Chapter 2
Information Extraction: Past, Present and Future

Jakub Piskorski and Roman Yangarber

Abstract In this chapter we present a brief overview of Information Extraction, which is an area of natural language processing that deals with finding factual information in free text. In formal terms, *facts* are structured objects, such as database records. Such a record may capture a real-world entity with its attributes mentioned in text, or a real-world event, occurrence, or state, with its arguments or actors: who did what to whom, where and when. Information is typically sought in a particular target setting, e.g., corporate mergers and acquisitions. Searching for specific, targeted factual information constitutes a large proportion of all searching activity on the part of information consumers. There has been a sustained interest in Information Extraction for over two decades, due to its conceptual simplicity on one hand, and to its potential utility on the other. Although the targeted nature of this task makes it more tractable than some of the more open-ended tasks in NLP, it is replete with challenges as the information landscape evolves, which also makes it an exciting research subject.

2.1 Introduction

The recent decades witnessed a rapid proliferation of textual information available in digital form in a myriad of repositories on the Internet and intranets. A significant part of such information—e.g., online news, government documents, corporate reports, legal acts, medical alerts and records, court rulings, and social media

J. Piskorski
Institute for Computer Science, Polish Academy of Sciences, Warsaw, Poland
e-mail: Jakub.Piskorski@ipipan.waw.pl

R. Yangarber (✉)
Department of Computer Science, University of Helsinki, Finland
e-mail: Roman.Yangarber@cs.helsinki.fi

T. Poibeau et al. (eds.), *Multi-source, Multilingual Information Extraction and Summarization 11*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-28569-1_2, © Springer-Verlag Berlin Heidelberg 2013

"Three bombs have exploded in north-eastern Nigeria, killing 25 people and wounding 12 in an attack carried out by an Islamic sect. Authorities said the bombs exploded on Sunday afternoon in the city of Maiduguri."

⇓

TYPE:	Crisis
SUBTYPE:	Bombing
LOCATION:	Maiduguri
DEAD-COUNT:	25
INJURED-COUNT:	12
PERPETRATOR:	Islamic sect
WEAPONS:	bomb
TIME:	Sunday afternoon

Fig. 2.1 Example of automatically extracted information from a news article on a terrorist attack

communication—is transmitted through *unstructured*, free-text documents and is thus hard to search in. This resulted in a growing need for effective and efficient techniques for analyzing free-text data and discovering valuable and relevant knowledge from it in the form of *structured* information, and led to the emergence of Information Extraction technologies.

The task of Information Extraction (IE) is to identify a predefined set of concepts in a specific domain, ignoring other irrelevant information, where a domain consists of a corpus of texts together with a clearly specified information need. In other words, IE is about deriving structured factual information from unstructured text. For instance, consider as an example the extraction of information on violent events from online news, where one is interested in identifying the main actors of the event, its location and number of people affected. Figure 2.1 shows an example of a text snippet from a news article about a terrorist attack and a structured information derived from that snippet. The process of extracting such structured information involves identification of certain small-scale structures like noun phrases denoting a person or a person group, geographical references and numeral expressions, and finding semantic relations between them. However, in this scenario some domain-specific knowledge is required (understanding the fact that terrorist attacks might result in people being killed or injured) in order to correctly aggregate the partially extracted information into a structured form.

Even in a limited domain, IE is a non-trivial task due to the complexity and ambiguity of natural language. There are many ways of expressing the same fact, which can be distributed across multiple sentences [26], documents, or even knowledge repositories. Further, a significant amount of relevant information might be implicit, which may be difficult to discern and an enormous amount of background knowledge is needed to infer the meaning of unrestricted natural language. However, the scope of IE is narrower than the scope of full text understanding—computing all possible interpretations and grammatical relations in natural language text—whose realization is, as of today, still impossible from the technical point

of view. As a consequence, the use of considerably less sophisticated linguistic analysis tools for solving IE tasks may be beneficial, since it might be sufficient for the extraction and assembly of relevant pieces of information and it requires less knowledge engineering. In particular, recent advances in the field of Natural Language Processing (NLP) in robust, efficient and high-coverage shallow text processing techniques, as opposed to deep linguistic analysis, have contributed to the spread of deployment of IE techniques in real-world applications for processing of vast amount of textual data.

Information Extraction has not received as much attention as Information Retrieval (IR) and is often confounded with the latter. The task of IR is to select from a collection of textual documents a subset which are relevant to a particular query, based on key-word search and possibly augmented by the use of a thesaurus. The IR process usually returns a ranked list of documents, where the rank corresponds to the relevance score that the system assigned to the document in response to the query. However, the ranked document list does not provide any detailed information on the content of those documents. The goal of IE is not to rank or select documents, but to extract from the documents salient facts about pre-specified types of events, entities, or relationships, in order to build more meaningful, rich representations of their semantic content, which can be used to populate databases that provide structured input for mining more complex patterns (e.g., trends, summaries) in text collections. To sum up, IE aims to process textual data collections into a shape that facilitates searching and discovering knowledge in such collections.

IE systems are in principle more difficult and knowledge-intensive to build than IR systems. However, IE and IR techniques can be seen as complementary and can potentially be combined in various ways. IR is often used in IE for pre-filtering a very large document collection to a manageable subset, to which IE techniques could be applied. Alternatively, IE could be used as a subcomponent of an IR system to identify structures for intelligent document indexing.

Recent advances in IE provide dramatic improvements in the conversion of the flow of raw textual information into structured data and are increasingly being deployed in commercial application, for example in the financial, medical, legal, and security domains. Furthermore IE can constitute a core component technology in many other NLP applications, such as Machine Translation, Question Answering, Text Summarization, Opinion Mining, etc.

This chapter provides a brief survey of Information Extraction, including past advances, current trends and future challenges, and is organized as follows. Section 2.2 introduces basic definitions, typical IE tasks, evaluation of IE systems, and IE competitions and challenges. Section 2.3 presents a generic architecture of an IE system. Early IE systems and approaches are addressed in Sect. 2.4. An overview of supervised and unsupervised machine-learning based techniques to develop IE systems is given in Sect. 2.5. Finally, we mention recent trends and topics in Information Extraction in Sect. 2.6, which covers multi-linguality, cross-document and cross-lingual IE, extraction from social media, and open-domain IE.

The reader may also refer to other, more in-depth surveys, such as [4, 32].

2.2 Information Extraction Tasks

2.2.1 Definitions

The task of Information Extraction is to identify instances of a particular pre-specified class of entities, relationships and events in natural language texts, and the extraction of the relevant properties (arguments) of the identified entities, relationships or events. The information to be extracted is pre-specified in user-defined structures called *templates* (or objects), each consisting of a number of *slots* (or attributes), which are to be instantiated by an IE system as it processes the text. The slots fills are usually: strings from the text, one of a number of pre-defined values, or a reference to a previously generated object template. One way of thinking about an IE system is in terms of database population, since an IE system creates a structured representation (e.g., database records) of selected information drawn from the analyzed text.

