations hold between textual event mentions. Given a document, the ES task requires systems to identify events within the same script and classify their inter-relations. These relations can be represented as labeled Directed Acyclic Graphs (DAGs).

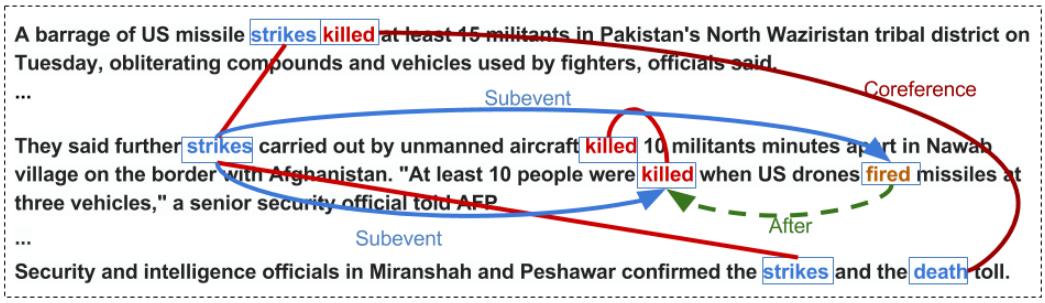


Figure 4.1: Example of Event Coreference and Sequence relations. Red lines are coreference links; solid blue arrows represent Subevent relations; dotted green arrows represent After relations.

There are two types of relations[1]: **After** relations connect events following script orders (e.g. *order* followed by *eating*); **Subevent** relations connect events to a larger event that contains them. In this paper, we focus only on the **After** relations.

Since script-based understanding is built in the ES task, it has some unique properties comparing to pure temporal ordering: 1. event sequences from different scripts provide separate logical divisions of text, while temporal ordering considers all events to lie on a single timeline; 2. temporal relations for events occurring at similar time points may be complicated. Script-based relations may alleviate the problem. For example, if a `bombing` `kills` some people, the temporal relation of the `bombing` and `kill` may be "inclusion" or "after". This is considered an **After** relation in ES because `bombing` causes the `kill`ing.

For structure prediction, decoding — recovering the complex structure from local decisions — is one of the core problems. The most successful decoding algorithm for coreference nowadays is mention ranking based [11, 50, 81]. These models rank the antecedents (mentions that appear earlier in discourse) and recover the full coreference clusters from local decisions. However, unlike coreference relations, sequencing relations are directed. Coreference decoding algorithms cannot be directly applied to such relations (§4.3.1). To solve this problem, we propose a unified graph-based framework that tackles both event coreference and event sequencing. Our method achieves state-of-the-art results on the event coreference task (§4.4.4) and beats an informed baseline on the event sequencing task (§4.4.5). Finally, we analyze the results and discuss the difficult challenges for both tasks (§4.5). Detailed definitions of these tasks can be found in the corresponding task documents[2].

4.2 Related Work

Many researchers have worked on event coreference tasks since Humphreys et al. [75]. In the previous chapter, we have summarized a variety of work under different settings on event coreference. Recent advances in event coreference have been promoted by the standardized annotation

[1]Detailed definition of relations can be found in http://cairo.lti.cs.cmu.edu/kbp/2016/after/
[2]`http://cairo.lti.cs.cmu.edu/kbp/2017/event/documents`

corpora. A lot of work is conducted on the popular ACE corpus [24, 25, 26, 27, 142]. Unlike the TAC KBP setting, the definition of event coreference in the ACE corpus requires strict argument matching.

In this chapter, we mainly follow the TAC-KBP event nugget tasks [103]. There is a small but growing amount of work on conducting event coreference on the TAC-KBP datasets [95, 96, 120]. The TAC dataset uses a relaxed coreference definition comparing to other corpora, requiring two event mentions to intuitively refer to the same real-world event despite differences of their participants.

For event sequencing, there are few supervised methods on script-like relation classification due to the lack of data. To the best of our knowledge, the only work in this direction is by Araki et al. [2]. This work focuses on the other type of relations in the event sequencing task: **Subevent** relations. There is also a rich literature on unsupervised script induction [18, 29, 57, 123, 139] that extracts scripts as a type of common-sense knowledge from raw documents. The focus of this work is to make use of massive collections of text documents to mine event co-occurrence patterns. In contrast, our work focuses on parsing the detailed relations between event mentions in each document.

Another line of work closely related to event sequencing is to detect other temporal relations between events. Recent computational approaches for temporal detection are mainly conducted on the TimeBank corpus [130]. There have been several studies on building automatic temporal reasoning systems [17, 46, 152]. In comparison, the Event Sequencing task is motivated by the Script theory, which places more emphasis on common-sense knowledge about event chronology.

4.3 Model

4.3.1 Graph-Based Decoding Model

In the Latent Antecedent Tree (LAT) model popularly used for entity coreference decoding [11, 55], each node represents an event mention and each arc a coreference relation, and new mentions are connected to some past mention considered most similar. Thus the LAT model represents the decoding structure as a tree. This can represent any coreference cluster, because coreference relations are by definition equivalence relations[3].

In contrast, tree structures cannot always fully cover an Event Sequence relation graph, because 1. the After links are directed, not symmetric, and 2. multiple event nodes can link to one node, resulting in multiple parents.

To solve this problem, we extend the LAT model and propose its graph version, namely the Latent Antecedent Graph (LAG) model. Figure 4.2 contrast LAT and LAG with decoding examples. The left box shows two example decoded trees in LAT, where each node has one single parent. The right box shows two example decoded trees in LAG, where each node can be linked to multiple parents.

Formally, we define the series of (pre-extracted) event mentions of the document as $M = \{m_0, m_1, \ldots, m_n\}$, following their discourse order. m_0 is an artificial root node preceding all mentions. For each mention m_j, let $A_j = \{m_0, m_1, \ldots, m_{j-1}\}$ be the set of its potential antecedents:

[3] An equivalence relation is reflexive, symmetric and transitive.

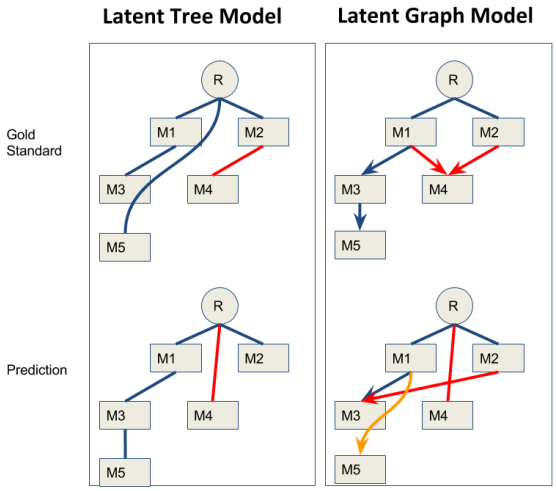


Figure 4.2: Latent Tree Model (left): tree structure formed by undirected links. Latent Graph Model (right): a DAG form by directed links. Dashed red links highlight the discrepancy between prediction and gold standard. The dotted yellow link (bottom right) can be inferred from other links.

Let $\mathcal{A}$ denotes the set of antecedents for all the mentions in the sequence $\{A_0, A_1, \ldots, A_n\}$. The two tasks in question can be considered as finding the appropriate antecedent(s) from $\mathcal{A}$. Similarly, we define the gold antecedent set $\tilde{\mathcal{A}} = \{\tilde{A}_0, \tilde{A}_1, \ldots, \tilde{A}_n\}$, where $\tilde{A}_i$ represent the set of antecedents of m_i allowed by the gold standard. In the coreference task, $\tilde{A}_i$ contains all antecedents that are coreferent with m_i. In the sequencing task, $\tilde{A}_i$ contains all antecedents that have an $After$ relation to m_i.

We can now describe the decoding process. We represent each arc as $\langle m_i, m_j, r\rangle (i < j)$, where r is the relation name. The relation direction can be specified in the relation name r (e.g. r can be *after.forward* or *after.backward*). Further, an arc from the root node m_0 to node m_j represents that m_j does not have any antecedent. The score of the arc is the dot product between the weight parameter $\vec{w}$ and a feature vector $\Phi(\langle m_i, m_j, r\rangle)$, where Φ is an arc-wise feature function. The decoded graph z can be determined by a set of binary variables $\vec{z}$, where $\vec{z}_{ijr} = 1$ if there is an arc $\langle m_i, m_j, r\rangle$ or 0 otherwise. The final score of z is the sum of scores of all arcs:

$$score(z) = \sum_{i,j,r} \vec{z}_{ijr} \vec{w} \cdot \Phi(\langle m_i, m_j, r\rangle) \tag{4.1}$$

The decoding step is to find the output $\hat{z}$ that maximizes the scoring function:

$$\hat{z} = \arg \max_{z \in \mathcal{Z}(\mathcal{A})} score(z) \tag{4.2}$$

where $\mathcal{Z}(\mathcal{A})$ denotes all possible decoding structures given the antecedent sets $\mathcal{A}$. It is useful to note that the decoding step can be applied in the same way to the gold antecedent set $\tilde{\mathcal{A}}$.

Algorithm 2 shows the Passive-Aggressive training algorithm [37] used in our decoding framework. Line 7 decodes the maximum scored structure from all possible gold standard

structures using the current parameters $\vec{w}$. Intuitively, this step tries to find **the "easiest" correct graph** — the correct graph with the highest score — for the current model. Several important components remain unspecified in algorithm 2: (1) the decoding step (line 5, 7); (2) the match criteria: whether to consider the system decoding structure as correct (line 6); (3) feature delta: computation of feature difference (line 8); (4) loss computation (line 9). We detail the actual implementation of these steps in §4.3.1.

Algorithm 2 PA algorithm for training

Input: Training data D, number of iterations T

1:

Output: Weight vector $\vec{w}$

2:

3: $\vec{w} = \vec{0}$;

4: $\langle \mathcal{A}, \tilde{\mathcal{A}} \rangle \in D$;

5: **for do** $t \leftarrow 1..T$ $\hat{z} = \arg\max_{\mathcal{Z}(\mathcal{A})} score(z)$

6: **if** $\neg Match(\hat{z}, \tilde{\mathcal{A}})$ **then**

7: $\tilde{z} = \arg\max_{\mathcal{Z}(\tilde{\mathcal{A}})} score(z)$

8: $\Delta = FeatureDelta(\tilde{z}, \hat{z})$

9: $\tau = \frac{loss(\tilde{z}, \hat{z})}{||\Delta||^2}$

10: $w = w + \tau\Delta$

return w

Minimum Decoding Structure

Similar to the LAT model, there may be many decoding structures representing the same configuration. In LAT, since there is exactly one link per node, the number of links in different decoding structures is the same, hence comparable. In LAG, however, one node is allowed to link to multiple antecedents, creating a potential problem for decoding. For example, consider the sequence $m_1 \xrightarrow{\text{after}} m_2 \xrightarrow{\text{after}} m_3$, both of the following structures are correct:

1. $\langle m_1, m_2, after \rangle$, $\langle m_2, m_3, after \rangle$
2. $\langle m_1, m_2, after \rangle$, $\langle m_2, m_3, after \rangle$, $\langle m_1, m_3, after \rangle$

However, the last relation in the second decoding structure can actually be inferred via transitivity. We do not intend to spend the modeling power on such cases. We empirically avoid such redundant cases by using the **transitive reduction graph** for each structure. For a directed acyclic graph, a transitive reduction graph contains the fewest possible edges that have the same reachability relation as the original graph. In the example above, structure 1 is a transitive reduction graph for structure 2. We call the decoding structures that corresponding to the reduction graphs as *minimum decoding structures*. For LAG, we further restrict $\mathcal{Z}(\mathcal{A})$ to contain only minimum decoding structures.

Training Details in Latent Antecedent Graph

In this section, we describe the decoding details for LAG. Note that if we enforce a single antecedent for each node (as in our coreference model), it falls back to the LAT model [11].

Decoding: We use a greedy **best-first decoder** [110], which makes a left-to-right pass over the mentions. The decoding step is the same for line 5 and 7. The only difference is that we will use gold antecedent set ($\tilde{\mathcal{A}}$) at line 7. For each node m_j, we keep all links that score higher than the root link $\langle 0, m_j, r \rangle$.

Cycle and Structure Check: Incremental decoding a DAG may introduce cycles to the graph, or violate the minimum decoding structure criterion. To solve this, we maintain a set $R(m_i)$ that is reachable from m_i during the decoding process. We reject a new link ($\langle m_j, m_i \rangle$ if $m_j \in R(m_i)$) to avoid cycles. We also reject a redundant link ($\langle m_i, m_j \rangle$ if $m_j \in R(m_i)$) to keep a minimum decoding structure. Our current implementation is greedy, we leave investigations of search or global inference based algorithms to future work.

Selecting the Latent Event Mention Graph: Note that sequence relations are on the event level. Given a unique event graph, it may still correspond to multiple mention graphs. In our implementation, we use a minimum set of event mentions to represent the full event graph by taking one single mention from each event. Following the "easiest" intuition, we select the single mention that will result in the highest score given the current feature weight w.

Match Criteria: We consider two graphs to match when their inferred graphs are the same. The inferred graph is defined by taking the transitive closure of the graph and propagate the links through the coreference relations. For example, in Figure 4.1, the mention `fired` will be linked to two `killed` mentions after propagation.

Feature Delta: In structural perceptron training [35], the weights are updated directly by the feature delta. For all the features $\tilde{f}$ of the gold standard graph $\tilde{z}$ and features $\hat{f}$ of a decoded graph $\hat{z}$, the feature delta is simply: $\Delta = \tilde{f} - \hat{f}$. However, a decoded graph may contain links that are not directly presented but inferable from the gold standard graph. For example, in Figure 4.2, the prediction graph has a link from $M5$ to $M1$ (the orange arc), which is absent but inferable from the gold standard tree. If we keep these links when computing Δ, the model does not converge well. We thus remove the features on the inferable links from $\hat{f}$ when computing Δ.

Loss: We define the loss to be the number of different edges in two graphs. Following Björkelund and Kuhn [11], we further penalize erroneous root attachment: an incorrect link to the root m_0 adds the loss by 2. For example, in Figure 4.2 the prediction graph (bottom right) incorrectly links m_4 to Root and misses a link to m_3, which cause a total loss of 3. In addition, to be consistent with the feature delta computation, we do not compute loss for predicted links that are inferable from the gold standard.

4.3.2 Features

Event Coreference Features

For event coreference, we design a simple feature set to capture syntactic and semantic similarity of arcs. The main features are summarized in Table 4.1. In the TAC KBP 2015 coreference task setting, the event mentions are annotated with two attributes. There are 38 event types and subtype

Head	Headword token and lemma pair, and whether they are the same.
Type	The pair of event types, and whether they are the same.
Realis	The pair of realis types and whether they are the same.
POS	POS pair of the two mentions and whether they are the same.
Exact Match	Whether the 5-word windows of the two mentions match exactly.
Distance	Sentence distance between the two mentions.
Frame	Frame name pair of the two mentions and whether they are the same.
Syntactic	Whether a mention is the syntactic ancestor of another.

Table 4.1: Coreference Features. Parsing is done using Stanford CoreNLP [98]; frame names are produced by Semafor [42].

pairs (e.g., *Business.Merge-Org, Conflict.Attack*). There also 3 realis type: events that actually occurred are marked as *Actual*; events that are not specific are marked as *Generic*; other events such as future events are marked as *Other*. For these two attributes, we use the gold annotations in our feature sets.

Event Sequencing Features

An event sequencing system needs to determine whether the events are in the same script and order them. We design separate feature sets to capture these aspects: the Script Compatibility set considers whether mentions should belong to the same script; the Event Ordering set determines the relative ordering of the mentions. Our final features are the cross products of features from the following 3 sets.

1. **Surface-Based Script Compatibility**: these features capture whether two mentions are script compatible based on the surface information, including:
    - Mention headword pair.
    - Event type pair.
    - Whether two event mentions appear in the same cluster in Chambers's event schema database [20].
    - Whether the two event mentions share arguments, and the semantic frame name of the shared argument (produced by the Semafor parser [42]).
2. **Discourse-Based Script Compatibility**: these features capture whether two event mentions are related given the discourse context.
    - Dependency path between the two mentions.
    - Function words (words other than Noun, Verb, Adjective and Adverb) in between the

two mentions.

- The types of other event mentions between the two mentions.
- The sentence distance of two event mentions.
- Whether there are temporal expressions in the sentences of the two mentions, extracted from the AGM-TMP slot using a PropBank parser [150]

3. **Event Ordering**: this feature set tries to capture the ordering of events. We use the discourse ordering of two mentions (forward: the antecedent is the parent; backward: the antecedent is the child), and temporal ordering produced by CAEVO [17].

Taking the *after* arc from `fired` to `killed` in Figure 4.1 as an example, a feature after the cross product is: Event type pair is *Conflict.Attack* and *Life.Die*, discourse ordering is *backward*, and sentence distance is 0.

4.4 Experiments

4.4.1 Dataset

We conduct experiments on the dataset released in Text Analysis Coreference (TAC-KBP) 2017 Event Sequencing task (released by LDC under the catalog name LDC2016E130). This dataset contains rich event relation annotations, with event mentions and coreference annotated in TAC-KBP 2015, and additional annotations on Event Sequencing[4]. There are 158 documents in the training set and 202 in the test set, selected from general news articles and forum discussion threads. The event mentions are annotated with 38 type-subtype and 3 realis status (Actual, Generic, Other). Event Hopper, After, and Subevent links are annotated between event mentions. For all experiments, we develop our system and conduct ablation studies using 5-fold cross-validation on the training set, and report performance on the test set.

