A Formal Framework for Evaluation of Information Extraction

An De Sitter*
Dept. of Mathematics and Computer Science
University of Antwerp
Middelheimlaan 1
2020 Antwerpen, Belgium
anneleen.desitter@ua.ac.be

Toon Calders†
Dept. of Mathematics and Computer Science
University of Antwerp
Middelheimlaan 1
2020 Antwerpen, Belgium
toon.calders@ua.ac.be

Walter Daelemans
CNTS
University of Antwerp
Universiteitsplein 1
2610 Antwerpen, Belgium
daelem@uia.ua.ac.be

Abstract

An important problem in the field of Information Extraction (IE) is the lack of clear guidelines for evaluating the correctness of the output generated by an extraction algorithm. This paper tries to handle this problem by providing a formal framework for IE and its evaluation. We define IE in two different, but frequently used approaches: the "All Occurrences" and the "One Best per Document" settings, and we give a formal approach for evaluating an IE system in both settings. Our approach is based on the observation that most commonly used evaluation measures use the confusion matrix as a basis for their computation. We also shortly discuss the most frequently used evaluation measures.

1 Introduction

Research on Information Extraction (IE) has been going on for about 15 years. During this period, research was focussed mostly on finding "good" algorithms for extraction. Datasets vary from structured over semi-structured to unstructured documents, from real-life applications to cleaned versions, ... The literature on IE is widely scattered. Workshops, tutorials and papers about IE may for example be found in machine learning, data mining, computational linguistics and AI conferences. A more important problem is, that there are no clear standards for the evaluation of IE algorithms, which makes comparison between them very difficult. Some questions that arise are:

- How is IE defined, i.e., is it about filling a template, or about finding all occurrences of a certain piece of information in a document?
- When is an extracted item correct? Since there is only about 70% - 85% inter-annotator agreement on a typical IE task, it is clear that there may be discussions about correctness.

*The authors gratefully acknowledge the support of the Special Research Fund of the University of Antwerp
†Research Assistant of the Fund for Scientific Research - Flanders.

- What measures should be used to compute "goodness" of an IE system? Most researchers report F_β-scores (e.g. [3], [9]), sometimes recall and precision are reported as well (e.g. [1]).

There is a need for clear guidelines concerning evaluation of IE. The field needs standard evaluation procedures, standard datasets which should be clearly described, a formal framework about how to decide when an item is (in)correct, and clear, informative, standard measures. Concerning the standard datasets, a lot of work has been, and is done by the RISE consortium [8]. Even though, it remains important to clearly describe the dataset(s) used in a paper discussing an IE system. Evaluation procedures are often copied from the IR community and concern mostly cross-validation, or splitting in training- and testsets. There as well, a clear explanation of the test strategy used is necessary. This paper deals with a formal framework on how to decide when an extracted item is (in)correct.

The organization of the paper is as follows: in Section 2, we introduce a formal definition of Information Extraction, where we need to make a distinction between the "All Occurrences"-approach and the "One Best per Document"-approach. In Section 3, we formalize the evaluation of an Information Extraction system. Section 4 discusses a few of the most used measures for evaluation, and finally we conclude the paper.

2 Defining Information Extraction

2.1 Preliminaries

In the literature, there are multiple proposals for a definition of IE. We shortly discuss three of them.

Wendy G. Lehnert gave following informal definition of an IE system.

- IE systems analyze unrestricted text in order to extract specific types of information.
- IE systems do not attempt to understand all of the text in all input documents, but they do analyze those portions of each document that contain relevant information.
- Relevance is determined by pre-defined domain guidelines which must specify, as accurately as possible, exactly what types of information the system is expected to find.

D. Freitag described the "Information Extraction task" he tried to solve in his work as follows: *Find the best unbroken fragment of text from a document that answers some domain-specific question* [2]. As we will see further on, this only applies on the One Best per Document approach.

A. McCallum and W. Cohen explain the IE task as *filling slots in a database from sub-segments of text* [6].

Before we are able to discuss evaluation of IE, it is important to agree on what we mean by IE, i.e. to define what is an IE task. In what follows, we will formally define an IE task and a solution to an IE task.

Depending on the intended purpose of the IE system, different approaches may be useful. A first possibility is the *One Best per Document* approach. In this case, we see the IE task as filling a given template structure. This is for example useful if the goal of the IE system is to fill a database in order to be able to query the information contained in the documents. For example, in the Seminar Announcements domain [8], we want to fill a database with start-times, end-times, locations and speakers of seminars. In the Job Advertisements domain [8], a logical application could be to set up a database containing all information (job title, salary, company,...) about the jobs in the corpus.

A second possibility is the *All Occurrences* approach. In this case, we want to find every occurrence of a certain item (e.g. speaker in the Seminar Announcements domain) in a document. This may be useful for e.g. annotation purposes.

Which approach should be used, usually depends on the format in which the corpus is given.