2.2.2 IE Task Types

Applying information extraction on text aims at creating a structured view—i.e., a representation of the information that is machine understandable. The classic IE tasks include:

- **Named Entity Recognition (NER)** addresses the problem of the identification (detection) and classification of predefined types of named *entities*, such as organizations (e.g., '*World Health Organisation*'), persons (e.g., '*Muammar Kaddafi*'), place names (e.g., '*the Baltic Sea*'), temporal expressions (e.g., '*1 September 2011*'), numerical and currency expressions (e.g., '*20 Million Euros*'), etc. NER task can additionally include extracting descriptive information from the text about the detected entities through filling of a small-scale template. For example, in the case of persons, it may include extracting the title, position, nationality, sex, and other attributes of the person. It is important to note that NER also involves lemmatisation (normalisation) of the named entities, which is particularly crucial in highly inflective languages. For example in Polish there are six inflected forms of the name '*Muammar Kaddafi*' depending on grammatical case: '*Muammar Kaddafi*' (nominative), '*Muammara Kaddafiego*' (genitive), *Muammarowi Kaddafiemu* (dative), '*Muammara Kaddafiego*' (accusative), *Muammarem Kaddafim* (instrumental), *Muammarze Kaddafim* (locative), *Muammarze Kaddafi* (vocative).
- **Co-reference Resolution (CO)** requires the identification of multiple (co-referring) mentions of the same entity in the text. Entity mentions can be:

  (a) Named, in case an entity is referred to by name; e.g., '*General Electric*' and '*GE*' may refer to the same real-world entity,

(b) Pronominal, in case an entity is referred to with a pronoun; e.g., in '*John bought food. But he forgot to buy drinks.*', the pronoun *he* refers to *John*,
(c) nominal, in case an entity is referred to with a nominal phrase; e.g., in '*Microsoft revealed its earnings. The company also unveiled future plans.*' the definite noun phrase *The company* refers to *Microsoft*, and
(d) Implicit, as in case of using zero-anaphora[1]; e.g., in the Italian text fragment '[*Berlusconi*]$_i$ *ha visitato il luogo del disastro.* ϕ_i *Ha sorvolato, con l'elicottero.*' (Berlusconi has visited the place of disaster. [He] flew over with a helicopter.) the second sentence does not have an explicit realisation of the reference to *Berlusconi*.

- **Relation Extraction (RE)** is the task of detecting and classifying predefined relationships between entities identified in text. For example:
  - `EmployeeOf(Steve Jobs,Apple)`: a relation between a person and an organisation, extracted from '*Steve Jobs works for Apple*'
  - `LocatedIn(Smith,New York)`: a relation between a person and location, extracted from '*Mr. Smith gave a talk at the conference in New York*',
  - `SubsidiaryOf(TVN, ITI Holding)`: a relation between two companies, extracted from '*Listed broadcaster TVN said its parent company, ITI Holdings, is considering various options for the potential sale*.

  Note, although in general the set of relations that may be of interest is unlimited, the set of relations within a given task is predefined and *fixed*, as part of the specification of the task.
- **Event Extraction (EE)** refers to the task of identifying events in free text and deriving detailed and structured information about them, ideally identifying *who did what to whom, when, where, through what methods (instruments), and why*. Usually, event extraction involves extraction of several entities and relationships between them. For instance, extraction of information on terrorist attacks from the text fragment '*Masked gunmen armed with assault rifles and grenades attacked a wedding party in mainly Kurdish southeast Turkey, killing at least 44 people.*' involves identification of perpetrators (*masked gunmen*), victims (*people*), number of killed/injured (*at least 44*), weapons and means used (*rifles and grenades*), and location (*southeast Turkey*). Another example is the extraction of information on new joint ventures, where the aim is to identify the partners, products, profits and capitalization of the joint venture. EE is considered to be the hardest of the four IE tasks.

While in the early years of IE the focus was on solving the IE tasks listed above at document level, research has shifted to cross-document information extraction, which will be addressed in Sect. 2.6.

[1]Zero-anaphora are typical in many languages—including the Romance and Slavic languages, Japanese, etc.—in which subjects may not be explicitly realized.

2.2.3 Evaluation in Information Extraction

Given an input text, or a collection of texts, the expected output of an IE system can be defined very precisely. This facilitates the evaluation of different IE systems and approaches. In particular, the *precision* and *recall* metrics were adopted from the IR research community for that purpose. They measure the system's effectiveness from the user's perspective, i.e., the extent to which the system produces all the appropriate output (recall) and only the appropriate output (precision). Thus, recall and precision can bee seen as measure of completeness and correctness, respectively. To define them formally, let #*key* denote the total number of slots expected to be filled according an *annotated* reference corpus, representing ground truth or a "gold-standard", and let #*correct* (#*incorrect*) be the number of correctly (incorrectly) filled slots in the system's response. A slot is said to be filled incorrectly either if it does not align with a slot in the gold standard (*spurious slot*) or if it has been assigned an invalid value. Then, precision and recall may be defined as follows:

$$precision = \frac{\#correct}{\#correct + \#incorrect} \qquad recall = \frac{\#correct}{\#key} \tag{2.1}$$

In order to obtain a more fine-grained picture of the performance of IE systems, precision and recall are often measured for each slot type separately.

The *f-measure* is used as a weighted harmonic mean of precision and recall, which is defined as follows:

$$F = \frac{(\beta^2 + 1) \times precision \times recall}{(\beta^2 \times precision) + recall} \tag{2.2}$$

In the above definition β is a non-negative value, used to adjust their relative weighting ($\beta = 1.0$ gives equal weighting to recall and precision, and lower values of β give increasing weight to precision).

Other metrics are used in the literature as well, e.g., the so called *slot error rate*, SER [40], which is defined as follows:

$$SER = \frac{\#incorrect + \#missing}{\#key} \tag{2.3}$$

where #*missing* denotes the number of slots in the reference that do not align with any slots in the system response. It reflects the ratio between the total number of slot errors and the total number of slots in the reference. Depending on the particular needs, certain error types, (e.g., spurious slots) may be weighted in order to deem them more or less important than others.

2.2.4 Competitions, Challenges and Evaluations

The interest and rapid advances in the field of IE has been essentially influenced by the DARPA-initiated series of Message Understanding Conferences (MUCs).[2] They were conducted with the intention to coordinate multiple research groups and US government agencies seeking to improve IE and IR technologies [23]. The MUCs defined several types of IE tasks and invited the scientific community to tune their systems to complete these tasks. MUC participants were initially given a detailed description of an IE task scenario, along with annotated training data to adapt their systems to this scenario within a limited time-period, of 1–6 months. During the testing phase, each participant received a new set of test documents, applied their systems on these documents, and returned the extracted templates to MUC organizers. These results were then compared to a set of templates manually filled by human annotators, similarly to how it is done in evaluation in other fields of NLP, including summarization, described in Chap. 1.

The IE tasks defined in MUC conferences were intended to be prototypes of extraction tasks that will arise in real-world applications. The first two MUC conferences (1987–1989) focused on automated analysis of military messages containing textual information about naval sightings and engagements, where the template to be extracted had ten slots. Since MUC-3 [35] the task shifted to the extraction from newswire articles, i.e., information concerning terrorist activities, international joint-venture foundations, corporate management succession events, and space vehicle and missile launches. Over time, template structures became more complex. Since MUC-5, nested template structure and multilingual IE were introduced. The later MUC conferences distinguished several different IE subtasks in order to facilitate the evaluation and identification of IE sub-component technologies which could be immediately useful. The generic IE tasks defined in MUC-7 (1998) provided progressively higher-level information about texts, which, basically correspond to the IE tasks described in Sect. 2.2.2.

The ACE (Automated Content Extraction) Program,[3] started in 1999, is a continuation of the MUC initiative, which aims to support the development of automatic content extraction technologies for automatic processing of natural language in text from various sources [15]. The ACE Program has defined new and harder-to-tackle IE tasks centered around extraction of entities, relations and events. In particular, the increased complexity resulted from: (a) inclusion of various information sources (e.g., news, web-logs, newsgroups) and quality of input data (e.g., telephone speech transcripts), (b) introduction of more fine-grained entity types (e.g., facilities, geo-political entities, etc.), template structures and relation types, and (c) widening the scope of the core IE tasks, e.g., the classic NER task has been converted to Entity

[2] http://www-nlpir.nist.gov/related_projects/muc/.

[3] http://www.itl.nist.gov/iad/mig/tests/ace/.

Detection and Tracking task, which includes detection of all mentions of entities within a document, whether named, nominal or pronominal.

The Linguistic Data Consortium[4] (LDC) develops annotation guidelines, corpora and other linguistic resources to support the ACE Program, which are used to evaluate IE systems in a similar manner to conducting MUC competitions. Effort has been put into preparing data for languages other than English, i.a., Spanish, Chinese, and Arabic. In the recent years, corpora for the evaluation of cross-document IE tasks (e.g., global entity detection and recognition task which requires cross-document co-reference resolution of certain types of entities) have been created.

Both MUC and ACE initiatives are of central importance to the IE field, since they provided a set of corpora that are available to the research community for the evaluation of IE systems and approaches.