4.4.2 Baselines and Benchmarks

Coreference: we compare our event coreference system against the top performing systems from TAC-KBP 2015 (LCC, UI-CCG, and LTI). In addition, we also compare the results against two official baselines [103]: the Singleton baseline that put each event mention in its own cluster and the Match baseline that creates clusters based on mention type and realis status match.
Sequencing: This work is an initial attempt to this problem, so there is currently no comparable prior work on the same task. We instead compare with a baseline using event temporal ordering systems. We use a state-of-the-art temporal system named Caevo [17]. To make a fair comparison, we feed the gold standard event mentions to the system along with mentions predicted by Caevo[5]. However, since the script-style After links are only connected between mentions in the same script, directly using the output of Caevo produces very low precision. Instead, we run a stronger baseline: we take the gold standard script clusters and then only ask Caevo to predict links within these clusters (Oracle Cluster + Temporal).

[4]http://cairo.lti.cs.cmu.edu/kbp/2016/after/
[5]We keep the mentions predicted by Caevo because its inference may be affected by these mentions.

4.4.3 Evaluation Metrics

Evaluating Event Coreference: We evaluate our results using the official scorer provided by TAC-KBP, which uses 4 coreference metrics: *BLANC* [132], *MUC* [30], B^3 [4] and *CEAF-E* [97]. Following the TAC KBP task, systems are ranked using the average of these 4 metrics.
Evaluating Event Sequencing: The TAC KBP scorer evaluates event sequencing using the metric of the TempEval task [151, 153]. The TempEval metric calculates special precision and recall values based on the closure and reduction graphs:

$$Precision = \frac{|Response^{-} \cap Reference^{+}|}{|Response^{-}|} \quad Recall = \frac{|Reference^{-} \cap Response^{+}|}{|Reference^{-}|}$$

where $Response$ represents the After link graph from the system response and $Reference$ represents the After link graph from the gold standard. G^{+} represents the graph closure for graph G and G^{-} represents the graph reduction for graph G. As preprocessing, relations are automatically propagated through coreference clusters (currently using gold standard clusters). The final score is the standard F-score: geometric mean of the precision and recall values.

4.4.4 Evaluation Results for Event Coreference

The test performance on Event Coreference is summarized in Table 4.2. Comparing to the top 3 coreference systems in TAC-KBP 2015, we outperform the best system by about 2 points absolute F-score on average. Our system is also competitive on individual metrics. Our model performs the best based on B^3 and CEAF-E, and is comparable to the top performing systems on MUC and BLANC.

Note that while the `Matching` baseline only links event mentions based on event type and realis status, it is very competitive and performs close to the top systems. This is not surprising since these two attributes are based on the gold standard. To take a closer look, we conduct an ablation study by removing the simple match features one by one. The results are summarized in Table 4.3. We observe that some features produce mixed results on different metrics: they provide improvements on some metrics but not all. This is partially caused by the different characteristics of different metrics. On the other hand, these features (parsing and frames) are automatically predicted, which make them less stable. Furthermore, the Frame features contain duplicate information to event types, which makes it less useful in this setting.

Besides the presented features, we have also designed features using event argument. However, we do not report the results since the argument features decrease the performance on all metrics.

4.4.5 Evaluation Results for Event Sequencing

The evaluation results on Event Sequencing is summarized in Table 4.4. Because the baseline system has access to the oracle script clusters, it produces high precision. However, the low recall value shows that it fails to produce enough After links. Our analysis shows that a lot of After relations are not indicated by clear temporal clues, but can only be solved with script knowledge. In Example 4.3, the baseline system is able to identify "fled" is after "ousted" from explicit marker

	B^3	CEAF-E	MUC	BLANC	AVG.
`Singleton`	78.10	68.98	0.00	48.88	52.01
`Matching`	78.40	65.82	**69.83**	76.29	71.94
LCC	82.85	74.66	68.50	**77.61**	75.69
UI-CCG	83.75	75.81	63.78	73.99	74.28
LTI	82.27	75.15	60.93	71.57	72.60
This work	**85.59**	**79.65**	67.81	77.37	**77.61**

Table 4.2: Test Results for Event Coreference with the `Singleton` and `Matching` baselines.

	B^3	CEAF-E	MUC	BLANC	AVG.
ALL	81.97	74.80	76.33	76.07	77.29
-Distance	81.92	74.48	76.02	77.55	77.50
-Frame	82.14	75.01	76.28	77.74	77.79
-Syntactic	81.87	74.89	75.79	76.22	77.19

Table 4.3: Ablation study for Event Coreference.

"after". However, it fails to identify that "extradited" is after "arrested", which requires knowledge about prototypical event sequences.

(4.3) Eight months after the [transport fled] Ivory Coast when Gbagbo, the former president, was [End.Position ousted] by the French military. Blé Goudé was subsequently [Jail arrested] in Ghana and [transport extradited] Megrahi,[Jail jailed] for [Attack killing] 270 people in 1988.[6]

In our error analysis, we noticed that our system produces a large number of relations due to coreference propagation. One single wrong prediction can cause the error to propagate.

Besides memorizing the mention pairs, our model also tries to capture script compatibility through discourse signals. To further understand how much these signals help, we conduct an ablation study of the features in the discoursed based compatibility features (see §4.3.2). Similarly, we remove each feature group from the full feature set one by one and observe the performance change.

The results are reported in Table 4.5. While most of the features only affect the performance by less than 1 absolute F1 score, the feature sets after removing *mention* or *sentences* show a significant drop in both precision and recall. This shows that discourse proximity is the most significant ones among these features. In addition, the *mention* feature set captures the following *explain away* intuition: the event mentions A and B are less likely to be related if there are similar mentions in between. One such example can be seen in Figure 4.1, the event mention `fired` is

[6]The small red text indicates the event type for each mention.

	Prec.	Recall	F-Score
Oracle Cluster+Temporal	**46.21**	8.72	14.68
Our Model	18.28	**16.91**	**17.57**

Table 4.4: Test results for event sequencing. The Oracle Cluster+Temporal system is running CAEVO on the Oracle Clusters.

	Prec.	Recall	F-Score	Δ
Full	37.92	36.79	36.36	
- Mention Type	32.78	29.81	30.07	6.29
- Sentence	33.90	30.75	31.00	5.36
- Temporal	37.21	36.53	35.81	0.55
- Dependency	38.18	36.44	36.23	0.13
- Function words	38.08	36.51	36.18	0.18

Table 4.5: Ablation Study for Event Sequencing.

more likely to relate to the closest `killed`, instead of the other `killed` in the first paragraph.

In addition, our performance on the development set is higher than the test set. Further analysis reveals two causes: 1. the coreference propagation step causes the scores to be very unstable, 2. our model only learns limited common sense ordering based on lexical pairs, which can easily overfit to the small training corpus. Since the annotation is difficult to scale, it is important to use methods to harvest script common sense knowledge automatically, as in the script induction work [18].

4.5 Discussion

4.5.1 Event Coreference Challenges

Although we have achieved good performance on event coreference, upon closer investigation we found that most of the coreference decisions are still made based on simple word/lemma matching (note that the type and realis baseline is as high as 0.72 F1 score). The system exploits little semantic information to resolve difficult event coreference problems. A major challenge is that our system is not capable of utilizing event arguments: in fact, Hasler and Orasan [71] found that only around 20% of the arguments in the same event slot are actually coreferent for coreferential event pairs in the ACE 2005 corpus. Furthermore, the TAC-KBP corpus uses a relaxed participant identity requirement for event coreference, which makes argument-based matching more difficult.

4.5.2 Event Sequencing Challenges

Our event sequencing performance is still low despite the introduction of many features. This task is inherently difficult because it requires a system to solve both the script clustering and event ordering tasks. The former task requires both common-sense knowledge and discourse reasoning. Reasoning is more important for long-term links since there are no explicit clues like prepositions and dependencies to be exploited. The ablation study shows that discourse features like sentence distance are more effective, which indicates that our model mainly relies on surface clues and has limited reasoning power.

Furthermore, we observe a strong locality property of After links by skimming the training data: most After link relations are found in a small local region. Since reasoning and coreference based propagation will accumulate local decisions, a system must be accurate on them.

The Ambiguous Boundary of a Script

Besides the above-mentioned challenges, a more fundamental problem is to define the boundary of scripts. Since the definition of scripts is only prototypical event sequences, the boundaries between them are not clear. In Example 4.3, the event `jailed` is considered to belong to a "Judicial Process" script and `killing` is considered to belong to an"Attack" script[7]. No link is annotated between these two mentions since they are considered to belong to different clusters, even though the "jailed" event is to punish the "killing". Therefore essentially, the current Event Sequencing task simply requires the system to fit these human defined boundaries. In principle, the "Judicial Process" script and the "Attack" script can form a larger script structure, on a higher hierarchical level.

While it is possible to manually define scripts and what kind of events they may contain specifically in a controlled domain, it is difficult to generalize the relations. Most previous work on script induction [18, 29, 57, 123, 139] treats scripts as statistical models where probabilities can be assigned, thereby avoiding the boundary problem. While the script boundaries may be application dependent, a possible solution may rely on the "Goals" in Schank's script theory. The Goal of a script is the final state expected (by the script protagonist) from the sequence of events. Goal oriented scripts may be able to help us explain whether `killing` and `jailed` should be separate: if we take the"killer" as the protagonist, the goal of "kill" is achieved at the point of the victim dying. We leave the investigation on proper theoretical justification to future work.

4.6 Conclusion

In this chapter, we present a unified graph framework to conduct event coreference and sequencing. We have achieved state-of-the-art results on event coreference and report the first attempt at event sequencing. While we only studied two types of relations, we believe the method can be adopted in broader contexts.

[7]Script names are taken from the annotation guideline: `http://cairo.lti.cs.cmu.edu/kbp/2016/after/annotation`

In general, analyzing event structure can bring new aspects of knowledge from text. For instance, Event Coreference systems can help group scattered information together. Understanding Event Sequencing can help clarify the discourse structure, which can be useful in other NLP applications, such as solving entity coreference problems [119]. However, in our investigation, we find that the linguistic theory and definitions for events are not adequate for the computational setting. For example, proper theoretical justification is needed to define event coreference, which should explain the problems, such as argument mismatches. In addition, we also need a theoretical basis for script boundaries. In the future, we will devote our effort to understanding the theoretical and computational aspects of events relations, and utilizing them for other NLP tasks.

Chapter 5

Verb Phrase Ellipsis

5.1 Introduction

Verb Phrase Ellipsis (VPE) is the anaphoric process where a verbal constituent is partially or totally unexpressed, but can be resolved through an antecedent in the context, as in the following examples:

(5.1) His wife also [antecedent *works for the paper*], as **did** his father.

(5.2) In particular, Mr. Coxon says, businesses are [antecedent *paying out a smaller percentage of their profits and cash flow in the form of dividends*] than they **have** historically.

In example 5.1, a light verb **did** is used to represent the verb phrase *works for the paper*; example 5.2 shows a much longer antecedent phrase, which in addition differs in tense from the elided one. Following Dalrymple et al. [39], we refer to the full verb expression as the "antecedent", and to the anaphor as the "target".

VPE resolution is necessary for deeper Natural Language Understanding, and can be beneficial for instance in dialogue systems or Information Extraction applications. Conceptually, VPE can be considered as an special event coreference task, where the antecedent event mention is generally a light verb phrase.

Computationally, VPE resolution can be modeled as a pipeline process: first detect the VPE targets, then identify their antecedents. Prior work on this topic [68, 115] has used this pipeline approach but without analysis of the interaction of the different steps.

In this chapter, we analyze the steps needed to resolve VPE. We preserve the target identification task, but propose a decomposition of the antecedent selection step in two subtasks. We use learning-based models to address each task separately, and also explore the combination of contiguous steps. Although the features used in our system are relatively simple, our models yield state-of-the-art results on the overall task. We also observe a small performance improvement from our decomposition modeling of the tasks.

There are only a few small datasets that include manual VPE annotations. While Bos and Spenader [12] provide publicly available VPE annotations for Wall Street Journal (WSJ) news documents, the annotations created by Nielsen [115] include a more diverse set of genres (e.g., articles and plays) from the British National Corpus (BNC).

We semi-automatically transform these latter annotations into the same format used by the

former. The unified format allows better benchmarking and will facilitate more meaningful comparisons in the future. We evaluate our methods on both datasets, making our results directly comparable to those published by Nielsen [115].

5.2 Related Work

Considerable work has been done on VPE in the field of theoretical linguistics: e.g., [39, 146]; yet there is much less work on computational approaches to resolving VPE.

Hardt [1992, 1997] presents, to our knowledge, the first computational approach to VPE. His system applies a set of linguistically motivated rules to select an antecedent given an elliptical target. Hardt [70] uses Transformation-Based Learning to replace the manually developed rules. However, in Hardt's work, the targets are selected from the corpus by searching for "empty verb phrases" (constructions with an auxiliary verb only) in the gold standard parse trees.

Nielsen [2005] presents the first end-to-end system that resolves VPE from raw text input. He describes several heuristic and learning-based approaches for target detection and antecedent identification. He also discusses a post-processing substitution step in which the target is replaced by a transformed version of the antecedent (to match the context). We do not address this task here because other VPE datasets do not contain relevant substitution annotations. Similar techniques are also described in Nielsen [2003, 2004, 2004].

Results from this prior work are relatively difficult to reproduce because the annotations on which they rely are inaccessible. The annotations used by Hardt [69] have not been made available, and those used by Nielsen [115] are not easily reusable since they rely on some particular tokenization and parser. Bos and Spenader [12] address this problem by annotating a new corpus of VPE on top of the WSJ section of the Penn Treebank, and propose it as a standard evaluation benchmark for the task. Still it is desirable to use Nielsen's annotations on the BNC which contain more diverse text genres with more frequent VPE.

5.3 Approaches

We focus on the problems of target detection and antecedent identification as proposed by Nielsen [115]. We propose a refinement of these two tasks, splitting them into these three:

1. **Target Detection (T)**, where the subset of VPE targets is identified.
2. **Antecedent Head Resolution (H)**, where each target is linked to the head of its antecedent.
3. **Antecedent Boundary Determination (B)**, where the exact boundaries of the antecedent are determined from its head.

The following sections describe each of the steps in detail.

5.3.1 Target Detection

Since the VPE target is annotated as a single word in the corpus[1], we model their detection as a binary classification problem. We only consider modal or light verbs (*be*, *do*, *have*) as candidates, and train a logistic regression classifier ($\mathbf{Log}^T$) with the following set of binary features:

1. The POS tag, lemma, and dependency label of the verb, its dependency parent, and the immediately preceding and succeeding words.
2. The POS tags, lemmas and dependency labels of the words in the dependency subtree of the verb, in the 3-word window, and in the same-size window after (as bags of words).
3. Whether the subject of the verb appears to its right (i.e., there is subject-verb inversion).

5.3.2 Antecedent Head Resolution

For each detected target, we consider as potential antecedent heads all verbs (including modals and auxiliaries) in the three immediately preceding sentences of the target word[2] as well as the sentence including the target word (up to the target[3]). This follows Hardt [68] and Nielsen [115].

We perform experiments using a logistic regression classifier ($\mathbf{Log}^H$), trained to distinguish correct antecedents from all other possible candidates. The set of features are shared with the Antecedent Boundary Determination task, and are described in detail in Section 5.3.3.

However, a more natural view of the resolution task is that of a ranking problem. The gold annotation can be seen as a partial ordering of the candidates, where, for a given target, the correct antecedent ranks above all other candidates, but there is no ordering among the remaining candidates. To handle this specific setting, we adopt a ranking model with domination loss [44].

Formally, for each potential target t in the determined set of targets T, we consider its set of candidates C_t, and denote whether a candidate $c \in C_t$ is the antecedent for t using a binary variable a_{ct}. We express the ranking problem as a bipartite graph $\mathcal{G} = (V^+, V^-, E)$ where vertices represent antecedent candidates:

$$\begin{aligned} V^+ &= \{(t,c) \mid t \in T, c \in C_t, a_{ct} = 1\} \\ V^- &= \{(t,c) \mid t \in T, c \in C_t, a_{ct} = 0\} \end{aligned}$$

and the edges link the correct antecedents to the rest of the candidates for the same target[4]:

$$E = \{((t,c^+),(t,c^-)) \mid (t,c^+) \in V^+, (t,c^-) \in V^-\}$$

We associate each vertex i with a feature vector $\mathbf{x}_i$, and compute its score s_i as a parametric function of the features $s_i = g(\mathbf{w}, \mathbf{x}_i)$. The training objective is to learn parameters $\mathbf{w}$ such that each positive vertex $i \in V^+$ has a higher score than the negative vertices j it is connected to, $V_i^- = \{j \mid j \in V^-, (i,j) \in E\}$.

[1] All targets in the corpus of Bos and Spenader [12] are single-word by their annotation guideline.
[2] Only 1 of the targets in the corpus of Bos and Spenader [12], has an antecedent beyond that window.
[3] Only 1% of the targets in the corpus are cataphoric.
[4] During training, there is always 1 correct antecedent for each gold standard target, with several incorrect ones.

Algorithm 3 Candidate generation

Input: a, the antecedent head
Input: t, the target
Output: B, the set of possible antecedent boundaries (begin, end)

$a_s \longleftarrow \text{SemanticHeadVerb}(a)$
$E \longleftarrow \{a_s\}$ *// the set of ending positions*
for $ch \in \text{RightChildren}(a_s)$ **do**
  $e \leftarrow \text{RightMostNode}(ch)$
  if $e < t \wedge \text{ValidEnding}(e)$ **then**
    $E \longleftarrow E \cup \{e\}$
$B \longleftarrow \emptyset$
for $e \in E$ **do**
  $B \longleftarrow B \cup \{(a, e)\}$
  if $a ==$ "be" **then**
    if $\text{IsVerb}(a+1)$ **then**
      $A \longleftarrow A \cup \{(a+1, e)\}$
    for $s \in \{a+1, a+2 \ldots e-1\}$ **do**
      if $\text{IsAdverb}(s) \wedge \text{IsVerb}(s+1)$ **then**
        $B \longleftarrow B \cup \{(s+1, e)\}$
return B

The combinatorial domination loss for a vertex $i \in V^+$ is 1 if there exists any vertex $j \in V_i^-$ with a higher score. A convex relaxation of the loss for the graph is given by [44]:

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log(1 + \sum_{j \in V_i^-} \exp(s_j - s_i + \Delta))$$

Taking $\Delta = 0$, and choosing g to be a linear feature scoring function $s_i = \mathbf{w} \cdot \mathbf{x_i}$, the loss becomes:

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log \sum_{j \in V_i^-} \exp(\mathbf{w} \cdot \mathbf{x}_j) - \mathbf{w} \cdot x_i$$

The loss over the whole graph can then be minimized using stochastic gradient descent. We will denote the ranker learned with this approach as **Rank**^{H}.

