Natural Language Engineering (2019), **25**, pp. 677–692
doi:10.1017/S1351324919000512

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Twenty-five years of information extraction

Ralph Grishman

Computer Science Dept., New York University, 60 Fifth Avenue, Room 300, New York NY 10011, USA
Email: grishman@cs.nyu.edu

(Received 21 July 2019; revised 20 August 2019)

Abstract
Information extraction is the process of converting unstructured text into a structured data base containing selected information from the text. It is an essential step in making the information content of the text usable for further processing. In this paper, we describe how information extraction has changed over the past 25 years, moving from hand-coded rules to neural networks, with a few stops on the way. We connect these changes to research advances in NLP and to the evaluations organized by the US Government.

Keywords: Information extraction; Message understanding

1. Introduction

This anniversary issue gives us an opportunity to look back at the past 25 years of *information extraction* (IE)—consider what has changed and *why* these changes have occurred.

First, a few definitions:

IE is the automatic identification and classification of instances of user-specified types of entities, relations, and events from text. The output is structured information (e.g., a database) which can be readily interpreted by other applications. The specification may take the form of examples or verbal descriptions of the information to be extracted. Texts which the user considers equivalent should be mapped to the same output structure.

Although there are exceptions, the information to be extracted is limited to specific individuals and specific events. Generic information, conditional information, statements of knowledge, and beliefs are excluded. These restrictions are intended to make the task more tractable and the output easier to interpret than general language understanding.

This is distinguished from *open IE*, which reduces text to a set of elementary sentences (subject–verb–object triples) for human consumption or search (but does not necessarily involve the collapsing of alternative verbal descriptions to a canonical form).

Although common evaluation corpora now are an integral component of most areas of NLP, shared US Government evaluations have played a particularly large role in the development of IE. Although these evaluations are referred to as "conferences," they involve much more: the specification of a task, the implementation of a system for the task by participants, the release of test data, and its processing and scoring. The past 30 years have seen three major series of evaluations:

MUC (Message Understanding Conference) Begun in 1988 to find a way to evaluate IE, the MUCs established IE as a major application of NLP (Sundheim 1996).

ACE (Automatic Content Extraction) Replaced the filling of one complex and task-specific template with a few dozen more general relations and events. Produced lots of annotated training data, fostering development of supervised methods (Doddington *et al.* 2004).

KBP (Knowledge Base Population) Increased the scale of the data to be processed, with the goal of creating a unified data base connecting tens of thousands of entities with about 40 relations and then answering questions about selected entities. Provided minimal annotated training data, thereby encouraging semi-supervised methods (Ji and Grishman 2011).

The regular evaluations of IE in turn have served as a model for evaluations in many other areas of NLP.

The frequent evaluations (every 1 or 2 years) make it possible to get an accurate picture of the varied approaches to IE over the past 30 years. Each participant in an evaluation was required to provide a (multi-page) system description. The participants included both universities and industry, and were motivated (by the possibilities of Government contracts) to incorporate what they believed was "best practice", not just the most publishable methods.

Since the conferences have all been organized by US Government agencies, it is not surprising that the initial participants were primarily from the US. But as the meetings progressed, they took on a more international character. By 2005, 6 out of 15 groups were non-US. By 2010, only 7 out of 20 KBP participants were from the US and 6 from Europe with the balance widely distributed.

2. Before corpora: rule-based systems

If we turn back the clock to 1994—25 years ago—and the start of this journal, we will find a new NLP technology being introduced to the wider world.

> The information explosion of the last decade has placed increasing demands on processing and analyzing large volumes of online data. In response, the Advanced Research Projects Agency (ARPA) has been supporting research to develop a new technology called IE. IE is a type of document processing which captures and outputs factual information contained within a document. Similar to an information retrieval system, an IE system responds to a user's information need. Whereas an Information Retrieval (IR) system identifies a subset of documents in a large text database or in a library scenario a subset of resources in a library, an IE system identifies a subset of information within a document (Okurowski 1993).

This announcement was based on a series of MUCs which had defined the task of IE and its evaluation. The conference series began in 1988 with invitations to a meeting ("MUC-1") at NOSC (Naval Ocean Systems Command) to discuss how IE might be evaluated. In order to be able to compare systems, there was agreement on the need for a shared template capturing the most important information in a document. Systems would be judged on how accurately they filled these template slots. MUC-2 represented a trial run of such an evaluation; MUC-3 agreed on scoring using recall, precision, and *F* measure. (*F* measure, the harmonic mean of recall and precision, was suggested as the primary metric for assigning a rank to participating systems.)

MUC-1 and MUC-2 both used Navy exercise message traffic ("rainforms" and "opreps") as the corpus. A typical message is as follows:

```
22.1    VISUAL SIGHTING OF PERISCOPE
        FOLLOWED BY ATTACK WITH ASROC AND TORPEDOS.
22.2    SUBMARINE WENT SINKER.
22.3    LOOSEFOOT 722/723 CONTINUE SEARCH.
22.4    FOUR BUOY ROAD PLACED BETWEEN CONSTELLATION AND DATUM.
```

The template planned for MUC-2 is shown in Appendix A.

MUC-3 and 4 used news about terrorism in Latin America (Chinchor *et al.* 1993). A sample message is shown in Appendix B along with one of the filled templates generated from this message.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

As the task got better defined, the number of participants grew. By MUC-5, in 1993 there were 16 participants, evenly divided between universities and companies (primarily defense contractors) (MUC 1993). The MUC tasks were getting larger in other respects as well. Participants in MUC-5 had a choice of two extraction topics (joint ventures or microelectronics) and two languages (English or Japanese). The templates were substantially more complex than in prior years.[a] Following MUC-5, the tasks were simplified to limit the effort required to participate. To emphasize faster development of IE systems for new domains, the time from the release of training material to the actual evaluation was reduced to one month. MUC-6 involved executive succession; MUC-7 involved rocket launches.

