ACE Event Extraction

- The Automatic Content Extraction (ACE) evaluations have included several information extraction tasks, including *entity recognition, value recognition, time recognition, relation extraction*, and *event extraction*.
- The 2005 ACE event extraction task included 8 general event types and 33 event subtypes, with 35 possible event roles.
- An event mention is limited in scope to one sentence. The role fillers for each event mention must be identified within the event sentence.
- Each event can have event role fillers, which are similar to relation arguments. But unlike relations, an event can have multiple fillers (arguments) for each role.

ACE Terminology

- **Entity**: an object that belongs to a semantic category.
- **Entity mention**: a reference to an entity
- **Timex**: a time expression (e.g., day, year, date)
- **Event mention**: a phrase or sentence that describes the occurrence of an event.
- **Event trigger**: the main word that most clearly expresses an occurrence of an event.
- **Event mention arguments (role fillers)**: entity mentions that are involved in an event and their relation to the event.

ACE 2005 Entity Types and Subtypes

Table 1 ACE05 Entity Types and Subtypes

Type	Subtypes
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity[3])	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

An **entity mention** can be a proper name, nominal, or pronoun.

ACE 2005 Event Types and Subtypes

Table 7 ACE05 Event Types and Subtypes

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

ACE 2005 Event Roles (from Liao's dissertation)

Person	Place	Buyer	Seller
Beneficiary	Price	Artifact	Origin
Destination	Giver	Recipient	Money
Org	Agent	Victim	Instrument
Entity	Attacker	Target	Defendant
Adjudicator	Prosecutor	Plaintiff	Crime
Position	Sentence	Vehicle	Time-After
Time-Before	Time-At-Beginning	Time-At-End	Time-Starting
Time-Ending	Time-Holds	Time-Within	

Table 1.2 - 35 Argument roles defined by ACE 2005

ACE Event Extraction Example

Barry Diller on Wednesday quit as chief of Vivendi Universal Entertainment.

Trigger = *quit*

Arguments (Roles)

Person: *Barry Diller*

Organization: *Vivendi Universal Entertainment*

Position: *chief*

Time-within: *Wednesday*

Multiple Triggers Example

Three murders occurred in France today, including the senseless slaying of Bob Cole and the assassination of Joe Westbrook. Bob was on his way home when he was attacked …

Type	Trigger	Place	Victim	Time
DIE	*murder*	*France*	--	*today*
DIE	*slay*	*France*	*Bob Cole*	*today*
DIE	*assassinate*	*France*	*Joe Westbrook*	*today*
ATTACK	*attack*	*France*	*he*	*today*

Sentence-level Event Extraction System

[Grishman, Westbrook, & Meyers, 2005] developed a sentence-level event extraction system for ACE 2005.

- Using training data, patterns are learned from sequences of constituent heads that separate the trigger and arguments for each event mention.
- An **Argument Classifier** is trained with MaxEnt to distinguish arguments from non-arguments with respect to a trigger.
- A **Role Classifier** is trained with MaxEnt to classify arguments with respect to event roles.
- A **Reportable-Event Classifier** is trained with MaxEnt to determine whether a potential trigger, event type, and set of arguments are describing a true event occurrence.

Sentence-level Extraction Procedure

1. Each test document is searched for instances of trigger words that occurred in the training documents.
2. For each trigger word, the patterns learned for that trigger are applied to identify arguments (with role labels) of the trigger.
3. The argument classifier is applied to the remaining entity mentions in the sentence to look for more arguments.
4. If new arguments are found, the role classifier is applied to assign a role to each one.
5. Finally, the reportable-event classifier is applied to the entire context to decide whether it is truly an event.

[Ji & Grishman, 2008] and [Liao & Grishman, 2010] use this system as a component in their document-level event extraction pipelines.

Sentence-Level vs. Document-Level

Sentence-Level Event Extraction:

Traditionally, most systems have extracted information about an event from an isolated sentence. Each sentence in a document is processed independently of the others.

