Lang Resources & Evaluation (2008) 42:361–393
DOI 10.1007/s10579-008-9079-3

Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations

Alberto Lavelli · Mary Elaine Califf · Fabio Ciravegna · Dayne Freitag · Claudio Giuliano · Nicholas Kushmerick · Lorenza Romano · Neil Ireson

Published online: 5 December 2008

Abstract We survey the evaluation methodology adopted in information extraction (IE), as defined in a few different efforts applying machine learning (ML) to IE. We identify a number of critical issues that hamper comparison of the results obtained by different researchers. Some of these issues are common to other NLP-related tasks: e.g., the difficulty of exactly identifying the effects on performance of the data (sample selection and sample size), of the domain theory (features selected), and of algorithm parameter settings. Some issues are specific to IE: how leniently to assess inexact identification of filler boundaries, the possibility of multiple fillers for a slot, and how the counting is performed. We argue that, when specifying an IE task, these issues should be explicitly addressed, and a number of methodological characteristics should be clearly defined. To empirically verify the practical impact of the issues mentioned above, we perform a survey of the results of different algorithms when applied to a few standard datasets. The survey shows a serious lack of consensus on these issues, which makes it difficult to draw firm conclusions on a comparative evaluation of the algorithms. Our aim is to elaborate a clear and

A. Lavelli (✉) · C. Giuliano · L. Romano
FBK-irst, via Sommarive 18, 38100 Povo, TN, Italy
e-mail: lavelli@fbk.eu

M. E. Califf
Illinois State University, Normal, IL, USA

F. Ciravegna · N. Ireson
University of Sheffield, Sheffield, UK

D. Freitag
Fair Isaac Corporation, San Diego, CA, USA

N. Kushmerick
Decho Corporation, Seattle, WA, USA

detailed experimental methodology and propose it to the IE community. Widespread agreement on this proposal should lead to future IE comparative evaluations that are fair and reliable. To demonstrate the way the methodology is to be applied we have organized and run a comparative evaluation of ML-based IE systems (the Pascal Challenge on ML-based IE) where the principles described in this article are put into practice. In this article we describe the proposed methodology and its motivations. The Pascal evaluation is then described and its results presented.

Keywords Evaluation methodology · Information extraction · Machine learning

1 Introduction

Evaluation has a long history in information extraction (IE), mainly thanks to the MUC conferences, where most of the IE evaluation methodology (as well as most of the IE methodology as a whole) was developed (Hirschman 1998). In this context, annotated corpora were produced and made available.

More recently, a variety of other corpora have been shared by the research community, such as Califf's job postings collection (Califf 1998), and Freitag's seminar announcements, corporate acquisition and university Web page collections (Freitag 1998). These more recent evaluations have focused not on the IE task *per se* (as in the MUC conferences), i.e. on the ability to extract information, but more on the ability to *learn* to extract information. This different focus on machine learning (ML) aspects has implications on the type of evaluation carried out. While a focus on IE means testing the extraction capabilities independently of the way in which results were obtained, an ML-oriented evaluation also focuses on the way results were obtained. For example it is important to focus on aspects such as the features used by the learner in order to understand if some results are obtained thanks to a new algorithm or thanks to a more powerful set of features (or maybe thanks to their combination). Also, the tasks that are possible to perform using ML (e.g., named entity recognition, implicit relation extraction) are definitely less complex than those possible when a human developer is in the loop (e.g., event extraction involving coreference resolution and domain-based reasoning). In this article we focus on evaluation of ML-oriented IE tasks, although many of the issues are relevant to IE in general.

In general, we claim that the definition of an evaluation methodology and the availability of standard annotated corpora do not guarantee that the experiments performed with different approaches and algorithms proposed in the literature can be reliably compared. Some obstacles to fair comparison are common to other ML-based NLP tasks, while some are specific to information extraction. In common with other NLP tasks, IE evaluation faces difficulties in exactly identifying the effects on performance of the data used (sample selection and sample size), of the information sources used (feature selection), and of algorithm parameter settings (Daelemans and Hoste 2002; Hoste et al. 2002; Daelemans et al. 2003).

Issues specific to IE evaluation include:

- *Fragment evaluation*: How leniently should inexact identification of filler boundaries be assessed?
- *Counting multiple matches*: When a learner predicts multiple fillers for a slot, how should they be counted?
- *Filler variation*: When text fragments having distinct surface forms refer to the same underlying entity, how should they be counted?
- *Evaluation platform*: Should researchers employ a previously implemented scorer or (as happens quite frequently) write their own?

Because of the complexity of the task, the limited availability of tools, and the difficulty of reimplementing published algorithms (usually quite complex and sometimes not fully described in papers), in IE there are very few comparative articles in the sense mentioned in Hoste (2002), Hoste et al. (2002), and Daelemans et al. (2003). Most of the papers simply present the results of the new proposed approach and compare them with the results reported in previous articles. There is rarely any detailed analysis to ensure that the same methodology is used across different experiments.

Given this predicament, it is obvious that a few crucial issues in IE evaluation need to be clarified. This article aims to provide a solid foundation for carrying out meaningful comparative experiments. To this end, we provide a critical survey of the different methodologies employed in the main IE evaluation tasks. In more detail, we make the following contributions:

1. We describe the IE evaluation methodology as defined in the MUC conference series and in related initiatives.
2. We identify a variety of methodological problems, some of which are common to many NLP tasks, and others of which are specific to IE.
3. We describe the main reference corpora used by IE researchers: their characteristics, how they have been evaluated, etc.
4. We propose an experimental methodology which future IE evaluations should follow in order to make comparisons between algorithms useful and reliable.
5. We describe an exercise of IE evaluation run as part of the Pascal European Network of Excellence to put the methodology into practice. 11 groups from the EU and the US participated in the evaluation.

The remainder of this article is organized as follows. First, we briefly identify the specific IE tasks with which we are concerned and briefly summarize prior IE research (Sect. 2). Then, we discuss in detail a variety of methodological problems that have hampered efforts to compare different IE algorithms (Sect. 3). We then describe in detail several benchmark corpora that have been used by numerous researchers to evaluate their algorithms (Sect. 4). Fourth, we spell out a recommended standard evaluation methodology that we hope will be adopted across the research community (Sect. 5). Fifth, we describe the way the methodology was implemented in the Pascal Challenge for ML-based IE evaluation. We conclude with an analysis of the lessons learned, and some suggestions for future work (Sect. 6).

2 What is "information extraction"?

In this section, we describe the specific kinds of information extraction tasks on which we focus in this article, and we clarify the relationship between IE and a variety of related natural language processing tasks.

As depicted in Fig. 1, we define ML-based IE as the process of identifying the specific fragments or substrings that carry a document's core meaning, according to some predefined information need or template. Depending on the requirements of the target application, the output of the IE process could be either annotations inserted into the original document, or external semantic references to spans of text from the original document. In general, these two methods are equivalent, and it is straightforward to translate back and forth.

It is essential to distinguish IE from information or document retrieval. Document retrieval systems identify entire documents from a large corpus that are relevant to a specific query. In contrast, IE highlights specific spans of text that have various semantic meanings.

As shown in Fig. 2, IE research has explored a spectrum of document classes. We do not claim that there are precise boundaries between one region of the spectrum and another, nor that IE tasks can be compared to one another on any single dimension. Rather, this spectrum helps to illuminate the relationship between

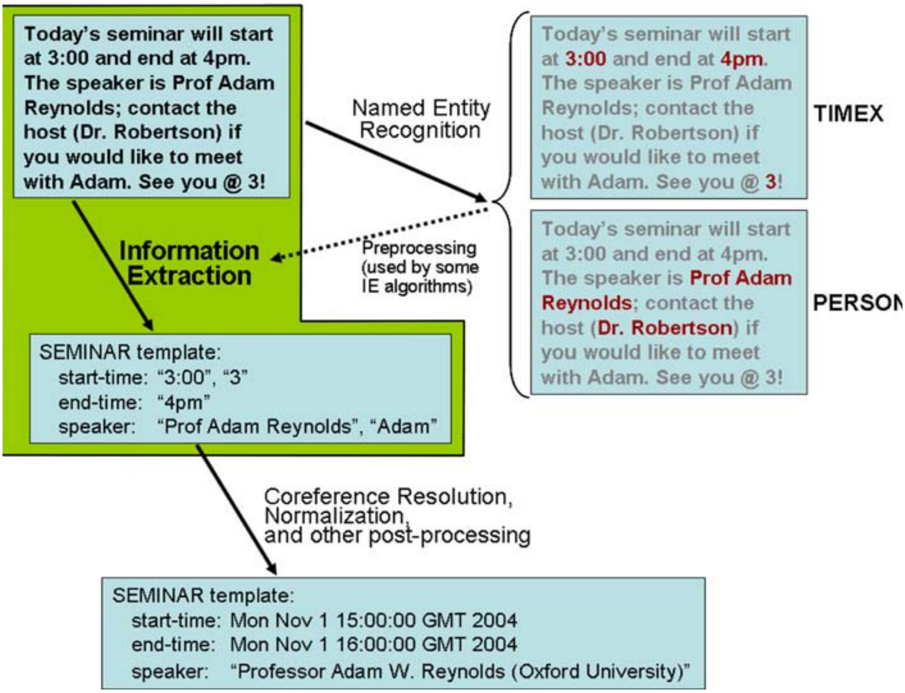
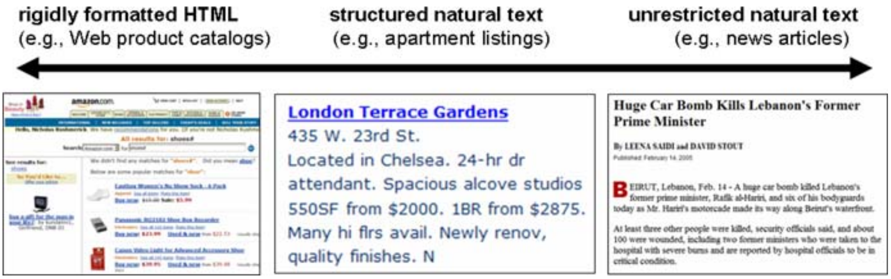

