# The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation

**George Doddington@NIST, Alexis Mitchell@LDC, Mark Przybocki@NIST, Lance Ramshaw@BBN, Stephanie Strassel@LDC, Ralph Weischedel@BBN**
BBN – Cambridge, MA; LDC – Philadelphia, PA; NIST – Gaithersburg, MD
doddington@nist.gov, amitche0@ldc.upenn.edu, mark.przybocki@nist.gov,
lramshaw@bbn.com, strassel@ldc.upenn.edu, weischedel@bbn.com

**Abstract**

The objective of the ACE program is to develop technology to automatically infer from human language data the *entities* being mentioned, the *relations* among these entities that are directly expressed, and the *events* in which these entities participate. Data sources include audio and image data in addition to pure text, and Arabic and Chinese in addition to English. The effort involves defining the research tasks in detail, collecting and annotating data needed for training, development, and evaluation, and supporting the research with evaluation tools and research workshops. This program began with a pilot study in 1999. The next evaluation is scheduled for September 2004.

## Introduction and Background

Today's global web of electronic information, including most notably the www, provides a resource of unbounded information-bearing potential. But to fully exploit this potential requires the ability to extract content from human language automatically. That is the objective of the ACE program – to develop the capability to extract meaning from multimedia sources. These sources include text, audio and image data.[1] The ACE program is a "technocentric" research effort, meaning that the emphasis is on developing core enabling technologies rather than solving the application needs that motivate the research.

The program began in 1999 with a study intended to identify those key content extraction tasks to serve as the research targets for the remainder of the program. These tasks were identified in general as the extraction of the entities, relations and events being discussed in the language. In general objective, the ACE program is motivated by and addresses the same issues as the MUC program that preceded it (NIST 1999). The ACE program, however, attempts to take the task "off the page" in the sense that the research objectives are defined in terms of the target objects (i.e., the entities, the relations, and the events) rather than in terms of the words in the text. For example, the so-called "named entity" task, as defined in MUC, is to identify those words (on the page) that are names of entities. In ACE, on the other hand, the corresponding task is to identify the entity so named. This is a different task, one that is more abstract and that involves inference more explicitly in producing an answer. In a real sense, the task is to detect things that "aren't there". Reference resolution thus becomes an integral and critical part of solving the problem.

During the period 2000-2001, the ACE effort was devoted solely to *entity* detection and tracking. During the period 2002-2003, *relations* were explored and added. Now, starting in 2004, *events* are being explored and added as the third of the three original tasks.

[1] While the ACE program is directed toward extraction of information from audio and image sources in addition to pure text, the research effort is restricted to information extraction from text. The actual transduction of audio and image data into text is not part of the ACE research effort, although the processing of ASR and OCR output from such transducers is.

## Task Definitions

The Automatic Content Extraction (ACE) program, a new effort to stimulate and benchmark research in information extraction, presents four challenges:

1. *Recognition of entities, not just names.* In the ACE entity detection and tracking (EDT) task, all mentions of an entity, whether a name, a description, or a pronoun, are to be found and collected into equivalence classes based on reference to the same entity. Therefore, practical co-reference resolution is fundamental.
2. *Recognition of relations.* The relation detection and characterization task (RDC) requires detection and characterization of relations between (pairs of) entities. There are five general types of relations, some of which are further sub-divided, yielding a total of 24 types/subtypes of relations:
   - **Role**, the role a person plays in an organization, which can be subtyped as Management, General-Staff, Member, Owner, Founder, Client, Affiliate-Partner, Citizen-Of, or Other,
   - **Part**, i.e., part-whole relationships, subtyped as Subsidiary, Part-Of, or Other,
   - **At**, location relationships, which can be subtyped Located, Based-In, or Residence,
   - **Near**, to identify relative locations and
   - **Social,** subtyped as Parent, Sibling, Spouse, Grandparent, Other-Relative, Other-Personal, Associate, or Other-Professional.
3. *Event extraction.* Though not in any previous ACE evaluation, event detection and characterization is planned for the 2004 evaluation (August-September, 2004). Details of the task definition, annotation guidelines, and scoring are being worked out at the time of writing this paper.
4. *Extraction is measured not merely on text, but also on speech and on OCR input.* Moving beyond name finding is a crucial leap for modalities other than text, since the ability to relate two strings (as in ACE) in very noisy input may degrade much more than

finding strings in isolation (as in named entity recognition.) Furthermore, the lack of case and punctuation, including the lack of sentence boundary markers, poses a challenge to full parsing of speech.