5.3.3 Antecedent Boundary Determination

From a given antecedent head, the set of potential boundaries for the antecedent, which is a complete or partial verb phrase, is constructed using Algorithm 3.

Informally, the algorithm tries to generate different valid verb phrase structures by varying the amount of information encoded in the phrase. To do so, it accesses the semantic head verb a_s of

the antecedent head a (e.g., *paying* for *are* in Example 5.2), and considers the rightmost node of each right child. If the node is a valid ending (punctuation and quotation are excluded), it is added to the potential set of endings E. The set of valid boundaries B contains the cross-product of the starting position $S = \{a\}$ with E.

For instance, from Example 5.2, the following boundary candidates are generated for *are*:

- are paying
- are paying out
- are paying out a smaller percentage of their profits and cash flow
- are paying out a smaller percentage of their profits and cash flow in the form of dividends

We experiment with both logistic regression (**Log**^{B}) and ranking (**Rank**^{B}) models for this task. The set of features is shared with the previous task, and is described in the following section.

Antecedent Features

The features used for antecedent head resolution and/or boundary determination try to capture aspects of both tasks. We summarize the features in Table 5.1. The features are roughly grouped by their type. **Labels** features make use of the parsing labels of the antecedent and target; **Tree** features are intended to capture the dependency relations between the antecedent and target; **Distance** features describe distance between them; **Match** features test whether the context of the antecedent and target are similar; **Semantic** features capture shallow semantic similarity; finally, there are a few **Other** features which are not categorized.

On the last column of the feature table, we indicate the design purpose of the feature: head selection (H), boundary detection (B) or both (B&H). However, we use the full feature set for all three tasks.

5.4 Joint Modeling

Here we consider the possibility that antecedent head resolution and target detection should be modeled jointly (they are typically separate). The hypothesis is that if a suitable antecedent for a target cannot be found, the target itself might have been incorrectly detected. Similarly, the suitability of a candidate as antecedent head can depend on the possible boundaries of the antecedents that can be generated from it.

We also consider the possibility that antecedent head resolution and antecedent boundary determination should be modeled independently (though they are typically combined). We hypothesize that these two steps actually focus on different perspectives: the antecedent head resolution (**H**) focuses on finding the correct antecedent position; the boundary detection step (**B**) focuses on constructing a well-formed verb phrase. We are also aware that **B** might be helpful to **H**, for instance, a correct antecedent boundary will give us correct context words, that can be useful in determining the antecedent position.

We examine the joint interactions by combining adjacent steps in our pipeline. For the combination of antecedent head resolution and antecedent boundary determination (**H+B**), we consider simultaneously as candidates for each target the set of all potential boundaries for all potential

Type	Feature Description	Purpose
Labels	The POS tag and dependency label of the antecedent head	H
	The POS tag and dependency label of the antecedent's last word	B
	The POS tag and lemma of the antecedent parent	H
	The POS tag, lemma and dependency label of within a 3 word around around the antecedent	B
	The pair of the POS tags of the antecedent head and the target, and of their auxiliary verbs	H
	The pair of the lemmas of the auxiliary verbs of the antecedent head and the target	H
Tree	Whether the antecedent and the target form a comparative construction connecting by *so*, *as* or *than*	H&B
	The dependency labels of the shared lemmas between the parse tree of the antecedent and the target	H
	Label of the dependency between the antecedent and target (if exists)	H
	Whether the antecedent contains any descendant with the same lemma and dependency label as a descendant of the target.	H
	Whether antecedent and target are dependent ancestor of each other	H
	Whether antecedent and target share prepositions in their dependency tree	H
Distance	The distance in sentences between the antecedent and the target (clipped to 2)	H
	The number of verb phrases between the antecedent and the target (clipped to 5)	H
Match	Whether the lemmas of the heads, and words in the the window (=2) before the antecedent and the target match respectively	H
	Whether the lemmas of the ith word before the antecedent and $i-1$th word before the target match respectively (for $i \in \{1,2,3\}$, with the 0th word of the target being the target itself)	H&B
Semantic	Whether the subject of the antecedent and the target are coreferent	H
Other	Whether the lemma of the head of the antecedent is *be* and that of the target is *do* (be-do match, used by Hardt and Nielsen)	H
	Whether the antecedent is in quotes and the target is not, or vice versa	H&B

Table 5.1: Antecedent Features

heads. Here too, a logistic regression model (**Log**$^{H+B}$) can be used to distinguish correct `(target, antecedent start, antecedent end)` triplets; or a ranking model (**Rank**$^{H+B}$)

	Documents		VPE Instances	
	Train	Test	Train	Test
WSJ	1999	500	435	119
BNC	12	2	641	204

Table 5.2: Corpus statistics

can be trained to rank the correct one above the other ones for the same target.

The combination of target detection with antecedent head resolution (**T+H**) requires identifying the targets. This is not straightforward when using a ranking model since scores are only comparable for the same target. To get around this problem, we add a "null" antecedent head. For a given target candidate, the null antecedent should be ranked higher than all other candidates if it is not actually a target. Since this produces many examples where the null antecedent should be selected, random subsampling is used to reduce the training data imbalance. The "null" hypothesis approach is used previously in ranking-based coreference systems [49, 131].

Most of the features presented in the previous section will not trigger for the null instance, and an additional feature to mark this case is added.

The combination of the three tasks (**T+H+B**) only differs from the previous case in that all antecedent boundaries are considered as candidates for a target, in addition to the potential antecedent heads.

5.5 Experiments

5.5.1 Datasets

We conduct our experiments on two datasets (see Table 5.2 for corpus counts). The first one is the corpus of Bos and Spenader [12], which provides VPE annotation on the WSJ section of the Penn Treebank. Bos and Spenader [12] propose a train-test split that we follow[5].

To facilitate more meaningful comparison, we converted the sections of the British National Corpus annotated by Nielsen [115] into the format used by Bos and Spenader [12], and manually fixed conversion errors introduced during the process[6] (Our version of the dataset is publicly available for research[7].) We use a train-test split similar to Nielsen [115][8].

[5]Section 20 to 24 are used as test data.

[6]We also found 3 annotation instances that could be deemed errors, but decided to preserve the annotations as they were.

[7]`https://github.com/hunterhector/VerbPhraseEllipsis`

[8]Training set is CS6, A2U, J25, FU6, H7F, HA3, A19, A0P, G1A, EWC, FNS, C8T; test set is EDJ, FR3

5.5.2 Evaluation

We evaluate and compare our models following the metrics used by Bos and Spenader [12].

VPE target detection is a per-word binary classification problem, which can be evaluated using the conventional precision (Prec), recall (Rec) and F1 scores.

Bos and Spenader [12] propose a token-based evaluation metric for antecedent selection. The antecedent scores are computed over the correctly identified tokens per antecedent: precision is the number of correctly identified tokens divided by the number of predicted tokens, and recall is the number of correctly identified tokens divided by the number of gold standard tokens. Averaged scores refer to a "macro"-average over all antecedents.

Finally, in order to asses the performance of antecedent head resolution, we compute precision, recall and F1 where credit is given if the proposed head is included inside the golden antecedent boundaries.

5.5.3 Baselines and Benchmarks

We begin with simple, linguistically motivated baseline approaches for the three subtasks. For target detection, we re-implement the heuristic baseline used by Nielsen [115]: take all auxiliaries as possible candidates and eliminate them using part-of-speech context rules (we refer to this as $\mathbf{Pos}^T$). For antecedent head resolution, we take the first non-auxiliary verb preceding the target verb. For antecedent boundary detection, we expand the verb into a phrase by taking the largest subtree of the verb such that it does not overlap with the target. These two baselines are also used in Nielsen [115] (and we refer to them as $\mathbf{Prev}^H$ and $\mathbf{Max}^B$, respectively).

To upper-bound our results, we include an oracle for the three subtasks, which selects the highest scoring candidate among all those considered. We denote these as $\mathbf{Ora}^T$, $\mathbf{Ora}^H$, $\mathbf{Ora}^B$.

We also compare to the current state-of-the-art target detection results as reported in Nielsen [115] on the BNC dataset ($\mathbf{Nielsen}^T$)[9].

5.6 Results

The results for each one of the three subtasks in isolation are presented first, followed by those of the end-to-end evaluation. We have not attempted to tune classification thresholds to maximize F1.

5.6.1 Target Detection

Table 5.3 shows the performance of the compared approaches on the Target Detection task. The logistic regression model $\mathbf{Log}^T$ gives relatively high precision compared to recall, probably because there are so many more negative training examples than positive ones. Despite a simple set of features, the F1 results are significantly better than Nielsen's baseline $\mathbf{Pos}^T$.

[9]The differences in the setup make the results on antecedent resolution not directly comparable.

Notice also how the oracle **Ora**^{T} does not achieve 100% recall, since not all the targets in the gold data are captured by our candidate generation strategy. The loss is around 7% for both corpora.

The results obtained by the joint models are low on this task. In particular, the ranking models **Rank**$^{T+H}$ and **Rank**$^{T+H+B}$ fail to predict any target in the WSJ corpus, since the null antecedent is always preferred. This happens because joint modeling further exaggerates the class imbalance: the ranker is asked to consider many incorrect targets coupled with all sorts of hypothesis antecedents, and ultimately learns just to select the null target. Our initial attempts at subsampling the negative examples did not improve the situation. The logistic regression models **Log**$^{T+H}$ and **Log**$^{T+H+B}$ are most robust, but still their performance is far below that of the pure classifier **Log**^{T}.

5.6.2 Antecedent Head Resolution

Table 5.4 contains the performance of the compared approaches on the Antecedent Head Resolution task, assuming oracle targets (**Ora**^{T}).

First, we observe that even the oracle **Ora**^{H} has low scores on the BNC corpus. This suggests that some phenomena beyond the scope of those observed in the WSJ data appear in the more general corpus (we developed our system using the WSJ annotations and then simply evaluated on the BNC test data).

Second, the ranking-based model **Rank**^{H} consistently outperforms the logistic regression model **Log**^{H} and the baseline **Prev**^{H}. The ranking model's advantage is small in the WSJ, but much more pronounced in the BNC data. These improvements suggest that indeed, ranking is a more natural modeling choice than classification for antecedent head resolution.

Finally, the joint resolution models **Rank**$^{H+B}$ and **Log**$^{H+B}$ give poorer results than their single-task counterparts, though **Rank**$^{H+B}$ is not far behind **Rank**^{H}. Joint modeling requires more training data and we may not have enough to reflect the benefit of a more powerful model.

5.6.3 Antecedent Boundary Determination

Table 5.5 shows the performance of the compared approaches on the Antecedent Boundary Determination task, using the soft evaluation scores (the results for the strict scores are omitted for brevity, but in general look quite similar). The systems use the output of the oracle targets (**Ora**^{T}) and antecedent heads (**Ora**^{H}).

Regarding boundary detection alone, the logistic regression model **Log**^{B} outperforms the ranking model **Rank**^{B}. This suggests that boundary determination is more a problem of determining the compatibility between target and antecedent extent than one of ranking alternative boundaries. However, the next experiments suggest this advantage is diminished when gold targets and antecedent heads are replaced by system predictions.

Non-Gold Antecedent Heads

Table 5.6 contains Antecedent Boundary Determination results for systems which use oracle targets, but system antecedent heads. When **Rank**^{H} or **Log**^{H} are used for head resolution, the

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{T}	100.00	93.28	96.52	100.00	92.65	96.18
Log^{T}	80.22	61.34	69.52	80.90	70.59	75.39
Pos^{T}	42.62	43.7	43.15	35.47	35.29	35.38
Log^{T+H}	23.36	26.89	25.00	12.52	38.24	18.86
Rank^{T+H}	0.00	0.00	0.00	15.79	5.88	8.57
Log^{T+H+B}	25.61	17.65	20.90	21.50	32.35	25.83
Rank^{T+H+B}	0.00	0.00	0.00	16.67	11.27	13.45
Nielsen^{T}	—	—	—	72.50	72.86	72.68

Table 5.3: Results for Target Detection

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}	94.59	88.24	91.30	79.89	74.02	76.84
Rank^{H}	70.27	65.55	67.83	52.91	49.02	50.89
Prev^{H}	67.57	63.03	65.22	39.68	36.76	38.17
Log^{H}	59.46	55.46	57.39	38.62	35.78	37.15
Rank^{H+B}	68.47	63.87	66.09	51.85	48.04	49.87
Log^{H+B}	39.64	36.97	38.26	30.16	27.94	29.01

Table 5.4: Results for Antecedent Head Resolution

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^{B}	89.47	83.46	86.36	81.10	75.13	78.00
Rank^{B}	83.96	78.32	81.04	75.68	70.12	72.79
Max^{B}	78.97	73.66	76.22	73.70	68.28	70.88

Table 5.5: Soft results for Antecedent Boundary Determination

difference between **Log**^{B} and **Rank**^{B} diminishes, and it is even better to use the latter in the BNC corpus. The models were trained with gold annotations rather than system outputs, and the ranking model is somewhat more robust to noisier inputs.

On the other hand, the results for the joint resolution model **Rank**$^{H+B}$ are better in this case than the combination of **Rank**^{H}**+Rank**^{B}, whereas **Log**$^{H+B}$ performs worse than any 2-step combination. The benefits of using a ranking model for antecedent head resolution seem thus to outperform those of using classification to determine its boundaries.

5.6.4 End-to-End Evaluation

Table 5.7 contains the end-to-end performance of different approaches, using the soft evaluation scores.

The trends we observed with gold targets are preserved: approaches using the **Rank**^{H} maintain an advantage over **Log**^{H}, but the improvement of **Log**^{B} over **Rank**^{B} for boundary determination is diminished with non-gold heads. Also, the 3-step approaches seem to perform slightly better than the 2-step ones. Together with the fact that the smaller problems are easier to train, this appears to validate our decomposition choice.

5.7 Discussion and Conclusion

In this chapter we have explored a decomposition of Verb Phrase Ellipsis resolution into subtasks, which splits antecedent selection in two distinct steps. By modeling these two subtasks separately with two different learning paradigms, we can achieve better performance then doing them jointly, suggesting they are indeed of different underlying nature.

Our experiments show that a logistic regression classification model works better for target detection and antecedent boundary determination, while a ranking-based model is more suitable for selecting the antecedent head of a given target. However, the benefits of the classification model for boundary determination are reduced for non-gold targets and heads. On the other hand, by separating the two steps, we lose the potential joint interaction of them. It might be possible to explore whether we can bring the benefits of the two side: use separate models on each step, but learn them jointly. We leave further investigation of this to future work.

We have also explored jointly training a target detection and antecedent resolution model, but have not been successful in dealing with the class imbalance inherent to the problem.

Our current model adopts a simple feature set, which is composed mostly by simple syntax and lexical features. It may be interesting to explore more semantic and discourse-level features in our system. We leave these to future investigation.

All our experiments have been run on publicly available datasets, to which we add our manually aligned version of the VPE annotations on the BNC corpus. We hope our experiments, analysis, and more easily processed data can further the development of new computational approaches to the problem of Verb Phrase Ellipsis resolution.

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Rank^{H}**+Log**^{B}	64.11	59.8	61.88	47.04	43.58	45.24
Rank^{H}**+Rank**^{B}	63.90	59.6	61.67	49.11	45.5	47.24
Log^{H}**+Log**^{B}	53.49	49.89	51.63	34.77	32.21	33.44
Log^{H}**+Rank**^{B}	53.27	49.69	51.42	36.26	33.59	34.88
Rank^{H+B}	67.55	63.01	65.20	50.68	46.95	48.74
Log^{H+B}	40.96	38.20	39.53	30.00	27.79	28.85

Table 5.6: Soft results for Antecedent Boundary Determination with non-gold heads

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{T}**+Ora**^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^{T}**+Rank**^{H}**+Rank**^{B}	52.68	40.28	45.65	43.03	37.54	40.10
Log^{T}**+Rank**^{H}**+Log**^{B}	52.82	40.40	45.78	40.21	35.08	37.47
Log^{T}**+Log**^{H}**+Rank**^{B}	49.45	37.82	42.86	33.12	28.90	30.86
Log^{T}**+Log**^{H}**+Log**^{B}	49.41	37.79	42.83	31.32	27.33	29.19
Pos^{T}**+Prev**^{H}**+Max**^{B}	19.04	19.52	19.27	12.81	12.75	12.78
Log^{T}**+Rank**$^{H+B}$	54.82	41.92	47.51	41.86	36.52	39.01
Log^{T}**+Log**$^{H+B}$	38.85	29.71	33.67	26.11	22.78	24.33

Table 5.7: Soft end-to-end results

Part II

Learning with Indirect Supervision

Chapter 6

Event Salience

6.1 Introduction

Automatic extraction of prominent information from text has always been a core problem in language research. While traditional methods mostly concentrate on the word level, researchers start to analyze higher-level discourse units in text, such as entities [48] and events [33].

Events are important discourse units that form the backbone of our communication. They play various roles in documents. Some are more central in discourse: connecting other entities and events, or providing key information of a story. Others are less relevant, but not easily identifiable by NLP systems. Hence it is important to be able to quantify the "importance" of events. For example, Figure 6.1 is a news excerpt describing a debate around a jurisdiction process: "*trial*" is central as the main discussing topic, while "*war*" is not.