Participation in multiple MUCs had led to some convergence of extraction architecture, a long pipeline including some familiar names (Hobbs 1993). It quickly became clear, for example, that a preprocessor to identify names was essential. But there were still basic areas of disagreement.

2.1 To parse or not to parse

One disagreement concerned full-sentence parsing. The job of IE is to analyze the structure of the input text and then, guided by that structure, to generate the specified output relations. The question is how much structure to build. One possible answer is to build a full parse tree, thus defining the role of every word in the sentence. However, this was not so easy to do in 1990. Grammars were constructed by hand and were either too tight (failed to parse 1/3 to 1/2 of sentences) or too loose (produced dozens of parses). The typical solution was to combine a tight grammar with a mechanism to recover a partial parse if no full sentence parse was possible. For MUC-5, half the participants (8) tried to generate full parses of each sentence; it's not always clear how successful they were. Most of these sites cited some linguistic formalism: GB (Government-Binding Theory), LFG (Lexical Functional Grammar), HPSG (Head-driven Phrase Structure Grammar), and CCG (Combinatory Categorial Grammar) were represented at MUC-5.[b]

The primary alternative to a full parse was partial parsing (chunking). This was faster and more reliable, but only generated some of the required structure; semantic patterns had to do the rest. Consider, for example, the "name" event for hiring an executive. It may appear as a simple active sentence, "IBM named Fred president" (pattern *company* `named` *person position*), a passive sentence, "Fred was named president of IBM," a relative clause, "Fred, who was named president of IBM," etc. This was OK for a sentence expressing a single event, but consider a sentence expressing two events:

> Fred, who was named president of IBM last year, suddenly resigned yesterday.

The pattern for the relative clause still matches, but the other event (Fred . . . resigned) is split in two. Creating a full set of patterns to handle all these cases is quite tricky.

The SRI team provided a neat solution. They implemented event rules which were applied nondeterministically and could skip selected constituents. For example, the simple active sentence pattern for "resigned" was extended to

> *person* relativePronoun (nounGroup | other)* verbGroup (nounGroup | other)* `resigned`

which could skip the relative clause, matching both "Fred" and "resigned."[c] Because the patterns are applied nondeterministically, both patterns would match and two events would be reported. The resulting system, FASTUS, was fast and effective (Hobbs *et al.* 1993 1997). The SRI researchers were careful to point out that this solution was suitable for IE but not for a general language understanding task which needs to capture the relation between events.

[a] This was a period of heightened concern in the US regarding commercial competition from Japan, in semiconductors and through joint ventures, and the possibility of using IE may have led to a more realistic and complex task.

[b] There is less emphasis on linguistic formalism in current research, presumably because the major decisions are now made at the time of treebank creation and are not easily revisited.

[c] This is a simplification of the actual SRI pattern.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

At this time, the first corpus trained systems, for part-of-speech tagging, became available (Church 1988). They were significantly more accurate than their rule-based predecessors and began to find some limited use in MUC-5.

2.2 Building a domain model

Once the input data have been syntactically analyzed, we must detect mentions of interest, identify their arguments, and generate the output structure. This was generally achieved through a process of semantic pattern matching, although described in different terms by different sites. The patterns consisted of English words, domain-specific word classes, and syntactic roles. If the system generated a full-sentence parse tree, the pattern had to match a subtree; if the system generated sequences of chunks, the pattern had to match a subsequence.

Studying the source texts and building the domain model remains something of a craft. If the word classes are too general or the patterns too brief, the system will overextract (low precision). More than likely, some patterns will be omitted and the system will underextract.

The one site which explored the possibility of (partially) automating this process was the group from the University of Massachusetts, Amherst, that participated in MUC-4. Most MUC task specifications included a small number (typically 100) of hand-tagged example documents. For MUC-3 and MUC-4, the Government provided these 100 annotated documents, but also over 1000 unlabeled documents, half of which were on the same topic. This provided an opening for a semi-supervised learner. The documents were divided into those that included a relevant event (in this case, a terrorist incident) and those that did not. This was a much smaller job than annotating the documents with their slot fillers. Meanwhile, the corpus was parsed and for every noun phrase in the corpus its immediate context (generally a subject–verb–object structure) was recorded. Then they computed, for every context, the fraction of documents containing that phrase which are relevant to the extraction task. These are ranked, and the top-ranked phrases are collected as promising extraction patterns (Riloff 1996). This set of patterns was as effective at IE as a set of manually selected patterns.

Completion of the Penn TreeBank in the mid-1990s (Marcus *et al.* 1993) led to a series of treebank-trained parsers of increasing accuracy (Collins 1996) and made full-sentence parsing more competitive. This came too late to have a significant influence on the remaining two MUCs—BBN was the only site to incorporate a treebank-based parser (Miller *et al.* 1998)—but it left the field well prepared for supervised methods which required accurate parsers.

2.3 Dividing the task

Up through MUC-5, the only way to participate in an MUC was to create a complete system to fill event templates, which might require several component subsystems. To encourage development of these components, MUC-6 split off three tasks, *named entity tagging*, *coreference*, and *template element*, with separate evaluations (Grishman and Sundheim 1996). These were seen as more general scenario-independent tasks. The original task was dubbed *scenario template*. This brought greater attention to these tasks and favored the rise of NLP specialists who concentrated on one task. Having a separate evaluation also made it feasible to “plug and play.” MUC-7 added a fifth task, the *template relation* task.

The named entity task in particular quickly took on a life of its own. It had lots of things going for it. It was easy to explain. It is not too difficult to implement a system (using hand-coded rules) which exhibits useful performance. It became a separate task at just the time that machine-learning methods were being introduced. And it was useful by itself.