IE systems and approaches are evaluated in other, broader venues as well. The Conference on Computational Natural Language Learning (CoNLL) has organized some "shared tasks" (competitions) on language-independent NER.[5] As part of the Text Analysis Conference (TAC)[6] a dedicated track, namely the Knowledge Base Population track, comprises evaluation of tasks centering around discovering information about entities (e.g., entity attribute extraction) as found in a large corpus and incorporating this information into a given knowledge base (*entity linking*), i.e., deciding whether a new entry needs to be created. Some IE-related tasks are being evaluated in the context of the Senseval[7] initiative, whose goal is to evaluate semantic analysis systems. For instance, the Web People Search Task [5] focused on grouping web pages referring to the same person, and extracting salient attributes for each of the persons sharing the same name. The task on Co-reference Resolution in Multiple Languages [55] focused on the evaluation of co-reference resolution for six different languages to provide, i.e., better insight into porting co-reference resolution components across languages.

2.2.5 Performance

The overall quality of extraction depends on multiple factors, including, i.e.: (a) the level of logical structures to be extracted (entities, co-reference, relationships, events), (b) the nature of the text source (e.g., online news, corporate reports, database textual fields, short text messages sent through mobile phones and/or social media), (c) the domain in focus (e.g., medical, financial, security, sports), and (d) the language of the input data (e.g., English vs. morphologically rich languages like Russian). To give an example, the precision/recall figures of the

[4] http://projects.ldc.upenn.edu.

[5] http://www.clips.ua.ac.be/conll2002/ner/.

[6] http://www.nist.gov/tac/.

[7] http://www.senseval.org.

top scoring IE systems for extracting information related to aircraft accidents from online English news evaluated at MUC-7[8] oscillated around: (a) 95/95% for NER task (b) 60/80% for co-reference, (c) 85/70% for relations, and (d) 70/50% for events. These figures reflect the relative complexity of the main IE tasks, with best performance achievable on NER[9] and event extraction being the most difficult task. It is important to note that a direct comparison between precision and recall figures obtained by IE systems in various competitions and challenges is difficult and not straightforward, due to the factors listed above, i.e., different set-up of the evaluation tasks w.r.t. template structure, domain, quality of input texts and language to be processed. In general, IE from English texts appears to be somewhat easier than for other languages which exhibit more complex linguistic phenomena, e.g., richer morphology and freer word order. Similarly for the extraction from grammatically correct texts, such as online news, versus extraction from short and often ungrammatical short messages, such as tweets and blog posts.

2.3 Architecture: Components of IE Systems

Although IE systems built for different tasks may differ from each other significantly, there are certain core components shared by most IE systems. The overall IE processing chain can be analyzed along several dimensions. The chain typically includes core linguistic components—which can be adapted to or be useful for NLP tasks in general—as well as IE-specific components, which address the core IE tasks. On the other hand, the chain typically includes domain-independent vs. domain-specific components. The domain-independent part usually consists of language-specific components that perform linguistic analysis in order to extract as much linguistic structure as possible. Usually, the following steps are performed:

- **Meta-data analysis:** Extraction of the title, body, structure of the body (identification of paragraphs), and the date of the document.
- **Tokenization:** Segmentation of the text into word-like units, called tokens and classification of their type, e.g., identification of capitalized words, words written in lowercase letters, hyphenated words, punctuation signs, numbers, etc.
- **Morphological analysis:** Extraction of morphological information from tokens which constitute potential word forms—the base form (or *lemma*), part of speech, other morphological tags depending on the part of speech: e.g., verbs have features such as tense, mood, aspect, person, etc. Words which are ambiguous with respect to certain morphological categories may undergo disambiguation. Typically part-of-speech disambiguation is performed.

[8] http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.htmlconference.

[9] Industrial-strength solutions for NER for various languages exist.

- **Sentence/Utterance boundary detection:** Segmentation of text into a sequence of sentences or utterances, each of which is represented as a sequence of lexical items together with their features.
- **Common Named-entity extraction:** Detection of domain-independent named entities, such as temporal expressions, numbers and currency, geographical references, etc.
- **Phrase recognition:** Recognition of small-scale, local structures such as noun phrases, verb groups, prepositional phrases, acronyms, and abbreviations.
- **Syntactic analysis:** Computation of a dependency structure (parse tree) of the sentence based on the sequence of lexical items and small-scale structures. Syntactic analysis may be deep or shallow. In the former case, one is interested in computing all possible interpretations (parse trees) and grammatical relations within the sentence. In the latter case, the analysis is restricted to identification of non-recursive structures or structures with limited amount of structural recursion, which can be identified with a high degree of certainty, and linguistic phenomena which cause problems (ambiguities) are not handled and represented with under-specified structures.

The extent of the domain-independent processing may vary depending on the requirements of the particular application. The core IE tasks—NER, co-reference resolution, and detection of relations and events—are typically domain-specific, and are supported by domain-specific system components and resources.

Domain-specific processing is typically also supported on a lower level by detection of specialized *terms* in text. For example, in domains related to medicine, extensive specialized lexicons, ontologies and thesauri of medical terms will be essential, whereas for IE in business-related domains, these are unnecessary.

A typical architecture of an IE system is depicted in Fig. 2.2. In the domain-specific core of the processing chain, a NER component is applied to identify the entities relevant in a given domain. Patterns may then be applied[10] to: (a) identify text fragments, which describe the target relations and events, and (b) extract the key attributes to fill the slots in the template representing the relation/event. A co-reference component identifies mentions that refer to the same entity. Finally, partially-filled templates are fused and validated using domain-specific inference rules in order to create full-fledged relation/event descriptions. The last step is crucial since the relevant information might be scattered over different sentences or even documents. It is important to note that, in practice, the borders between domain-independent linguistic analysis components and core IE components may be blurred, e.g., there may be a single NER component which performs domain-independent and domain-specific named-entity extraction simultaneously.

There are several software packages, both freely available for research purposes and commercial use, which provide various tools that can be used in the process

[10]Various formalisms are used to encode patterns, ranging from character-level regular expressions, through context-free grammars to unification-based formalisms.

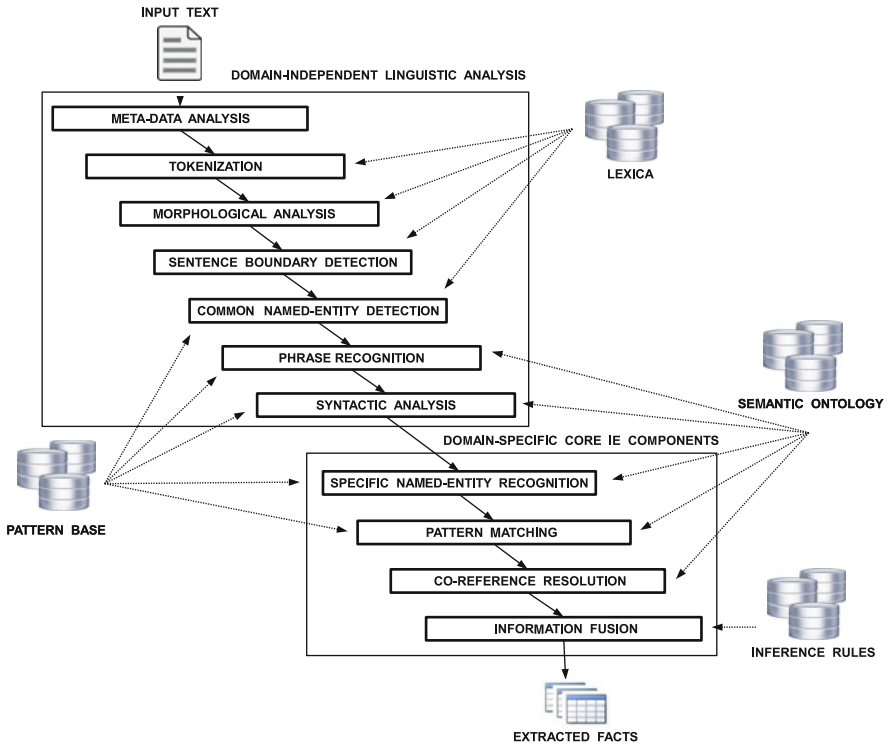

Fig. 2.2 Typical architecture of an information extraction system

of developing an IE system, ranging from core linguistic processing modules (e.g., language detectors, sentence splitters), to general IE-oriented NLP frameworks.[11]