## Data Representation

The ACE research targets, namely *entities*, *relations*, and *events*, are represented in terms of their underlying attributes and constituents. This information is output in XML format, by both LDC annotators and system developers, according to an "apf" document type definition (LDC 2004).

For entities, there is a direct connection with the source data, in terms of the "mentions" of the entity. The identity of the entity is inferred from these mentions and from the entity attributes. The entity attributes are the type (person, organization, geo-political, location, facility, vehicle, weapon) and subtype of the entity, the entity class (specific, generic), and the name(s) of the entity that appear in the source data.

Relations are represented in terms of their attributes and their (two) arguments. The arguments are the ACE entities that are related by the relation. The attributes are the relation type and subtype.

Events are represented in terms of their attributes and their participants. The participants are the ACE entities that participate in the event. ACE events are in essence a generalization of ACE relations. An ACE event can have a number of participants, and each participant is characterized by a role that it plays in the event (agent, object, source, target). Currently the event attributes are event type (destroy, create, transfer, move, interact) and event modality (real, not real).

## Data Annotation

Under the ACE (NIST 2003) and DARPA TIDES (TIDES 2004) Programs, the Linguistic Data Consortium at the University of Pennsylvania develops annotation guidelines, corpora and other linguistic resources to support information extraction research (LDC 2004). LDC's ACE annotators tag broadcast transcripts, newswire and newspaper data in English, Chinese and Arabic, producing both training and test data for common research task evaluations.

### Annotation Tasks

There are three primary ACE annotation tasks corresponding to the three research tasks: Entity Detection and Tracking (EDT), Relation Detection and Characterization (RDC), and Event Detection and Characterization (VDC). A fourth annotation task, Entity Linking (LNK), establishes co-reference between entity mentions.

EDT is the core annotation task, providing the foundation for all remaining tasks. The current ACE task identifies seven types of entities: Person, Organization, Location, Facility, Weapon, Vehicle and Geo-Political Entity (GPEs). Each type is further divided into subtypes (for instance, Organization subtypes include Government, Commercial, Educational, Non-profit, Other). Annotators tag all mentions of each entity within a document, whether named, nominal or pronominal. For every mention, the annotator identifies the maximal extent of the string that represents the entity and labels the head of each mention. Nested mentions are also captured. Each entity is classified according to its type and subtype. Each entity mention is further tagged according to its class – specific, generic, attributive, negatively quantified or underspecified. During the LNK annotation task, annotators review the entire document to group mentions of the same entity together; they also label cases of metonymy, where the name of one entity is used to refer to another entity (or entities) related to it.

During RDC tagging, annotators identify relations that exist between the entities tagged during the EDT task. There are five relation types in ACE: Role, Part, Located, Near, and Social. The Role relation links people to an organization to which they belong, own, founded, or provide some service. The Part relation indicates subset relationships, such as a state to a nation, or a subsidiary to its parent company. The At relation indicates the location of a person or organization at some location. The Near relation indicates the proximity of one location to another. The Social relation links two people in personal, familial or professional relationships. For each type there is a set of possible subtypes. Every relation takes two primary arguments: the two entities that it links. Relations that are supported by explicit textual evidence are distinguished from those that depend on contextual inference on the part of the reader. For explicit relations annotators also identify any temporal attributes. Annotators do not include relationships dependent on a reader's knowledge of the world. All relations are based on textual or contextual evidence found within the scope of the document.

In VDC, annotators identify and characterize five types of events in which EDT entities participate. Targeted types include Interaction, Movement, Transfer, Creation and Destruction events. Annotators tag the textual mention or anchor for each event, and categorize it by type and subtype. They further identify event participants (agent, object, source and target) and attributes (temporal, locative as well as others like instrument or purpose) according to a type-specific template. In future phases of ACE, annotators will identify additional event types as well as characterizing relations between events.