Today's seminar will start at 3:00 and end at 4pm. The speaker is Prof Adam Reynolds; contact the host (Dr. Robertson) if you would like to meet with Adam. See you @ 3!
Named Entity Recognition
TIMEX
PERSON
Information Extraction
Preprocessing (used by some IE algorithms)
SEMINAR template:
start-time: "3:00", "3"
end-time: "4pm"
speaker: "Prof Adam Reynolds", "Adam"
Coreference Resolution, Normalization, and other post-processing
SEMINAR template:
start-time: Mon Nov 1 15:00:00 GMT 2004
end-time: Mon Nov 1 16:00:00 GMT 2004
speaker: "Professor Adam W. Reynolds (Oxford University)"

Fig. 1 We define ML-based information extraction as the task of identifying specific fragments from text documents using ML means only


rigidly formatted HTML
(e.g., Web product catalogs)
structured natural text
(e.g., apartment listings)
unrestricted natural text
(e.g., news articles)
amazon.com

London Terrace Gardens
435 W. 23rd St.
Located in Chelsea. 24-hr dr attendant. Spacious alcove studios 550SF from $2000. 1BR from $2875. Many hi flrs avail. Newly renov, quality finishes. N

Huge Car Bomb Kills Lebanon's Former Prime Minister

By LEENA SAIDI and DAVID STOUT
Published February 14, 2005

BEIRUT, Lebanon, Feb. 14 - A huge car bomb killed Lebanon's former prime minister, Rafik al-Hariri, and six of his bodyguards today as Mr. Hariri's motorcade made its way along Beirut's waterfront.

At least three other people were killed, security officials said, and about 100 were wounded, including two former ministers who were taken to the hospital with severe burns and are reported by hospital officials to be in critical condition.

Fig. 2 Information extraction has explored a spectrum of document classes, from rigidly structured HTML to free-formatted natural text

IE as defined in this article, and other similar forms of natural language processing or document analysis.

At one end of the spectrum lie rigidly formatted texts, such as HTML, that are automatically created by instantiating a template with objects selected from a database. The term "wrapper induction" has been used for the application of ML techniques to IE from highly structured documents such as product catalogs, search engine result lists, etc. (Kushmerick 2000). Wrapper induction is an interesting and practical special case of IE, but we ignore it in this article, because the evaluation issues that we discuss rarely arise. For example, in most wrapper induction applications, the structures to be extracted are easily specifiable, and the applications typically require perfect extraction, so evaluation questions such as how to define precision/recall simply do not arise.

At the other end of the spectrum are loosely structured natural language texts, such as news articles. These documents are characterized by degrees of inherent ambiguity (syntactic and semantic) and variation in word choice, complicating the information extraction process. On the other hand, these texts are usually highly grammatical, so that natural language processing techniques can be applied to help processing.

In the middle of the spectrum lie structured natural language text documents. For example, apartment listings and job advertisements usually employ a restricted vocabulary and telegraphic syntax that substantially simplifies the extraction process.

Having broadly identified the kind of tasks we are interested in the spectrum shown in Fig. 2, we now further restrict our area of interest. For the purposes of this article, we restrict the analysis to the task of *implicit relation extraction*. Implicit relation extraction is the task mainly dealt with by the wrapper induction community and the ML-based IE community. It requires the identification of implicit events and relations. For example Freitag (1998) defines the task of extracting speaker, start-time, end-time and location from a set of seminar announcements. No explicit mention of the event (the seminar) is done in the annotation. Implicit event extraction is simpler than full event extraction, but has important applications whenever either there is just one event per text or it is easy to devise extraction strategies for recognizing the event structure from the document

(Ciravegna and Lavelli 2004). This task is different from named-entity recognition (NER). The aim of NER is to recognize instances of common data types such as people, locations, organizations, or dates. As shown in Fig. 1, the IE we refer to may use the results of NER but it needs to make use of further contextual information to distinguish, for example, the speaker of a seminar from other people mentioned in a seminar announcement. Other tasks which are beyond the scope of this article are various forms of post-processing such as coreference resolution or normalization.

Moreover, in the kind of IE we are interested in, there is usually the simplifying assumption that each document corresponds to a single *event* (seminar announcement, job posting). The objective is to produce a structured summary (fill a *template*), the typed elements of which (*slots*) are the various details that make up the event in question. Since only a single event is involved, it is possible to identify the different elements of the template independently. However, even in this simplified type of IE a number of problematic issues arise and may hamper the comparative evaluation of different approaches and algorithms.

An event is a specific relation that holds among certain entities mentioned in a document. Our focus on single-event extraction excludes from consideration what is commonly called *relation extraction*. Relation extraction refers to the identification of certain relations that commonly hold between named entities (e.g., “ORGANIZATION *is located in* LOCATION”). Such relations are typically, though not necessarily, binary. Recently there has been a lot of activity in this field because of its practical importance. However, while the evaluation of relation extraction shares some challenges with single-event IE, it also introduces other challenges (among them the lack of widely accepted reference corpora) which are beyond the scope of this article.

2.1 A short history of information extraction

In what follows, we briefly summarize the main milestones in IE research, from the MUC conferences to the ACE program (Automatic Content Extraction) more recently carried out by NIST. Although none of them specifically focused on ML-based IE tasks and they used tasks far more complex than implicit relation recognition, it is useful to look at these experiences.

2.1.1 MUC conferences

The MUC conferences can be considered the starting point of IE evaluation methodology as currently defined. The MUC participants borrowed the Information Retrieval concepts of precision and recall for scoring filled templates. Given a system response and an answer key prepared by a human, the system’s precision was defined as the number of slots it filled correctly, divided by the number of fills it attempted. Recall was defined as the number of slots it filled correctly, divided by the number of possible correct fills, taken from the human-prepared key. All slots were given the same weight. F-measure, a weighted combination of precision and recall, was also introduced to provide a single figure to compare different systems’ performance. In Makhoul et al. (1999) some limitations of F-measure are underlined, and a new measure, slot error rate, is proposed. Although the proposal is interesting, it does not seem to have had any impact on the IE community, which continues to employ F-measure as the standard way of comparing systems' performance.

Apart from the definition of precise evaluation measures, the MUC conferences made other important contributions to the IE field: the availability of a large amount of annotated data (which has made possible the development of ML based approaches), the emphasis on domain-independence and portability, and the identification of a number of different tasks which can be evaluated separately.

In particular, the MUC conferences made available annotated corpora for training and testing,[1] along with evaluation software (i.e., the MUC scorer (Douthat 1998)).

MUC-7 defined and evaluated the following tasks (description taken from Hirschman (1998)):

Named Entity: Identification of person (PERSON), location (LOC) and organization (ORG) names, as well as time, date and money expressions. At MUC-6 the highest performing automated Named Entity system was able to achieve a score comparable to human-human interannotator agreement. At MUC-7 the results were lower because of the absence of training data for the satellite launch domain.

Coreference: Identification of coreferring expressions in the text, including name coreference (*Microsoft Corporation* and *Microsoft*), definite reference (*the Seattle-based company*) and pronominal reference (*it, he, she*). This was the most difficult of the tasks.

Template Element: Identification of the main entities (persons, organizations, locations), with one template per entity including its name, other "aliases" or shortened forms of the name, and a short descriptive phrase useful in characterizing it. The template elements constituted the building blocks for the more complex relations captured in template relation and scenario template tasks.

Template Relation: Identification of properties of Template Elements or relations among them (e.g., *employee_of* connecting person and organization, or *location_of* connecting organization and location). This task was introduced in MUC-7.

Scenario Template: Extraction of predefined event information and link of the event information to particular organization, person or artifact entities involved in the event. At MUC-7 the scenario concerned satellite launch events and the event template consisted of 7 slots.

It should be noticed that MUC evaluation concentrated mainly on IE from relatively unrestricted text, i.e. newswire articles.

2.1.2 ML-based IE evaluations

In independent efforts, other researchers created and made available annotated corpora developed from somewhat more constrained texts where the task was

[1] The corpora for MUC-3 and MUC-4 are freely available in the MUC web site (http://www-nlpir.nist.gov/related_projects/muc), while those of MUC-6 and MUC-7 can be purchased via the Linguistic Data Consortium (http://ldc.upenn.edu).

mainly related to the main topic of this article: ML-based IE for implicit relation extraction. Califf compiled and annotated a set of 300 job postings from the Internet (Califf 1998), and Freitag compiled corpora of seminar announcements and university web pages, as well as a corporate acquisitions corpus from newswire texts (Freitag 1998). Several of these corpora are available from the RISE repository (RISE 1998) where a number of tagged corpora have been made available by researchers in Machine Learning for IE: e.g., Seminar Announcements (Freitag 1998), Job Postings (Califf 1998). Further specific details about such corpora will be provided in Sect. 4.

In the Seminar Announcement collection (Freitag 1998), the templates are simple and include slots for the seminar speaker, location, start time, and end time. This is in strong contrast with what happened in the last MUC conferences (such as MUC-6 and MUC-7) where templates might be nested (i.e., the slot of a template may take another template as its value), or there might be several templates from which to choose, depending on the type of document encountered. In addition, MUC data sets include irrelevant documents which the extraction system should ignore. A template slot may be filled with a lower-level template, a set of strings from the text, a single string, or an arbitrary categorical value that depends on the text in some way (a so-called “set fill”).

Califf (1998) takes an approach that is somewhat in-between Freitag’s approach and more complex MUC extraction tasks. All of the documents are relevant to the task, and the assumption is that there is precisely one template per document, but that many of the slots in the template can have multiple fillers.

Although the tasks to be accomplished are different, the methodologies adopted by Freitag (1998) and Califf (1998) are similar to the one used in the MUC competition: precision, recall, and F-measure are employed as measures of the performance of the systems.

In cases where elaborate representations (nested templates, set fills) are required of a system, the task’s difficulty may approach that of full NLP. In general, the challenges facing NLP cannot be circumvented in IE. Some semantic information and discourse-level analysis is typically required. To this are also added sub-problems unique to IE, such as slot filling and template merging.