2.4 Early Years: Knowledge Engineering Approaches

The strong application potential of IE was recognized already in the late 1980s. One of the first attempts to apply IE in the financial domain to extract information from messages regarding money transfers between banks was the ATRANS system [38], based on simple NLP techniques and script-frames approach. JASPER was an IE system, described in [1], which extracted information from reports on corporate earnings from small sentence fragments using robust NLP methods. SCISOR [28] was an integrated system incorporating IE for the extraction of facts related to corporate mergers and acquisitions from online news. Examples of IE system in

[11] Some examples of NLP tools which are relevant in the context of developing IE systems can be found at: http://alias-i.com/lingpipe/web/competition.html.

the security domain are described, for example, in [34, 35]. All the above systems and other early IE systems were developed using the Knowledge Engineering (KE) approach [4], where the creation of linguistic knowledge in the form of *rules*, or patterns, for detecting and extracting the target information from text is performed by human experts, through inspection of the test corpus and intuition. This is usually done in an iterative manner, starting with a small set of extraction rules which are tested on the available corpora and extended until a desired trade-off between precision and recall is reached. Most of the early IE systems had a major shortcoming: they exhibited a non-modular black-box character, were mono-lingual, and were not easily adaptable to new scenarios. However, they did demonstrate that relatively simple NLP techniques may be sufficient for solving real-world IE tasks that are narrow in scope.

The subsequent efforts in the area of KE-based approaches aimed at moving toward general-purpose IE systems and frameworks, which are modular and easier to adapt to new domains and languages. The FASTUS system described in [24] is an example of an efficient and robust general-purpose IE system developed at SRI International. FASTUS was able to process English and Japanese, and was among the top-scoring system in the MUC competitions. It was designed as a set of cascaded nondeterministic finite-state transducers, where each stage of the processing is represented as a finite-state device. The output structures of the successive stages serve as the input for the next stage. Many other IE systems developed by that time were implemented as cascades of finite-state devices, e.g., the SPPC system described in [45], that processes German text. The emergence of finite-state based IE systems was mainly motivated by two reasons. From the computational perspective, finite-state devices are time- and space-efficient due to their closure properties and the existence of efficient optimization and transformation algorithms. From the linguistic perspective, the vast majority of local linguistic phenomena encountered in IE can be easily expressed as finite-state devices, and in particular, they can be usually broken down into a number of autonomous finite-state sub-descriptions. In this manner, a finite-state cascade allows for a strong decomposition of the linguistic analysis and of the entire IE process into many subtasks, ordered by increasing complexity. The incremental character of this approach yields simplicity, higher modularity and ease of maintenance of the underlying linguistic resources. Although more powerful formalisms exist—e.g., context-free or unification grammars—the finite-state-based formalisms for extraction grammars became prevalent, due to the characteristics mentioned. In addition, finite-state approximation grammars [43] proved to provide surprisingly effective partial parsers. Finite-state based formalisms are used in the popular general-purpose text engineering platform GATE [14], which is deployed for the development of IE systems [42], as well as in other IE-oriented text engineering frameworks such as SPROUT [17] and EXPRESS [49].

Although IE systems based on shallow linguistic analysis implemented as finite-state cascades became popular, the later MUC conferences saw some integration of shallow and deep linguistic analysis in IE systems in order to obtain a better performance. For instance, deep linguistic analysis could be applied on text

fragments deemed to be highly relevant or on those which could not have been processed successfully through shallow linguistic analysis. An example of a system that attempts to find a pragmatic middle way in the shallow vs. deep analysis debate is LASIE-II [25]. It deploys an eclectic mixture of techniques ranging from finite-state recognizers for detection of domain-specific lexical patterns to restricted context-free grammars for partial parsing. This system exhibited highly modularized architecture and was equipped with visual tools for selecting the control flow through different module combinations, which became an essential feature of many IE systems to enable one to obtain deeper insight into strengths and weaknesses of the particular subcomponents and their interactions. Two other examples of modular IE systems based on the KE approach, namely, IE2 and REES are described in [2] and [3], respectively. The former achieved the highest scores in almost all IE tasks in MUC-7, whereas the latter was a first attempt at constructing large-scale event and relation extraction system based on shallow text analysis methods, which covered more than 100 types of relations and events related to the area of business, finance and politics.

2.5 Emergence of Trainable IE Systems

Research in the late 1990s and the beginning of the twenty-first century resulted in significant advances in IE (and in NLP in general) in terms of emergence of KE-based, modular IE systems that were able to process vast amounts of textual data robustly and efficiently, including in languages other than English. By the turn of the century, the modular design of IE systems was widely accepted; the design consisted of a *generic core engine* on one hand, and language- and domain-specific components on the other, which may be called *knowledge bases*. The knowledge bases are further modularized according to the kind of knowledge needed to perform the IE tasks. Information contained in the different knowledge bases is of different forms and different levels of complexity. On the lowest-level are specialized lexicons and gazetteers. We may view ontologies, semantic networks and thesauri as being one level above, in that they specify inter-relationships among concepts and organize knowledge into more complex objects. Many of the described systems are pattern-based: they employ a pattern-matching mechanism to identify constituents in text that have semantic significance to the task—entities or events—and large parts of the pattern base are generally highly domain-specific. Patterns are much more easily constructed and understood when they are encoded in a declarative fashion, and languages for defining patterns emerged, in the systems mentioned above. Finally, certain *reasoning* operations that need to be performed on extracted facts are also highly domain-specific in nature—these operations can sometimes be encoded in declarative fashion, or as sub-programs; in either case, they can be viewed as a parameterization of the core processing chain.

Abstracting the domain-specific knowledge into knowledge bases and away from the core processing engine allows for much more flexible re-configuration of an

existing IE engine from one task to another. However, the process of handcrafting language- and domain-specific resources and components remained a very time-consuming and difficult task. This stimulated research on trainable IE systems that deploy machine-learning (ML) techniques to offload some of the burden of customizing a general-purpose IE engine to a new domain or task.

The first "wave" in the shift away from KE-based approaches and toward trainable systems focused on supervised machine-learning approaches. The principal motivation behind this trend is to shift the human effort in customization of the IE knowledge bases away from knowledge engineering and toward annotating training data, which serves as input to machine-learning algorithms. This promotes further modularity in development, since knowledge engineering requires effort from both the system developer and the domain expert, whereas data annotation, in principle, requires mostly an effort on the part of the latter.

The broad array of tools and algorithms from supervised learning that were already in wide use in other areas of NLP came to be applied to IE tasks as well. This trend affected all IE-related tasks. For example, hidden Markov models (HMMs), at the core of many NLP tasks, including speech processing and part-of-speech tagging, found application in IE. An example of a system for named entity detection that uses HMMs is Nymble [9]. An HMM consists of states and probabilistic transitions between the states. The simple idea behind using the HMM for finding named entities is to associate one state with being inside a name, and another state with being outside a name, while scanning or generating the words of a sentence. This basic scheme is generalized by adding states to accommodate multiple *types* of names, etc. Given some amount of text annotated with labels that indicate which words constitute names, and what type of names, the HMM is trained and applied using standard algorithms well-known from the literature.