Finally after MUC-7 questions were raised about the value of continued MUCs. Some of the templates were very specific; MUC-5 included a template with more than 40 slots. This led to a lot of work not directly relevant to IE technology. Scores of the top performers seemed to have

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

topped out at $F = 50$–60. A working group was formed which recommended extracting a set of elementary events and their arguments rather than a monolithic template (Hirschman *et al.* 1999). This became a basic theme of the ACE program, which started in 2001.

3. Supervised methods: ACE

3.1 Entity, relation, and event

In ACE, the information in each document is represented by a set of entities, relations, and events. There are seven types of entities, six types of relations, and eight types of events. The types are shown in Appendix C; each type is further divided into subtypes (not shown). Relations are binary; events may have any number of arguments. With minor exceptions, arguments must be entities or temporal expressions (thus excluding relations or events which take other events as arguments). The arguments to a relation or event must appear in the same sentence; this makes annotation more tractable. It also simplifies modeling because it reduces relation tagging to a classification task (classifying all pairs of entities in the same sentence).

The annotated corpora of the ACE evaluations are still widely used as benchmarks for IE. In particular, the three types of data structures produced for the 2005 evaluation are still being used to annotate additional data (Aguilar *et al.* 2014).

Another basic theme was supervised training. It had become clear from the MUCs and contemporaneous NLP research that annotating training data could be an effective way of improving extraction performance. To support such training, a sizable investment was made in corpus annotation. New corpora were released annually. The largest, for ACE 2005, was 300,000 words of English and comparable amounts of Chinese and Arabic.

In addition, to gauge the robustness of the extraction, one release included noisy output from audio transcripts and OCR (optical character recognition), but this was not further pursued.

As we have already noted, in the early 1990s there was a shift in the core NLP tasks to corpus-trained models, initially for part-of-speech tagging and then for parsing, which greatly improved the quality of intermediate results.

We will consider in turn the most popular models for each type of IE structure: named entities, entities, relations, and events.

3.2 Named entity

The general role of this component is to identify and classify all the names in our corpus. More abstractly, its job is to encapsulate all the messy, ad hoc structures which are not part of the core language. In addition to names, this may include addresses, times of day, and chemical formulas (Nadeau and Sekine 2007).

This is essentially a sequence labeling problem and is typically solved by an MEMM (Maximum Entropy Markov Model) or a CRF (Conditional Random Field) at the token level (Nadeau and Sekine 2007). There is a small benefit from taking into account global features which capture name consistency across documents: if the same name appears in two documents, we favor the analysis which assigns the two instances the same name type (Finkel *et al.* 2005). A lot of features are required to classify names not seen in training—primarily shape, prefixes, and suffixes. In some of the top systems, this feature-based approach has been replaced by a system which operates dual sequence models, one at the token level, one at the character level (Klein *et al.* 2003).

3.3 Entities

Entity generation will typically operate on parser output. It has two principal functions: grouping together coreferential phrases and assigning each group a semantic type. ACE has seven entity

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

semantic types, shown in Appendix C. Groups not in one of these seven types are dropped. What remains is a set of entities, each consisting of a set of entity mentions.

Several types of models have been used for coreference, principally mention-mention models (which first classify each pair of entity mentions as to the probability of coreference and then resolves conflicts) and mention-entity models (which make a single pass over a document, processing entity mentions in text order, either assigning the mention to a previously created entity or constructing a new entity) (Ng 2017).

3.4 Relations

As we noted earlier, because relations are between pairs of entities in the same sentence, it is possible to treat relation tagging as a classification problem, classifying each pair as a relation type or NONE. Extensive studies were made using maximum entropy methods and trying a wide variety of features, including words, entity types, and dependency relations (Kambhatla 2004; Jiang and Zhai 2007). Kernel methods have also successfully been used (Zhao and Grishman 2005).

3.5 Events

A proper treatment of events is more challenging because it involves the interaction of the trigger (the principal word defining the event) and multiple arguments. It is consequently a structured prediction task. The simplest solution is to decide first on the type of event, if any, and then to analyze the arguments (Ahn 2006). This, however, loses considerable accuracy because for many common verbs their meaning depends on the arguments it takes. For example, firing a person is a different type of event than firing a rocket.

A better solution is to use joint inference: optimize for a combination of label choices if these choices interact. Besides the interaction of event type with event arguments just noted, there are interactions between the types of adjacent events (attacks often co-occur with deaths) (Li *et al.* 2013).

Event extraction is followed by event coreference, whose role is to identify multiple mentions of the same event. As was the case for entity coreference, there are several viable strategies, including mention-pair models and mention-ranking models (Lu and Ng 2018). These models rely on the argument structures of the mentions: they classify a pair of event mentions as potentially coreferential if the event types are consistent and the argument values are compatible. Some examples of compatible arguments can be learned through bootstrapping, but performance is modest (Huang *et al.* 2019). The problem in part is that many cases of event coreference are complicated, involving containment or partial overlap.

4. Semi-supervised methods

ACE was a success in terms of producing annotated corpora and research results, but there were issues it did not address. In particular, it treated documents separately, whereas many realistic tasks involved large numbers of interrelated documents. Information about an individual may need to be pieced together from several documents. To address these questions, NIST (the US National Institute of Standards and Technology) organized the annual "Text Analysis Conference" and its central task, "Knowledge Base Population" (KBP) (Ji and Grishman 2011). Starting in 2009, the KBP task added additional components year by year. We will describe the "Cold Start" variant as of 2017, when the data sets were largely complete.