2.1.3 ACE program

More recently, NIST started the ACE (Automatic Content Extraction) program.[2] The objective of the ACE program is to develop automatic content extraction technology to support automatic processing of human language in text form. The corpora used in the program include different source types: newswire, broadcast news, broadcast conversation, weblog, usenet (newsgroups/discussion forum), conversational telephone speech.

[2] http://www.nist.gov/speech/tests/ace.

The ACE research objectives are viewed as the detection and characterization of *entities* (Entity Detection and Recognition, EDR), *relations* (Relation Detection and Recognition, RDR), and *events* (Event Detection and Recognition, VDR). In each of the above tasks certain specified types of entities (relations, events) that are mentioned in the source language data have to be detected and selected information about these entities (relations, events) has to be recognized and merged into a unified representation for each detected entity (relation, event).

Entity Detection and Recognition (EDR) is the core annotation task, providing the foundation for all remaining tasks. The goal of this task is to identify seven types of entities: Person, Organization, Location, Facility, Weapon, Vehicle and Geo-Political Entity (GPEs). Each type may be further divided into subtypes (for instance, Organization subtypes include Government, Commercial, Educational, ...). Annotators tag all mentions of each entity within a document.

Relation Detection and Recognition (RDR) involves the identification of relations between entities. The definition of RDR targets physical relations (e.g., Located, Near and Part-Whole), social/personal relations (e.g., Business, Family and Other), a range of employment or membership relations, relations between artifacts and agents (including ownership), affiliation-type relations like ethnicity, relationships between persons and GPEs like citizenship, and finally discourse relations. For every relation, annotators identify two primary arguments (namely, the two ACE entities that are linked) as well as the relation's temporal attributes.

Event Detection and Recognition (VDR) This is the most experimental ACE task and it was performed for the first time during the 2005 evaluation (only for Chinese and English). It requires the recognition of events involving entities and time expressions.

The ACE 2007 evaluation included four languages (English, Chinese, Arabic, and Spanish) and the recognition of temporal expressions was added, while in 2008 the number of languages was reduced (English and Arabic only) and the tasks were modified (with both within-document and cross-document recognition). Concerning the 2008 tasks, only entities (EDR) and relations (RDR) were considered. Only the original five ACE entities were addressed for within-document EDR, while cross-document EDR was limited only to entities of type Person and Organization. The ACE training and development annotated data are made available via the Linguistic Data Consortium (http://ldc.upenn.edu).

3 Critical issues in ML-based IE evaluation

Despite the definition of an evaluation methodology and the availability of standard annotated corpora, there is no guarantee that the experiments performed by different researchers using various algorithms can be reliably compared. In this section we discuss obstacles standing in the way of transparent comparisons. Some of these problems are common to many types of empirical research. Others bedevil any kind of work in IE and may have been partially addressed in the context of MUC and ACE. Still others are particular to the evaluation of machine learning-based approaches to IE.

There are three broad categories into which these challenges fall:

- Data problems.
- Problems of experimental design.
- Problems of presentation.

In this section we consider each of these categories in turn, enumerating the questions and challenges specific to each. Some of these questions do not have an easy answer. Some, however, can be addressed by community consensus.

3.1 Data problems

Many of the problem domains shared by the IE community were contributed by individual researchers who, identifying underexplored aspects of the IE problem, produced reference corpora on their own initiative, following conventions and procedures particular to their own experiments. It was perhaps inevitable that subsequent use of these corpora by other parties identified errors or idiosyncrasies.

Errors in data: *Errors* range from illegal syntax in the annotation (e.g., a missing closing tag in XML) to unidentified or mis-identified slot fillers, to inconsistently applied slot definitions. The most frequently used corpora have undergone considerable scrutiny over the years, and in some cases corrected versions have been produced.

Branching Corpora: While correction of data errors can only lend clarity, incomplete follow-through leads to the problem of *branching corpora*. A privately corrected corpus raises questions concerning the extent to which any observed improvements are due to improvements in the training data.

Mark-up vs. Templates: There are at least two ways in which the annotations required for IE may be provided: either through annotation of textual extents in the document (e.g., using tags), or in the form of a populated template. These two alternatives are each employed by one of the two most frequently used reference corpora: the Seminar Announcements corpus employs tags, while the Job Postings corpus uses templates. While transforming tagged texts into templates can be considered straightforward, the reverse is far from obvious and differences in the annotations can produce relevant differences in performance. For example, in one of the tagged versions of the Job Postings corpus, a document's string `NNTP` in the email headers was inadvertently tagged as `N<platform>NT</platform>P`, because the string `NT` appeared in the “‘platform” slot of the document's template.

Common Format: This leads to the more general issue of data format. In an ideal world, the community would agree on a single, well-documented format (e.g., XML with in-line annotation) and common software libraries would be provided to factor out any differences due to format. Note that annotation format can have subtle influences on performance. The in-line annotation in Fig. 3 (forward reference) may be inadvertently used by a text tokenizer, leading to skewed test results.

```
Synergistic transactivation of the BMRF1 promoter by
the <protein>Z</protein>/<protein>c-myb</protein>
combination ...
```

Fig. 3 An example of protein annotation taken from the BioCreAtIvE corpus

3.2 Problems of experimental design

Given reasonably clean training data, there are many ways in which an empirical study in IE can be structured and conducted. This section analyzes challenges of experimental design, some of them common to other NLP tasks (e.g., see (Daelemans and Hoste 2002; Hoste et al. 2002; Daelemans et al. 2003)) and in general to any empirical investigation, others particular to IE and related endeavors. These challenges include exactly identifying the effects on performance of the data used (the sample selection and the sample size) or of representation (the features selected), choosing appropriate parameter settings, and using metrics that yield the greatest insight into the phenomena under study. For any given challenge in this category, there are typically many valid answers; the critical thing is that the researcher explicitly specify how each challenge is met.

Training/Testing Selection: One of the most relevant issues is that of the exact split between training set and test set, considering both the numerical proportions between the two sets (e.g., a 50/50 split *vs*. a 80/20 one) and the procedure adopted to partition the documents (e.g., *n* repeated random splits *vs*. *n*-fold cross-validation).

Tokenization: Another relevant concern is tokenization, which is often considered something obvious and non-problematic. However, it has a larger influence on performance than is often acknowledged (Habert et al. 1998), and can certainly affect the performance of IE algorithms. As in other areas of NLP, consistency in tokenization is required. In the worst case, if the tokenizer does not adopt the right policy, correct identification of slot fillers may be impossible. Consider, for example, the protein identification example shown in Fig. 3 (sampled from the BioCreAtIvE corpus[3]). Here, the handling of characters such as “-” and “/” certainly has an impact on performance.

Impact of Features: In accounting for the performance of an approach, it is also important to distinguish between the learning algorithm and the features employed. In IE, for instance, some approaches have employed simple orthographic features, while others have used more complex linguistic features, such as part-of-speech tags or semantic labels extracted from gazetteers (e.g., Califf 1998; Ciravegna 2001b; Peshkin and Pfeffer 2003).

Fragment Evaluation: A first issue is related to how to evaluate an extracted fragment—e.g., if an extra comma is extracted should it count as correct, wrong, partially correct? This issue is related to the question of how relevant is the exact identification of the boundaries of the extracted items. Freitag (1998) proposes three different criteria for matching reference instances and extracted instances:

Exact: The predicted instance matches exactly an actual instance.

Contains: The predicted instance strictly contains an actual instance, and at most *k* neighboring tokens.

Overlap: The predicted instance overlaps an actual instance.

Each of these criteria can be useful, depending on the situation, and it can be interesting to observe how performance varies with changing criteria. De Sitter and

[3] http://biocreative.sourceforge.net.

Daelemans (2003) mention such criteria and present the results of their algorithm for all of them.

Scorer: A second issue concerns which software has been used for the evaluation. The only such tool that is widely available is the MUC scorer. Usually IE researchers have implemented their own scorers, relying on a number of implicit assumptions that may have a strong influence on performance's evaluation.

How to Count Matches: When multiple fillers are possible for a single slot, there is an additional ambiguity—usually glossed over in papers—that can influence performance. For example, Califf and Mooney (2003) remark that there are differences in counting between RAPIER (Califf 1998), SRV (Freitag 1998), and WHISK (Soderland 1999). In his test on Job Postings, Soderland (1999) does not eliminate duplicate values. When applied to Seminar Announcements SRV and RAPIER behave differently: SRV assumes only one possible answer per slot, while RAPIER makes no such assumption since it allows for the possibility of needing to extract multiple independent strings.

De Sitter and Daelemans (2003) also discuss this question and note that in such cases there are two different ways of evaluating performance in extracting slot fillers: to find *all occurrences* (AO) of an entity (e.g. every mention of the job title in the posting) or only one occurrence for each template slot (one best per document, OBD). The choice of one alternative over the other may have an impact on the performance of the algorithm. De Sitter and Daelemans (2003) provide results for the two alternative ways of evaluating performance. This issue is often left underspecified in papers and, given the lack of a common software for evaluation, this further amplifies the uncertainty about the reported results.

Even in domains in which all slots are typically defined to be OBD, textual realities may deviate from this specification. While the seminar announcement problem was originally evaluated as OBD, Fig. 4 shows that, for some seminar announcements, this specification is not completely appropriate. Clearly, the performance recorded for such documents will depend on how these multiple slot fillers are accounted. Under AO, an algorithm must identify both speakers in order to be 100% correct.

Filler Variations: A problem closely related to but distinct from the issue of multiple fillers is that of multiple textual realizations for a single underlying entity. Figure 4 also shows examples of this phenomenon (“`Joel S. Birnbaum, Ph.D`”, “`Dr. Birnbaum`”, etc.). Such variations are common with people's names, but not limited to them (e.g., “7:00 P.M.”, “7pm”). Leaving aside the problem of normalization, how such variations are counted may also affect scores.