Document-Level Event Extraction:

Recently, researchers have begun to incorporate discourse properties and information about associations across sentences in a document to improve event extraction performance.

Enforcing Consistency

[Gale, Church, & Yarowsky, 1992] identified the widely recognized **One Sense Per Discourse** heuristic**:** within a discourse, instances of the same word have a strong tendency to share the same sense.

[Ji & Grishman] made a related observation that strong sense and event role consistency exists across related documents.

One Trigger Sense Per Cluster: In topically-related documents, event trigger words have a strong tendency to share the same sense.

One Argument Role Per Cluster: in topically-related documents, an entity has a strong tendency to participate in the same event role (argument).

Shared Trigger Sense Example

Test Sentence:

Most US army commanders believe it is critical to pause the breakneck advance towards Baghdad to secure the supply lines and make sure weapons are operable and troops resupplied …

Related Document:

British and US forces report gains in the advance on Baghdad and take control of Umm Qasr, despite a fierce sandstorm which slows another flank.

"*Advance toward*" wasn't in the training data, but "*advance on*" was. Identifying "*advance*" as a Movement_Transport event in a related document suggests the same trigger sense for in the new document.

Correcting Trigger Senses

Test Sentence:

But few at the Kremlin forum suggested that Putin's own standing among voters will be hurt by Russia's apparent diplomacy failures.

Related Document:

Putin boosted ties with the United States by throwing his support behind its war on terrorism after the Sept. 11 attacks, but the Iraq war has hurt the relationship.

"*hurt*" was mistakenly identified as a Life_Injure event in the test sentence because it is a common trigger word for that event type. But *hurt* is never a trigger for Life_Injure events in the related documents, so that trigger label can be discarded.

Shared Argument Role Example

Test Sentence:

Vivendi earlier this week confirmed months of press speculation that it planned to shed its entertainment assets by the end of the year.

Related Documents:

Vivendi has been trying to sell assets to pay off huge debt, estimated at the end of last month at more than $13 billion.

Under the reported plans, Blackstone group would buy Vivendi's theme park division, including Universal Studios Hollywood, …

"Vivendi" was not recognized as a seller in the test document. But it was extracted as a seller in several related documents, which suggests it may be a seller in the test document too.

Empirical Evidence for Cluster Heuristics

Candidate Triggers		Event Type	Perc./Freq. as trigger in ACE training corpora	Perc./Freq. as trigger in test document	Perc./Freq. as trigger in test + related documents
Correct Event Triggers	*advance*	Movement_Transport	31% of 16	50% of 2	88.9% of 27
	fire	Personnel_End-Position	7% of 81	100% of 2	100% of 10
	fire	Conflict_Attack	54% of 81	100% of 3	100% of 19
	replace	Personnel_End-Position	5% of 20	100% of 1	83.3% of 6
	form	Business_Start-Org	12% of 8	100% of 2	100% of 23
	talk	Contact_Meet	59% of 74	100% of 4	100% of 26
Incorrect Event Triggers	*hurt*	Life_Injure	24% of 33	0% of 2	0% of 7
	execution	Life_Die	12% of 8	0% of 4	4% of 24

Cross-Sentence and Cross-Document Extraction

- [Ji & Grishman, 2008] created a system that improved event extraction output by making inferences that enforce consistency across within-document sentences as well as related documents.
- A pipeline architecture gradually refines the output:
  1. *sentence-level event extraction*
  2. *document-level (cross-sentence) inferences*: rules enforce consistency across sentences in the same document.
  3. *cross-document inferences*: an IR system retrieves related documents and rules enforce consistency with these documents.