Supervised learning is similarly applied to other IE tasks, mentioned above.[12] As we may do with NER, we may also re-cast the task of relation or event extraction as a classification problem. This may be done as follows. Given a segment (or a window) of text, such as a sentence, or a paragraph, etc., pose the problem as: does this segment contain a relation of type R? The key to making the problem tractable as stated is in defining a set of independent variables, or *features*, of the text segment that may help us to determine the presence or absence of the relation. These features will typically include words and terms (possibly consisting of multiple words) present (or absent) in the segment, entities found in the segment, other relations, and other syntactic and semantic information extracted from the segment. Some of the earlier literature on application of supervised learning to IE tasks includes, e.g., [56, 57]. Conditional random fields (CRFs), another supervised learning method that has become popular in the recent years, has also been applied to IE tasks. CRFs are similar to HMMs in that they are good at modeling local dependencies, and they perform especially well on the "lower-level" tasks—e.g.,

[12]Standard references are available for application of supervised learning algorithms, e.g., [70], and in particular to NLP tasks, cf. [41].

named entity classification, [60], as they are well suited for sequence-labeling tasks in general. CRFs can be used to model certain variants of the relation detection task as well, as shown, e.g., by [7] which applies CRFs to the task of relation *labeling* in an open setting. More on this latter topic is mentioned in Sect. 2.6.2.[13]

The features used by a learning algorithm do not need to be limited to the simplest kinds of features of the text, such as "bags of words", but they do need to be more readily extractable from text than the classification problem we are trying to solve—i.e., they have to be handled by lower-level processing within the IE pipeline. This approach fits well with the overall cascaded design of the IE pipeline, where the text is first analyzed into small constituent pieces, and then successive stages extract higher-level information over the same text.

Co-reference resolution can be similarly re-cast as a classification problem. For example, for a given pair of entity mentions extracted from a text—entities that were extracted in an earlier processing phase by a component that precedes co-reference resolution in the IE pipeline—we may ask whether these mentions co-refer, i.e., whether they refer to the same real-world object. The features, in this case, may include bags of words found in the sentences containing the mentions, as well as other, more complex and abstract aspects of the text, including the distance between the entities, the words and/or parts of speech found along the path in the parse tree leading from one entity to the other (if the entities are in the same sentence, and the sentence has been fully parsed), the layout of the segments containing the two entities, etc.[14]

Once we have chosen a set of representative features, we can use supervised learning algorithms, such as Naive Bayes (NB), or Bayes network (BN), to train classifiers based on a set of annotated training data.

The choice of features also depends on the type of classifier we wish to train. Some of the simpler classifiers, such as NB, may suffer degraded performance if the features are not independent, while others, such as support-vector machines (SVM), may be are less sensitive to complex dependencies among feature, which may simplify feature engineering. Several tradeoffs need to be considered when choosing the features and the learning algorithms. On one hand we wish incorporate as many features as possible, including ones that are possibly not independent, and possibly not informative. On the other hand, we need to limit the computational cost. Methods for automatic feature selection (such as leave-one-out cross-validation) can be employed to reduce large sets of non-independent and possibly non-informative features, but then the feature selection process itself can become computationally expensive.

[13]This kind of labeling is actually more similar in nature to named entity classification, except that the information extracted is about not the nature of node in a semantic network—i.e., a (possibly named) entity—but the nature of an edge, i.e., a link between two nodes. The information extracted focuses mostly on what label the system should attach to the analyzed object.

[14]For an example of learning for anaphora resolution, cf. [46].

In this fashion, the burden on the human developer of the IE system shifts away from manual knowledge engineering, and toward feature engineering on one hand, and data annotation on the other. Although feature engineering is a complex task, to some extent we may expect that once appropriate features are chosen for one domain, they will work reasonably well for other domains as well, or at least for "similar" domains.

Thus, the effectiveness (quality of performance) of supervised learning depends on the complexity of the task and on the amount of available annotated training data. The larger the amount of annotated text, the better the performance of the learner. On the other hand, the more complex IE tasks, such as relation extraction, will require more annotated data than the simpler tasks (such as name entity recognition).[15] For the higher-level complex IE tasks, data sparseness becomes a serious problem, which requires still more annotated data. Similarly to supervised learning for syntactic parsing, (e.g., [11]) where tree-banks containing millions of annotated words are used for training statistical parsers, supervised learning for IE tasks can also require massive amounts of annotated data. To alleviate the data annotation bottleneck, recent research in IE has followed several approaches.

Active learning is an area of machine learning which aims to reduce the amount of annotation required from the human annotator, by "active" participation on the part of the learner in the learning process.[16] The idea is that the annotator initially provides a small set of examples, based on which the learner actively decides which other examples—from among a very large set of possible candidate examples—the human teacher/annotator should annotate next, for maximum gain. Intuitively, this means selecting the examples on which the learner's uncertainty is greatest. In practice, it means selecting such examples, on which the conditional distribution of the target class—e.g., a sought name type, or whether a relation is present or absent—has the highest entropy, given the feature values. These are examples from which the learner will benefit *most*, assuming that an annotator is willing to spend only a fixed amount of time on annotating data. Another way of viewing this is that the learner benefits less if the annotator marks up many examples which are very similar among themselves (in terms of the feature values), and provides redundant information. For examples of application of active learning in IE, cf., e.g., [31].

Active learning is a "human-in-the-loop" approach to learning, used to support supervised learning methods in IE, which results in a "hybrid" approach to building IE systems: the human developer is still involved in the knowledge engineering process but less directly and with substantial automated assistance.

Bootstrapping is another kind of learning that aims to reduce the amount of human involvement in knowledge engineering. Similarly to active learning, the

[15]Typically a substantial subset of the annotated data will need to be reserved for testing and evaluation purposes—otherwise we cannot estimate the quality of performance on unseen data. We cannot test on the same data on which the learning algorithm was trained.

[16]In contrast to a "passive" learner, which learns from a set of annotated examples—negative as well as positive—provided by the teacher.

system developer provides an initial set of annotated examples—the "seeds—but in bootstrapping the learning process proceeds without further supervision, until a convergence criterion is reached. Such *weakly supervised* methods,[17] require some amount of human supervision, in the initial phase—providing the seeds—as well as possibly in the final phase—checking the result of the learning, to remove candidates that appear "bad", or not useful. Methods for learning domain-specific names (such as names of companies for the domain of financial news, and names of medical conditions for the domain of epidemic surveillance) are described, e.g., in [13, 48, 74]. Application to learning domain-specific patterns for event extraction is described, e.g., in [63, 66, 71].

The main motivation for bootstrapping is that we expect that once the learner converges and produces a large set of candidate elements for the knowledge bases—e.g., for named entities or for relation extraction patterns—it is easier, or takes less time, for the system developer to go rank the set of candidates by their quality (even if the set is large), than to construct a set candidates from scratch. Thus we can hope to reduce the expense of knowledge engineering.

2.6 Current Trends, Challenges and the Future

We have presented IE in the "classic" setting, as it has been studied traditionally. We turn now to the challenges and directions that we anticipate will grow in importance and will dominate the IE research landscape in the coming years. In the context of currently emerging novel kinds of large-scale corpora, IE assumes new dimensions and reinvents itself.

For over a decade the bulk of research focused on information extraction in English. The growing amount of textual data available in other languages resulted in shifting of the focus to other non-English IE and language-independent (multilingual) IE techniques. While much of the attention in non-English IE focused on NER (cf. [44] which includes a collection of references to work on non-English NER), relatively little work has been reported on higher-level IE tasks, including relation and event extraction.

IE in languages other than English is, in general, harder, and the performance of non-English IE systems is usually lower. This is mainly due to lack of core NLP components and underlying linguistic resources for many languages, but most of all due to the various linguistic phenomena that are non-existent in English, which include, *inter alia*:

- Lack of whitespace, which complicates word boundary disambiguation (e.g., in Chinese [22]);
- Productive compounding, which complicates morphological analysis, as in German, where circa 10–15% words are compounds whose decomposition is

[17] These bootstrapping-type methods are sometimes called "unsupervised", but that is rather a misnomer; a more appropriate term would be *weakly* or *minimally* supervised learning.

crucial for higher-level NLP processing [45]; Finnic languages exhibit similar phenomena;
- Complex proper name declension, which complicates named-entity normalisation. Typical for Slavic languages and exemplified in [52], which describes techniques for lemmatisation and matching Polish person names;
- Zero anaphora, whose resolution is crucial in the context of CO task. This is typical for Japanese, Romance and Slavic languages, [27];
- Free word order and rich morphology, which complicates relation extraction. This is typical, e.g., for Germanic, Slavic and Romance languages, and exemplified in [76], which proposes more linguistically sophisticated extraction patterns for relation extraction than the classical surface-level linear extraction patterns applied in many English IE systems.