In light of these observations, we note that there are actually three ways to count:

- One Answer per Slot—OAS (where “2pm” and “2:00” are considered one correct answer)
- One Answer per Occurrence in the Document—OAOD (each individual appearance of a string has to be extracted in the document where two separate occurrences of “2pm” would be counted separately).[4]

[4] Note that the occurrences considered here are only those that can be interpreted without resorting to any kind of contextual reasoning. Hence, phenomena related to coreference resolution are not considered at all.

```
...
Who: <speaker>Joel S. Birnbaum, Ph.D</speaker>
Senior Vice President of Research and Development
Director, Hewlett-Packard Laboratories
...
Here are highlights of what you'll be hearing from some of
the speakers:

<speaker>Joel S. Birnbaum, Ph.D</speaker>
Senior Vice President of Research and Development
Director, Hewlett-Packard Laboratories

<speaker>Dr. Birnbaum</speaker>'s talk will set the stage
for the rest of the NTU broadcast.
...
<speaker>Jeff Eastman, Ph.D.</speaker>
Consulting Engineer, Hewlett-Packard
Software Engineering Systems Division

<speaker>Dr. Eastman</speaker> is the chief architect
of HP's Distributed Smalltalk.
...
```

Fig. 4 An example of multiple speaker tags in a seminar announcement

- One Answer per Different String—OADS (where two separate occurrences of "2pm" are considered one answer, but "2:00" is yet another answer)

Freitag takes the first approach, Soderland takes the second, and Califf takes the third.

3.3 Problems of presentation

Once experiments are run and the results gathered, the researcher faces the question which information to include in a report. While this is partly a question of style, choices in this area can affect the extent to which results from two papers can be compared. A lack of consensus concerning best practices may ultimately impede progress.

Learning Curve: The question of how to formalize the learning-curve sampling method and its associated cost-benefit trade-off may cloud comparison. For example, the following two approaches have been used: (1) For each point on the learning curve, train on some fraction of the available data and test on the remaining fraction; or (2) Hold out some fixed test set to be used for all points on the learning curve.

Statistical Significance: All too often, IE research merely reports numerical performance differences between algorithms, without analyzing their statistical properties. The most important form of analysis is whether some reported numerical difference is in fact statistically significant. One reason for this may be the occasional use of complicated scoring functions without an obvious formula for confidence bounds.

Slot or Domain Omission: One very common problem that complicates a sound comparison between different algorithms is the fact that some papers present results only on one of the major reference corpora (e.g., Seminar Announcements, Job Postings, etc.). For example, Roth and Yih (2001), Chieu and Ng (2002), and Peshkin and Pfeffer (2003) report results only on the Seminar Announcements[5] and Kosala and Blockeel (2000) and De Sitter and Daelemans (2003) only on the Job Postings. On the other hand, Freitag (1998) presents results on Seminar Announcements, corporate acquisition, and university web page collection, Califf (1998) on Seminar Announcements, corporate acquisition and also on Job Postings, and Ciravegna (2001a), Freitag and Kushmerick (2000), Finn and Kushmerick (2004b), and Finn and Kushmerick (2004a) on both Seminar Announcements and Job Postings.

F-measure but not Precision/Recall: Related to this issue is the fact that sometimes papers report only F-measure but not precision and recall, while the trade-off between precision and recall is a fundamental aspect of performance.

Complexity and Efficiency: A further issue concerns the computational complexity of the algorithms. It sometimes can be difficult to evaluate the complexity of the algorithms proposed because of the lack of a detailed enough description. And it is obviously difficult to fairly compare the practical performance in time and space of algorithms running with different hardware and software configurations. However, from the perspective of practical application, this is a relevant aspect to evaluate. For example, Kosala and Blockeel (2000) report that they used approximately one fifth to one half of the available training examples for the Job Postings dataset due to insufficient memory.

4 Reference corpora for IE

The datasets used more often in IE[6] are Job Postings (Califf 1998), Seminar Announcements, Reuters corporate acquisition, and the university web page collections (Freitag 1998). In the following we will describe the main characteristics of the first two of these corpora (set of fields to extract, standard train/test split, ...) together with tables showing the results published so far (precision, recall and F_1 on a per-slot basis as well as microaveraged over all slots[7]). In addition to reporting the results, we specify how the matches were counted by the algorithms, given that this issue turned out to be the most crucial difference between the different experiments.

[5] Although in Roth and Yih (2002) the results for Job Postings are also included. Moreover, Chieu and Ng (2002) report also results on Management Succession.

[6] Note that here we are not taking into account the corpora made available during the MUC conferences which, because of the complexity of the IE tasks, have been not very often used in IE experiments after the MUC conferences. Hirschman (1998) provides an overview of such corpora and of the related IE tasks.

[7] See footnote 14.

In Appendix a glossary listing the names/acronyms of the systems mentioned in the paper together with their full names and bibliographical references is provided.

4.1 Seminar announcements

The Seminar Announcement collection (Freitag 1998) consists of 485 electronic bulletin board postings distributed in the local environment at Carnegie Mellon University.[8] The purpose of each document in the collection is to announce or relate details of an upcoming talk or seminar. The documents were annotated for four fields: *speaker*, the name of seminar's speaker; *location*, the location (i.e., room and number) of the seminar; *stime*, the start time; and *etime*, the end time. Figure 5 shows an example taken from the corpus.

4.1.1 Methodology and results

Freitag (1998) randomly partitions the entire document collection five times into two sets of equal size, training and testing. The learners are trained on the training documents and tested on the corresponding test documents from each partition. The resulting numbers are averages over documents from all test partitions. In Freitag (1997), however, the random partitioning is performed ten times (instead of five). Later experiments have followed alternatively one of the two setups: e.g., Califf (1998), Freitag and Kushmerick (2000), Ciravegna (2001a), Finn and Kushmerick (2004b), Finn and Kushmerick (2004a), Li et al. (2005a) and Iria and Ciravegna (2006) follow the ten run setup;[9] Roth and Yih (2001), Chieu and Ng (2002) and Sigletos et al. (2005) follow the five run one; Peshkin and Pfeffer (2003) do the same as well[10] and provide results on each single slot but showing only F-measure. Sutton and McCallum (2004) and Finkel et al. (2005) report performance using 5-fold cross validation (but showing only F-measure). Finally, Soderland (1999) reports WHISK performance using 10-fold cross validation on a randomly selected set of 100 texts, instead of using the standard split for training and test sets.

In Table 1 we list the results obtained by different systems on Seminar Announcements, together with the information about how matches are counted (when available).

4.1.2 Learning curve

Peshkin and Pfeffer (2003) provides also the learning curves for precision and recall and F-measure on the Seminar Announcement collection. Trained on a small sample, BIEN rarely tries to tag, resulting in high precision and poor recall. When

[8] Downloadable from the RISE repository: http://www.isi.edu/info-agents/RISE/repository.html.

[9] Califf (1998), Freitag and Kushmerick (2000), and Finn and Kushmerick (2004a, b) use exactly the same partitions as Freitag (1997).

[10] What is written in their paper is not completely clear but they have confirmed to us that they have adopted the five run setup (personal communication).

```
<0.6.1.94.14.16.40.xu+@IUS4.IUS.CS.CMU.EDU (Yangsheng Xu).0>
Type:     cmu.cs.robotics
Who:      <speaker>Ralph Hollis</speaker>
          Senior Research Scientist
          The Robotics Institute
          Carnegie Mellon University
Topic:    Lorentz Levitation Technology:
          a New Approach to Fine Motion Robotics, Teleoperation
          Haptic Interfaces, and Vibration Isolation
Dates:    15-Jan-94
Time:     <stime>3:30 PM</stime> - <etime>5:00 PM</etime>
Place:    <location>ADAMSON WING Auditorium in Baker Hall</location>
Host:     Yangsheng Xu (xu@cs.cmu.edu)
PostedBy: xu+ on 6-Jan-94 at 14:16 from IUS4.IUS.CS.CMU.EDU (Yangsheng Xu)
Abstract:

                       RI SEMINAR

 WHEN:    Friday, Jan 15, 1994; <stime>3:30 pm</stime> - <etime>5:00 pm</etime>
          Refreshments will be served starting at 3:15 pm

 WHERE:   <location>ADAMSON WING Auditorium in Baker Hall</location>

 SPEAKER: <speaker>Ralph Hollis</speaker>
          Senior Research Scientist
          The Robotics Institute
          Carnegie Mellon University

 TITLE:   Lorentz Levitation Technology:
          a New Approach to Fine Motion Robotics, Teleoperation
          Haptic Interfaces, and Vibration Isolation
```

Fig. 5 An excerpt from the seminar announcement `cmu.cs.robotics−1018 : 0`

the size of the sample increases, BIEN learns to generalize and tags many more entities, obtaining lower precision and higher recall.

Ciravegna et al. (2002) traced the learning curve for $(LP)^2$. It shows that the algorithm learns with a very limited number of examples. `stime` and `etime` tend to reach excellent accuracy after a couple of dozens of examples, while `speaker` and `location` reach reasonable accuracy after between 50 and 80 examples.