Participants were given a large collection of unannotated documents, a mix of newspaper articles and blogs, two to four million documents in each of English, Chinese, and Spanish. A small

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

portion of these, 30,000 documents in each language, served as the test corpus; sites were expected to build a graph in which each node represented an individual, organization, GPE (Geo-Political Entity), location, or facility mentioned in the test collection. Associated with each type of node were a set of properties. whose value could be a number, a date, a string, or another node in the network. For example, a person node would have an age property whose value is an integer and whose city_of_birth was a GPE node.

In addition, sites had to link the entities to the arguments of events appearing in the test collection.

Compared to ACE, the test corpora were about two orders of magnitude larger. At this scale, complete manual annotation of the test corpus was not feasible. Scoring was done by sampling: NIST selected some names mentioned in the test corpus and checked whether (1) the system had created a node for this name and (2) the node had the desired property. Training documents for the various annotation tasks were minimal—small samples the first year a task was run, augmented in subsequent years by the annotations required for scoring.

The large volumes of unannotated data and the lack of annotated training encouraged experimentation with semi-supervised methods—learning from partially labeled data. Most direct was the generalization of the earlier work in MUC-4 to bootstrapping, an iterative strategy starting from a small labeled seed. Bootstrapping was successfully applied to scenario template (Yangarber *et al.* 2000), named entities (Collins and Singer 1999), and relations (Agichtein and Gravano 2000). However, success was not always assured; adding an incorrect element might lead the bootstrapping badly astray.

Participants were also provided with a large data base, BaseKB. This enabled researchers to explore an approach to training a relation classifier termed *distant supervision* (Mintz *et al.* 2009). The basic idea of distant supervision is to convert an existing set of facts into an annotated corpus and then use the annotated corpus to train a classifier in the usual way. Suppose we have a database with a relation R consisting of pairs $< x_1, y_1 >, < x_2, y_2 >, \ldots$, and that some of these pairs appear in the corpus separated by word sequences w_i. We will annotate every sequence w_i as expressing relation R.

The basic model makes strong assumptions which are not satisfied by realistic data. It assumes that if a pair $< x_i, y_i >$ matches a sentence in the corpus, then that sentence expresses relation R. Violations of this assumption lead to a noisy annotated corpus, with many false positives and false negatives. An alternative MIML (MultiInstance MultiLabel) model requires only that at least one instance of the pair represent a relation and that the pair may represent more than one relation label. This model leads on average to cleaner annotations (Surdeanu *et al.* 2012). Further improvements can be made by combining the distant supervision with some manually annotated data (Pershina *et al.* 2014).

More radical approaches are also being tried, including few-shot methods and even zero-shot methods. These address the situation where you have an event extractor which can recognize N event types and now want to add the ability to recognize an N + first event type. In a few-shot method, a small amount of training data is provided; in a zero-shot method, no additional training data is provided. Huang *et al.* (2018) propose to ground the event types and event instances in a shared semantic space based on the arguments to the event and then, given a new event instance, assign it to the closest type. Levy *et al.* (2017) convert a relation into a set of questions and then rely on a reading comprehension system to answer these questions.

Whether distant supervision can outperform hand-built patterns or supervised training depends on several factors. Preparing patterns by hand requires considerable skill and insight but may yield a relatively clean (high precision) system. The preparation of an annotated corpus may require less skill but more time. Distant supervision requires the least labor but may produce the noisiest model. Most likely the best method will involve some combination of these approaches.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

5. Deep learning

Advances in deep learning (multilayer neural networks) have had a dramatic effect on all of NLP over the past few years; IE was no exception.

Neural networks provide a major advantage over the trainable models which preceded them (primarily maximum entropy models): given sufficient training data and time, they can capture arbitrary functions of their inputs. That means that they do not require manual feature engineering. On the other hand, the time factor may be significant; citing training times of one or two weeks is not unusual.

The most widespread change brought about by neural networks was the way in which words are represented. Although prior models had made some use of smoothing lexical dependencies, words were generally treated as discrete symbols. If a vector representation were required it would take the form of a sparse 1-hot vector. Practical neural networks, however, required a representation using continuous-valued, low-dimension vectors. In effect, each word is represented by a point in d-space, termed its word embedding. Several methods were developed which captured the semantic properties of the vocabulary, in particular that words which were semantically similar would appear close-by in d-space.

Wt note here some aspects of the deep learning IE models. The primary network types currently in use are CNNs (Convolutional Neural Networks) and RNNs (Recurrent Neural Networks) using LSTMs (Long Short-Term Memories) (Yin *et al.* 2017).

5.0.0.1 Named entities. The best named entity performance is currently obtained by combining a dual token/character model with contextualized word embeddings (Akbik *et al.* 2019) Performance on the standard test set (the Reuters newswire used by CONLL for the 2003 evaluation) has improved from an F measure of 89 in 2003 to an F of 93 (Li *et al.* 2018).

5.0.0.2 Relations. CNNs offer a particularly simple network structure but the convolution operates within a fixed window size, which may limit the ability to capture dependencies spanning the entire sentence. ACE relations are mostly realized at close range, with the entities separated by fewer than four words. This makes it reasonable to implement relation extraction using a CNN; Nguyen and Grishman (2015) reported good results with windows of two, three, and four tokens.

5.0.0.3 Events. As noted above, event extraction can involve multiple interactions which may benefit from joint inference. In a neural network, these interactions can be captured directly through a set of "memory matrices" whose values are assigned as part of the network training and then used for event trigger and argument prediction (Nguyen *et al.* 2016).

Event extraction is in substantial part a matter of word sense disambiguation. But until recently each word was assigned a single word embedding and so did not capture sense distinctions. Contextualized word embeddings relax that constraint, making the embedding dependent on context. Using contextualized word embeddings on the ACE corpus improves event classification by about two points *F*-measure (Lu and Nguyen 2018).