### The Annotation Process

The complexity of ACE annotation requires annotators with a solid background in linguistics, particularly syntax and semantics. The ACE project manager works with language-specific lead annotators to develop and maintain the formal ACE annotation task definitions and guidelines, train annotators and monitor annotation quality. The work environment is designed to encourage regular discussion and "groupthink" among the annotation teams in order to support consistent treatment of the data across team members and languages. Each annotation task requires at least one pass over the data; a second pass reviews the existing annotation for consistency and accuracy. Second passing is typically conducted by more experienced senior annotators.

| Corpus/ Project Phase | Data Amount (words/language) | Tasks | Languages | Evaluation | Availability |
|---|---|---|---|---|---|
| ACE-Pilot | 15K training | entities | English | May, Nov 2000 | Available 2004 |
| ACE-1 | 180K training, 45K evaluation | entities | English | Feb 2000 | Available 2004 |
| ACE-2 | 180K training, 45K dev, 45K eval | entities, relations | English, Chinese | Sept 2000 | LDC Catalog # LDC2003T11 |
| ACE 2003 | 100K training, 50K evaluation | entities, relations | English, Chinese, Arabic | Sept 2003 | LDC Catalog # LDC2004T09 |
| ACE 2004 | 300K training, 50K evaluation | entities, relations, events | English, Chinese, Arabic | Fall 2004 | Under development |

Table 1 List of Corpora developed for and used to support ACE research

In addition to multiple passes over all ACE data, an additional 5% to 10% of the data is completely re-annotated from scratch by different annotators. Results of this dual annotation are compared and discrepancies adjudicated in order to establish inter-annotator agreement scores and identify areas of lingering confusion or inconsistency. Rates of inter-annotator agreement for ACE named entities are comparable to rates shown in previous programs like MUC (NIST 1999). The results for the more complex annotation tasks are somewhat lower. In 2002, inter-annotator agreement for the English EDT task showed an overall value score of 86, whereas English RDC was only 35. In 2003, the overall value score for EDT was 88 in English, 87 in Chinese and 74 in Arabic. RDC agreement had improved to 52 for English and 45 for Chinese. Particular challenges to annotators include the coreference of generic entities and the use of metonymy, characterization of GPEs, distinguishing certain relation types, and identifying implicit vs. explicit relations. After the 2003 Extraction evaluation, LDC worked to redefine ACE annotation tasks with an eye to improving annotator consistency.

### Corpora

As part of the ACE and TIDES information extraction programs, LDC has developed a number of annotated corpora. These corpora all draw on broadcast news, newspaper and newswire data. Sources include data from the Topic Detection and Tracking corpora, Chinese Treebank, Arabic Treebank and other news data. Table 1 summarizes data developed thus far for ACE.

## Evaluation and Scoring

ACE evaluation requires meaningful and helpful scoring of entities, relations and events. Each of these tasks is essentially a detection and recognition task – the target objects are *detected* in the input language stream and the various attributes and characteristics of these objects are *recognized*.

Evaluation requires, as a preliminary step, that a correspondence (mapping) be established between ACE system output objects and reference (true) objects. This mapping is chosen so that the performance measure used for system evaluation is maximized. The performance measure for all three tasks is formulated in terms of a synthetic application *value*, where value is accrued by correctly detecting the target objects and correctly recognizing their attributes, and where value is lost by falsely detecting target objects or incorrectly determining attributes of the target objects. The value formulas are given below:

### Entity scoring

The entity evaluation score is defined to be the sum of the values of all system output entities:

$$EDT\_Value_{sys} = \sum_i value\_of\_sys\_entity_i$$

The value of each system output entity is defined to be the product of an inherent entity value and the sum of the values of the entity's mentions:

$$Value_{sys\_entity} = Entity\_Value(sys\_entity) \cdot \sum_m Mention\_Value(sys\_mention_m)$$

The *Entity_Value* of a system output entity is a function of its type. If the output entity is mapped, then the minimum value for the sys entity and its corresponding ref entity is used. For unmapped system entities, *Entity_Value* is weighted by a false alarm penalty. For mapped output entities, *Entity_Value* is discounted for errors in entity type, subtype and class.

The *Mention_Value* of a system entity mention is also a function of its type. If the mention is mapped, then the minimum value for the sys mention and its corresponding ref mention is used.[2] For mapped system mentions, *Mention_Value* is discounted for errors in mention type, role and style. For unmapped system mentions[3], *Mention_Value* is weighted by a false alarm penalty and a coreference discount[4].

[2] The mapping of system output mentions to reference mentions is chosen so as to maximize the total value of the mentions.

[3] All mentions of a system output entity are unmapped for entities that are themselves unmapped.

[4] The coreference discount is intended to reduce the penalty for mentions that are valid mentions of an entity but that are incorrectly associated at the entity level. This is because such mentions have already been penalized by virtue of not having

For cross-document entities (i.e., for entities that are mentioned in multiple documents), the *Value* of each system entity is accumulated over all documents being evaluated.