Though this is only a partial list of phenomena that complicate non-English IE, significant progress in non-English IE could be observed in the recent years.

Apart from tackling multi-linguality, one central characteristic theme that runs through current research in IE, as distinguished from classic IE presented so far, is the extraction of information from *multiple sources*—in contrast from finding information within a single piece of text. The clear motivation for this is that most given facts or stories typically do not exist in an isolated report, but are rather carried by many on-line sources, possibly in different languages. In particular, the information redundancy resulting from the access to multiple sources reporting can be exploited for the validation of facts. Further, the fact or story may undergo evolution through time, with elaborations, modifications, or possibly even contradictory related information emerging over time.

Examples of approaches to IE from multiple sources and existing IE systems that exploit such approaches may be found in [16, 21, 30, 33, 36, 47, 51, 62, 65, 69, 72, 73]. In the area of cross-lingual IE, experiments on cross-lingual bootstrapping of ML-based event extraction systems have been reported in [12, 33, 64], and on cross-lingual information fusion in [50]. An overview on the challenges in IE, in particular addressing cross-document IE, cross-lingual IE, and cross-media IE and fusion may be found, e.g., in [29].

There have been several workshops dedicated to research on the various aspects of IE from multiple sources, in particular [10, 53, 54]. The latter two include the body of work which was invited to be presented in this volume, as mentioned in the Preface. This is the primary purpose and focus of this volume—to detail the research in the direction of these current and future needs served and supported by IE; the subsequent chapters provide a wealth of references to a wide range of recent and current work.

We conclude this survey with an excursion into two more closely related aspects of state-of-the-art IE from large-scale text collections—applying IE to novel types of media and attempting to broaden the application of IE to any number of domains simultaneously.

2.6.1 Information Extraction from Social Media

The advent of social media resulted in new forms of individual expression and communication. At present, an ever-growing amount of information is being transferred through social media, including blogs, Web fora, and micro-blogging services like Facebook[18] or Twitter.[19] These services allow users to write large numbers of short text messages in order to communicate, comment and chat about current events of any kind, current events, politics, products, etc. The trend of using social media has been significantly boosted by the low-barrier cross-platform access to such media and global proliferation of mobile devices. It is worth noting that social media can in many situations provide information that is more up-to-date than conventional sources of information, such as online news—e.g., in the context of natural disasters, as the earthquakes in Japan in 2011, or the Arab Spring—which makes them a particularly attractive source of information in these settings. As a consequence, some work was reported by the research community on automated analysis of social media content, including also attempts to information extraction from social media.

Extraction of information from social media is more challenging than classic IE, i.e., extraction from trusted sources and well-formed grammatical texts. The major problems encountered when processing social media content are: (a) texts are typically very short—e.g., Facebook limits status updates to 255 characters, whereas Twitter limits messages ("tweets") to 140 characters, (b) texts are noisy and written in an informal setting, include misspellings, lack punctuation and capitalisation, use non-standard abbreviations, and do not contain grammatically correct sentences, and (c) high uncertainty of the reliability of the information conveyed in the text messages, e.g., compared to the news media. A straightforward application of standard NLP tools on social media content typically results in significantly degraded performance. The mentioned characteristics influenced a new line of research in IE, which focuses on developing methods to extract information from short and noisy text messages.

Some recent work reports on NER from Tweets. Locke and Martin [39] presents results of tuning a SVM-based classifier for classifying persons, locations and organisations names in Twitter, which achieves relatively poor results, i.e., precision and recall around 75% and 50% respectively. Ritter et al. [58] describes an approach to segmentation and classification of a wider range of names in tweets based on CRFs (using POS and shallow parsing features) and Labeled Latent Dirichlet Allocation respectively. The reported precision and recall figures oscillate around 70% and 60% respectively. Liu et al. [37] proposed NER (segmentation and classification) approach for tweets, which combines KNN and CRFs paradigms and achieves precision and recall figures around 80% and 70% respectively.

[18] http://www.facebook.com/.

[19] http://twitter.com/.

Higher level IE tasks such as event extraction from Twitter or Facebook are more difficult than NER since not all information on an event may be expressed within a single short message. Most of the reported work in the area of event extraction from short messages in social media focus on event detection, e.g., [59] presents a SVM-based approach to classify tweets on earthquakes and typhoons (80–90% recall and 60–65% precision) and a probabilistic spatio-temporal model to find the centre and the trajectory of the event, [67] reports on an approach of using linguistic features to detect tweets with content relevant to situational awareness during mass emergencies (emergency event detection), which achieves 80% accuracy. Apart from research on pure event detection from micro-blogs, some work reported on extracting structured information on events. For instance, [68] studies the content of tweets during natural hazards events in order to identify event features which can be automatically extracted. Benson et al. [8] presents a CRF-based approach to extracting entertainment events with 85% precision (including extraction of location, event name, artist name, etc.) through aggregating information across multiple messages, similar in spirit to the approaches referred to above.

Research on IE from social media is still in its early stages, and focuses mainly on processing English. Future research will likely focus on further adaptation of classic IE techniques to extract information from short messages used in micro-blogging services, tackling non-English social media, and techniques for aggregation and fusion of information extracted from conventional text documents (e.g., news) and short messages posted through social media, e.g., for using Twitter to enhance event descriptions extracted from classic online news with situational updates, etc.

2.6.2 Open-Domain Information Extraction

In the last two decades IE has moved from mono-lingual, domain-tailored, knowledge-based IE systems to multilingual trainable IE systems that deploy weakly supervised ML techniques which automate to a large extent the entire IE customisation process. In [6] the paradigm of *Open Information Extraction* (OIE) was introduced, that aims to facilitate domain-independent discovery of relations extracted from texts and to scale to heterogeneous and large-size corpora such as the Web. An OIE system takes as input only a corpus of texts without any a priori knowledge or specification of the relations of interest and outputs a set of all extracted relations. The main rationale behind introducing the concept of OIE was to move away from the classic IE systems, where the relations of interest have to be specified prior to posing a query to the system, since adapting such a system to the extraction of a new relation type requires a training phase. Instead, an OIE system should: (a) discover all possible relations in the texts, using time linear in the number of documents in the corpus, and (b) create an IE-centric index that would support answering a broad range of unanticipated questions over arbitrary relations. Transforming classic IE methods and systems into fully unsupervised OIE technologies that can handle large-size and diverse corpora such as Web

would have various immediate applications, e.g., factual web-based Q/A systems capable of handling complex relational queries, and intelligent indexing for search engines.

A first step towards OIE, namely tackling automated customisation and handling heterogeneity, was presented with KnowItAll [18] and similar systems. KnowItAll is a self-supervised Web extraction system that uses a small set of domain-independent extraction patterns (e.g., `<A> is a <B>`) to automatically instantiate "reliable", relation-specific extraction rules (training data), which are than used to learn domain-specific extraction rules iteratively in a bootstrapping process. It relies solely on part-of-speech tagging, and requires neither a NER component nor a parser, i.e., no features which rely on syntactic or semantic analysis are required, which helps to better tackle the problem of heterogeneity and scaling to other languages. However, the system requires: (a) a large number of search engine queries in the process of assessing the reliability of the candidate relation-specific extraction rules, (b) providing the name of the relation of interest to the system, and (c) a new learning cycle in case of adding a new relation. An OIE system TextRunner, described in [75], addresses the aforementioned problems, i.e., it needs just one pass through the corpus without the need to name any relations of interest in advance. TextRunner first learns a general model of how relations are expressed in a particular language using CRF paradigm.[20] The creators of the system claimed that most relations in English can be characterized by a set of several lexico-syntactic patterns, e.g., `Entity-1 Verb Prep Entity-2`. In the second phase, the system scans each sentence and uses the model to assign to each word labels that denote the beginning/end of an entity or string describing the relation. Analogously to KnowItAll, the model uses only low-level linguistic features such as part-of-speech, token type (e.g., capitalisation), closed-word classes, etc., which is attractive from the perspective of handling genre diversity and different languages. For each sentence the system returns one or more triples, each representing a binary relation between two entities (e.g., `(Paris, capital-of, France)`), which is accompanied by a probability of the triple (relation) being extracted correctly, relying primarily on information about the frequency of occurrence on the Web. An evaluation of TextRunner in [7] revealed that it achieves on average 75% precision, however, the output of the system is unnormalised to a large extent, e.g., problems of matching names referring to the same real-world entity or identifying synonymous relations were not handled. Some improvements of the system in terms of handling relation synonymy problem, which resulted in improving the overall recall, are also described in [7]. Related work on OIE, similar in spirit, is reported in [61], focusing on avoiding relation-specificity in the relation extraction task, but it does not scale well to Web size.