4.1.3 Different versions

During their experiments using Seminar Announcements, Fabio Ciravegna and Leon Peshkin produced their own "improved" versions of the corpus. These two versions were used as a starting point to produce a new revised version. This version is now publicly available on the web site of the EU Dot.Kom project (http://www.dot-kom.org) and referenced in the RISE repository. Such version mainly fixes obvious annotation errors. E.g., errors in the inexact identification of `stime` and `etime` boundaries; usually, a missing final dot "." at the right boundary (see Fig. 6 for an example of such changes to the annotation). More than 80 corrections of such

Table 1 Results obtained by different systems on CMU seminar announcements. Note that in the experiments with SNoW the matches were counted adopting the One Answer per Slot criterion for all the slots but for `speaker`, for which the One Answer per Occurrence in the Document criterion was used. Results for $(LP)^2$ are taken from http://nlp.shef.ac.uk/amilcare/results.html

	SRV			RAPIER			WHISK		
Matching	OAS			OADS			OAOD		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Speaker	54.4	58.4	56.3	80.9	39.4	53.0	52.6	11.1	18.3
Location	74.5	70.1	72.3	91.0	60.5	72.7	83.6	55.4	66.6
Stime	98.6	98.4	98.5	96.5	95.3	95.9	86.2	100	92.6
Etime	67.3	92.6	77.9	95.8	96.6	96.2	85.0	87.2	86.1
All slots	–	–	77.1	–	–	77.3	–	–	64.9

	BWI			$(LP)^2$			SNoW		
Matching	OAS			OAS			OAS-OAOD		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Speaker	79.1	59.2	67.7	86.64	84.33	85.44	83.3	66.3	73.8
Location	85.4	69.6	76.7	85.51	70.37	77.18	90.9	64.1	75.2
Stime	99.6	99.6	99.6	94.59	92.08	93.32	99.6	99.6	99.6
Etime	94.4	94.9	94.6	97.11	95.91	96.50	97.6	95.0	96.3
All slots	–	–	83.9	–	–	88.11	–	–	–

	ME_2	BIEN	T-Rex	Elie		
Matching	OAOD		OAS	OAOD		
Slot	F_1	F_1	F_1	Prec	Rec	F_1
Speaker	72.6	76.9	85.9	84.6	85.1	84.8
Location	82.6	87.1	84.9	89.9	82.2	85.9
Stime	99.6	96.0	93.1	84.7	96.3	90.1
Etime	94.2	98.8	93.6	94.8	94.4	94.6
All slots	86.9	–	87.2	89.4	89.8	88.5

	Sutton	Finkel	Sigletos	Li-SVMUM
Matching	OAOD	OAOD	OAS	OAS
Slot	F_1	F_1	F_1	F_1
Speaker	88.1	84.16	75.40	69.0
Location	80.4	90.0	81.83	81.3
Stime	96.7	97.11	99.51	94.8
Etime	97.2	97.89	96.68	92.7
All slots	90.6	92.29	–	84.5

```
Version 1.0 (RISE version)
will be given at <stime>10:45 a.m</stime>., Tuesday,

Version 1.2
will be given at <stime>10:45 a.m.</stime>, Tuesday,
```

Fig. 6 An example of how SA annotation was modified

errors were performed on the dataset. No correction of errors in the text of the original announcements was performed. Moreover, three further changes were made: (1) file names were modified to make them Windows-compliant; (2) all <sentence> and <paragraph> tags were stripped from the corpus; (3) the documents were made XML-legal (i.e., special characters such as ampersand were replaced with their XML entity references).

Moreover, there is also the Seminar Announcements corpus with associated templates produced by Mary Elaine Califf to run RAPIER.

Finally, Peshkin and Pfeffer (2003) created a derivative dataset in which documents are stripped of headers and two extra fields are sought: *date* and *topic*.

4.2 Job postings

The Job Posting collection (Califf 1998) consists of a set of 300 computer-related job postings from the Usenet newsgroup `austin.jobs`.[11] The IE task is to identify the types of information that would be useful in creating a searchable database of such jobs, with fields like *message-id* and the posting *date* which are useful for maintaining the database, and then fields that describe the job itself, such as the job *title*, the *company*, the *recruiter*, the *location*, the *salary*, the *language*s and *platform*s used, and required years of experience and degrees. Some of these slots can take only one value, but for most of the slots a job posting can contain more than one appropriate slot-filler. There are a total of 17 different slots for this task. Figure 7 shows an example taken from the corpus. Note that, differently from the Seminar Announcements, the annotations of the Job Postings in RISE (1998) are provided as separate templates associated with each text.

4.2.1 Methodology and results

Califf (1998) performs experiments randomizing the collection, dividing it into 10 parts and doing 10-fold cross-validation; she also trained RAPIER on subsets of the training data at various sizes in order to produce learning curves. Freitag and Kushmerick (2000), Kosala and Blockeel (2000), and Roth and Yih (2002) adopt the same 10-fold cross-validation methodology. Ciravegna (2001a), Finn and Kushmerick (2004a), and Li et al. (2005b) randomly partition the entire document

[11] Available from the RISE repository: http://www.isi.edu/info-agents/RISE/repository.html. The collection we refer to in the article is the following: http://www.isi.edu/info-agents/RISE/Jobs/SecondSetOfDocuments.tar.Z.

```
From: spectrum@onramp.net
Newsgroups: austin.jobs
Subject: US-TX-Austin - VISUAL BASIC Developers $50K to $70K
Date: Sat, 23 Aug 97 09:52:21
Organization: OnRamp Technologies, Inc.; ISP
Lines: 65
Message-ID: <NEWTNews.872347949.11738.consults@ws-n>
Content-Type: TEXT/PLAIN; charset=US-ASCII
X-Newsreader: NEWTNews & Chameleon -- TCP/IP for MS Windows from
NetManage
Xref: cs.utexas.edu austin.jobs:119473

US-TX-Austin - VISUAL BASIC Developers $50K to $70K

POSTING I.D. D05

Major corporations have immediate openings for Visual Basic
programmers. 2-5 years experience; Oracle or SQL Server helpful.
Windows 95 and Windows NT programming a plus. Please contact Bill
Owens at (972) 484-9330; FAX (972) 243-0120 at Resource Spectrum.
To review several hundred positions with similar requirements please
visit our web site at www.spectrumm.com. Please reference Posting ID
and position title when contacting us. Qualified, experienced people
from all over the world will be considered. You must speak and write
English fluently. You must be a US citizen, a Permanent Resident, and
meet all job requirements.

Resource Spectrum
5050 Quorum Dr., Ste 700
Dallas, Texas 75240
Internet Address: spectrum@onramp.net (We prefer this transmission)
Fax: (972) 243-0120
Voice (972)484-9330
Contact: Bill Owens
```

```
computer_science_job

id: NEWTNews.872347949.11738.consults@ws-n
title: Developers
salary: $50K to $70K
company:
recruiter: Resource Spectrum
state: TX
city: Austin
country: US
language: VISUAL BASIC
platform: Windows NT  Windows 95
application: SQL Server  Oracle
area:
req_years_experience: 2
desired_years_experience: 5
req_degree:
desired_degree:
post_date: 23 Aug 97
```

Fig. 7 An excerpt from the job posting `job119473` together with its associated template

collection ten times into two sets of equal size, training and testing. Sigletos et al. (2005) uses 5-fold cross-validation and reports only overall f-measures, not the figures for the individual slots. Soderland (1999) reports WHISK performance using 10-fold cross validation on a randomly selected set of 100 texts instead of using the standard split for training and test sets. Moreover, he reports only the overall figures for precision and recall and not the figures for the single slots.

De Sitter and Daelemans (2003) use a Job Posting collection which is different from the one described above and consists of 600 postings.[12] As a matter of fact, this version includes 600 postings with templates associated, while the tagged postings are 300 only and they are exactly those of the Job Postings collection available in RISE. De Sitter and Daelemans perform their evaluation using 10-fold cross-validation.

In Table 2 we list the results obtained by different systems on Job Postings, together with the information about how matches are counted (when available). We do not list systems that either did not report results slot by slot but only overall figures (Soderland 1999) or reported results only on few slots (Freitag and Kushmerick 2000; Kosala and Blockeel 2000).

4.2.2 Learning curve

Califf (1998) provides also the learning curves for precision, recall and F-measure on the Job Posting collection.

4.2.3 Different versions

Given the fact that some IE algorithms need a tagged corpus (rather than an external annotation as provided by the version of Job Postings available in the RISE repository), some researchers produced their own tagged version: we have found four different versions produced by Mary Elaine Califf, Fabio Ciravegna, Scott Wen-tau Yih, and Georgios Sigletos. The creation of a standard "tagged" version is rather complex and its preparation will need some time.

4.3 Corporate acquisition

The Acquisition collection contains 600 articles on corporate acquisitions taken from the Reuters-21578 data set[13] (a standard source of data for experiments in Text Categorization consisting of 21,578 newswire articles produced by the Reuters press service in 1987). Note that the Acquisition collection was not available in the RISE repository and was recently made publicly available in the Dot.Kom web site in the context of the work reported in this article.

The task of the IE is to identify the following information: *acquired* (Entity that is purchased), *purchaser* (Purchasing company or person), *seller* (Selling

[12] Available from ftp://ftp.cs.utexas.edu/pub/mooney/job-data/job600.tar.gz.

[13] http://www.daviddlewis.com/resources/testcollections/reuters21578.

Table 2 Results obtained by different systems on job postings

Matching	RAPIER (OADS)			$(LP)^2$(OAOD)			SNoW (OAS)		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Id	98.0	97.0	97.5	98.80	99.07	98.93	99.7	99.7	99.7
Title	67.0	29.0	40.5	56.75	38.38	45.71	62.0	45.9	52.7
Company	76.0	64.8	70.0	80.09	76.07	77.86	89.7	65.1	75.4
Salary	89.2	54.2	67.4	80.13	62.71	70.24	89.3	61.6	72.9
Recruiter	87.7	56.0	68.4	83.52	73.50	78.15	89.4	81.5	85.3
State	93.5	87.1	90.2	94.78	98.57	96.63	91.7	91.8	91.7
City	97.4	84.3	90.4	93.79	96.55	95.15	90.1	87.9	89.0
Country	92.2	94.2	93.2	98.16	98.66	98.41	95.6	95.4	95.5
Language	95.3	71.6	80.6	76.65	74.83	75.70	83.5	81.6	82.5
Platform	92.2	59.7	72.5	69.18	65.89	67.41	74.4	73.8	74.1
Application	87.5	57.4	69.3	76.29	77.07	76.64	84.7	47.5	60.9
Area	66.6	31.1	42.4	59.23	46.33	51.95	63.5	43.4	51.6
Req-years-e	80.7	57.5	67.1	82.40	84.15	83.12	90.2	78.4	83.9
Des-years-e	94.6	81.4	87.5	87.28	90.96	88.97	75.3	83.1	79.0
Req-degree	88.0	75.9	81.5	92.88	87.02	89.84	86.3	80.9	83.5
Des-degree	86.7	61.9	72.2	85.64	48.67	61.55	81.0	48.8	60.9
Post date	99.3	99.7	99.5	97.98	100.00	98.97	99.0	99.3	99.2
All slots	89.4	64.8	75.1	83.15	77.55	79.72	–	–	–