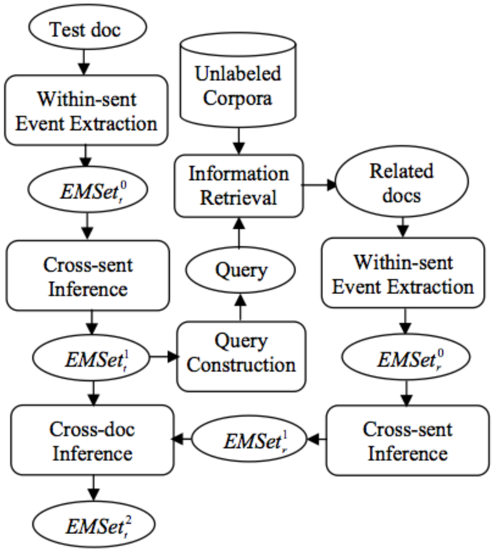

Figure 1. Cross-doc Inference for Event Extraction

Rules for Within-Document Consistency

Rule (1): Remove Triggers and Arguments with Low Local Confidence

If *LConf(trigger, etype)* < δ_1, then delete the whole event mention *EM*;
If *LConf(arg, etype)* < δ_2 or *LConf(arg, etype, role)* < δ_3, then delete *arg*.

Rule (2): Adjust Trigger Classification to Achieve Document-wide Consistency

If *XSent-Trigger-Margin(trigger)* > δ_4, then propagate the most frequent *etype* to all event mentions with *trigger* in the document; and correct roles for corresponding arguments.

Rule (3): Adjust Trigger Identification to Achieve Document-wide Consistency

If *LConf(trigger, etype)* > δ_5, then propagate *etype* to all unlabeled strings *trigger* in the document.

Rule (4): Adjust Argument Identification to Achieve Document-wide Consistency

If *LConf(arg, etype)* > δ_6, then in the document, for each sentence containing an event mention *EM* with *etype*, add any unlabeled mention in that sentence with the same head as *arg* as an argument of *EM* with *role*.

Rules for Cluster-Wide Consistency

Rule (5): Remove Triggers and Arguments with Low Cluster-wide Confidence

If *XDoc-Trigger-Freq (trigger, etype)* < δ_7, then delete *EM*;
If *XDoc-Arg-Freq(arg, etype)* < δ_8 or *XDoc-Role-Freq(arg, etype, role)* < δ_9, then delete *arg*.

Rule (6): Adjust Trigger Classification to Achieve Cluster-wide Consistency

If *XDoc-Trigger-Margin(trigger)* > δ_{10}, then propagate most frequent *etype* to all event mentions with *trigger* in the cluster; and correct roles for corresponding arguments.

Rule (7): Adjust Trigger Identification to Achieve Cluster-wide Consistency

If *XDoc-Trigger-BestFreq (trigger)* > δ_{11}, then propagate *etype* to all unlabeled strings *trigger* in the cluster, override the results of Rule (3) if conflict.

Rule (8): Adjust Argument Classification to Achieve Cluster-wide Consistency

If *XDoc-Role-Margin(arg)* > δ_{12}, then propagate the most frequent *etype* and *role* to all arguments with the same head as *arg* in the entire cluster.

Rule (9): Adjust Argument Identification to Achieve Cluster-wide Consistency

If *XDoc-Role-BestFreq(arg)* > δ_{13}, then in the cluster, for each sentence containing an event mention *EM* with *etype*, add any unlabeled mention in that sentence with the same head as *arg* as an argument of *EM* with *role*.

Experimental Results

Performance / System/Human	Trigger Identification +Classification			Argument Identification			Argument Classification Accuracy	Argument Identification +Classification		
	P	R	F	P	R	F		P	R	F
Within-Sentence IE with Rule (1) (Baseline)	67.6	53.5	59.7	47.8	38.3	42.5	86.0	41.2	32.9	36.6
Cross-sentence Inference	64.3	59.4	61.8	54.6	38.5	45.1	90.2	49.2	34.7	40.7
Cross-sentence+ Cross-doc Inference	**60.2**	**76.4**	**67.3**	**55.7**	**39.5**	**46.2**	**92.1**	**51.3**	**36.4**	**42.6**
Human Annotator1	59.2	59.4	59.3	60.0	69.4	64.4	85.8	51.6	59.5	55.3
Human Annotator2	69.2	75.0	72.0	62.7	85.4	72.3	86.3	54.1	73.7	62.4
Inter-Annotator Agreement	41.9	38.8	40.3	55.2	46.7	50.6	91.7	50.6	42.9	46.4

Cross-Event Inference for Event Extraction

[Liao & Grishman, 2010] observed that certain types of events frequently co-occur, so they incorporated **cross-event information** into their event extraction system.