A comparison of the performance of an OIE system versus a traditional relation extraction system is given in [7]. In particular, variants of the aforementioned TextRunner system were compared against a relation extraction system that uses

[20] Trained on extractions heuristically generated from PennTreebank.

the same CRF-based extraction model, but which was trained from hand-labelled data and which used a richer feature set, including lexical features. Experiments on extracting binary relations such as "corporate acquisitions" or "inventors of products" revealed that both systems achieve comparable precision (circa 75%), whereas the traditional extraction system achieved significantly higher recall, i.e., circa 60% vs. 20% achieved by OIE system. A hybrid system, combining the two using stacked generalisation, was reported to achieve a 10% relative improvement in precision with slight deterioration of the recall over the classic extraction system. In case higher level of recall is more important, in the context of extraction of binary relations, as the ones mentioned above, using traditional relation extraction is still by far more effective, although an OIE system might be deployed to reduce the number of hand-labeled training data. If, however, precision is more important and the number of relations is large, then an OIE system might potentially constitute a better alternative. One of the most frequent errors of the early OIE systems such as TEXTRUNNER were: (a) incoherent extractions (i.e., no meaningful interpretation of the relation phrase), (b) uninformative extractions (critical information omitted in relation phrases), and (c) incorrect or improperly scoped arguments (early OIE systems did not handle complex arguments, e.g., complex NPs with relative clauses). In [19], new OIE systems were introduced which try to tackle the aforementioned problems based on more fine-grained linguistic analysis of the structure of strings/phrases denoting relations and their arguments in English, which lead to further improvements.

The area of OIE, briefly described here, and referred work indicate progress in tackling Web-scale IE and further reduction of human involvement in the customization process, however, it is important to note that most of the work on OIE exhibits certain limitations since it is mainly focused on the extraction of binary relations within the scope of sentence boundaries. Attempts at devising OIE methods to extract more complex structures have been reported; e.g., [20] addresses the application of OIE techniques to the extraction of ordered sequences from the Web—sequence name and a set of ordered pairs, where the first element is the string naming the member of the sequence and the second element represents the position of that member, e.g., ordered list of the presidents of a given country. OIE for events, which have time, location, and potentially involve extraction of more than one relation that go beyond sentence level, constitutes a challenging task to be studied in the future. Since most of the work on OIE focused on English, the portability of existing OIE algorithms to other languages also needs to be investigated. Furthermore, the applicability of OIE to corpora consisting of short text messages might constitute an interesting area for research. Finally, another line of research might focus on the integration of some inference mechanisms in OIE systems, that would allow reasoning based on the facts they extract from texts.

References

1. Andersen, P., Hayes, P., Huettner, A., Schmandt, L., Nirenburg, I., Weinstein, S.: Automatic extraction of facts from press releases to generate news stories. In: Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLC '92, Trento, pp. 170–177. Association for Computational Linguistics, Stroudsburg (1992)
2. Aone, C., Halverson, L., Hampton, T., Ramos-Santacruz, M., Hampton, T.: SRA: description of the IE2 system used for MUC-7 In: Proceedings of MUC-7. Morgan Kaufmann, Columbia (1999)
3. Aone, C., Ramos-Santacruz, M.: REES: a large-scale relation and event extraction system. In: Proceedings of the 6th Conference on Applied Natural Language Processing, ANLP 2000, Seattle, pp. 76–83. Association for Computational Linguistics, Stroudsburg (2000)
4. Appelt, D.: Introduction to information extraction. AI Commun.**12**, 161–172 (1999)
5. Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., Amigï£¡, E.: WePS-3 evaluation campaign: overview of the web people search clustering and attribute extraction tasks. In: Braschler, M., Harman, D., Pianta, E. (eds.) CLEF (Notebook Papers/LABs/Workshops), Padua (2010)
6. Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of the 20th International Joint Conference on Artifical intelligence, Hyderabad, pp. 2670–2676. Morgan Kaufmann, San Francisco (2007)
7. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Proceedings of ACL-08: HLT, Columbus, pp. 28–36. Association for Computational Linguistics, Columbus (2008)
8. Benson, E., Haghighi, A., Barzilay, R.: Event discovery in social media feeds. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Vol. 1, pp. 389–398. Association for Computational Linguistics, Stroudsburg (2011)
9. Bikel, D., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Proceedings of the 5th Applied Natural Language Processing Conference, Washington. Association for Computational Linguistics, Washington, DC (1997)
10. Califf, M.E., Greenwood, M.A., Stevenson, M., Yangarber, R. (eds.): In: Proceedings of the Workshop on Information Extraction Beyond The Document. COLING/ACL, Sydney (2006)
11. Charniak, E.: Statistical Language Learning. MIT, Cambridge (1993)
12. Chen, Z., Ji, H.: Can one language bootstrap the other: a case study on event extraction. In: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, SemiSupLearn '09, Boulder, pp. 66–74. Association for Computational Linguistics, Stroudsburg (2009)
13. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. University of Maryland, College Park (1999)
14. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia (2002)
15. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: Automatic content extraction (ACE) program – task definitions and performance measures. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004) (2004)
16. Downey, D., Etzioni, O., Soderland, S.: A probabilistic model of redundancy in information extraction. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05, Edinburgh, pp. 1034–1041. Morgan Kaufmann, San Francisco (2005)
17. Drożdżyński, W., Krieger, H.U., Piskorski, J., Schäfer, U., Xu, F.: Shallow processing with unification and typed feature structures – foundations and applications. Künstliche Intell. **1/04**, 17–23 (2004)

18. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D., Alexander, A.: Unsupervised named-entity extraction from the web: an experimental study. Artif. Intell. **165**, 91–134 (2005)
19. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam: Open information extraction: the second generation. In: Proceedings of IJCAI 2011, Barcelona, pp. 3–10 (2011)
20. Fader, A., Soderland, S., Etzioni, O.: Extracting sequences from the web. In: Proceedings of the ACL 2010 Conference Short Papers, Uppsala, pp. 286–290. Association for Computational Linguistics, Uppsala (2010)
21. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Michigan, pp. 363–370. Association for Computational Linguistics, Stroudsburg (2005)
22. Gao, J., Wu, A., Li, M., ning Huang, C.: Chinese word segmentation and named entity recognition: a pragmatic approach. Comput. Linguist. **31**, 574 (2005)
23. Grishman, R., Sundheim, B.: Message understanding conference – 6: a brief history. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING), Kopenhagen, pp. 466–471. The Association for Computational Linguistics, Stroudsburg (1996)
24. Hobbs, J.R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., Tyson, M.: FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In: Roche, E., Schabes, Y. (eds.) Finite State Language Processing. MIT, Cambridge (1997)
25. Humphreys, K., Gaizauskas, R., Huyck, C., Mitchell, B., Cunningham, H., Wilks, Y.: University of sheffield: description of the LaSIE-II system and used for MUC-7. In: Proceedings of MUC-7, Virginia. SAIC (1998)
26. Huttunen, S., Yangarber, R., Grishman, R.: Complexity of event structure in information extraction. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002). Taipei (2002)
27. Iida, R., Poesio, M.: A cross-lingual ILP solution to zero anaphora resolution. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, The Association for Computer Linguistics, Portland, Oregon, 19–24 June 2011, pp. 804–813 (2011)
28. Jacobs, P., Rau, L.: SCISOR: extracting information from on-line news. Commun. ACM **33**, 88–97 (1990)
29. Ji, H.: Challenges from information extraction to information fusion. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, pp. 507–515. Association for Computational Linguistics, Stroudsburg (2010)
30. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. In: Proceedings of ACL-08: HLT, pp. 254–262. Association for Computational Linguistics, Columbus (2008)
31. Jones, R., Ghani, R., Mitchell, T., Riloff, E.: Active learning for information extraction with multiple view feature sets. ECML-03 Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik (2003)
32. Kaiser, K., Miksch, S.: Information extraction – a survey. Tech. Rep. Asgaard-TR-2005-6, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna (2005)
33. Lee, A., Passantino, M., Ji, H., Qi, G., Huang, T.S.: Enhancing multi-lingual information extraction via cross-media inference and fusion. In: COLING (Posters), Beijing, pp. 630–638 (2010)
34. Lehnert, W., Cardie, C., Fisher, D., McCarthy, J., Riloff, E., Soderland, S.: University of Massachusetts: MUC-4 test results and analysis. In: Proceedings of the 4th Message Understanding Conference. Morgan Kaufmann, McLean (1992)
35. Lehnert, W., Cardie, C., Fisher, D., Riloff, E., Williams, R.: University of Massachusetts: Description of the CIRCUS system as used for MUC-3. In: Proceedings of the 3rd Message Understanding Conference. Morgan Kaufmann, San Diego (1991)

36. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of ACL 2010, Uppsala, pp. 789–797. ACL (2010)
37. Liu, X., Zhang, S., Wei, F., Zhou, M.: Recognizing named entities in tweets. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. Human Language Technologies, Vol. 1, pp. 359–367. Association for Computational Linguistics, Stroudsburg (2011)
38. Llytinen, S., Gershman., A.: ATRANS: automatic processing of money transfer messages. In: Proceedings of the 5th National Conference of the American Association for Artificial Intelligence. IEEE Computer Society Press (1986)
39. Locke, B., Martin, J.: Named entity recognition: adapting to microblogging. Senior Thesis, University of Colorado, Colorado (2009)
40. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of DARPA Broadcast News Workshop, Herndon (1999)
41. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT, Cambridge, MA (1999)
42. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural elements of language engineering robustness. J Nat. Lang. Engin. **8**(2/3), 257–274 (2002)
43. Mohri, M., Nederhof, M.: Regular approximation of context-free grammars through transformations. In: Junqua, J., van Noord, G. (eds.) Robustness in Language and Speech Technology, pp. 153–163. Kluwer, The Netherlands (2001)
44. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguist. Investig. **30**(1), 3–26 (2007)
45. Neumann, G., Piskorski, J.: A shallow text processing core engine. Comput. Intell. **18**, 451–476 (2002)
46. Ng, V., Cardie, C.: Combining sample selection and error-driven pruning for machine learning of coreference rules. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, Philadelphia, pp. 55–62 (2002)
47. Patwardhan, S., Riloff, E.: Effective information extraction with semantic affinity patterns and relevant regions. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 717–727 (2007)
48. Phillips, W., Riloff, E.: Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002) (2002)
49. Piskorski, J.: ExPRESS – extraction pattern recognition engine and specification suite. In: Proceedings of FSMNLP 2007 (2007)
50. Piskorski, J., Belayeva, J., Atkinson, M.: On refining real-time multilingual news event extraction through deployment of cross-lingual information fusion techniques. In: EISIC, pp. 38–45. IEEE (2011)
51. Piskorski, J., Tanev, H., Atkinson, M., van der Goot, E., Zavarella, V.: Online news event extraction for global crisis surveillance. Trans. Comput. Collectiv. Intell. (5) (2011)
52. Piskorski, J., Wieloch, K., Sydow, M.: On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. Inf. Retr. **12**(3), 275–299 (2009)
53. Poibeau, T., Saggion, H. (eds.): In: Proceedings of the MMIES Workshop, RANLP: International Conference on Recent Advances in Natural Language Processing. Borovets, Bulgaria (2007)
54. Poibeau, T., Saggion, H., Yangarber, R. (eds.): In: Proceedings of the MMIES Workshop, COLING: International Conference on Computational Linguistics. Manchester (2008)
55. Recasens, M., Marquez, L., Sapena, E., Martï£¡, A., Taule, M., Hoste, V., Poesio, M., Versley, Y.: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010), ACL 2010. Uppsala, Sweden. In: SemEval-2010 Task 1: Coreference Resolution in Multiple Languages, pp. 1–8 (2010)
56. Riloff, E.: Automatically constructing a dictionary for information extraction tasks. In: Proceedings of Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington, DC, pp. 811–816. AAAI/MIT (1993)

57. Riloff, E.: Automatically generating extraction patterns from untagged text. In: Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, pp. 1044–1049. AAAI/MIT (1996)
58. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pp. 1524–1534. Association for Computational Linguistics, Edinburgh/Scotland (2011)
59. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of WWW 2010, Raleigh, pp. 851–860. ACM (2010)
60. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, pp. 104–107. NLPBA (2004)
61. Shinyama, Y., Sekine, S.: Preemptive information extraction using unrestricted relation discovery. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06, pp. 304–311. Association for Computational Linguistics, Stroudsburg (2006)
62. Sidner, C.L., Schultz, T., Stone, M., Zhai, C. (eds.): Multi-Document Relationship Fusion via Constraints on Probabilistic Databases. The Association for Computational Linguistics (2007)
63. Stevenson, M., Greenwood, M.A.: A semantic approach to IE pattern induction. In: Knight, K., Ng, H.T., Oflazer, K. (eds.) ACL. The Association for Computer Linguistics (2005)
64. Sudo, K., Sekine, S., Grishman, R.: Cross-lingual information extraction system evaluation. In: Proceedings of the 20th International Conference on Computational Linguistics, COLING '04. Association for Computational Linguistics, Stroudsburg (2004)
65. Tanev, H., Piskorski, J., Atkinson, M.: Real-time news event extraction for global crisis monitoring. In: Proceedings of NLDB 2008, London, pp. 207–218 (2008)
66. Thelen, M., Riloff, E.: A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002) (2002)
67. Verma, S., Vieweg, S., Corvey, W., Palen, L., Martin, J., Palmer, M., Schram, A., Anderson, K.: Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In: Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM 2011), Barcelona, pp. 385–392. AAAI (2011)
68. Vieweg, S., Hughes, A., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, pp. 1079–1088. ACM, New York (2010)
69. Wagner, E.J., Liu, J., Birnbaum, L., Forbus, K.D., Baker, J.: Using explicit semantic models to track situations across news articles. In: Proceedings of the 2006 AAAI Workshop on Event Extraction and Synthesis, pp. 42–47 (2006)
70. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques, 2nd edn. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)
71. Yangarber, R.: Counter-training in discovery of semantic patterns. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo (2003)
72. Yangarber, R.: Verification of facts across document boundaries. In: Proceedings IIIA-2006: International Workshop on Intelligent Information Access, IIIA-2006 (2006)
73. Yangarber, R., Jokipii, L.: Redundancy-based correction of automatically extracted facts. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pp. 57–64. Association for Computational Linguistics, Stroudsburg (2005)
74. Yangarber, R., Lin, W., Grishman, R.: Unsupervised learning of generalized names. In: Proceedings of COLING: the 19th International Conference on Computational Linguistics, Taipei (2002)

75. Yates, A., Banko, M., Broadhead, M., Cafarella, M.J., Etzioni, O., Soderland, S.: Textrunner: Open information extraction on the web. In: HLT-NAACL (Demonstrations), Rochester, pp. 25–26 (2007)
76. Zavarella, V., Tanev, H., Piskorski, J.: Event extraction for Italian using a cascade of finite-state grammars. In: Proceedings of FSMNLP 2008, Ispra (2008)

http://www.springer.com/978-3-642-28568-4

Multi-source, Multilingual Information Extraction and Summarization

Poibeau, T.; Saggion, H.; Piskorski, J.; Yangarber, R. (Eds.)

2013, XX, 324 p., Hardcover
ISBN: 978-3-642-28568-4