## Relation scoring

The relation evaluation score is defined to be the sum of the values of all system output relations:

$$RDC\_Value_{sys} = \sum_i value\_of\_sys\_relation_i$$

The value of each system output relation is defined to be the product of an inherent relation value and the sum of the values of the relation's entity arguments:

$$Value_{sys\_relation} = Relation\_Value(sys\_relation) \cdot \sum_a Argument\_Value(sys\_argument_a)$$

The *Relation_Value* of a system output relation is a function of its type. If the output relation is mapped, then the minimum value for the sys relation and its corresponding ref relation is used. For unmapped system relations, *Relation_Value* is weighted by a false alarm penalty. For mapped output relations, *Relation_Value* is discounted for errors in relation type and subtype.

The *Argument_Value* of a system relation argument is the *Entity_Value* of that entity argument, where the entity argument of the system relation is mapped to the corresponding argument of the reference relation:[5]

$$Argument\_Value = Entity\_Value(sys)$$

Mapped arguments with an "unacceptably" small *Argument_Value* are assigned an *Argument_Value* of zero.[6]

For cross-document relations (i.e., for relations that are mentioned in multiple documents), the *Value* of each system relation is accumulated over all documents being evaluated. Only those argument entity mentions that appear in these documents are used to compute *Argument_Value*, however.[7]

## Event scoring

The event evaluation score is defined to be the sum of the values of all system output events:

$$VDC\_Value_{sys} = \sum_i value\_of\_sys\_event_i$$

The value of each system output event is defined to be the product of an inherent event value and the sum of the values of the event's entity participants:

$$Value_{sys\_event} = Event\_Value(sys\_event) \cdot \sum_p Participant\_Value(sys\_participant_p)$$

The *Event_Value* of a system output event is a function of its type and its modality. If the output event is mapped, then the minimum value for the sys event and its corresponding ref event is used. For unmapped system events, *Event_Value* is weighted by a false alarm penalty. For mapped output events, *Event_Value* is discounted for errors in event type and modality.

The *Participant_Value* of a system event participant is the *Entity_Value* of that entity participant, where the entity participant of the system event is mapped to the corresponding participant of the reference event.[8] For mapped participants, *Participant_Value* is discounted for errors in participant role. For unmapped system participants, *Participant_Value* is weighted by a false alarm penalty.

Participants with zero *Participant_Value* are considered to be unmapped. Further, mapped participants with an "unacceptably" small *Participant_Value* are assigned a *Participant_Value* of zero.[9]

For cross-document events (i.e., for events that are mentioned in multiple documents), the *Value* of each system event is accumulated over all documents in which the event is mentioned. Only those event entity mentions that appear in these documents are used to compute *Participant_Value*, however.[10]

## References


- LDC, 2004, Automatic Content Extraction [www.ldc.upenn.edu/Projects/ACE/]
- NIST, 1999, Message Understanding Conference [www.itl.nist.gov/iaui/894.02/related_projects/muc/]
- NIST, 2003, Automatic Content Extraction [www.nist.gov/speech/tests/ace]
- NIST, 2004, Automatic Content Extraction [www.nist.gov/speech/tests/ace/ace04]
- TIDES, 2004, DARPA Program in Translingual Information Detection Extraction and Summarization [www.darpa.mil/ipto/programs/tides/index.htm]


contributed value to the correct entity which they should have been (but were not) affiliated with.

[5] For symmetric relations, argument order is not fixed. In this case, the order used is the order which maximizes the sum of argument values is the order used.

[6] In order for a system output argument to be reasonably considered to represent its corresponding reference argument it is required to exhibit a reasonable overlap with the reference, in terms of *Entity_Value*. Specifically, the *Entity_Value* of the system output argument (mapped to its corresponding reference argument) is compared to the (self-referenced) *Entity_Value* of the corresponding reference argument. A reasonable overlap exists whenever this ratio is greater than or equal to $\Theta_{Amin}$.

[7] The mapping of system arguments to reference arguments is done globally, however, and considers all mentions of the entity arguments. Thus the mapping, while globally optimum, may be suboptimum when considering only a single document.

[8] The mapping of the participants of a system output event to those of a reference event is done so as to maximize the sum of the participant values.

[9] As with relations, a parameter $\Theta_{Pmin}$ determines the minimum overlap for event participants.

[10] The mapping of system participants to reference participants is done globally, as is done with relations arguments. Thus the mapping, while globally optimum, may be suboptimum when considering only a single document.