For example:

S1: He left the company.

S2: He planned to go shopping before heading home.
left ➡ TRANSPORT EVENT

S2: His colleagues threw a retirement party for him.
left ➡ END-POSITION EVENT

Motivating Observations

1. *"Within a document, there is a strong trigger consistency: if one instance of a word triggers an event, other instances of the same word will trigger events of the same type."*

   True > 99.4% of the time in the ACE corpus.

2. *"Normally one entity, if it appears as an argument of multiple events of the same type in a single document, is assigned the same role each time."*

   True > 97% of the time in the ACE corpus.

3. *There are also very strong correlations between different types of events, and the role fillers across different types of events.*

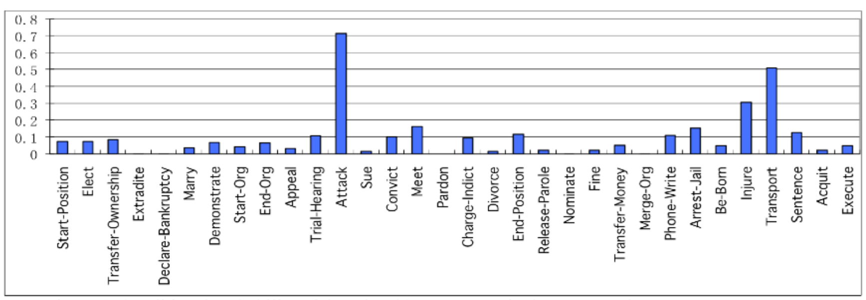

Figure 1. Conditional probability of the other 32 event types in documents where a *Die* event appears

Co-occurring Events

Event	Cond. Prob.
Attack	0.714
Transport	0.507
Injure	0.306
Meet	0.164
Arrest-Jail	0.153
Sentence	0.126
Phone-Write	0.111
End-Position	0.116
Trial-Hearing	0.105
Convict	0.100

Table 3. Events co-occurring with *die* events with conditional probability > 10%

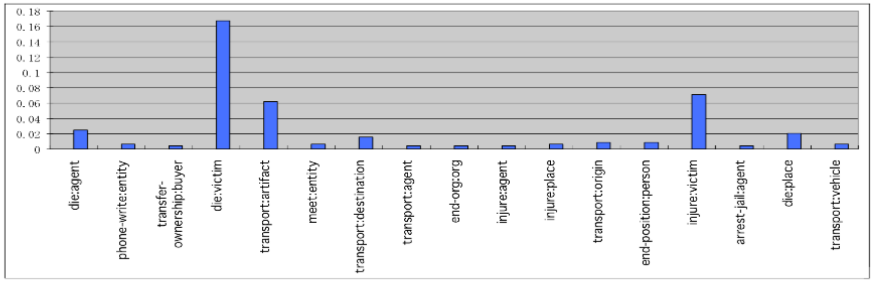

Figure 3. Conditional probability of all possible roles in other event types for entities that are the *Target*s of *Attack* events (roles with conditional probability below 0.002 are omitted)

Two-Pass Approach

- Liao & Grishman adopt a two-pass approach that first identifies the "easy cases" and then uses that knowledge to help identify the harder cases.

*The pro-reform director of Iran's biggest-selling daily newspaper and official organ of Tehran's municipality has **stepped** down following the **appointment** of a conservative … it was **founded** a decade ago … but a conservative city council was **elected** in the February 28 municipal polls … Mahmud Ahmadi-Nejad, reported to be a hardliner among conservatives, was **appointed** mayor on Saturday … **Founded** by former mayor Gholamhossein Karbaschi, Hamshahri …*

*British officials say they believe **Hassan** was a blindfolded woman seen being **shot** in the head by a hooded militant on a video obtained but not aired by the Arab television station Al-Jazeera. **She** would be the first foreign woman to **die** in the wave of kidnappings in Iraq … **she's** been **killed** by (men in pajamas) …*

Confident Event Table

- The sentence-level event extraction system is applied to a document to identify high confidence predictions of event triggers and arguments.
- Event triggers and arguments (roles) that are labeled with high confidence are stored in a **Confident Event Table**.
- A word is assumed to be a trigger for only one type of event, and an entity is assumed to belong to just one role for an event trigger.