Researchers are aware of the need to identify central events in applications like detecting salient relations [161], and identifying climax in storyline [155]. Generally, the salience of discourse units is important for language understanding tasks, such as document analysis [9], information retrieval [159], and semantic role labeling [28]. Thus, proper models for finding important events are desired.

In this chapter, we study the task of **event salience detection**, to find events that are most relevant to the main content of documents. To build a salience detection model, one core observation is that salient discourse units are forming discourse relations. In Figure 6.1, the "*trial*" event is connected to many other events: "*charge*" is pressed before "*trial*"; "*trial*" is being "*delayed*".

We present two salience detection systems based on the observations. First is a feature based learning to rank model. Beyond basic features like frequency and discourse location, we design features using cosine similarities among events and entities, to estimate the *content organization* [64]: how lexical meaning of elements relates to each other. Similarities from within-sentence or across the whole document are used to capture interactions on both local and global aspects (§6.4). The model significantly outperforms a strong "Frequency" baseline in our experiments.

However, there are other discourse relations beyond lexical similarity. Figure 6.1 showcases

Federal prosecutors **urged** a **trial** **judge** today to **deny** defense **requests** to **delay** the **trial** of Zacarias Moussaoui and suggested that Mr. Moussaoui, the only person **charged** in the Sept. 11 **attacks**, was to **blame** for many of the **delays** so far. The **attacks** ''were volleys in a declared war against the United States and were more than just acts of terror,'' the prosecutors said in a filing to the Federal District Court in Alexandria, Va. ''Thus, the victims' and the nation's interest in a fair and speedy **trial** is beyond dispute.''.Last week, court-appointed defense lawyers asked that the starting date of the **trial**, now set for Sept. 30, be **delayed** by at least two months to allow them to wade through volumes of evidence that prosecutors have presented to them, including more than 1,300 computer discs.

Figure 6.1: Examples annotations. Underlying words are annotated event triggers; the red bold ones are annotated as salient.

some: the script relation [143][1] between "*charge*" and "*trial*", and the frame relation [6] between "*attacks*" and "*trial*" ("*attacks*" fills the "charges" role of "*trial*"). Since it is unclear which ones contribute more to salience, we design a Kernel based Centrality Estimation (KCE) model (§6.5) to capture salient specific interactions between discourse units automatically.

In KCE, discourse units are projected to embeddings, which are trained end-to-end towards the salience task to capture rich semantic information. A set of soft-count kernels are trained to weigh salient specific latent relations between discourse units. With the capacity to model richer relations, KCE outperforms the feature-based model by a large margin (§6.7.1). Our analysis shows that KCE is exploiting several relations between discourse units: including script and frames (Table 6.5). To further understand the nature of KCE, we conduct an *intrusion test* (§6.6.2), which requires a model to identify events from another document. The test shows salient events form tightly related groups with relations captured by KCE.

The notion of salience is subjective and may vary from person to person. We follow the empirical approaches used in entity salience research [48]. We consider the *summarization test*: an event is considered salient if a summary written by a human is likely to include it, since events about the main content are more likely to appear in a summary. This approach allows us to create a large-scale corpus (§6.3).

This chapter makes three main contributions. First, we present two event salience detection systems, which capture rich relations among discourse units. Second, we observe interesting connections between salience and various discourse relations (§6.7.1 and Table 6.5), implying potential research on these areas. Finally, we construct a large scale event salience corpus,

[1]Scripts are prototypical sequences of events: a *restaurant* script normally contains events like "order", "eat" and "pay".

providing a testbed for future research. Our code, dataset and models are publicly available[2].

6.2 Related Work

Events have been studied on many aspects due to their importance in language. To name a few: event detection [84, 111, 120], coreference [90, 95], temporal analysis [17, 46], sequencing [2], script induction [7, 18, 122, 139].

However, studies on event salience are premature. Some previous work attempts to approximate event salience with word frequency or discourse position [155, 161]. Parallel to ours, Choubey et al. [33] propose a task to find the most dominant event in news articles. They draw connections between event coreference and importance, on hundreds of closed-domain documents, using several oracle event attributes. In contrast, our proposed models are fully learned and applied on more general domains and at a larger scale. We also do not restrict to a single most important event per document.

There is a small but growing line of work on entity salience [47, 48, 126, 159]. In this work, we study the case for events.

Text relations have been studied in tasks like text summarization, which mainly focused on cohesion [67]. Grammatical cohesion methods make use of document level structures such as anaphora relations [8] and discourse parse trees [99]. Lexical cohesion based methods focus on repetitions and synonyms on the lexical level [52, 108, 147]. Though sharing similar intuitions, our proposed models are designed to learn richer semantic relations in the embedding space.

Comparing to the traditional summarization task, we focus on events, which are at a different granularity. Our experiments also unveil interesting phenomena among events and other discourse units.

6.3 The Event Salience Corpus

This section introduces our approach to construct a large-scale event salience corpus, including methods for finding event mentions and obtaining saliency labels. The studies are based on the Annotated New York Times corpus [141], a newswire corpus with expert-written abstracts.

6.3.1 Automatic Corpus Creation

Event Mention Annotation: Despite many annotation attempts on events [14, 130], automatic labeling of them in general domain remains an open problem. Most of the previous work follows empirical approaches. For example, Chambers and Jurafsky [18] consider all verbs together with their subject and object as events. Do et al. [45] additionally include nominal predicates, using the nominal form of verbs and lexical items under the *Event* frame in FrameNet [6].

There are two main challenges in labeling event mentions. First, we need to decide which lexical items are event triggers. Second, we have to disambiguate the word sense to correctly identify events. For example, the word "phone" can refer to an entity (a physical phone) or an

[2] `https://github.com/hunterhector/EventSalience`

	Train	Dev	Test
# Documents	526126	64000	63589
Avg. # Word	794.12	790.27	798.68
Avg. # Events	61.96	60.65	61.34
Avg. # Salience	8.77	8.79	8.90

Table 6.1: Dataset Statistics.

event (a phone call event). We use FrameNet to solve these problems. We first use a FrameNet based parser: Semafor [42], to find and disambiguate triggers into frame classes. We then use the FrameNet ontology to select event mentions.

Our frame based selection method follows the Vendler classes [154], a four way classification of eventuality: *states*, *activities*, *accomplishments* and *achievements*. The last three classes involve state change, and are normally considered as events. Following this, we create an "event-evoking frame" list using the following procedure:

1. We keep frames that are subframes of *Event* and *Process* in the FrameNet ontology.
2. We discard frames that are subframes of state, entity and attribute frames, such as *Entity*, *Attributes*, *Locale*, etc.
3. We manually inspect frames that are not subframes of the above-mentioned ones (around 200) to keep event related ones (including subframes), such as *Arson*, *Delivery*, etc.

This gives us a total of 569 frames. We parse the documents with Semafor and consider predicates that trigger a frame in the list as candidates. We finish the process by removing the light verbs[3] and reporting events[4] from the candidates, similar to previous research [135].

Salience Labeling: For all articles with a human written abstract (around 664,911) in the New York Times Annotated Corpus, we extract event mentions. We then label an event mention as salient if we can find its lemma in the corresponding abstract (Mitamura et al. [103] showed that lemma matching is a strong baseline for event coreference.). For example, in Figure 6.1, event mentions in bold and red are found in the abstract, thus labeled as salient. Data split is detailed in Table 6.1 and §6.6.

6.3.2 Annotation Quality

While the automatic method enables us to create a dataset at scale, it is important to understand the quality of the dataset. For this purpose, we have conducted two small manual evaluation study.

Our lemma-based salience annotation method is based on the assumption that lemma matching being a strong detector for event coreference. In order to validate this assumption, one of the

[3]Light verbs carry little semantic information: "appear", "be", "become", "do", "have", "seem", "do", "get", "give", "go", "have", "keep", "make", "put", "set", "take".

[4]Reporting verbs are normally associated with the narrator: "argue", "claim", "say", "suggest", "tell".

Name	Description
`Frequency`	The frequency of the event lemma in document.
`Sentence Location`	The location of the first sentence that contains the event.
`Event Voting`	Average cosine similarity with other events in document.
`Entity Voting`	Average cosine similarity with other entities in document.
`Local Entity Voting`	Average cosine similarity with entities in the sentence.

Table 6.2: Event Salience Features.

authors manually examined 10 documents and identified 82 coreferential event mentions pairs between the text body and the abstract. The automatic lemma rule identifies 72 such pairs: 64 of these matches human decision, producing a precision of 88.9% $(64/72)$ and a recall of 78% $(64/82)$. There are 18 coreferential pairs missed by the rule.

The next question is: *is an event really important if it is mentioned in the abstract?* Although prior work [48] shows that the assumption to be valid for entities, we study the case for events. We asked two annotators to manually annotate 10 documents (around 300 events) using a 5-point Likert scale for salience. We compute the agreement score using Cohen's Kappa [34]. We find the task to be challenging for human: annotators don't agree well on the 5-point scale (Cohen's Kappa = 0.29). However, if we collapse the scale to binary decisions, the Kappa between the annotators raises to 0.67. Further, the Kappa between each annotator and automatic labels are 0.49 and 0.42 respectively. These agreement scores are also close to those reported in the entity salience tasks [48].

While errors exist in the automatic annotation process inevitably, we find the error rate to be reasonable for a large-scale dataset. Further, our study indicates the difficulties for human to rate on a finer scale of salience. We leave the investigation of continuous salience scores to future work.

6.4 Feature-Based Event Salience Model

This section presents the feature-based model, including the features and the learning process.

6.4.1 Features

Our features are summarized in Table 6.2.

Basic Discourse Features: We first use two basic features similar to Dunietz and Gillick [48]: *Frequency* and *Sentence Location*. *Frequency* is the lemma count of the mention's syntactic head word [98]. *Sentence Location* is the sentence index of the mention, since the first few sentences are normally more important. These two features are often used to estimate salience [9, 155].

Content Features: We then design several lexical similarity features, to reflect Grimes' content relatedness [64]. In addition to events, the relations between events and entities are also important. For example, Figure 6.1 shows some related entities in the legal domain, such as "*prosecutors*" and "*court*". Ideally, they should help promote the salience status for event "*trial*".

Lexical relations can be found both within-sentence (local) or across sentence (global) [67]. We compute the local part by averaging similarity scores from other units in the same sentence. The global part is computed by averaging similarity scores from other units in the document. All similarity scores are computed using cosine similarities on pre-trained embeddings [102].

These lead to 3 content features: *Event Voting*, the average similarity to other events in the document; *Entity Voting*, the average similarity to entities in the document; *Local Entity Voting*, the average similarity to entities in the same sentence. Local event voting is not used since a sentence often contains only 1 event.

6.4.2 Model

A Learning to Rank (`LeToR`) model [88] is used to combine the features. Let ev_i denote the ith event in a document d. Its salience score is computed as:

$$f(ev_i, d) = W_f \cdot F(ev_i, d) + b \tag{6.1}$$

where $F(ev_i, d)$ is the features for ev_i in d (Table 6.2); W_f and b are the parameters to learn.

The model is trained with pairwise loss:

$$\sum_{ev^+, ev^- \in d} \max(0, 1 - f(ev^+, d) + f(ev^-, d)), \tag{6.2}$$
$$\text{w.r.t. } y(ev^+, d) = +1 \ \& \ y(ev^-, d) = -1.$$

$$y(e_i, d) = \begin{cases} +1, & \text{if } e_i \text{ is a salient entity in } d, \\ -1, & \text{otherwise.} \end{cases}$$

where ev^+ and ev^- represent the salient and non-salient events; y is the gold standard function. Learning can be done by standard gradient methods.

6.5 Neural Event Salience Model

As discussed in §6.1, the salience of discourse units is reflected by rich relations beyond lexical similarities, for example, script ("*charge*" and "*trial*") and frame (a "*trial*" of "*attacks*"). The relations between these words are specific to the salience task, thus difficult to be captured by raw cosine scores that are optimized for word similarities. In this section, we present a neural model to exploit the embedding space more effectively, in order to capture relations for event salience estimation.

6.5.1 Kernel-based Centrality Estimation

Inspired by the kernel ranking model [158], we propose Kernel-based Centrality Estimation (KCE), to find and weight semantic relations of interests, in order to better estimate salience.

Formally, given a document d, the set of annotated events $\mathbb{V} = \{ev_1, \dots ev_i \dots, ev_n\}$, KCE first embed an event into vector space: $ev_i \xrightarrow{Emb} \overrightarrow{ev_i}$. The embedding function is initialized with pre-trained embeddings. It then extract K features for each ev_i:

$$\Phi_K(ev_i, \mathbb{V}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_k(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_K(\overrightarrow{ev_i}, \mathbb{V})\}, \tag{6.3}$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{V}) = \sum_{ev_j \in \mathbb{V}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{ev_j}) - \mu_k)^2}{2\sigma_k^2}\right). \tag{6.4}$$

$\phi_k(\overrightarrow{ev_i}, \mathbb{V})$ is the k-th Gaussian kernel with mean μ_k and variance σ_k^2. It models the interactions between events in its kernel range defined by μ_k and σ_k. $\Phi_K(ev_i, \mathbb{V})$ enforces multi-level interactions among events — relations that contribute similarly to salience are expected to be grouped into the same kernels. Such interactions greatly improve the capacity of the model with negligible increase in the number of parameters. Empirical evidences [158] have shown that kernels in this form are effective to learn weights for task-specific term pairs.

The final salience score is computed as:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + b, \tag{6.5}$$

where W_v is learned to weight the contribution of the certain relations captured by each kernel.

We then use the exact same learning objective as in equation (6.2). The pairwise loss is first back-propagated through the network to update the kernel weights W_v, assigning higher weights to relevant regions. Then the kernels use the gradients to update the embeddings, in order to capture the meaningful discourse relations for salience.

Since the features and KCE capture different aspects, combining them may give superior performance. This can be done by combining the two vectors in the final linear layer:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{6.6}$$

6.5.2 Integrating Entities into KCE

KCE is also used to model the relations between events and entities. For example, in Figure 6.1, the entity "*court*" is a frame element of the event "*trial*"; "*United States*" is a frame element of the event "*war*". It is not clear which pair contributes more to salience. We again let KCE to learn it.

Formally, let $\mathbb{E}$ be the list of entities in the document, i.e. $\mathbb{E} = \{en_1, \dots, en_i, \dots, en_n\}$, where en_i is the ith entity in document d. KCE extracts the kernel features about entity-event relations as follows:

$$\Phi_K(ev_i, \mathbb{E}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{E}), \ldots, \phi_k(\overrightarrow{ev_i}, \mathbb{E}), \ldots, \phi_K(\overrightarrow{ev_i}, \mathbb{E})\}, \tag{6.7}$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{E}) = \sum_{en_j \in \mathbb{E}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{en_j}) - \mu_k)^2}{2\sigma_k^2}\right) \tag{6.8}$$

similarly, en_i is embedded by: $en_i \xrightarrow{Emb} \overrightarrow{en_i}$, which is initialized by pre-trained entity embeddings.

We reach the full KCE model by combining all the vectors using a linear layer:

$$f(ev_i, d) = W_e \cdot \Phi_K(ev_i, \mathbb{E}) + W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{6.9}$$

The model is again trained by equation (6.2).

6.6 Experimental Methodology

This section describes our experiment settings.

6.6.1 Event Salience Detection

Dataset: We conduct our experiments on the salience corpus described in §6.3. Among the 664,911 articles with abstracts, we sample 10% of the data as the test set and then randomly leave out another 10% documents for development. Overall, there are 4359 distinct event lexical items, at a similar scale with previous work [18, 45]. The corpus statistics are summarized in Table 6.1.
Input: The inputs to models are the documents and the extracted events. The models are required to rank the events from the most to least salience.
Baselines: Three methods from previous researches are used as baselines: *Frequency*, *Location* and *PageRank*. The first two are often used to simulate saliency [9, 155]. The *Frequency* baseline ranks events based on the count of the headword lemma; the *Location* baseline ranks events using the order of their appearances in discourse. Ties are broken randomly.

Similar to entity salience ranking with PageRank scores [159], our *PageRank* baseline runs PageRank on a fully connected graph whose nodes are the events in documents. The edges are weighted by the embedding similarities between event pairs. We conduct supervised PageRank on this graph, using the same pairwise loss setup as in KCE. We report the best performance obtained by linearly combining *Frequency* with the scores obtained after a one-step random walk.
Evaluation Metric: Since the importance of events is on a continuous scale, the boundary between "important" and "not important" is vague. Hence we evaluate it as a ranking problem. The metrics are the precision and recall value at 1, 5 and 10 respectively. It is adequate to stop at 10 since there are less than 9 salient events per document on average (Table 6.1). We also report Area Under Curve (AUC). Statistical significance values are tested by permutation (randomization) test with $p < 0.05$.

Implementation Details: We pre-trained word embeddings with 128 dimensions on the whole Annotated New York Times corpus using Word2Vec [102]. Entities are extracted using the TagMe entity linking toolkit [56]. Words or entities that appear only once in training are replaced with special "unknown" tokens.

The hyper-parameters of the KCE kernels follow previous literature [158]. There is one exact match kernel ($\mu = 1, \sigma = 1e^{-3}$) and ten soft-match kernels evenly distributed between $(-1, 1)$, i.e. $\mu \in \{-0.9, -0.7, \ldots, 0.9\}$, with the same $\sigma = 0.1$.

The parameters of the models are optimized by Adam [78], with batch size 128. The vectors of entities are initialized by the pre-trained embeddings. Event embeddings are initialized by their headword embedding.

6.6.2 The Event Intrusion Test: A Study

KCE is designed to estimate salience by modeling relations between discourse units. To better understand its behavior, we design the following **event intrusion test**, following the word intrusion test used to assess topic model quality [23].