Matching	DeSitter-AO (OAOD)			DeSitter-OBD (OAS)			Elie (OAOD)		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Id	97	98	97	99	96	97	100.0	99.7	99.9
Title	31	43	36	35	35	35	57.3	54.6	55.9
Company	45	78	57	26	74	38	90.1	71.3	79.6
Salary	56	70	62	62	72	67	71.2	62.0	66.3
Recruiter	40	79	53	44	74	55	86.9	77.6	82.0
State	77	97	86	93	95	94	92.4	93.1	92.8
City	84	95	89	90	92	91	95.1	94.9	95.0
Country	92	98	95	91	94	92	97.4	94.2	95.8
Language	25	27	26	33	34	33	91.4	84.7	87.9
Platform	31	34	32	35	38	36	84.9	75.2	79.8
Application	32	29	30	31	30	30	80.7	61.3	69.7
Area	16	17	16	16	18	17	61.9	40.2	48.7
Req-years-e	50	80	62	72	81	76	80.6	79.3	79.9
Des-years-e	33	55	41	36	66	47	92.8	74.7	82.8
Req-degree	29	43	35	41	51	45	85.0	74.9	79.6
Des-degree	28	45	35	29	37	33	66.6	50.5	57.5
Post date	84	99	91	99	97	98	95.1	100.0	97.5
All slots	–	–	–	–	–	–	84.6	74.6	79.3

Table 2 continued

Matching Slot	Li—SVMUM OAOD F_1	Li—PAUM OAOD F_1
Id	97.7	97.4
Title	49.6	53.1
Company	77.2	78.4
Salary	86.5	86.4
Recruiter	78.4	81.4
State	92.8	93.6
City	95.9	95.2
Country	96.2	96.5
Language	86.9	87.3
Platform	80.1	78.4
Application	70.2	69.7
Area	46.8	54.0
Req-years-e	80.8	80.0
Des-years-e	81.9	85.6
Req-degree	87.5	87.9
Des-degree	59.2	62.9
Post date	99.2	99.4
Macro-avg	80.8	81.6

company), *acqabr* (Short name for *acquired*), *purchabr* (Short name for *purchaser*), *sellerabr* (Short name for *seller*), *acqloc* (Location of *acquired*), *acqbus* (Description of *acquired*'s business), *dlramt* (Purchasing price), *status* (Status of negotiations).

Freitag (1998) randomly partitions ten times the entire document collection into two sets of equal size, training and testing, a partitioning that Freitag and Kushmerick (2000) also observes. The learners were trained on the training documents and tested on the corresponding test documents for such partition. The resulting numbers are averages over documents from all test partitions. Both Califf (1998) and Finn and Kushmerick (2004b) observe the same experimental regime, but it is unclear whether they use the same partitions.

This dataset has not been used as widely as the two previously described but it represents a richer, harder problem than either Seminar Announcements or Job Postings. In Table 3 we list the results obtained by different systems on Corporate Acquisition.

5 A proposal

In order to achieve the goal of making comparisons between IE algorithms fair and reliable, a number of guidelines and resources need to be made available. They include:

Table 3 Results obtained by different systems on corporate acquisition: SRV from Freitag (1998); RAPIER from Califf (1998); BWI from Freitag and Kushmerick (2000); ELIE$_{L2}$ from Finn and Kushmerick (2004b); and CProb from Freitag (1998). Note that this last entry is not a stand-alone learner, but a combination of methods

	SRV			RAPIER			BWI		
Matching	OAS			OAOD			OAS		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Acquired	42.7	35.0	38.5	57.3	19.2	28.8	55.5	24.6	34.1
Purchaser	47.4	43.0	45.1	50.0	20.5	29.1	–	–	–
Seller	21.3	26.1	23.4	32.4	10.0	15.3	–	–	–
Acqabr	37.0	39.2	38.1	43.6	18.5	26.0	–	–	–
Purchabr	44.7	53.0	48.5	42.8	16.7	24.0	–	–	–
Sellerabr	20.7	32.1	25.1	10.5	7.3	8.6	–	–	–
Acqloc	22.7	22.0	22.3	46.9	16.3	24.2	–	–	–
Dlramt	66.1	58.1	61.8	63.3	28.5	39.3	63.4	42.6	50.9
Status	59.1	39.0	47.0	67.3	29.8	41.3	–	–	–

	ELIE$_{L2}$			CProb
Matching	OAOD			OAS
Slot	Prec	Rec	F_1	F_1
Acquired	57	37	43	45.6
Purchaser	51	42	47	53.0
Seller	32	11	17	–
Acqabr	65	29	40	43.1
Purchabr	54	20	29	–
Sellerabr	38	8	12	–
Acqloc	47	28	35	–
Dlramt	55	63	59	64.3
Status	52	48	50	59.5

Improved versions of corpora: We are collecting the different versions of the standard corpora produced by researchers in order to compare the corrections introduced and produce new versions which take such corrections into account. The final aim is to distribute new, “improved” versions of the annotated corpora. The new version of Seminar Announcements is already available in the Dot.Kom web site and referenced in the RISE repository (http://www.isi.edu/info-agents/RISE/repository.html).

Explicit versioning: The community needs to take an active role in managing the versioning of widely used corpora. We recommend that any changes to a corpus be given a unique label by the author of the changes (e.g., “Seminar Announcements corpus, Lavelli version”) and released to the community. Ideally, these corpora, with their sub-versions, should be made freely available in a single location (such as the Dot.Kom site).

Shared format: Annotation should be made available as legal XML and should be as information-preserving as possible. We argue for mark-up over filled templates. There are two ways in which a document may be marked up to identify the relevant textual extents. In *in-line* mark-up, tags are inserted directly into the text, individual slot fillers bracketed by a pair of begin and end tags, where the name of the tags corresponds to the type of the slot. In *stand-off* mark-up, a separate annotation file is created for each file in the corpus, and textual extents are expressed in terms of character offsets. Because it can be difficult to write a parser of in-line mark-up that does not take subtle hints about tokenization from the embedded tags, stand-off annotation, though not the most common form, is to be preferred.

Exact definition of the corpus partition: Researchers should make use of existing training/testing splits when using standard corpora. (Note that all of the corpora described in Sect. 4 include such splits, and the corpora on the Dot.Kom site will include them.) When releasing a new corpus, splits should similarly be specified. If experiments are conducted on a corpus which cannot be made publicly available, the procedure by which splits are generated should be explicitly described. It is desirable in this case to use multiple splits of the data in the interest of greater statistical significance.

Task definition: The parameters of the task should be described as explicitly as possible. At a minimum, this description should include the following:

1. A set of fields to extract.
2. The legal numbers of fillers for each field, such as "exactly one value" (1), "zero or one values" (0-1), "zero or more values" (0+), or "one or more values" (1+).
3. The possibility of multiple varying occurrences of any particular filler (e.g., "7:00 PM" vs. "7pm").
4. How stringently matches are evaluated (exact, overlap or contains).

The practice of the community on these questions has been inconsistent to date. While Item #1 above is always specified, Items #2, #3 and #4 are usually specified only implicitly based on inspecting a sample of the labeled fields, intuition, common sense, etc.

Scorer: If possible, use the MUC scorer to evaluate performance. If for some reason, the MUC scorer is not used, the counting procedure must be carefully described. Ideally in this case, the substitute scorer should also be released as source code for use by other researchers.

Definition of preprocessing tasks: Some of the preparation subtasks (e.g., tokenization) may influence the performance of the algorithms. Therefore, when possible, we will provide an annotated version of the corpora with, for example, tokens, Part-of-Speech tagging, gazetteer lookup and named entity recognition in order to allow fair comparison of the different algorithms. This will also facilitate the comparison of the impact of different features in the learning phase.

Learning curve: Learning curves are almost always of interest. Since developing annotated data for IE is such a laborious process, most IE problems are data-limited. Thus, the performance of a learner in sparse-data conditions is clearly relevant. Learning curves should be generated by fixing the test set and sampling successive supersets of the training data. This procedure may be repeated multiple times using different train/test splits.

Statistical significance: Estimates of statistical significance should be provided in all cases. When a study compares two or more algorithms, or variants of an algorithm, approximate randomization may be used to assess the observed improvement. When such comparison is inappropriate or impossible, confidence bounds may be calculated using the bootstrap procedure. A succinct introduction to these two procedures is provided in Appendix.

Reporting scores: When using a "standard" corpus, such as those described in Sect. 4, report results for all slots defined for the corpus, unless the algorithm or subject of the paper precludes this (e.g., the algorithm is suitable only for strings structured like protein names). In addition to F_1, report precision and recall. Furthermore, research should report performance both on a per-slot basis as well as microaveraged over all slots.[14]

Work in this direction was done in the framework of the EU Dot.Kom project and resulted in the PASCAL Challenge described below.

5.1 The PASCAL challenge

As a result of the activities described above, a PASCAL[15] challenge on the evaluation of ML-based IE techniques was organized and run. The proposal was jointly supported by the EU PASCAL Network of Excellence and by the EU Dot.Kom project (http://www.dot-kom.org). The evaluation had four primary motivations:

– Fair comparison of ML algorithms for IE through a controlled application of the methodology described in this article.
– Summary assessment of the general benefit of state-of-the-art ML to the problem of IE.
– Identification of any challenges not adequately addressed by current ML approaches.
– Publication of an extensive testbed to enable comprehensive, comparable research beyond the lifetime of the current challenge.

A corpus of 1,100 documents was collected from various sources; it comprises of 850 Workshop Call for Papers (CFPs) and 250 Conference CFPs. The majority of the documents come from the field of Computer Science, due to the readily available archives, although other fields, such as biomedicine and linguistics, are also represented. Care was taken to ensure each document relates to a unique call. The documents are divided into three corpora:

[14] The "all slots" figures are obtained by aggregating the confusion matrices over all fields, rather than averaging results from field-specific confusion matrices. This approach is called "microaveraging" in the text classification literature.

[15] PASCAL was a Network of Excellence on "Pattern Analysis, Statistical Modelling and Computational Learning" funded by the European Commission as part of FP6. In March 2008 the follow-up Network of Excellence PASCAL2 was started as part of FP7.