6. User-generated media

Another significant addition of the last few years was the processing of user-generated data. Twitter was founded in 2006; currently about 500,000,000 tweets are sent each day. Automatically monitored tweets provide a source of current activities second to none, so they have become a target for NLP developers (Panem *et al.* 2014). There is now a annual workshop on the analysis of such informal communication, WNUT (Workshop on Noisy User-Generated Text, web site `http://noisy-text.github.io/`).

But the tweets are quite different from the well-edited texts of newswires which had been the target of most NLP. The tweets may contain many variant spellings, little or no punctuation, and newly coined terms. In consequence, taggers which were trained on edited text performed poorly on tweets (e.g., a top-ranked named entity tagger which obtained an *F* score over 90% on the standard Reuters test corpus obtained an *F* score of about 40% on tweet corpora).

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

The WNUT workshops include an annual multi-site evaluation, but the performance of these tweet-optimized systems was not much better; top performance in the 2016 evaluation was F = 52% (Strauss *et al.* 2016). Generally the taggers used similar designs to those described above, principally CRFs and RNNs built using LSTMs. Because individual tweets provide much less context, tweet taggers must rely more on name lists (e.g., gazetters). Taking advantage of global consistency—a preference for assigning the same token the same tag in different tweets—is also important (Ritter *et al.* 2011; Liu *et al.* 2011; Cherry and Guo 2015). (As we noted earlier, global consistency also plays a role, but a smaller one, in tagging edited text.)

7. Evaluation

At first glance, IE evaluation seems rather straightforward. We agreed already at MUC-3 to score using recall, precision, and F measure. We prepare a key and compare it to the IE system's response.

$$recall = \frac{\text{number of slots correctly filled in system response}}{\text{number of slots filled in key}}$$

$$precision = \frac{\text{number of slots correctly filled in system response}}{\text{number of slots filled in system response}}$$

and then compute the F-measure using

$$1/F = \frac{1/recall + 1/precision}{2}$$

It quickly became clear that things would not be so simple. Systems were supposed to generate one template per event; if a document reported two events, two templates should be filled. However, the system response did not explicitly specify how to pair up the templates in the key and response. To address this issue, possible alignments of key and response templates were generated and scored, and the maximum score was reported (MUC no date). A similar problem arose at a smaller scale if there were multiple participants in an event. In general this recall/precision model provided satisfactory and intuitive scores when new tasks were added to MUC. The one exception was the coreference task. One scoring scheme was originally designed and an elegant alternative was proposed at the MUC conference, but neither seemed intuitive.[d] To this day there are disagreements regarding coreference scoring metrics (Luo 2005).

When MUC was divided into four and later five tasks, each was given its own scoring metric, which made sense since each task might be used independently. The ACE evaluation, in contrast, was based on a set of parallel *cost models* for entities, relations, and events. Each model combined detection, classification, clustering (i.e., coreference), and additional features. The official score ("ACE value") was based on all these factors, suitably weighted. A positive value is assigned to each element correctly recognized and a false alarm penalty is charged for each incorrect output. The score could be negative if the number of errors exceeded the number of correctly identified elements (Doddington *et al.* 2004). This is a standard ROC (Receiver Operating Characteristic) model but was not intuitive to the participants; in consequence, it was used for formal Government reports but little used in the published literature.[e]

In place of the cost model, most researchers report recall/precision scores for relations and events. These scores are highly dependent on the accuracy of entity extraction since only entities can serve as arguments of relations and events. To isolate improvements to relation and event extraction, most researchers assume that the relation or event extractor is provided with perfect

[d] By *intuitive* we mean that the ranking of scores generally corresponded to peoples notion of better output.

[e] Possibly also because the raw value scores were so low for events—below 15%—and participants felt embarrassed to report such a score.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

information about entities. This has the benefit of producing higher (more optimistic) scores than running a real entity extractor.

With the shift to deep-learning taggers which are capable of representation learning, some researchers now assume the relation tagger has minimal information regarding entities—only their position in the sentence, not their semantic type. These shifts—reflecting changing research goals—must be taken into account when comparing tagger performance.

8. Looking ahead

We have briefly described the wide range of approaches that have been developed over the past 25 years for building IE systems, and the gradual rise in task performance which has accompanied the introduction of these approaches. The result is a growing set of applications in finance (Ding *et al.* 2015), medicine (Wang *et al.* 2018), and science (Peters *et al.* 2014). Still, performance (*F* score) after more than 25 years of development has only advanced from the low 60s to the low 70s on standard event classification benchmarks, and there are serious obstacles to be faced in further improving the scores. What are our prospects?

(1) In some regards, the standard benchmarks (drawn from newswires and blogs) are particularly difficult because the range of topics is so broad, increasing the risk of event misclassification Most applications involve a narrower range of topics and so yield higher performance than the benchmarks.

(2) There will be errors and uncertainties in the human annotations which limit the score we can get. This applies even to texts carefully prepared using dual annotation and reconciliation, such as ACE corpora. Annotating relations requires identifying two endpoints which are easily missed. Relatively abstract categories will lead to uncertainties in classification for both relations and events (Min and Grishman 2012). We should embrace this vagueness as part of the power of natural language and take account of it in our evaluations.

(3) There will be examples which require world knowledge and inference. For instance, the ACE events include a *phone* event (a subtype of *contact*). Given the sentence “Fred phoned Jim and he later returned the call.” the system must be able to infer that Jim later called Fred. Handling such cases properly may require a deeper modeling of the events. This is much more feasible in a narrow domain.

(4) Insufficient training data. We expect that we would get several percent improvement in event extraction just by doubling the amount of ACE training data. But “just” may not be an appropriate word when the data were a major government investment. Going forward, we could not afford similar investment for everyone who wants an IE system of their own. Here we may be saved by semi-supervised or unsupervised methods. At a minimum the unsupervised systems could provide cores of relation and event types, which then can be extended and adjusted for particular users, using some form of domain adaptation.