Event Intrusion Test: The test will present to a model a set of events, including: the **origins**, all events from one document; the **intruders**, some events from another document. Intuitively, if events inside a document are organized around the core content, a model capturing their relations well should easily identify the intruder(s).

Specifically, we take a bag of unordered events $\{O_1, O_2, \ldots, O_p\}$, from a document O, as the origins. We insert into it intruders, events drawn from another document, I: $\{I_1, I_2, \ldots, I_q\}$. We ask a model to rank the mixed event set $M = \{O_1, I_1, O_2, I_2, \ldots\}$. We expect a model to rank the intruders I_i below the origins O_i.

Intrusion Instances: From the development set, we randomly sample 15,000 origin and intruding document pairs. To simplify the analysis, we only take documents with at least 5 salient events. The intruder events, together with the entities in the same sentences, are added to the origin document.

Metrics: AUC is used to quantify ranking quality, where events in O are positive and events in I are negative. To observe the ranking among the salient origins, we compute a separate AUC score between the intruders and the salient origins, denoted as SA-AUC. In other words, SA-AUC is the AUC score on the list with non-salient origins removed.

Experiments Details: We take the full KCE model to compute salient scores for events in the mixed event set M, which are directly used for ranking. Frequency is recounted. All other features (Table 6.2) are set to 0 to emphasize the relational aspects,

We experiment with two settings: 1. adding only the salient intruders. 2. adding only the non-salient intruders. Under both settings, the intruders are added one by one, allowing us to observe the score change regarding the number of intruders added. For comparison, we add a *Frequency* baseline, that directly ranks events by the Frequency feature.

6.7 Evaluation Results

This section presents the evaluations and analyses.

Method	P@01		P@05		P@10		AUC	
Location	0.3555	–	0.3077	–	0.2505	–	0.5226	–
PageRank	0.3628	–	0.3438	–	0.3007	–	0.5866	–
Frequency	0.4542	–	0.4024	–	0.3445	–	0.5732	–
LeToR	$0.4753^{\dagger}$	+4.64%	$0.4099^{\dagger}$	+1.87%	$0.3517^{\dagger}$	+2.10%	$0.6373^{\dagger}$	+11.19%
KCE (-EF)	0.4420	−2.69%	0.4038	+0.34%	$0.3464^{\dagger}$	+0.54%	$0.6089^{\dagger}$	+6.23%
KCE (-E)	$0.4861^{\dagger\ddagger}$	+7.01%	$0.4227^{\dagger\ddagger}$	+5.04%	$0.3603^{\dagger\ddagger}$	+4.58%	$0.6541^{\dagger\ddagger}$	+14.12%
KCE	$0.5049^{\dagger\ddagger}$	+11.14%	$0.4277^{\dagger\ddagger}$	+6.29%	$0.3638^{\dagger\ddagger}$	+5.61%	$0.6557^{\dagger\ddagger}$	+14.41%

Method	R@01		R@05		R@10		W/T/L
Location	0.0807	–	0.2671	–	0.3792	–	–/–/–
PageRank	0.0758	–	0.2760	–	0.4163	–	–/–/–
Frequency	0.0792	–	0.2846	–	0.4270	–	–/–/–
LeToR	$0.0836^{\dagger}$	+5.61%	$0.2980^{\dagger}$	+4.70%	$0.4454^{\dagger}$	+4.31%	8037 / 48493 / 6770
KCE (-EF)	0.0714	−9.77%	0.2812	−1.18%	$0.4321^{\dagger}$	+1.20%	6936 / 48811 / 7553
KCE (-E)	$0.0925^{\dagger\ddagger}$	+16.78%	$0.3172^{\dagger\ddagger}$	+11.46%	$0.4672^{\dagger\ddagger}$	+9.41%	11676 / 43294 / 8330
KCE	$0.0946^{\dagger\ddagger}$	+19.44%	$0.3215^{\dagger\ddagger}$	+12.96%	$0.4719^{\dagger\ddagger}$	+10.51%	12554 / 41461 / 9285

Table 6.3: Event salience performance. (-E) and (-F) marks removing Features and Entity information from the full KCM model. The relative performance differences are computed against Frequency. W/T/L are the number of documents a method wins, ties, and loses compared to Frequency. † and ‡ mark the statistically significant improvements over $\texttt{Frequency}^{\dagger}$, $\texttt{LeToR}^{\ddagger}$ respectively.

6.7.1 Event Salience Performance

We summarize the main results in Table 6.3.

Baselines: *Frequency* is the best performing baseline. Its precision at 1 and 5 are higher than 40%. *PageRank* performs worse than *Frequency* on all the precision and recall metrics. *Location* performs the worst.

Feature Based: LeToR outperforms the baselines significantly on all metrics. Particularly, its P@1 value outperforms the *Frequency* baseline the most (4.64%), indicating a much better estimation on the most salient event. In terms of AUC, LeToR outperforms *Frequency* by a large margin (11.19% relative gain).

Feature Ablation: To understand the contribution of individual features, we conduct an ablation study of various feature settings in Table 4.5. We gradually add feature groups to the *Frequency* baseline. The combination of *Location* (sentence location) and *Frequency* almost sets the performance for the whole model. Adding each voting feature individually produces mixed results. However, adding all voting features improves all metrics. Though the margin is small, 4 of them are statistically significant over *Frequency*+*Location*.

Kernel Centrality Estimation: The KCE model further beats LeToR significantly on all metrics, by around 5% on AUC and precision values, and by around 10% on the recall values. Notably, the P@1 score is much higher, reaching 50%. The large relative gain on all the recall metrics and

Feature Groups	P@1	P@5	P@10	R@1	R@5	R@10	AUC
SL	0.3548	0.3069	0.2497	0.0807	0.2671	0.3792	0.5226
Frequency	0.4536	0.4018	0.3440	0.0792	0.2846	0.4270	0.5732
+ SL	0.4734	0.4097	0.3513	0.0835	0.2976	0.4436	0.6354
+ SL + Event	0.4726	0.4101†	0.3516	0.0831	0.2969	0.4431	0.6365†
+ SL + Entity	0.4739	0.4100	0.3518	0.0812	0.2955	0.4418	0.6374
+ SL + Entity + Event	0.4739	0.4100	0.3518†	0.0832	0.2974	0.4452†	0.6374†
+ SL + Entity + Event + Local	0.4754†	0.4100	0.3517†	0.0837	0.2981	0.4454†	0.6373†

Table 6.4: Event Salience Feature Ablation Results. The + sign indicates adding feature groups to Frequency. SL is the sentence location feature. Event is the event voting feature. Entity is the entity voting feature. Local is the local entity voting feature. † marks the statistically significant improvements over +SL.

the high performance on precision show that KCE works really well on the top of the rank list.

Kernel Ablation: To understand the source of performance gain of KCE, we conduct an ablation study by removing its components: −E removes of entity kernels; −EF removes the entity kernels and the features. We observe a performance drop in both cases. Without entities and features, the model only using event information still performs similarly to *Frequency*. The drops are also a reflection of the small number of events ($\approx$ 60 per document) comparing to entities ($\approx$ 200 per document). The study indicates that the relational signals and features contain different but both important information.

Discussion: The superior results of KCE demonstrate its effectiveness in predicting salience. So what additional information does it capture? We revisit the changes made by KCE: 1. it adjusts the embeddings during training. 2. it introduces weighted soft count kernels. However, the *PageRank* baseline also does embedding tuning but produces poor results, thus the second change should be crucial. We plot the learned kernel weights of KCE in Figure 6.2. Surprisingly, the salient decisions are not linearly related, nor even positively correlated to the weights. In fact, besides the "Exact Match" bin, the highest absolute weights actually appear at 0.3 and -0.3. This implies that embedding similarities do not directly imply salience, breaking some assumptions of the feature based model and *PageRank*.

Case Study: We inspect some pairs of events and entities in different kernels and list some examples in Table 6.5. The pre-trained embeddings are changed a lot. Pairs of units with different raw similarity values are now placed in the same bin. The pairs in Table 3 exhibit interesting types of relations: e.g.,"*arrest-charge*" and "*attack-kill*" form script-like chains; "911 attack" forms a quasi-identity relation [133] with "attack"; "business" and "increase" are candidates as frame-argument structure. While these pairs have different raw cosine similarities, they are all useful in predicting salience. KCE learns to gather these relations into bins assigned with higher weights, which is not achieved by pure embedding based methods. The KCE has changed the embedding space and the scoring functions significantly from the original space after training.

		Word2Vec	Kernel
attack	kill	0.69	0.3
arrest	charge	0.53	0.3
USA (**E**)	war	0.46	0.3
911 attack (**E**)	attack	0.72	0.3
attack	trade	0.42	0.9
hotel (**E**)	travel	0.49	0.9
charge	murder	0.49	0.7
business(**E**)	increase	0.43	0.7
attack	walk	0.44	-0.3
people (**E**)	work	0.40	-0.3

Table 6.5: Examples of pairs of Events/Entities in the kernels. The **Word2vec** column shows the cosine similarity using pre-trained word vectors. The **Kernel** column shows the closest kernel they belong after training. Items marked with (**E**) are entities.

This partially explains why the raw voting features and PageRank are not as effective.

6.7.2 Intrusion Test Results

Figure 6.3 plots results of the intrusion test . The left figure shows the results of setting 1: adding non-salient intruders. The right one shows the results of setting 2: adding salient intruders. The AUC is 0.493 and the SA-AUC is 0.753 if all intruders are added.

The left figure shows that KCE successfully finds the non-salient intruders. The SA-AUC is higher than 0.8. Yet the AUC scores, which include the rankings of non-salience events, are rather close to random. This shows that the salient events in the origin documents form a more cohesive group, making them more robust against the intruders; the non-salient ones are not as cohesive.

In both settings, KCE produces higher SA-AUC than *Frequency* at the first 30%. However, in setting 2, KCE starts to produce lower SA-AUC than *Frequency* after 30%, then gradually drops to 0.5 (random). This phenomenon is expected since the asymmetry between origins and intruders allow KCE to distinguish them at the beginning. When all intruders are added, KCE performs worse because it relies heavily on the relations, which can be also formed by the salient intruders. This phenomenon is observed only on the salient intruders, which again confirms the cohesive relations are found among salient events.

In conclusion, we observe that the salient events form tight groups connected by discourse relations while the non-salient events are not as related. The observations imply that the main scripts in documents are mostly anchored by small groups of salient events (such as the "Trial" script in Example 6.1). Other events may serve as "backgrounds" [29]. Similarly, Choubey et al. [33] find that relations like event coreference and sequence are important for saliency.

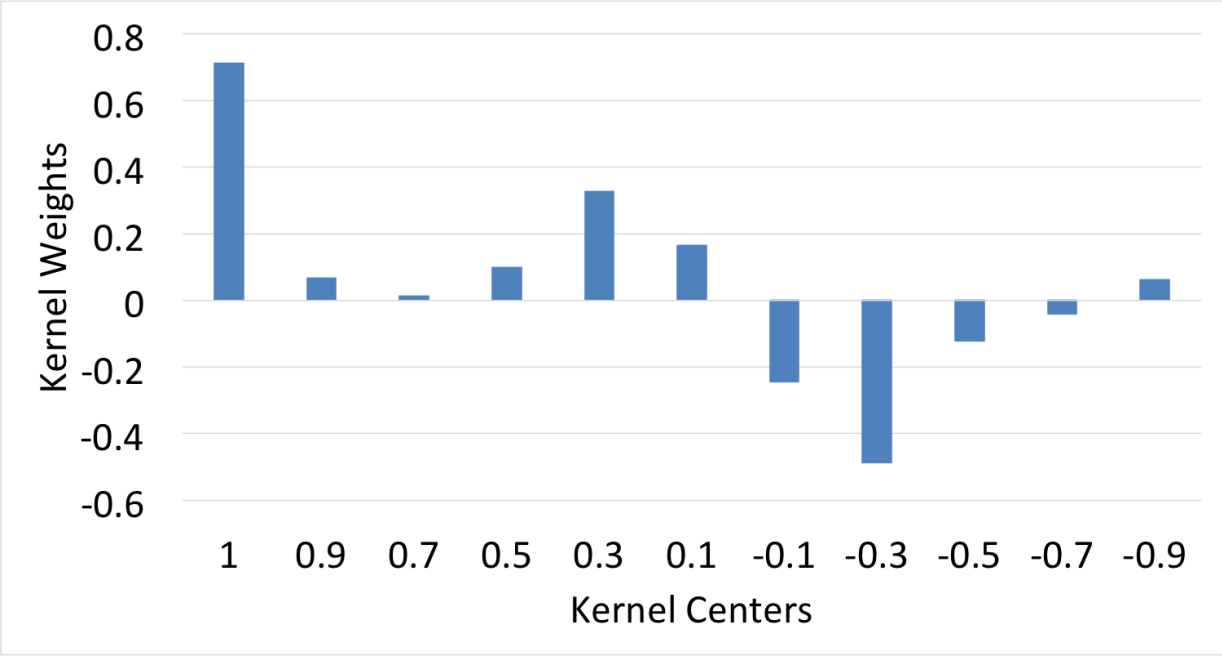


Figure 6.2: Learned Kernel Weights of KCE

6.8 Conclusion

In this chapter we describe two salient detection models, based on lexical relatedness and semantic relations. The feature-based model with lexical similarities is effective, but cannot capture semantic relations like scripts and frames. The KCE model uses kernels and embeddings to capture these relations, thus outperforms the baselines and feature-based models significantly. All the results are tested on our newly created large-scale event salience dataset. While the automatic method inevitably introduces noises to the dataset, the scale enables us to study complex event interactions, which is infeasible via costly expert labeling.

Our case study shows that the salience model finds and utilize a variety of discourse relations: script chain (*attack* and *kill*), frame argument relation (*business* and *increase*), quasi-identity (*911 attack* and *attack*). Such complex relations are not as prominent in the raw word embedding space. The core message is that a salience detection module automatically discovers connections between salience and relations. This goes beyond prior centering analysis work that focuses on lexical and syntax and provide a new semantic view from the script and frame perspective.

In the intrusion test, we observe that the small number of salient events are forming tight connected groups. While KCE captures these relations quite effectively, it can be confused by salient intrusion events. The phenomenon indicates that the salient events are tightly connected, which form the main scripts of documents.

In this chapter, we have shown that we can use indirect supervision signals to reveal many interesting semantic relations between discourse phenomena and salience. Some of them are not directly related to our proposed task. For example, our study suggests that core script information

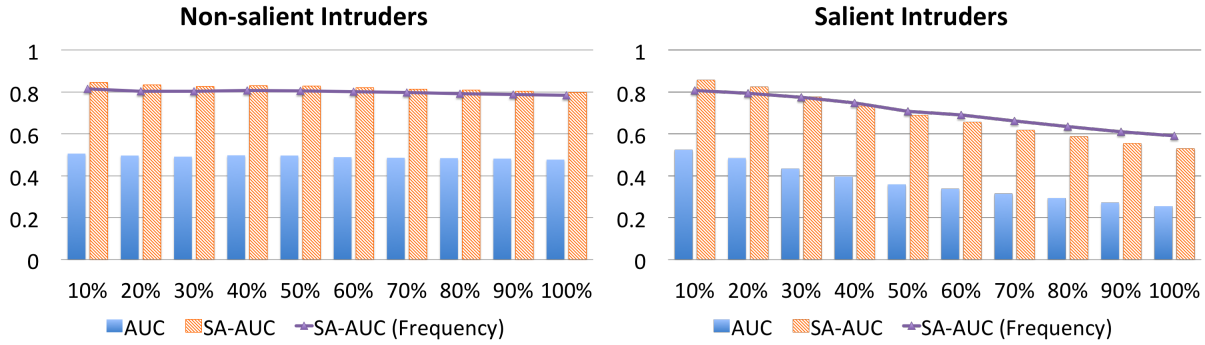


Figure 6.3: Intruder study results. X-axis shows the percentage of intruders inserted. Y-axis is the AUC score scale. The left and right figures are results from salient and non-salient intruders respectively. The blue bar is AUC. The orange shaded bar is SA-AUC. The line shows the SA-AUC of the frequency baseline.

may reside mostly in the salient events. In this proposal, we plan to use similar methods to reveal other semantic knowledge from data, with a focus on frames and scripts.

Part III

Proposed Work

Chapter 7

Proposed Theoretical Framework

7.1 Introduction

The information that can be convey via natural language is limited by its expressiveness. Fauconnier [53] has pointed out that language is a "superficial manifestation of hidden, highly abstract, cognitive constructions", and that "a natural language sentence is a completely different king of thing from a sentence in a logical calculus". Natural language can not provide precise and unambiguous information. Instead, language only has the potential to convey meaning, and that meaning is only created after projection to the mental space. Only part of meanings implied by a natural language expression are intended to be communicated.

Textual mentions of discourse elements (DEs), such as events and entities, are no exception. The DEs often have rich denotation, a textual mention normally only focuses on part of the whole denotation space. Prior theoretical work [134] has shown that it is important to focus on the partial denotation when modeling coreference between DEs. In this chapter, we will discuss that focusing on the correct parts is also important for modeling other relations among DEs.

7.1.1 The Exact Argument Identity Assumption and Its Failures

There are two levels of event relations that we are mostly interested in. The **mention level** studies the predicate argument relations for an event mention, and the **script level** considers the relations of elements across events (e.g. whether the arguments in different mentions are filling the same frame slot). Arguments are one of the key connections between these two levels, since they may be shared by multiple events in a script. For example, in a restaurant script, the customer may enter the restaurant, orders, ate and left the restaurant. The "customer" is a shared argument among these subevents. To differentiate with the semantic frame roles, we refer to the shared role as a **script role** and the filler as a **script argument**.