- Training Corpus (400 Workshop CFPs): The documents in the training corpus are randomly divided into 4 sets of 100 documents. Each of these sets is further randomly divided into 10 subsets of 10 documents. Each document relates to a workshop held between 1993 and 2000.
- Test Corpus (200 Workshop CFPs): The documents in the training corpus relate to workshops held between 2000 and 2005.
- Enrich Corpus (250 Workshop CFPs & 250 Conference CFPs): The documents in the enrich corpus relate to workshops held between 2000 and 2005 & conferences held between 1997 and 2005.

Note that the training and test data is largely temporally distinct. Thus there will be less differentiation between the 4-fold cross-validation training and test data, as these are randomly sampled from the same timeframe. The Test Corpus may exhibit differences introduced by the temporal disparity providing a more rigorous test of a learning system's ability to generalise. As the Enrich Corpus offers documents taken from the same timeframe as the Test Corpus, it can potentially be exploited to uncover the temporal differences.

The data was preprocessed using the GATE (http://www.gate.ac.uk) system, which provides tokenization, orthography, Part-of-Speech tagging and named-entity recognition (Location, Person, Date, etc.) text features. The features selected are a fairly basic set in terms of linguistic processing. All participants were required to use only those features in order to separate the influence of the features from the learning capabilities.

The annotation exercise took place over roughly three months and involved a series of consultations between the challenge organizers and the annotators to determine the final annotations. The general methodology adopted was one of maximizing the information provided by the annotations whilst minimizing ambiguity during annotating. This meant that whilst it would have been desirable to extract the list of people on the organizing committee, in the initial studies the annotators found it very difficult to determine whether a name should or should not be included, and thus this annotation was removed from consideration. For the final annotation exercise 10 people annotated an overlapping set of documents, with each document being annotated by two people. Conflicts were resolved by the overseeing annotator. An annotation tool (Melita: Ciravegna et al. 2002) was used to aid the process, although all automatic pattern matching was switched off, except for exact-string matching, so that the data was not biased towards the matching algorithm. Each document can have 11 annotation types; 8 relating to the workshop itself (name, acronym, homepage, location, date, paper submission date, notification date and camera-ready copy date) and 3 relating to the associated conference (name, acronym and homepage).

The following tasks have been proposed:

- *Task1*: Given all the available training documents (i.e. 300 documents for the 4-fold cross-validation and 400 documents for the Test Corpus experiment), learn the textual patterns necessary to extract the annotated information.
- *Task2a* (Learning Curve): Examine the effect of limited training resources on the learning process by incrementally adding the provided subsets to the training

data. Thus there are 9 experiments; for the four-fold cross-validation experiment the training data has 30, 60, 90, 120, 150, 180, 210, 240 and 270 documents, and for the Test Corpus experiment the training data has 40, 80, 120, 160, 200, 240, 280, 320 and 360 documents.

- *Task2b* (Active Learning): Examine the effect of selecting which documents to add to the training data. Given each of the training data subsets used in Task2a, select the next subset to add from the remaining training documents. Thus a comparison of the Task2b and Task2a performance will show the advantage of the active learning strategy.
- *Task3a* (Enrich Data): Perform the above tasks exploiting the additional 500 unannotated documents. In practice only one participant attempted this task and only to enhance Task1 on the Test Corpus.
- *Task3b* (Enrich WWW Data): Perform either of the above tasks but using any other (unannotated) documents found on the WWW. In practice only one participant attempted this task and only to enhance Task1 on the Test Corpus.

Coherently with the analysis of critical issues outlined in this article, the PASCAL challenge was based on a precise evaluation methodology: each system was evaluated on its ability to identify every occurrence of an annotation and only exact matches were scored. Performance is reported using the standard IE measures of Precision, Recall and F-measure. The systems' overall performance was calculated by micro-averaging the performance on each of the eleven slots. All participants were required to submit their blind results to an evaluation server in order to maintain regularity in the result scoring.

In Table 4 the results on the test corpus of the systems that participated in Task1 are shown. Further details on the Pascal challenge and on the results obtained by the participants can be found in Ireson et al. (2005)

6 Conclusions

In this article we have surveyed the evaluation methodology adopted in IE identifying a number of critical issues that hamper comparison of the results obtained by different researchers.

The "ideal" long-term goal would be to provide a flexible unified tool that could be used to recreate many of the previous algorithms (e.g., BWI (the original C version, or TIES, the Java reimplementation carried on at FBK-irst[16]), RAPIER, $(LP)^2$, etc); along with standard code for doing test/train splits, measuring accuracy, etc. In short, we envision a sort of "Weka for IE".[17] However, this goal is very challenging because it would involve either integrating legacy code written in

[16] http://tcc.itc.it/research/textec/tools-resources/ties.html.

[17] Weka is a collection of open source software implementing ML algorithms for data mining tasks, http://www.cs.waikato.ac.nz/ml/weka

Table 4 Task1 results for individual slots on the test data experiment. Only those systems which provided the highest *F*-measure for at least one slot are shown. The italized figures highlight the best results on each slot

	Amilcare system1			Yaoyong system1			Stanford system1			Yaoyong system2			ITC-irst system2		
Slot	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
ws name	65.6	24.1	35.2	62.9	53.9	58.0	61.8	*57.6*	59.6	71.3	43.7	54.2	*85.2*	53.9	*66.0*
ws acronym	*88.7*	*84.4*	*86.5*	73.8	52.3	61.2	80.6	35.8	49.6	79.6	48.1	60.0	73.3	25.9	38.3
ws date	76.9	63.2	69.4	81.0	66.6	73.1	82.2	69.3	*75.2*	83.8	58.6	69.0	85.0	45.1	58.9
ws home	*86.4*	61.9	72.1	65.6	87.0	*74.8*	67.8	66.5	67.1	73.4	67.9	70.5	67.2	41.9	51.6
ws location	62.1	40.2	48.8	61.1	*67.4*	64.1	73.7	57.6	64.7	71.7	61.2	*66.0*	81.2	40.6	54.2
ws submission	*87.6*	*85.1*	*86.4*	71.9	76.37	74.0	74.7	68.0	71.2	76.7	63.6	69.6	84.1	61.7	71.2
ws notification	88.9	*88.9*	*88.9*	86.7	82.1	84.3	87.0	77.4	81.9	*94.3*	78.4	85.6	92.1	79.5	85.3
ws cameraready	87.6	*86.5*	*87.0*	76.4	73.6	75.0	77.7	79.1	78.4	84.5	66.9	74.7	91.1	68.7	78.3
conf name	79.2	42.2	*55.1*	64.9	41.1	50.3	64.3	40.0	49.3	77.5	34.4	47.7	79.5	34.4	48.1
conf acronym	*92.2*	*88.8*	*90.5*	61.9	34.8	44.5	57.6	42.8	49.1	63.4	27.8	38.7	66.7	23.5	34.8
conf home	65.6	28.0	*39.3*	36.8	9.3	14.9	38.9	9.3	15.1	45.5	6.7	11.6	55.6	6.7	11.9

different programming languages, or reimplementing published algorithms, whose details are subtle and sometimes not described in complete detail.

The work reported in this article addresses a more practical mid-term goal: to elaborate a clear and detailed experimental methodology and propose it to the IE community. The aim is to reach a widespread agreement so that future IE evaluations will adopt the proposed methodology, making comparisons between algorithms fair and reliable. In order to achieve this goal, we have developed and made available to the community a set of tools and resources that incorporate a standardized IE methodology as part of the Pascal challenge. This includes a web site (http://nlp.shef.ac.uk/pascal), with a standardized corpus, a scorer (derived from the MUC scorer and adaptable to other tasks) and a precise description of a set of tasks, with standardized results for a set of algorithms.

While the methodological issues that we have discussed are important, the good news is that in most cases it is quite straightforward for researchers to fix these problems, either while planning and conducting the research, or during the peer review process prior to publication. Unfortunately, when a reviewer is examining any single submission in isolation, the methodological problems may be difficult to spot. We hope that this article helps researchers design their experiments so as to avoid these problems in the first place, and assists reviewers in detecting methodological flaws.

This article has focused specifically on methodological problems in IE research. Some of the issues are relevant only to IE, but others apply to other topics in empirical natural language processing, such as question answering, summarization or document retrieval. Some of the issues apply to many technologies based on ML. We hope that the lessons we have learned in the context of IE might assist in resolving methodological difficulties in other fields.

Finally, our focus has been on traditional performance measures such as precision and recall. As we have seen, it can be quite difficult to determine whether they are calculated consistently by different researchers. Nevertheless, it is important to bear in mind that these measures are just a means to an end. The ultimate goal is to increase end users' satisfaction with an application, but a user's experience is unlikely to be related to these traditional measures in a simple manner; for example, a 5% increase in precision is unlikely to mean that the user is 5% more satisfied. Therefore, while we strongly advocate the methodology described in this paper, we also caution that methodological hygiene in and of itself does not guarantee that a particular approach offers a tangible benefit to end users.

Acknowledgements F. Ciravegna, C. Giuliano, N. Ireson, A. Lavelli and L. Romano were supported by the IST-Dot.Kom project (http://www.dot-kom.org), sponsored by the European Commission as part of the Framework V (grant IST-2001-34038). N. Kushmerick was supported by grant 101/F.01/C015 from Science Foundation Ireland and grant N00014-03-1-0274 from the US Office of Naval Research. We would like to thank Leon Peshkin for kindly providing us his own corrected version of the Seminar Announcement collection and Scott Wen-Tau Yih for his own tagged version of the Job Posting collection. We would also like to thank Hai Long Chieu, Leon Peshkin, and Scott Wen-Tau Yih for answering our questions concerning the settings of their experiments. We are also indebted to the anonymous reviewers of this article for their valuable comments.

Appendix

Statistical significance testing

The objective in many papers on IE is to show that some innovation leads to better performance than a reasonable baseline. Often this involves the comparison of two or more system variants, at least one of which constitutes the baseline, and one of which embodies the innovation. Typically, the preferred variant achieves the highest scores, if only by small margins, and often this is taken as sufficient evidence of general improvement, even though the test sets in many IE domains are relatively small.