(5) Pipeline problems. IE remains a multi-stage process where earlier stages may introduce errors which are magnified by later stages. Joint inference strategies can reduce this effect.

And we should keep in mind that deep learning is still a young technology from which we can expect continuing improvements in machine learning just as the advent of Bidirectional Encoder Representations from Transformers (BERT) and contextualized embeddings has given many systems a boost of late (Devlin *et al.* 2018). So our prospects for continued improvement seem pretty good.

As performance improves, the number of applications which become commercially viable will continue to grow. To maintain market share for their platforms, every one of the “tech giants” (along with multiple start-ups) is now counted on to provide an NLP API including all of the elements of the pipeline, and to update it steadily, bringing state-of-the-art NLP components much closer to IE applications. In this market-driven environment there may be less demand for the Government to guide research by funding fresh evaluations.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

Acknowledgements. This work was supported in part by DARPA/I2O and US Army Research Office Contract No. W911NF-18-C-0003 under the World Modelers program. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Department of Defense or the US Government. This document does not contain technology or technical data controlled under either the US International Traffic in Arms Regulations or the US Export Administration Regulations. The author wishes to thank the reviewers for their suggestions regarding topics to include in the paper.

Note. MUC 3-7 proceedings are available through the ACL Anthology at https://www.aclweb.org/anthology/

References

Agichtein E. and Gravano L. (2000). Snowball: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries, DL '00*. New York, NY, USA: ACM, pp. 85–94.

Aguilar J., Beller C., McNamee P., Van Durme B., Strassel S., Song Z. and Ellis J. (2014). A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation, Baltimore, MD, USA*. Association for Computational Linguistics, pp. 45–53.

Ahn D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events, ACL*, pp. 1–8.

Akbik A., Bergmann T. and Vollgraf R. (2019). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, USA*. Association for Computational Linguistics, pp. 724–728.

Cherry C. and Guo H. (2015). The unreasonable effectiveness of word representations for twitter named entity recognition. In *HLT-NAACL, ACL*.

Chinchor N., Hirschman L. and Lewis D.D. (1993). Evaluating message understanding systems: an analysis of the third Message Understanding Conference (MUC-3). *Computational Linguistics* **19**(3), 409–450.

Church K.W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing, Austin, Texas, USA*. Association for Computational Linguistics, pp. 136–143.

Collins M. and Singer Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, ACL*.

Collins M.J. (1996). A new statistical parser based on bigram lexical dependencies. In *34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, California, USA*. Association for Computational Linguistics, pp. 184–191.

Devlin J., Chang M., Lee K. and Toutanova K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ding X., Zhang Y., Liu T. and Duan J. (2015). Deep learning for event-driven stock prediction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), AAAI*.

Doddington G., Mitchell A., Przybocki M., Ramshaw L., Strassel S. and Weischedel R. (2004). The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC '04), Lisbon, Portugal*. European Language Resources Association (ELRA).

Finkel J.R., Grenager T. and Manning C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05), Ann Arbor, Michigan, USA*. Association for Computational Linguistics, pp. 363–370.

Grishman R. and Sundheim B. (1996). Message Understanding Conference- 6: a brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Hirschman L., Robinson P., Ferro L., Chinchor N., Brown E., Grishman R. and Sundheim B. (1999). Hub-4 Event'99 general guidelines and templettes. In *Broadcast News Workshop '99 Proceedings*. Morgan Kaufman.

Hobbs J.R. (1993). The generic information extraction system. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, Morgan Kaufmann, August 25–27, 1993*.

Hobbs J.R., Appelt D., Bear J., Israel D., Kameyalna M. and Tyson M. (1993). FASTUS: a system for extracting information from text. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, Morgan Kaufmann, March 21–24, 1993*.

Hobbs J.R., Appelt D.E., Bear J., Israel D.J., Kameyama M., Stickel M.E. and Tyson M. (1997). FASTUS: a cascaded finite-state transducer for extracting information from natural-language text. *CoRR*, cmp-lg/9705013.

Huang L., Ji H., Cho K., Dagan I., Riedel S. and Voss C. (2018). Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 2160–2170.

Huang Y.J., Lu J., Kurohashi S. and Ng V. (2019). Improving event coreference resolution by learning argument compatibility from unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, USA*. Association for Computational Linguistics, pp. 785–795.

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

Ji H. and Grishman R. (2011). Knowledge base population: successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA*. Association for Computational Linguistics, pp. 1148–1158.

Jiang J. and Zhai C. (2007). A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, Rochester, New York, USA*. Association for Computational Linguistics, pp. 113–120.

Kambhatla N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions, Barcelona, Spain*. Association for Computational Linguistics, pp. 178–181.

Klein D., Smarr J., Nguyen H. and Manning C.D. (2003). Named entity recognition with character-level models. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 180–183.

Levy O., Seo M., Choi E. and Zettlemoyer L. (2017). Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada*. Association for Computational Linguistics, pp. 333–342.

Li J., Sun A., Han J. and Li C. (2018). A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

Li Q., Ji H. and Huang L. (2013). Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria*. Association for Computational Linguistics, pp. 73–82.

Liu X., Zhang S., Wei F. and Zhou M.T. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL*.

Lu J. and Ng V. (2018). Event coreference resolution: a survey of two decades of research. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18*. AAAI Press, pp. 5479–5486.

Lu W. and Nguyen T.H. (2018). Similar but not the same: word sense disambiguation improves event detection via neural representation matching. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium*. Association for Computational Linguistics, pp. 4822–4828.

Luo X. (2005). On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*. Association for Computational Linguistics, pp. 25–32.

Marcus M.P., Santorini B. and Marcinkiewicz M.A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* **19**(2), 313–330.