A popular approach of script modeling uses script argument to connect the events [15, 119, 124]. This approach typically represents the scripts in the form of ⟨event, frame role⟩ chains[1]. For example, a shared script role "criminal suspect" may connect a chain of events like: ⟨*search,*

[1]Some approaches also use multiple arguments instead of one, but our discussion still hold in that case

Arg1⟩, ⟨*arrest, Arg1*⟩, ⟨*plead, Arg0*⟩, ⟨*convict, Arg1*⟩, ⟨*sentence, Arg1*⟩. The traditional approaches identify such chains using entity coreference chains, which in fact use the following assumption.

> **Assumption 7.1.1** (Exact Argument Identity Assumption)**.** *The shared script arguments of the event mentions in a script should be identical.*

As a special case of this assumption, if a pair of event mentions are coreferent, then the fillers in their respective argument slots (e.g. the agent for both events) should be coreferent.

At the first glance, this assumption is reasonable. However, it does not always hold, especially when we use the current widely-accepted definition of coreference (we will give a formal definition in the next section). This can be demonstrated by analyzing the scripts in the following paragraph:

(7.1) The police `warned` the crowd to disperse, but the protesters `refused` to leave Taksim Square and chanted anti-government slogans. Dozens of police officers then moved toward the crowd and began `spraying` the protesters with water cannons. ... said as she held red carnations in her hand. "I wanted to hand these to them, but instead they `pushed` me away with their shields and said our right to protest was over."

The paragraph describes a `protest` script with several events and their participants. Following the Exact Argument Identity Assumption, the fillers in the shared arguments slots should be identical. One chain of shared frame elements are the "police" (⟨*warn, Arg0*⟩, ⟨*refuse, Arg1*⟩, ⟨*spray, Arg0*⟩, ⟨*push, Arg0*⟩). Another chain is about the "protestor" (⟨*warn, Arg1*⟩, ⟨*refuse, Arg0*⟩, ⟨*spray, Arg1*⟩, ⟨*push, Arg1*⟩). Such ideal setting is shown in Figure 7.1a.

However, the actual analysis should be described as in Figure 7.1b. While one chain is still about the "police", it is unclear whether "Dozens of police officers" should be identical to "The Police" — we do not have enough information to determine whether the two sets contain exactly the same members. In the "protestors" chain, it is also inappropriate to treat "she", a single protestor, to be identical to the group of protestors.

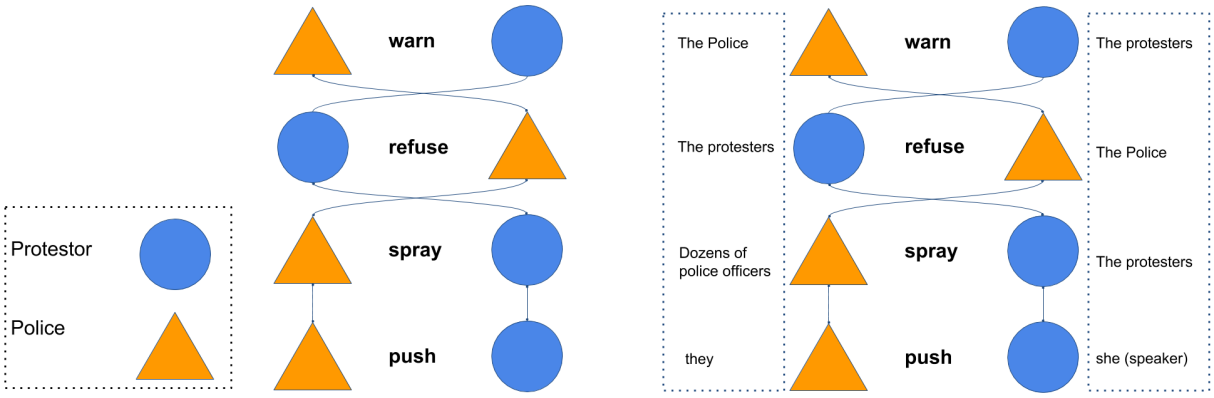
Similar violations are also observed our prior experiments, such as in event coreference (Chapter 3). Example 7.2 is taken from the TAC-KBP event coreference data, the two `arrested` event mentions are considered as coreference, although their corresponding locations are not exactly the same: they are in the same scope, but described with different level of granularity.

(7.2)
1. Man Linked by U.S. to Hezbollah is `arrested` in `Brazil`.
2. The suspect was `arrested` Thursday in the `city of Curitiba in southern Brazil`.

Since argument sharing in coreference can be viewed as a special case in script modeling, we can summarize the problem as followed:

> *Problem* 7.1.1 (Script Argument Inconsistency). The same script argument may be instantiated as multiple mentions across a script. These mentions may not be coreferent under the exact coreference definition.

In the remaining of this chapter, we will first discuss the theoretical analysis for the problem. We then introduce our proposed theoretical framework targeting at solving it. The framework is inspired by the quasi-identity theory [134]. We further discuss that this framework have implications to boarder event semantics, especially on event state modeling.



(a) The ideal script model

(b) The actual script model.

Figure 7.1: While the ideal script model (Left) assumes the shared frame elements to be exactly identical, actual text (Right) demonstrates more complexities.

7.2 The Complexity about Identity

In this section, we argue that Problem 7.1.1 arises from the definition of coreference: the identity of discourse elements. To be exact, the gaps are inevitable under the widely accepted notion of *exact coreference*, which is phrased by Recasens et al. [134] as:

> **Definition 7.2.1** (Exact Coreference)**.** Coreference is a relation holding between two (or more) linguistic expressions that refer to the same entity in the real world.[2]

The major problem of this definition is that discourse elements are treated as holistic units. Thus when an entity or event is mentioned several times, its entire denotation is assumed for producing the interpretation. This contradicts with the real-world usage, where people often focus on different facets of the denotation. So even if two entity or event mentions are technically coreferent, in reality only some facets might be intended both times, while some facets may be intended in one mention and others in the other.

Recasens et al. [134] pointed out the problem and argue that identity should be treated as a scalar value instead of a boolean decision. They formally define it as followed:

> **Definition 7.2.2** (Quasi Coreference)**.** Coreference is a scalar relation holding between two (or more) linguistic expressions that refer to discourse entities considered to be at the same granularity level relevant to the linguistic and pragmatic context.

The Quasi Coreference definition differs Exact Coreference from several aspects. First, it considers coreference as a scalar relation. Second, it states that coreference happens in *the discourse model* instead of *the real world*: the discourse model is a projected world, a partial mental replica of the actual world built by language users (see [134, 156] for more details). Third,

[2]Their discussion on coreference are general and can be applied to other discourse units, such as events.

the coreference criteria depends on the context. Here we briefly introduce this new coreference definition with the following examples (taken from [134]):

(7.3) Last night in Tel Aviv, *Jews* attacked a restaurant that employs Palestinians. "*We* want war," *the crowd* chanted.

(7.4) The *plant* colonized the South of France, from where *it* entered Catalonia in the 80s, spreading quickly.

In 7.3, the mention *Jews* is a conceptual set that contain the participants of the attacks. The mention *we* is a generic set of all the members involved in this incident. The mention *the crowd* only refer to members who are "chanting". It is not likely that we can establish a full set equality between these groups. However, Recasens et al. [134] argue that the sets themselves lose their distinctive features given the purpose of the discourse, thus quasi-coreference can be established between them. In 7.4, the mention *plant* are described twice with different locations (the ones in the South of France and the ones in Catalonia). However, quasi-coreference can be established because the purpose of the discourse is trying to emphasize that the same plant colonized and then entered and spread.

We argue that the shared arguments in a script should also be modelled by scalar relations, unlike the boolean decision used in the Exact Argument Identity Assumption. Our proposed solution to Problem 7.1.1 is to incorporate quasi-identity to Assumption 7.1.1:

> **Assumption 7.2.1** (Quasi Argument Identity Assumption)**.** *The shared frames elements in a script have non-zero quasi-coreferent value.*

In the coreference Example 7.2, the arguments can be classified as quasi-coreferent by unifying the granularity differences. Similarly, in the script Example 7.1, we can establish quasi Set.Set and Part.Whole[3] relations between the conflicting fillers. [4]

Computationally, *Frame Element Inconsistency* creates difficulties for models to exploit the mutual benefits between entity and event relations. For example, it may have reduced the effectiveness of entity coreference for within-document event coreference [32]. Script mining based on entity coreference will be sparser than expected [160]. Or that script information is not helpful to general entity coreference cases [119]. Besides these computational evidences, we also find that the problems have affected corpus annotation on events and entities.

7.2.1 Evidence from Corpus Annotation

The frame inconsistency phenomenon can be frequently observed in datasets under two popular annotation schemes for events and entities: RichERE [87] and ACE [86]. These schemes take different approaches to handle the issue.

The RichERE annotation scheme defines a new relation named *Event Hopper*, which is a relaxed version for Exact Coreference, particularly for dealing with granularity variation of event and entity mentions [87]. The guideline makes several relaxations, such as:

1. Trigger granularity can be different (assaulting 32 people vs. wielded a knife)
2. Event arguments may be non-coreferential or conflicting (18 killed vs. dozens killed)

[3]These cases belongs to different types of Meronymy quasi-identity as classified by Recasens et al. [133].

[4]Readers who are interested in the details of determining quasi-identity types should refer to Recasens et al. [134].

The former rule relaxes the identity decisions between event predicates, which actually is a evidence for the continuum on event identity. The latter rule relaxes the identity decision between the entity pairs in the event arguments. However, given the relaxation rules, the guideline is under-specified in terms of coreference definition: *Event hoppers contain mentions of events that "feel" coreferential to the annotator even if they do not meet the earlier*[5] *strict event identity requirement.* The definition is ad-hoc and may result in some arbitrary annotation decisions.

Unlike RichERE, the ACE corpus [86] takes a strict approach to both event and entity coreference. Generally, the ACE annotation guideline discourages marking any ambiguous coreference links. Such approach reduces the ambiguity during annotation, but also lessen the connections between events and entities. Furthermore, even under the strict constraints, the same frame inconsistency problem can still occur. Hasler and Orasan [71] have done a thorough annotation study on the NP4E corpus based on the ACE ontology, following the guideline of ACE. Their study still reports that the arguments of coreferential event mentions are not coreferent in many cases. To be specific, they found that indirect anaphoric relations (which is defined similar to quasi-identity) between arguments exist in 70% of the event chains annotated. They advocate that indirect referential relations among arguments should be taken into account in building event coreference chains, which is one of the goals we are pursuing in this thesis.

7.3 The Facet View of Discourse Elements

We find prior work on quasi-identity inspiring. The continuum definition of identity is theoretically sound and elegant. However, the continuous scalar value termed in definition 7.2.2 is difficult to be obtained in practice. There is a need for a more **feasible computational view** than the full continuum view.

One approach along this line taken by Recasens et al. [133] is taxonomy-based. They defined 4 types and 15 subtypes for quasi-identity. For example, Example 7.3 shows a quasi-identical link with the Set.Set subtype under Meronymy. However, while the taxonomy provides guidance on which entity pairs can be considered as quasi-identity, the taxonomy type itself is inadequate to demonstrate the nature of the relation: two DEs fitting in the taxonomy can be interpreted as near identical only when both are functionally equivalent given the context. For example, not all meronymy relations are quasi-identical. We propose to seek for an approach with more **explainable power**.

In this section, we are proposing an interpretation of quasi-coreference, based on *facets*, which also provides explanations for coreference decisions. The proposed facet language not only provides a new way to describe and model quasi-coreference, but also has further implications on modeling event entity interactions. In §7.3.1 we will describe how they can help us describe the static properties of entities, with applications on identity/coreference problems. In §7.3.2 we will discuss how the new view may have interesting extensions for modelling events as dynamic functions.

[5]The LightERE guideline, see Song et al. [148].

7.3.1 Facets for Static Identity Analysis

The traditional fully identical view considers a DE as a non-dividable whole, and the continuum view of quasi-coreference considers a DE to be fully continuous. There exists, obviously, a middle ground in between these two. A DE can be viewed as a collection of discrete units, which we will refer to as *facets*[6] in this thesis:

> **Definition 7.3.1** (Facet)**.** The facets of an discourse element are semantic units corresponding to the possible interpretations given the textual description in a discourse model.

For example, the mention of a person can be interpreted as his/her social role or family role. The mention of a organization can be interpreted as a collection of members, or a collection of facilities, or its social function. These are all different facets described by the same mention. In natural language, the mentions of DEs serve the purpose for communication, and generally, only partial facets of the mention is to be communicated. These facets should possibly be identified given the linguistic and pragmatic context, which we call *active facets*:

> **Definition 7.3.2** (Active Facet)**.** The active facets of a mention of a DE are the facets that are considered to be relevant to the language user given the linguistic and pragmatic context.

In other words, the *active facets* are the facets that the language users are trying to communicate. Note that similar to quasi-coreference, the interpretations may subject to the language user, they happen in the discourse model [156] projected by the user instead of the real world. Figure 7.2 provides one example, the mention "paper" may have facets like "burnable", "foldable", but in this sentence, only some of them are relevant, such as "cuttable".

From the facet perspective, we can now describe the identity between discourse elements: If two DEs refer to exactly the same set of facets, they are fully identical. If they are only identical on part of the facets, we enter the realm of quasi-identity and more analysis are needed. Since the active facets are the ones related to the context, a quasi-identity relation should be established when they are they same. We can rephrased the quasi-coreference definition a little using the facet language:

> **Definition 7.3.3** (Facet Based Quasi Coreference)**.** Coreference is a scalar relation holding between two (or more) linguistic expressions when the active facets of the discourse elements are considered to be the same.

We can explain the quasi-coreference examples using the facet language. In 7.4, the *active facets* of the *plant* are the organization perspective, instead of its particular facilities. This can be induced from the uses of the verbs: colonize, enter and spread, since these verbs are not compatible with physical facilities in terms of selectional preference, but are compatible to organizations. Since the active facets of *plant* and *it* are the organization, which are the same one, thus we can establish quasi-coreference between them.

The facet language can be used to model the interaction between events and entities. We can rephrase Assumption 7.2.1 as followed:

[6]Some work uses the word "facet" to represent the taxonomic features for DEs belonging to more than one taxonomy, we overload "facet" to represent various features for DEs, including the taxonomic ones

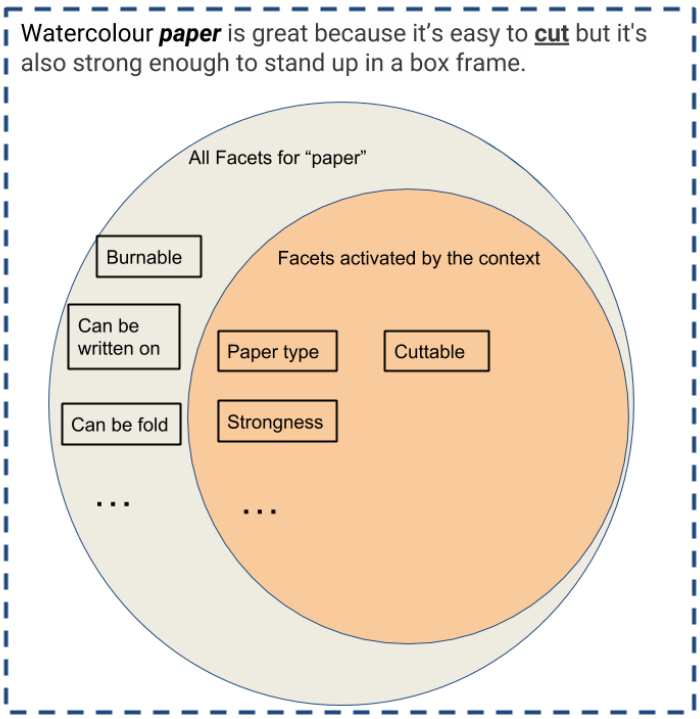


Figure 7.2: Example of active facets for an entity mention of "paper".

Assumption 7.3.1 (Facet based Quasi Argument Identity Assumption)**.** *The shared script arguments in a script share active facets given the context of the script.*

In this new assumption, we have replaced "non-zero coreference value" to "sharing of active facets", which we believe has more explainable power. For example, in 7.1, the script emphasizes the roles of the two parties. Thus the shared active facets for both set of frame elements are their role in protest (e.g. police force and protesters). In 7.2, the purpose of the script can be met without emphasizing the granularity differences between a country and a city. Thus the active facets of the two locations may not share the granularity hence can be considered identical.

Note that we emphasize "context", but we restrict ourselves to the script context only (i.e. event related). There are other context to analyze these facets. For example, the "18 killed vs. dozens killed" example illustrated in the RichERE guideline are due to reporting variations. The active facets of the mention "dozens" can be considered compatible to those of the mention "18" because "dozens" is used as an approximate expression here. We are aware of these complex interpretations for mentions, but studying all of them are out of the scope of this thesis. Our work is mostly under the umbrella of script-based interpretations.

So far, we focus our discussions on the static properties of the entities and events, such as the role of the entities. However, one of the main aspects of events is that they are about the changes of states (such as entity properties). These are the dynamic perspective of events. In the next section, we will discuss how facets are also relevant from the dynamic perspective.

7.3.2 Facets and the Dynamic Interactions

The configurations of the entity properties in the underlying world (the real world or a discourse model) are called *states*. In the state view, entities are objects with properties, and events are functions that change the properties of these objects[7]. State modeling is a very important process for event understanding, which can help use trace the change of the script arguments.

(7.5) Federal investigators say a stolen Horizon Air turboprop *plane* `broke` into *many pieces* when it `crashed` into an area of thick brush on a small island in the Puget Sound.

For example, in 7.5, the state change indicated by `crashed` and `broke` allow us to infer that part of the *plane* become the *pieces*: the plane change from a whole to a collection of parts.

In our analysis, we consider events as functions that take a state as input, and produce a new state. In the discussion on entity understanding, we argue that not all facets of the entities are activated. Similarly, when analyzing the the event functions, we do not need to take the whole entities as arguments. The inference procedure should be more efficient if we focus only on the most relevant facets.