  If multiple labels are assigned to the same word/entity, then the highest scoring label is chosen if the difference between scores is large. If the difference between scores is small, then there is a conflict so the information is recorded in a separate **Conflict Table**.

Confident table		
Event type table		
Trigger	**Event Type**	
Met	Meet	
Exploded	Attack	
Went	Transport	
Injured	Injure	
Attacked	Attack	
Died	Die	
Argument role table		
Entity ID	**Event type**	**Role**
0004-T2	Die	Time Within
0004-6	Die	Place
0004-4	Die	Victim
0004-7	Die	Agent
0004-11	Attack	Target
0004-T3	Attack	Time Within
0004-12	Attack	Place
0004-10	Attack	Attacker
Conflict table		
Entity ID	**Event type**	**Roles**
0004-8	Attack	Victim, Agent

Table 4. Example of document-level confident-event table (event type and argument role entries) and conflict table

Document-Level Trigger Classifier

- A MaxEnt classifier is trained to predict whether a word is the trigger of an event, and if so, what type.
- The information in the confident event table is used to create features representing the other event types that have been found in the document.
- Each feature is the conjunction of:
  - the base form of the word
  - for each of the 33 event types, a binary value indicating whether this event type is present elsewhere in the document.

Document-Level Argument (Role) Classifier

- A MaxEnt classifier is trained to predict whether a given mention is an argument of a given event, and if so, what role it plays.
- The information in the confident event table is used to create features for other event roles associated with this entity in the document.
- Each feature is the conjunction of:
  - the type of the given event
  - for each of the other 32 event types, the role of the given entity with respect to that event type (if one exists) or else *null*.

Putting it All Together

- First, the sentence-level event extraction system is applied and high-confidence triggers and arguments are labeled.
- Next, the document-level trigger classifier is applied to all words that do not already have a label. This will often identify some additional event triggers.
- Finally, the document-level argument tagger is applied to all event triggers. Only entity mentions in the same sentence that have not already been assigned a role are considered.

  This tagger can identify arguments for the newly identified triggers as well as new arguments for the previously identified triggers.

Experiments

- The 2005 ACE data set was used for evaluation: 549 training texts, 10 tuning texts, 40 test texts.
- Two baseline systems were evaluated:
  - the sentence-level event extraction component by itself
  - the [Ji & Grishman, 2008]'s approach for cross-sentence and cross-document ("within-event-type") inference rules.
- They also looked at the performance of two human annotators on 28 documents in the test set.

Evaluation Results

performance / system/human	**Trigger classification**			**Argument classification**			**Role classification**		
	P	R	F	P	R	F	P	R	F
Sentence-level baseline system	67.56	53.54	59.74	46.45	37.15	41.29	41.02	32.81	36.46
Within-event-type rules	63.03	59.90	61.43	48.59	46.16	47.35	43.33	41.16	42.21
Cross-event statistical model	68.71	68.87	**68.79**	50.85	49.72	**50.28**	45.06	44.05	**44.55**
Human annotation1	59.2	59.4	59.3	60.0	69.4	64.4	51.6	59.5	55.3
Human annotation2	69.2	75.0	72.0	62.7	85.4	72.3	54.1	73.7	62.4

Table 5. Overall performance on blind test data

Conclusions

- Event extraction is a difficult problem – recall and precision are still only mediocre.
- But a variety of recent systems have shown that considering the entire document, as well as related documents, can be beneficial.
- These systems are still relatively shallow in their understanding of event descriptions. More explicit, richer event representations are probably needed to push performance to a higher level.