Miller S., Crystal M., Fox H., Ramshaw L., Schwartz R., Stone R., Weischedel R. and The Annotation Group (1998). BBN: description of the SIFT system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29–May 1, 1998*.

Min B. and Grishman R. (2012). Compensating for annotation errors in training a relation extractor. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France*. Association for Computational Linguistics, pp. 194–203.

Mintz M., Bills S., Snow R. and Jurafsky D. (2009). Distant supervision for relation extraction without labeled data. In *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, Suntec, Singapore*. Association for Computational Linguistics, pp. 1003–1011.

MUC (no date). The message understanding conference scoring software user's manual. Available at https://www-nlpir.nist.gov/related_projects/muc/muc_sw/muc_sw_manual.html

MUC (1991). Appendix H: Text and answer key templates for TST1-MUC3-0099. In *THIRD MESSAGE UNDERSTANDING CONFERENCE (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21–23, 1991*.

MUC (1993). *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25–27, 1993*.

Nadeau D. and Sekine S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes* **30**(1), 3–26.

Ng V. (2017). Machine learning for entity coreference resolution: a retrospective look at two decades of research. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI*, pp. 4877–4884.

Nguyen T.H., Cho K. and Grishman R. (2016). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California*. Association for Computational Linguistics, pp. 300–309.

Nguyen T.H. and Grishman R. (2015). Relation extraction: perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, Colorado, USA*. Association for Computational Linguistics, pp. 39–48.

Okurowski M.E. (1993). Information extraction overview. In *TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19–23, 1993, Fredericksburg, Virginia, USA*. Association for Computational Linguistics, pp. 117–121.

Panem S., Gupta M. and Varma V. (2014). Structured information extraction from natural disaster events on twitter. Available at https://www.microsoft.com/en-us/research/publication/structured-information-extraction-from-natural-disaster-events-on-twitter/

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

Pershina M., Min B., Xu W. and Grishman R. (2014). Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, Maryland, USA*. Association for Computational Linguistics, pp. 732–738.

Peters S.E., Zhang C., Livny M. and R.C. (2014). A machine reading system for assembling synthetic paleontological databases. *PLOS*. Available at https://doi.org/10.1371/journal.pone.0113523

Riloff E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA*, AAAI, *August 4–8, 1996*, Vol. 2, pp. 1044–1049.

Ritter A., Clark S., Mausam and Etzioni O. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL*, pp. 1524–1534.

Strauss B., Toma B., Ritter A., de Marneffe M.-C. and Xu W. (2016). Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT), Osaka, Japan*. The COLING 2016 Organizing Committee, pp. 138–144.

Sundheim B.M. (1996). The message understanding conferences. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6–8, 1996, Vienna, Virginia, USA*. Association for Computational Linguistics, pp. 35–37.

Surdeanu M., Tibshirani J., Nallapati R. and Manning C.D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea*. Association for Computational Linguistics, pp. 455–465.

Wang Y., Wang L., Rastegar-Mojarad M., Moon S., Shen F., Afzal N., Liu S., Zeng Y., Mehrabi S., Sohn S. and Liu H. (2018). Clinical information extraction applications: a literature review. *Journal of Biomedical Informatics* **77**, 34–49.

Yangarber R., Grishman R., Tapanainen P. and Huttunen S. (2000). Automatic acquisition of domain knowledge for information extraction. In *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31–August 4, 2000*. Saarbrücken, Germany: Universität des Saarlandes, pp. 940–946.

Yin W., Kann K., Yu M. and Schütze H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv.1702.01923v1. 17 Feb 2017.

Zhao S. and Grishman R. (2005). Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05), Ann Arbor, Michigan*. Association for Computational Linguistics, pp. 419–426.

Appendix A. MUC-2 template

MUC-1 and 2 involved Navy messages. MUC-1 was exploratory and did not involve a shared template. The first shared template, developed for MUC-2, is shown here (Sundheim 1996).

```
MESSAGE ID

EVENT: HIGHEST LEVEL OF ACTION  DETECT, TRACK, TARGET,
                                HARASS, ATTACK, OTHER
FORCE INITIATING EVENT:         FRIENDLY, HOSTILE, NO DATA
CATEGORY(S) OF EVENT AGENT(S):  AIR, SURF, SUB, NO DATA
CATEGORY(S) OF EVENT OBJECT(S): AIR, SURF, SUB, LAND, NO DATA
ID(S) OF 0-TH LEVEL AGENT(S):
ID(S) OF 0-TH LEVEL OBJECT(S):
INSTRUMENT(S) OF 0-TH AGENT(S):
LOC OF OBJECT(S) AT EVENT TIME:
TIME(S) OF EVENT:
RESULT(S) OF EVENT:             1. RESPONSE BY OPPOSING FORCE
                                2. HOLDING CONTACT, LOST CONTACT
                                3. CONTINUING TO TRACK,
                                   STOPPED TRACKING
                                4. HOLDING TARGET, LOST TARGET
                                5. (NO) DAMAGE OR LOSS TO AGENT,
                                   (NO) DAMAGE OR LOSS TO OBJECT -
                                6. else, NO DATA
```

Some of the slots in the template are multiple choices, such as the FORCE INITIATING EVENT slot; alternative fills are separated by commas. Other slots, such as the ID slots, prefer to be filled with specific vessel IDs, locations and times when those are available.