Conveniently, the active facets are relevant to the context by definition. Since events are part of the context, the facets needed to model the event functions should be a subset of the active facets. The set relations can be illustrated by the Venn Graph in Figure 7.3. A few of the "paper" facets are activated by the context, for example, the "strong-ness" facet is triggered by the adjective "strong" and event mention "stand"; The "cuttable" facet is triggered by the event mention "cut". The "cuttable" facet should be the relevant one when analyzing the "cut" event.

By considering the partial facets of event arguments, we can focus our modeling effort on state changes. In the previous example, the state changes triggered by the event mention "cut" can be reduced to applying "cut" to an "cuttable" object. Modeling state change on the facet level makes it easy to generalize the learned knowledge: if we can understand the states after applying "cut" to "paper", we may be able to generalize it to other "cuttable" objects.

7.3.3 Facets for Events

We have not discussed much about the facets for events, which is a not that obvious comparing to entities. One type of multi-facet for events is that they have affected states of multiple objects. In example 7.6, we can identify two active facets from the event "pay": the first one shows that the "payer" will give up the money, and the second one shows that the "goal" to resolve claims is to be achieve. Similarly, there are facets that are not emphasized in this sentence, such as who are the "receiver".

(7.6) Google agreed to pay 125 million dollars to resolve outstanding claims and establish an independent "Book Rights Registry".

Readers who are familiar with frame semantics will realize that "payer", "goal" and the "receiver" are the core frame elements. We find the core frame elements to be closely related to the event facets. This is reasonable since events are representations of state changes. The facets of events should be the different state changes caused (or to be caused) by the event. Since each core argument may have been changed by the event, each one of them have the potential to be

[7]Note that we also slightly overload *states* to include all kind of properties such as mental states

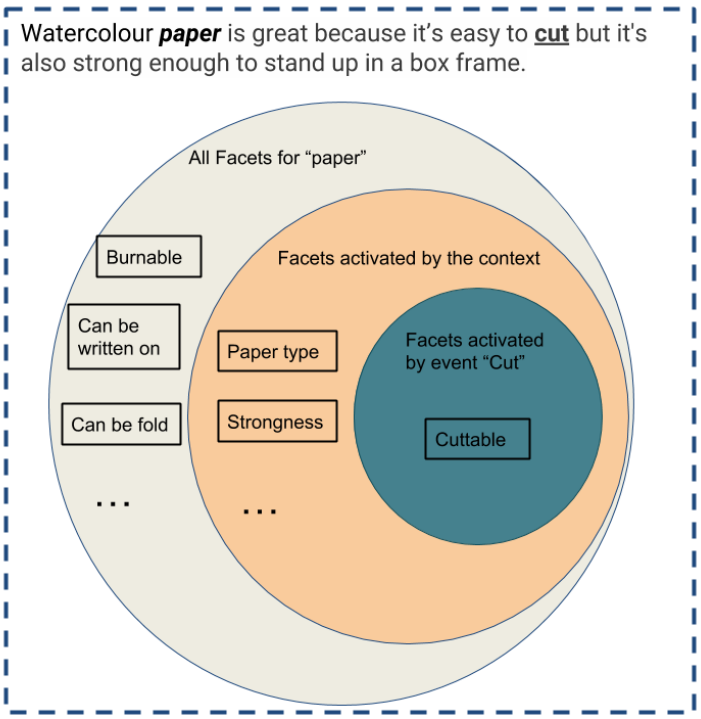


Figure 7.3: A Venn Graph showing the relations between different sets of facets.

a different facet. The event facets are reflected a little in the RichERE annotation as multi-type annotations. For example, a predicate "kill" may have types "Conflict.Attack" and "Life.Death": the former one is more related to the state of the "Attacker" while the latter one relates to the state of the "Victim". Coreference links are also annotated on the type level, which is the same as our proposed facet-based linking. However, there are only a few possible multi-typed event mentions, because the types are restricted by a small number of pre-defined event ontology (35 types).

7.4 Relations to Prior Research

The identity problems have been studied directly or indirectly from many different perspectives. Here we introduce some related research to the best of our knowledge.

7.4.1 Quasi-Identity

Recasens et al. [134] first formally propose the notion of quasi-identity (a.k.a. near-identity), which fills the gap between full coreference and non-coreference. Recasens et al. [133] provides a list of typology for quasi-entity coreference, which list 15 possible subtypes of quasi-identity under 4 high-level types (i.e. Name metonymy, meronymy, class and spatio-temporal function).

Relevant corpus study have been conducted guided by the theory and the typology. Our theoretical analyses are largely inspired and based on their theoretical foundation.

Similar analyses have later been applied to events by Hovy et al. [74]. Besides full coreference (exact coreference), they propose quasi-identity should be establish between event mentions when "most aspects are the same, but some additional information is provided for one or the other that is not shared". They further identify 2 types of quasi-identity between event mentions, namely *membership* and *subevent*. A corpus study is also performed, and later computationally modelled by Araki et al. [2]. The event sequencing work in Chapter 4 also follows this line. The prior work, the interaction between event and entity mentions are not adequately addressed, which is instead the focus of the proposed work.

7.4.2 Script Modeling

One research line relevant to the quasi-identity of events is on script modeling, which is similar to the *subevent* relation defined in [74]. Recent work on script modeling is pioneered by [15, 18, 22]. Rich computational work has then been devoted to this area [7, 16, 19, 21, 29, 76, 77, 85, 106, 107, 116, 118, 122, 123, 124, 137, 138, 139, 144, 145, 157, 160]. The common nature of these work uses the event context to predict other events, which is quite similar to language modelling. The relations between the events can be sometimes considered as quasi-identical.

On the other hand, the facet view can also help fill some gaps in script modeling. As discussed above, the script modeling work often uses the exact argument identity assumption to connect the events. This can potentially be improved by our facet based approach.

7.4.3 Area Similar to Quasi-Identity

There are several specific types of relation between discourse elements being studied. Some of them are very similar to quasi-identity, and can be roughly classified into the quasi-identity typology. It is important to note the connections and differences of these relations and our problem.

Bridging, or associative anaphora refers to linking of distinct entities or events. Linguistic studies of the phenomenon are discussed in Prince [129]. It is also computationally modelled via traditional feature based approaches [72, 125]. Example 7.7 is an excerpt taken from the OntoNotes [73] corpus. The phrases *the windows*, *the carpets* and *walls* are all related to the antecedent *the Polish center* via a part-of relation.

(7.7) . . . as much as possible of the Polish center will be made from aluminum, steel and glass recycled from Warsaw's abundant rubble. . . . The windows will open. The carpets won't be glued down and walls will be coated with non-toxic finishes.

The bridging relation shares some similar relations with quasi-identity, that the expressions are not fully identical. However, quasi-identity relations further require the expressions should be functionally the same given the context, which may not happen in bridging relations.

Metonymy Resolution is another problem that shares some common characteristics with quasi-identity. Metonymy stands for the substitution of a concept with a semantically related one. For example, in "Moscow traded gas and aluminum with Beijing." (example borrowed from [65]), the locations names are used as substitutions of the government. Metonymy resolution studies

the problem of recovering the true facet of the mention, which sometimes require resolving the figurative references. This problem is similar, or can be considered as a sub problem of facet identification.

7.4.4 Context Dependent Modeling

More generally, modeling the facets of discourse elements always require modeling the context. There are rich prior work on context dependent modeling in NLP. To name a few, Gillick et al. [62] introduces a task named context dependent entity type detection. Dasigi et al. [43] determine the granularity of meaning of a word in context, in order to improve prepositional phrase attachment. Peters et al. [121] introduce a new type of deep contextualized word representation to incorporate context information, such as language usage and polysemy. Though the goals of this line of work are slightly different from our purpose, their methods can be inspiring to our problem.

Chapter 8

Proposed Experiments

In this chapter, we propose the experiments to validate the theoretical hypotheses discussed in Chapter 7, and to show their potential impact to some existing NLP problems.

One challenge to model complex semantic phenomenon is data sparsity. Though this is a general problem for NLP, it becomes severe for complex semantic tasks such as coreference and event sequencing: large amount of training data is required for them, but the annotation tasks are also very difficult for human at the same time. For example, the facets of entities and events are not easy to be defined from the first place, which makes manual annotation almost infeasible. In this section, we sketch our proposed solution on how to solve these problems. We try to seek indirect supervision signals, which often come in abundance and is closer to the application end. We also attempt to incorporate existing knowledge sources to simplify the problem.

8.1 Facets Understanding and Linking

In this section, we propose several computational experiments to validate the first hypothesis: *only partial facets of entities and events are being activated given a discourse context*. To achieve this, we first need to identify the facets of the entity and events, and then show that by focusing on some facets, we can improve the performance of some downstream tasks.

8.1.1 Facet Representation

To the best of our knowledge, there is few direct annotation for facets. Instead, we propose to use some indirect methods to represent the facets. At this step, our main goal is to learn a space for facets, where similar facets are close by.

Entity Event Dual Representation

In our previous chapter, we have shown that events are useful for identifying the active facets for the entities. We can collect facets about the entity via events. These facets can be also represented using the events. We call these as "event dual representation" for the facets. For example, 8.1 shows that "*paper*" is burnable (flammable) as indicated by the event "burning". 8.2 shows another facet that "*paper*" can be written on.

(8.1) As others have pointed out, *burning paper* inside can be dangerous.

(8.2) I found that *writing* on *paper* helped me to quit analyzing and create faster.

Such knowledge is in the form of *predicate*-able (e.g. burnable, writable), which can be obtained via raw text documents. We propose to map the *predicate*-able facets of the entities to a low dimensional embedding space, mimicking word embeddings. This is close to the line of work on learning frame embeddings. But one major difference is that we are connecting the frame embedding knowledge to the entities, instead of tying them to the predicates. Further, we are plan to further connect these frame embeddings with some existing background knowledge.

Incorporating Background Knowledge

Mining patterns from raw text arguments is often limited. The distribution learnt about the entities are often skewed. For example, common sense knowledge, which is quite important for reasoning, is rarely mentioned in text. Further, the proposed dual representation only covers sparse predicate related information, we further propose to incorporate other sources of knowledge to this work, with a special focus to common sense. The background knowledge can help form the facets for entities, while the event centric approach will help identify active ones. Learning both knowledge into the same space should help showing the connection between the predicate knowledge and the entity facets.

Specifically, we have investigated and found the following resources to be useful for our purpose.

Wikipedia Sections: Wikipedia articles contain rich information for many real world entities. For example, the entity *paper* has an entry in Wikipedia (https://en.wikipedia.org/wiki/Paper). Each section in the Wikipedia page describes different facets of *paper*, including: *History*, *Early sources of fibre*, *Etymology*, *Papermaking*, *Applications*, *Types, thickness and weight*, *Paper stability*, *Environmental impact*. The content in each section contains rich information around the particular facet. We can utilize the additional entities and events in the sections to get more information of the entity of interest.

ConceptNet: While Wikipedia provides detailed descriptions to entities, there are some knowledge that are more subtle to human. For example, the *paper* entry in Wikipedia does not mention that paper is flammable. Alternatively, we propose to augment the data with ConceptNet, which is a knowledge graph of common sense entities. In ConceptNet, the entry for *paper* (http://www.conceptnet.io/c/en/paper) contains interesting facts. For example, it shows that *paper* can be burnt, cut, written on and more. It also shows that *paper* is related to *writing*, *academic*, *book*, which covers a variety of aspects of *paper*.

Procedural Knowledge Base: There are also rich procedural knowledge datasets available online, such as WikiHow. Such datasets provide rich information on common sense events. This will be another useful source of information to augment event facets.

8.1.2 Facet Identification

This step is closely connected to the facet representation step. In fact, if we can successfully learn some embedding representations for the facets, it will be easier to identify the active facets given

the context. However, since the active facets are mostly related to the context and purpose, the validation tasks will affect the active ones. We now describe several candidate validation tasks to prove utility of the facet representations. Note that we are not trying to perform all three tasks in our thesis, but will focus on whether a task can prove our hypotheses.

Event Coreference

The core reason for active facet hypothesis is actually to solve the "Frame Inconsistences" problem. Event coreference is clearly a great candidate to validate our approach. We plan to improve our event coreference system by incorporating the argument information in the facet form.

Entity and Quasi-Entity Coreference

Since the facet based representation is inspired by quasi-entity coreference. It would also be useful to show that the facet representation can help establish an estimation for quasi-entity coreference. The NIDENT [136] corpus contains a set of 60 documents with quasi-coreference links. These links are assigned with an "identity degree" value from 1 to 3 (with 3 being exact coreference). We plan to investigate how well can our facet based representation approximate these annotations.

Implicit Semantic Role Labeling (SRL)

Directly modeling quasi-entity coreference is informative. However, the NIDENT dataset is small and may be difficult to use even as a test set. We find implicit SRL to be another useful validation task. The implicit SRL task studies the following problem: the argument for a predicate may not appear in the sentence, but it can be found in the same document. The task thus requires a system to find such arguments. Unlike traditional SRL, a system has no access to the syntax information under the setting of implicit SRL. The task can be considered as a "Null coreference problem", where the missing argument in a sentence can be considered as a "Null argument filler", and the task is to find in the document which other argument phrase can be coreferent with it.

There are several dataset available for this task, including the implicit NomBank [61] and the SemEval 2010 SRL task [140]. The identity between a Null argument and the filler argument is closer to our definition of quasi-identity: they are only required to be functionally identical.

8.2 Facets and State Modeling

By modeling state, one could understand the configuration of entities in a document according to the script timeline. We propose to work on one of the following validation task.

Script Induction

Prior work on Script Induction or Event Schema Induction mainly uses the entities as connecting points between the events. States, which are normally latent in text, are actually particularly appropriate as connecting points. Furthermore, the states can also provide explanations for the upcoming events. We propose to use a state-based approach to model script induction.

State Aware Implicit SRL

This experiment is an extension to the implicit SRL problem. The states of the entities are not considered in prior models or our proposed models. However, they may have affected which argument to choose. We may be able to use the same implicit SRL datasets in the prior experiments. However, there may not be many cases need state modeling to determine SRL in such datasets, and there is no dataset specifically annotated for it. However, following prior work [28], it is convenient to use a cloze style task to validate such argument filling problem. We plan to apply our model on state critical documents, such as cooking recipes (the state of the ingredients are important).

Chapter 9

Timeline

- Event Detection (Done)
  - System for TAC-KBP 2015, 2016 [89, 91]
- Event Coreference (Done)
  - System for TAC-KBP 2016 [91]
  - Published at LREC 2014 [90]
- Verb Phrase Ellipsis (Done)
  - Work done at Google
  - Published at NAACL 2016 workshop [92]
- Event Sequencing (Done)
  - Published at COLING 2018 [94]
- Facet Identification:
  - Implicit Argument Modeling (In Progress), Target: NAACL 2019
  - Facet Aware Event Coreference (In Progress), Target: ACL 2019
- State Modeling (Not Started)
  - Application: argument selection or script modeling
  - Target: EMNLP 2019
- Thesis writing: Summer 2019
- Thesis defense: Fall 2019

Bibliography

[1] David Ahn. 2006. The stages of event extraction. *ARTE '06 Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. 15, 16, 17, 18, 20, 21

[2] Jun Araki, Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014. Detecting Subevent Structure for Event Coreference Resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4553–4558, Reykjavik, Iceland. European Language Resources Association (ELRA). 27, 55, 78

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 722–735. 20, 23

[4] A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics. 20, 33

[5] Amit Bagga and Bagga Baldwin. 1999. Cross - Document Event Coreference: Annotations, Experiments, and Observations. In *ACL-99 Workshop on Coreference and Its Applications*. 16, 17

[6] CF Baker, CJ Fillmore, and JB Lowe. 1998. The berkeley framenet project. *Proceeding ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90. 54, 55

[7] Niranjan Balasubramanian, Stephen Soderland, and OE Mausam. 2013. Generating Coherent Event Schemas at Scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 55, 78

[8] Breck Baldwin and Thomas S Morton. 1998. Dynamic Coreference-Based Summarization. *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, pages 1–6. 55

[9] Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 1–34. 53, 57, 60

[10] Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for the Association for Computational Linguistics*, July, pages 1412–1422. 15,

16, 17, 18, 19, 20, 21

[11] Anders Björkelund and Jonas Kuhn. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 47–57. 26, 27, 30

[12] Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, 45(4):463–494. 39, 40, 41, 45, 46

[13] L Breiman. 2001. Random forests. *Machine learning*, pages 5–32. 19

[14] Susan Windisch Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The Rich Event Ontology. In *Proceedings of the Events and Stories in the News Workshop*, pages 87–97. 55

[15] Nathanael Chambers. 2011. *Inducing Event Schemas and Their Participants from Unlabeled Text*. Doctoral, Stanford University. 69, 78

[16] Nathanael Chambers. 2013. Event Schema Induction with a Probabilistic Entity-Driven Model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,*, pages 1797–1807. 78

[17] Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284. 27, 32, 55

[18] Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL '08 Meeting of the Association for Computational Linguistics*, pages 789–797. 27, 35, 36, 55, 60, 78

[19] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610. Association for Computational Linguistics. 78

[20] Nathanael Chambers and Dan Jurafsky. 2010. A Database of Narrative Schemas. In *LREC 2010, Seventh International Conference on Language Resources and Evaluation*. 31

[21] Nathanael Chambers and Dan Jurafsky. 2011. Template-Based Information Extraction without the Templates. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 976–986. 78

[22] Nathanael Chambers and Dan Jurafsky. 2012. Learning the Central Events and Participants in Unlabeled Text. In *International Conference on Machine Learning (ICML 2012)*. 78

[23] Jonathan Chang, Sean Gerrish, Chong Wang, and David M Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*, pages 288—-296. 61