Approximate randomization is a computer-intensive procedure for estimating the statistical significance of a score difference in cases where the predictions of two systems under comparison are aligned at the unit level (Noreen 1989). For example, Chinchor et al. (1993) used this procedure to assess the pairwise separation among participants of MUC3.

Table 5 presents pseudocode for the approximate randomization procedure. The procedure involves a large number (*M*) of passes through the test set. Each pass involves swapping the baseline and preferred outcomes on approximately half of the test documents, yielding two new "swapped" scores.[18] The fraction of passes for which this procedure widens the gap between systems is an estimate of the *p* value associated with the observed score difference. If this computed fraction is less than or equal to the desired confidence level (typically 0.05), we are justified in concluding that the observed difference in scores between baseline and preferred is significant.

In many cases, a relevant baseline is difficult to establish or acquire for the purpose of a paired comparison. Often the most salient comparison is with numbers reported only in the literature. Confidence bounds are critical in such cases to ascertain the level of significance of a result. However, calculating confidence bounds on a score such as the F-measure is cumbersome and possibly dubious, since it is unclear what parametric assumptions to make. Fortunately, we can apply *the bootstrap*, another computer-intensive procedure, to model the distribution of possible F-measures and assess confidence bounds (Efron and Tibshirani, 1993).

Table 6 sketches this procedure. As in approximate randomization, we iterate a large number (*M*, typically at least 1000) of times. With each iteration, we calculate the statistic of interest (e.g., the F-measure) on a set of documents from the test set formed by sampling with replacement. The resulting score sample may then be used to assess confidence bounds. In an approach called *the percentile bootstrap*, these scores are binned by quantile. The upper and lower values of the confidence interval may then be read from this data. For example, the lower bound of the 90% confidence interval lies between the maximum score among the lowest 5% and the next score in an ordering from least to greatest. Obviously, in order for this computation to be valid, *M* must be sufficiently large. Additional caveats apply, and interested readers are referred to the Efron and Tibshirani introduction (1993).

[18] Note that the swap of the outcomes is performed at the document level and not at the level of the single markup.

Table 5 The approximate randomization procedure

```
Given $S$, the score of the baseline
Given $S'$, the score of the preferred variant
$\delta \leftarrow |S' - S|$
$C \leftarrow 0$
**for** $i$ in 1 to $M$ **do**
    **for** Each document in the test set **do**
        Swap document outcome of baseline and preferred with probability 0.5
    **end for**
    Calculate scores $S'_i$ and $S_i$ scores on “swapped” result sets
    $\delta_i \leftarrow |S'_i - S_i|$
    **if** $\delta_i \geq \delta$ **then** Increment $C$
**end for**
Return $p$-value $= (C + 1)/(M + 1)$
```

Table 6 The bootstrap procedure

```
Given $D$, a set of test documents
$N \leftarrow |D|$
**for** $i$ in 1 to $M$ **do**
    $D_i \leftarrow$ documents by sampling $D$ $N$ times *with replacement*
    $S_i \leftarrow$ the score of sample $D_i$
**end for**
Return $\{S_i | 1 \leq i \leq M\}$
```

Glossary

In the table below, we have listed the names/acronyms of the systems mentioned in the paper together with their full names and bibliographical references.

BIEN	Bayesian Information Extraction Network (Peshkin and Pfeffer 2003)
BWI	Boosted Wrapper Induction (Freitag and Kushmerick 2000)
CProb	Bayesian Prediction Combination (Freitag 1998)
Elie	Adaptive Information Extraction Algorithm (Finn and Kushmerick 2004a, b)
$(LP)^2$	Adaptive Information Extraction Algorithm (Ciravegna 2001a)
ME_2	Maximum Entropy Classifier (Chieu and Ng 2002)
PAUM	Perceptron Algorithm with Uneven Margins (Li et al. 2005b)
RAPIER	Robust Automated Production of Information Extraction Rules (Califf 1998)
SNoW	Sparse Network of Winnows (Roth and Yih 2001, 2002)
SRV	Symbolic Relational Learner (Freitag 1998)
SVMUM	Support Vector Machine with Uneven Margins (Li et al. 2005a)
TIES	Trainable Information Extraction System
T-Rex	Trainable Relation Extraction (Iria and Ciravegna 2006)
WHISK	(Soderland 1999)

References

Califf, M. E. (1998). Relational learning techniques for natural language information extraction. Ph.D. thesis, University of Texas at Austin.

Califf, M., & Mooney, R. (2003). Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research, 4*, 177–210.

Chieu, H. L., & Ng, H. T. (2002). Probabilistic reasoning for entity and relation recognition. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2002).*

Chinchor, N., Hirschman, L., & Lewis, D. D. (1993). Evaluating message understanding systems: An analysis of the third Message Understanding Conference (MUC-3). *Computational Linguistics, 19*(3), 409–449.

Ciravegna, F. (2001a). Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI-01).* Seattle, WA.

Ciravegna, F. (2001b). $(LP)^2$, an adaptive algorithm for information extraction from web-related texts. In *Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*. Seattle, WA.

Ciravegna, F., Dingli, A., Petrelli, D., & Wilks, Y. (2002). User-system cooperation in document annotation based on information extraction. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02).*

Ciravegna, F., & Lavelli, A. (2004). LearningPinocchio: Adaptive information extraction for real world applications. *Journal of Natural Language Engineering, 10*(2), 145–165.

Daelemans, W., & Hoste, V. (2002). Evaluation of machine learning methods for natural language processing tasks. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002).* Las Palmas, Spain.

Daelemans, W., Hoste, V., Meulder, F. D., & Naudts, B. (2003). Combined optimization of feature selection and algorithm parameters in machine learning of language. In *Proceedings of the 14th European Conference on Machine Learning (ECML 2003).* Cavtat-Dubronik, Croatia.

De Sitter, A., & Daelemans, W. (2003). Information extraction via double classification. In *Proceedings of the ECML/PKDD 2003 Workshop on Adaptive Text Extraction and Mining (ATEM 2003).* Cavtat-Dubronik, Croatia.

Douthat, A. (1998). The Message Understanding Conference scoring software user's manual. In *Proceedings of the 7th Message Understanding Conference (MUC-7).* http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_sw/muc_sw_manual.html.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York: Chapman and Hall.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005).*

Finn, A., & Kushmerick, N. (2004a). Information extraction by convergent boundary classification. In *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM 2004).* San Jose, California.

Finn, A., & Kushmerick, N. (2004b). Multi-level boundary classification for information extraction. In *Proceedings of the 15th European Conference on Machine Learning*. Pisa, Italy.

Freitag, D. (1997). Using grammatical inference to improve precision in information extraction. In *Proceedings of the ICML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition*. Nashville, Tennessee.

Freitag, D. (1998). Machine learning for information extraction in informal domains. Ph.D. thesis, Carnegie Mellon University.

Freitag, D., & Kushmerick, N. (2000). Boosted wrapper induction. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000).* Austin, Texas.

Habert, B., Adda, G., Adda-Decker, M., de Mareuil, P. B., Ferrari, S., Ferret, O., Illouz, G., & Paroubek, P. (1998). Towards tokenization evaluation. In *Proceedings of 1st International Conference on Language Resources and Evaluation (LREC-98).* Granada, Spain.

Hirschman, L. (1998). The evolution of evaluation: Lessons from the Message Understanding Conferences. *Computer Speech and Language, 12*(4), 281–305.

Hoste, V., Hendrickx, I., Daelemans, W., & van den Bosch, A. (2002). Parameter optimization for machine-learning of word sense disambiguation. *Natural Language Engineering, 8*(4), 311–325.

Ireson, N., Ciravegna, F., Califf, M. E., Freitag, D., Kushmerick, N., & Lavelli, A. (2005). Evaluating machine learning for information extraction. In *Proceedings of 22nd International Conference on Machine Learning (ICML 2005)*. Bonn, Germany.

Iria, J., & Ciravegna, F. (2006). A methodology and tool for representing language resources for information extraction. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy.

Kosala, R., & Blockeel, H. (2000). Instance-based wrapper induction. In *Proceedings of the Tenth Belgian-Dutch Conference on Machine Learning (Benelearn 2000)*. Tilburg, The Netherlands, pp. 61–68.

Kushmerick, N. (2000). Wrapper induction: Efficency and expressiveness. *Artificial Intelligence, 118*(1–2), 15–68.

Li, Y., Bontcheva, K., & Cunningham, H. (2005a). SVM based learning system for information extraction. In J. Winkler, M. Niranjan, & N. Lawrence (Eds.), *Deterministic and statistical methods in machine learning*, Vol. 3635 of *LNAI*. (pp. 319–339). Springer Verlag.

Li, Y., Bontcheva, K., & Cunningham, H. (2005b). Using uneven margins SVM and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CONLL 2005)*.

Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999), Performance measures for information extraction. In *Proceedings of the DARPA Broadcast News Workshop*. http://www.nist.gov/speech/publications/darpa99/pdf/dir10.pdf.

Noreen, E. W. (1989). *Computer Intensive Methods for Testing Hypotheses: An Introduction*. New York: Wiley.

Peshkin, L., & Pfeffer, A. (2003). Bayesian information extraction network. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*. Acapulco, Mexico.

RISE. (1998). A repository of online information sources used in information extraction tasks. [http://www.isi.edu/info-agents/RISE/index.html] Information Sciences Institute/USC.

Roth, D., & Yih, W. (2001). Relational learning via propositional algorithms: An information extraction case study. In *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*. Seattle, WA.

Roth, D., & Yih, W. (2002). Relational learning via propositional algorithms: An information extraction case study. Technical Report UIUCDCS-R-2002-2300, Department of Computer Science, University of Illinois at Urbana-Champaign.

Sigletos, G., Paliouros, G., Spyropoulos, C., & Hatzopoulos, M. (2005). Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research, 6*, 1751–1782.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning, 34*(1–3), 233–272.

Sutton, C., & McCallum, A. (2004). Collective segmentation and labeling of distant entities. In *Proceedings of the ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*.