Appendix B. Sample message and template for MUC-3

B.1 Message

```
TST1-MUC3-0099

   LIMA, 25 OCT 89 (EFE) -- [TEXT] POLICE HAVE REPORTED THAT
TERRORISTS TONIGHT BOMBED THE EMBASSIES OF THE PRC AND THE SOVIET
UNION.  THE BOMBS CAUSED DAMAGE BUT NO INJURIES.

   A CAR-BOMB EXPLODED IN FRONT OF THE PRC EMBASSY, WHICH IS IN THE
LIMA RESIDENTIAL DISTRICT OF SAN ISIDRO.  MEANWHILE, TWO BOMBS WERE
THROWN AT A USSR EMBASSY VEHICLE THAT WAS PARKED IN FRONT OF THE
EMBASSY LOCATED IN ORRANTIA DISTRICT, NEAR SAN ISIDRO.

   POLICE SAID THE ATTACKS WERE CARRIED OUT ALMOST SIMULTANEOUSLY AND
THAT THE BOMBS BROKE WINDOWS AND DESTROYED THE TWO VEHICLES.

   NO ONE HAS CLAIMED RESPONSIBILITY FOR THE ATTACKS SO FAR. POLICE
SOURCES, HOWEVER, HAVE SAID THE ATTACKS COULD HAVE BEEN CARRIED OUT BY
THE MAOIST "SHINING PATH" GROUP OR THE GUEVARIST "TUPAC AMARU
REVOLUTIONARY MOVEMENT" (MRTA) GROUP.  THE SOURCES ALSO SAID THAT THE
SHINING PATH HAS ATTACKED SOVIET INTERESTS IN PERU IN THE PAST.

   IN JULY 1989 THE SHINING PATH BOMBED A BUS CARRYING NEARLY 50
SOVIET MARINES INTO THE PORT OF EL CALLAO. FIFTEEN SOVIET MARINES WERE
WOUNDED.

   SOME 3 YEARS AGO TWO MARINES DIED FOLLOWING A SHINING PATH BOMBING
OF A MARKET USED BY SOVIET MARINES.

   IN ANOTHER INCIDENT 3 YEARS AGO, A SHINING PATH MILITANT WAS KILLED
BY SOVIET EMBASSY GUARDS INSIDE THE EMBASSY COMPOUND.  THE TERRORIST
WAS CARRYING DYNAMITE.

   THE ATTACKS TODAY COME AFTER SHINING PATH ATTACKS DURING WHICH
LEAST 10 BUSES WERE BURNED THROUGHOUT LIMA ON 24 OCT.
```

B.2 A filled scenario template

This is one of three templates which should be generated for this message. The full set appears in MUC (1991). The “/” separates alternative correct slot fills.

```
0. MESSAGE ID                     TST1-MUC3-0099
1. TEMPLATE ID                    1
2. DATE OF INCIDENT               24 OCT 89 - 25 OCT 89
3. TYPE OF INCIDENT               BOMBING
4. CATEGORY OF INCIDENT           TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)    "TERRORISTS "
6. PERPETRATOR: ID OF ORG(S)      "SHINING PATH"
                                  "TUPAC AMARU REVOLUTIONARY MOVEMENT" /
                                  "MRTA"
```

```
7. PERPETRATOR: CONFIDENCE      POSSIBLE: "SHINING PATH"
                                POSSIBLE: "TUPAC AMARU REVOLUTIONARY MOVEMENT / "MRTA"
8. PHYSICAL TARGET: ID(S)       "EMBASSIES" / "EMBASSIES OF THE PRC"
9. PHYSICAL TARGET: TOTAL NUM   1 / PLURAL
10. PHYSICAL TARGET: TYPE(S)    DIPLOMAT OFFICE OR RESIDENCE: "EMBASSIES" / "EMBASSIES OF THE PRC"
11. HUMAN TARGET: ID(S)         -
12. HUMAN TARGET: TOTAL NUM     -
13. HUMAN TARGET: TYPE(S)       -
14. TARGET : FOREIGN NATION(S) "PEOPLES REP OF CHINA : "EMBASSIES" / "EMBASSIES OF THE PRC"
15. INSTRUMENT : TYPE(S)        -
16. LOCATION OF INCIDENT        PERU: LIMA (CITY): SAN ISIDRO (NEIGHBORHOOD)
17. EFFECT ON PHYSICAL TARGET(S) SOME DAMAGE: "EMBASSIES" / "EMBASSIES OF THE PRC"
18. EFFECT ON HUMAN TARGET     NO INJURY : "-"
```

Appendix C. ACE entities, relations, and events

C.1 Entities

A GPE is a location with a government, such as a city, state, or country. Mentions of a GPE may refer to the land mass (“He traveled to Florida”), the population (“Floida loves orange juice.”), or the government (“Florida declared a state of emergency”).

Type	Examples
person	Fred Smith; the undertaker
organization	Ford; San Francisco 49ers; a car manufacturer
GPE	France; Los Angeles
location	Nile; Mt. Everest; southern Africa
facility	Disneyland; the Berlin Wall; Aden’s streets
vehicle	the U.S.S. Cole; the train; the helicopter
weapon	Anthrax; bullets; tear gas

C.2 Relations

A relation expresses a relationship between two entities which are mentioned in the same sentence.

Type	Examples
physical	location of a person: Fred was in France
part-whole	the lobby of the hotel; Paris, France
personal-social	his lawyer; his wife; his neighbor
org-affiliation	the CEO of Microsoft; a student at Harvard
agent-artifact	my home; my car
gen-affiliation	a Methodist minister; American troops

https://doi.org/10.1017/S1351324919000512 Published online by Cambridge University Press

C.3 Events

These 8 event types are divided into 33 subtypes.

Type	Examples
life	is born; marries; dies
movement	transport; travel
transaction	sell; purchase; acquire
business	found; merge
conflict	attack; demonstrate
contact	meet; phone; write
personell	hired; fired; elected
justice	arrest; trial; convict

Cite this article: Grishman R (2019). Twenty-five years of information extraction. *Natural Language Engineering* **25**, 677–692. https://doi.org/10.1017/S1351324919000512