[24] Chen Chen and Vincent Ng. 2013. Chinese Event Coreference Resolution: Understanding the State of the Art. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, October, pages 822–828. 27

[25] Chen Chen and Vincent Ng. 2015. Chinese Event Coreference Resolution : An Unsupervised Probabilistic Model Rivaling Supervised Resolvers. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1097–1107. 27

[26] Zheng Chen and H Ji. 2009. Graph-based event coreference resolution. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, (August):54–57. 16, 17, 27

[27] Zheng Chen, H Ji, and Robert Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, 3, pages 17–22. 17, 19, 27

[28] Pengxiang Cheng and Katrin Erk. 2018. Implicit Argument Prediction with Event Knowledge. In *NAACL 2018*, 2012. 53, 84

[29] JC Kit Cheung, H Poon, and Lucy Vanderwende. 2013. Probabilistic Frame Induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT 2013)*. 27, 36, 64, 78

[30] Nancy Chinchor. 1992. MUC-5 EVALUATION METRIC. In *Proceedings of the 5th Conference on Message Understanding*, pages 69–78. 20, 33

[31] Timothy Chklovski and Patrick Pantel. 2004. VERB OCEAN : Mining the Web for Fine-Grained Semantic Verb Relations. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 33–40. 23

[32] Prafulla Kumar Choubey and Ruihong Huang. 2017. Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events. In *EMNLP 2017*. 72

[33] Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the Most Dominant Event in a News Article by Mining Event Coreference Relations. In *NAACL 2018*. 53, 55, 64

[34] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. 57

[35] Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in NLP (EMNLP 2002)*, July, pages 1–8. 8, 30

[36] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*. 23

[37] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585. 28

[38] Agata Cybulska and Piek Vossen. 2012. Using Semantic Relations to Solve Event Coreference in Text. In *SemRel2012 in conjunction with LREC2012*, pages 60–67. 15, 16, 18, 19, 20, 21

[39] Mary Dalrymple, Stuart M. Shieber, and Fernando C N Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452. 39, 40

[40] Laurence Danlos. 2003. Event coreference between two sentences. *Computing Meaning*, 77:271–288. 17

[41] D Das, Nathan Schneider, Desai Chen, and NA Smith. 2010. Probabilistic frame-semantic parsing. In *In Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies Conference (NAACL 2010)*, volume 3, Los Angeles. 23

[42] Dipanjan Das and NA Smith. 2011. Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, volume 1, pages 1435–1444. xi, 9, 31, 56

[43] Pradeep Dasigi, Waleed Ammar, Chris Dyer, and Eduard Hovy. 2017. Ontology-Aware Token Embeddings for Prepositional Phrase Attachment. In *ACL 2017*. 79

[44] Ofer Dekel, C Manning, and Yoram Singer. 2004. Log-linear models for label ranking. *Advances in neural information processing systems*, 16:497–504. 41, 42

[45] Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. 55, 60

[46] Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint Inference for Event Timeline Construction. *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (July):677–687. 27, 55

[47] M. Dojchinovski, D. Reddy, T. Kliegr, T. Vitvar, and H. Sack. 2016. Crowdsourced corpus with entity salience annotations. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC*, pages 3307–3311. 55

[48] Jesse Dunietz and Daniel Gillick. 2014. A New Entity Salience Task with Millions of Training Examples. In *Proceedings of the European Association for Computational Linguistics*, pages 205–209. 53, 54, 55, 57

[49] Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982. 45

[50] Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Proceedings of the Transactions of the Association for Computational Linguistics*. 26

[51] Aymen Elkhlifi and Rim Faiz. 2009. Automatic annotation approach of events in news articles. *International Journal of Computing & Information Sciences*, 7(1):40–50. 16, 17

[52] Günes Erkan and Dragomir R Radev. 2004. LexRank : Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479. 55

[53] Gilles Fauconnier. 1997. *Mappings In Thought and Language*. Cambridge University Press. 69

[54] Christiane Fellbaum. 1998. WordNet. 9

[55] Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. *Joint Conference on {EMNLP} and {CoNLL-Shared} Task*, pages 41–48. 27

[56] Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM 2010*. 61

[57] Francis Ferraro and Benjamin Van Durme. 2016. A Unified Bayesian Model of Scripts , Frames and Language. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pages 2601–2607. 27, 36

[58] JR Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*. 20

[59] Joseph Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76. 18

[60] Radu Florian, John F Pitrelli, Salim Roukos, and Imed Zitouni. 2010. Improving mention detection robustness to noisy input. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, (October):335–345. 23

[61] M Gerber and J Y Chai. 2010. Beyond NomBank: A Study of Implicit Arguments for Nominal Predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, July, pages 1583–1592. 83

[62] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-Dependent Fine-Grained Entity Type Tagging. Technical report. 79

[63] Kartik Goyal, Sujay Kumar Jauhar, Mrinmaya Sachan, Shashank Srivastava, Huiying Li, and Eduard Hovy. 2013. A Structured Distributional Semantic Model for Event Co-reference. In *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, Bulgaria. 17, 23

[64] Joseph Evans Grimes. 1975. *The Thread of Discourse*. New York. 53, 58

[65] Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. 2017. Vancouver Welcomes You! Minimalist Location Metonymy Resolution. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1248–1259. 78

[66] Mark Hall, Geoffrey Holmes, Bernhard Pfahringer, Ian Witten, Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software : An update The WEKA Data Mining Software : An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18. 19

[67] Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. 55, 58

[68] Daniel Hardt. 1992. An algorithm for VP ellipsis. In *Proceedings of the 30th annual*

meeting on Association for Computational Linguistics, January, pages 9–14. 39, 40, 41

[69] Daniel Hardt. 1997. An empirical approach to VP ellipsis. *Computational Linguistics*, 23(4):525–541. 40

[70] Daniel Hardt. 1998. Improving Ellipsis Resolution with Transformation-Based Learning. *AAAI Fall Symposium*, pages 41–43. 40

[71] Laura Hasler and Constantin Orasan. 2009. Do coreferential arguments make event mentions coreferential? In *Proceedings of DAARC*, pages 151–163. 35, 73

[72] Yufang Hou, Katja Markert, and Michael Strube. 2013. Global Inference for Bridging Anaphora Resolution. *Naacl-Hlt*, (June):907–917. 78

[73] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, June, pages 57–60. 78

[74] Eduard Hovy, T Mitamura, F Verdejo, J Araki, and Andrew Philpot. 2013. Events are Not Simple: Identity, Non-Identity, and Quasi-Identity. In *The 1st Workshop on EVENTS: Definition, Detection, Coreference and Representation, NAACL-HLT 2013 Workshop*, pages 21–28, Atlanta. 2, 15, 18, 78

[75] Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, 35 th Annual Meeting of Assoc. for Computational Linguistics*, pages 75–81, Madrid. 26

[76] Bram Jans, Steven Bethard, I Vulić, and MF Moens. 2012. Skip n-grams and ranking functions for predicting script events. *EACL '12 Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344. 78

[77] Niels Kasch and Tim Oates. 2010. Mining script-like structures from the web. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 34–42. 78

[78] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*. 61

[79] Heeyoung Lee, Yves Peirsman, and Angel Chang. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proc. of the 15th Conference on Computational Natural Language Learning: Shared Task*, June, pages 28–34. xi, 23

[80] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*. 16, 17, 19, 21

[81] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *EMNLP 2017*. 26

[82] Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in Chinese event extraction. In *EMNLP-CoNLL '12*

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, July, pages 1006–1016. 10

[83] Peifeng Li, Qiaoming Zhu, and Xiaoxu Zhu. 2011. A Clustering and Ranking Based Approach for Multi-document Event Fusion. *12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 159–165. 16, 18

[84] Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*. 55

[85] Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing Narrative Event Evolutionary Graph for Script Event Prediction. In *IJCAI 2018*. 78

[86] Linguistic Data Consortium. 2008. ACE (Automatic Content Extraction) English Annotation Guidelines for Entities, Version 6.6 2008.06.13. Technical report. 72, 73

[87] Linguistic Data Consortium. 2015. DEFT Rich ERE Annotation Guidelines: Events v.2.6. Technical report. 72

[88] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331. 58

[89] Zhengzhong Liu, Jun Araki, Dheeru Dua, Teruko Mitamura, and Eduard Hovy. 2015. CMU-LTI at KBP 2015 Event Track. In *KBP TAC 2015*. NIST. 85

[90] Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4539–4544, Reykjavik, Iceland. European Language Resources Association (ELRA). 55, 85

[91] Zhengzhong Liu, Jun Araki, Teruko Mitamura, and Eduard Hovy. 2016. CMU-LTI at KBP 2016 Event Nugget Track. In *TAC 2016*. 11, 85

[92] Zhengzhong Liu, Edgar Edgar Gonzàlez, and Dan Gillick. 2016. Exploring the steps of Verb Phrase Ellipsis. In *NAACL Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*. 85

[93] Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2015. Evaluation Algorithms for Event Nugget Detection : A Pilot Study. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 53–57. 11

[94] Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2018. Graph-Based Decoding for Event Sequencing and Coreference Resolution. In *Proceedings of the 27th International Conference on Computational Linguistics*. 85

[95] Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 90–101. 27, 55

[96] Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint Inference for Event Reference Resolution. In *Proceedings of the 26th International Conference on*

Computational Linguistics. 27

[97] Xiaoqiang Luo. 2005. On coreference resolution performance metrics. *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (October):25–32. 20, 33

[98] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the ACL: System Demonstrations*, pages 55–60. xi, 9, 31, 57

[99] Daniel Marcu. 1999. Discourse Trees are Good Indicators of Importance in Text. *Advances in Automatic Text Summarization*, pages 123–136. 55

[100] Katie McConky, Rakesh Nagi, Moises Sudit, and William Hughes. 2012. Improving event co-reference by context extraction and dynamic feature weighting. *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 38–43. 16, 17

[101] PN Mendes, Max Jakob, Andres Garcia-silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. *Proceedings of the 7th Conference on Semantic Systems*, pages 1–8. 20

[102] Tomas Mikolov, I Sutskever, and Kai Chen. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 58, 61

[103] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2015. Overview of TAC KBP 2015 Event Nugget Track. In *TAC KBP 2015*, pages 1–31. 7, 11, 27, 32, 56

[104] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2017. Overview of TAC-KBP 2016 Event Nugget Track. In *TAC 2016*. 7, 11, 12

[105] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2018. Events Detection, Coreference and Sequencing: What's next? Overview of the TAC KBP 2017 Event Track. In *TAC 2017*, pages 1–42. 7, 25

[106] A Modi and Ivan Titov. 2014. Inducing Neural Models of Script Knowledge. In *Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*. 78

[107] Ashutosh Modi and Ivan Titov. 2013. Learning Semantic Script Knowledge with Event Embeddings. 78

[108] Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48. 55

[109] Martina Naughton. 2009. *Sentence Level Event Detection and Coreference Resolution*. Ph.D. thesis, National University of Ireland, Dublin. 16, 17

[110] Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, July, pages 104–111, Philadelphia. 20, 30

[111] Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation

with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371. 55

[112] Leif Arda Nielsen. 2003. Using Machine Learning techniques for VPE detection. In *Proceedings of Recent Advances in Natural Language Processing*, pages 339–346. 40

[113] Leif Arda Nielsen. 2004. Robust VPE detection using automatically parsed text. In *Proceedings of the Student Workshop, ACL*, pages 31–36. 40

[114] Leif Arda Nielsen. 2004. Using Automatically Parsed Text for Robust VPE detection. In *Proceedings of the fifth Discourse Anaphor and Anaphora Reolution conference (DAARC)*. 40

[115] Leif Arda Nielsen. 2005. *A corpus-based study of Verb Phrase Ellipsis Identification and Resolution*. Doctor of philosopy, King's College London. 39, 40, 41, 45, 46

[116] Simon Ostermann, Michael Roth, Stefan Thater, and Manfred Pinkal. 2017. Aligning Script Events with Narrative Texts. In *Conference: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*. 78

[117] Ted Pedersen and Jason Michelizzi. 2004. WordNet :: Similarity - Measuring the Relatedness of Concepts. *HLT-NAACL–Demonstrations '04 Demonstration Papers at HLT-NAACL 2004*, (July):38–41. 23

[118] Haoruo Peng, Snigdha Chaturvedi, and Dan Roth. 2017. A Joint Model for Semantic Sequences : Frames, Entities, Sentiments. *CoNLL 2017*, (CoNLL):173–181. 78

[119] Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving Hard Coreference Problems. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 809–819, Denver, Colorado. 37, 69, 72

[120] Haoruo Peng, Yangqi Song, and Dan Roth. 2016. Event Detection and Co-reference with Minimal Supervision. In *EMNLP 2016*. 27, 55

[121] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL 2018*. 79

[122] Karl Pichotta. 2016. Learning Statistical Scripts With LSTM Recurrent Neural Networks. In *In Proceedings of the 30th AAAI Conference on Artificial Intelligence*, February. 55, 78

[123] Karl Pichotta and Raymond J. Mooney. 2016. Using Sentence-Level LSTM Language Models for Script Inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 279–289. 27, 36, 78

[124] Karl Pichotta and RJ Mooney. 2013. Statistical Script Learning with Multi-Argument Events. In *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, 2012, pages 220–229, Gothenburg,Sweden. 69, 78

[125] Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pages 143–es. 78

[126] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. 2018. SWAT: A System for Detecting Salient Wikipedia Entities in Texts. 55

[127] Sameer Pradhan, Alessandro Moschitti, and Olga Uryupina. 2012. CoNLL-2012 Shared Task : Modeling Multilingual Unrestricted Coreference in OntoNotes. In *Conll*, June, pages 1–40. 20

[128] Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. *International Conference on Semantic Computing (ICSC 2007)*, pages 446–453. 15, 16

[129] Ellen F Prince. 1981. Toward a taxonomy of given-new information. *Radical pragmatics*, 3(1979):223–255. 78

[130] James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Beth Sundheim, David Day, Lisa Ferro, and Dragomir. 2002. The TIMEBANK Corpus. *Natural Language Processing and Information Systems*, 4592:647–656. 27, 55

[131] Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, August, pages 968–977. 45

[132] Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 1(1). 20, 33

[133] Marta Recasens, Eduard Hovy, and M. Antònia Martí. 2010. A Typology of Near-Identity Relations for Coreference (NIDENT). *7th International Conference on Language Resources and Evaluation (LREC-2010)*, (i):149–156. 63, 72, 73, 77

[134] Marta Recasens, Eduard Hovy, and M. Antonia Marti. 2011. Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138–1152. 2, 15, 69, 70, 71, 72, 77

[135] Marta Recasens, Mc De Marneffe, and Christopher Potts. 2013. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June, pages 627–633. 56

[136] Marta Recasens, M Antònia Martí, and Constantin Orasan. 2012. Annotating Near-Identity from Coreference Disagreements. In *Proceedings of The Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 165–172. 83

[137] Michaela Regneri, A Koller, and M Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988. 78

[138] Michaela Regneri, Alexander Koller, J Ruppenhofer, and Manfred Pinkal. 2011. Learning Script Participants from Unlabeled Data. In *Proceedings of Recent Advances in Natural Language Processing*, pages 463–470. 78

[139] Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script Induction as Language Modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686. 27, 36, 55, 78

[140] Joseph Ruppenhofer, Caroline Sporleder, Roser Morante, Collin F Baker, and Martha Palmer. 2010. SemEval-2010 Task 10: Linking Events and Their Participants in Discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, July, pages 45–50. 83

[141] Evan Sandhaus. 2008. The New York Times Annotated Corpus. 55

[142] S Sangeetha and Michael Arock. 2012. Event Coreference Resolution using Mincut based Graph Clustering. *International Journal of Computing & Information Sciences*, pages 253–260. 16, 17, 27

[143] Roger C Schank and Robert P Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates. 2, 25, 54

[144] Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui, and Computer Science. 2008. Jointly Learning Templates and Slots for Event Schema Induction. 78

[145] Tomohide Shibata and Sadao Kurohashi. 2011. Acquiring Strongly-related Events using Predicate-argument Co-occurring Statistics and Case Frames. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1028–1036. 78

[146] Stuart M. Shieber, Fernando C. N. Pereira, and Mary Dalrymple. 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19(5):527–552. 40

[147] E Skorochod'ko. 1971. Adaptive Method of Automatic Abstracting and Indexing. In *Proceedings of the IFIP Congress 71*. 55

[148] Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 89–98. 73

[149] A Sun, R Grishman, and S Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 521–529. 9

[150] Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2010, pages 1257–1268. 9, 19, 32

[151] Naushad UzZaman. 2012. *Interpreting the Temporal Aspects of Language*. Ph.D. thesis, University of Rochester. 33

[152] Naushad Uzzaman and James F Allen. 2010. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, July, pages 276–283. 27

[153] Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. Semeval-2013 task 1: {T}empeval-3: Evaluating time expressions, events, and temporal relations. In *Second joint conference on lexical and computational semantics (* SEM)*, volume 2, pages 1–9. 2, 33

[154] Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, 66(2):143–160. 56

[155] Piek Vossen and Tommaso Caselli. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Story Lines*, pages 40–49. 53, 55, 57, 60

[156] Bonnie Lynn Webber. 1978. *A Formal Approach to Discourse Anaphora*. Garland Press, New York. 71, 74

[157] Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical Quantized Representations for Script Generation. In *EMNLP 2018*. 78

[158] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard Hovy. 2017. JointSem: Combining Query Entity Linking and Entity based Document Ranking. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*, pages 2391–2394. 59, 61

[159] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling. In *SIGIR 2018*. 53, 55, 60

[160] Wenlin Yao and Ruihong Huang. 2018. Temporal Event Knowledge Acquisition via Identifying Narratives. In *ACL 2018*. 72, 78

[161] Congle Zhang, Stephen Soderland, and Daniel S Weld. 2015. Exploiting Parallel News Streams for Unsupervised Event Extraction. volume 3, pages 117–129. 53, 55