Example 1 *The document in Figure 1 was taken from the Seminar Announcements corpus. It was slightly edited for readability and illustrative purpose. The subscripts indicate the word numbers. The tags between "<" and ">" do not belong to the original document, but indicate the correct solution in the* All Occurrences *approach. There are four kinds of information to extract: speaker, location, start-time (stime) and end-time (etime). In this particular document, no end-time is provided.*

In the One Best per Document *approach, a correct solution is the following template:*

stime	5:00 PM
etime	
speaker	Al Roth, Ido Erev
location	CMU , Adamson Wing (Baker Hall)

We do not need the annotations in the document for this approach.

Type$_1$:$_2$ cmu.andrew.org.heinz$_3$
Topic$_4$:$_5$ Pitt-CMU$_6$ Rationality$_7$ Seminar$_8$
Dates$_9$:$_{10}$ 23-Feb-95$_{11}$
Time$_{12}$:$_{13}$ <stime> 5:00$_{14}$ PM$_{15}$ </stime>
Place$_{16}$:$_{17}$ <location> CMU$_{18}$,$_{19}$ Adamson$_{20}$ Wing$_{21}$ ($_{22}$ Baker$_{23}$ Hall$_{24}$)$_{25}$ </location>
.$_{26}$
PostedBy$_{27}$:$_{28}$ Cristina$_{29}$ Bicchieri$_{30}$ on$_{31}$ 10-Feb-95$_{32}$ at$_{33}$ 16:32$_{34}$ from$_{35}$
andrew.cmu.edu$_{36}$
Abstract$_{37}$:$_{38}$

Please$_{39}$ disregard$_{40}$ the$_{41}$ previous$_{42}$ announcement$_{43}$,$_{44}$ the$_{45}$ date$_{46}$ was$_{47}$ wrong$_{48}$
!$_{49}$

Pitt-CMU$_{50}$ Rationality$_{51}$ Seminar$_{52}$:$_{53}$
<speaker> Al$_{54}$ Roth$_{55}$ </speaker> and$_{56}$ <speaker> Ido$_{57}$ Erev$_{58}$ </speaker> will$_{59}$
speak$_{60}$ on$_{61}$ "$_{62}$ Low$_{63}$ versus$_{64}$ high$_{65}$ game$_{66}$ theory$_{67}$,$_{68}$ part$_{69}$ I$_{70}$:$_{71}$ Boundedly$_{72}$
rational$_{73}$ adaptive$_{74}$ behavior$_{75}$ in$_{76}$ the$_{77}$ intermediate$_{78}$ term$_{79}$ "$_{80}$.$_{81}$

Thursday$_{82}$,$_{83}$ February$_{84}$ 23$_{85}$
Time$_{86}$:$_{87}$ <stime> 5:00$_{88}$ pm$_{89}$ </stime> --$_{90}$
Place$_{91}$:$_{92}$ <location> CMU$_{93}$,$_{94}$ Adamson$_{95}$ Wing$_{96}$ ($_{97}$ Baker$_{98}$ Hall$_{99}$)$_{100}$ </location>
.$_{101}$

Figure 1: Document cmu.andrew.org.heinz-8113-0 from the Seminar Announcements corpus.

In what follows, we formally define the IE task and solution for both approaches.
First, we define a document as follows:

Definition 1 *A* document *is a sequence of tokens* $(t_1; \ldots; t_N)$*, where a token may be any non-empty sequence of characters, and usually will be a word, number, or punctuation mark.* N *is the length of the document.*

We will use $\mathcal{D}$ to indicate a set of documents, and D for a single document. With "*text fragment*", we mean a contiguous subsequence of a document.

2.2 All Occurrences

The *All Occurrences* (AO) approach can be seen as annotating every occurrence of the information needed (e.g. speaker or start-time in the Seminar Announcements domain) in a document. For evaluation of an IE system following the AO approach, we need both the exact locations of the correct answers, and of the predictions made by the system.

Definition 2 *An* All Occurrences Information Extraction task for a particular type of information *is a function*

$\mathcal{F} : \mathcal{D} \rightarrow 2^{N \times N} : D \mapsto \{(i_k,j_k);\ 1 \leq i_k \leq j_k \leq |D|\}$

- i_k *and* j_k *indicate the left and right boundaries of a text fragment;*
- (i_k,j_k) *is called an* answer*;*
- $|D|$ *is the number of tokens in the document.*

An AO IE task *is a set of functions* $\mathcal{F}$*, one for each type of information* [1].

Example 2 *The AO IE task for the item 'location' gives for the document D in Figure 1:* $\mathcal{F}_{loc}(D) = \{(18, 25), (93, 100)\}$.

The AO IE task for the item 'speaker' gives for the document D in Figure 1: $\mathcal{F}_{sp}(D) = \{(54, 55), (57, 58)\}$.

Note that in the first case ('location'), both answers indicate different occurrences of the same location. In the second case ('speaker'), both answers indicate a different speaker. However, since we want to find every occurrence of every instance, this doesn't matter in the formal setting.

Definition 3 *A* solution for an AO IE task $\mathcal{F}$ for a particular type of information *is a function*

$\widehat{\mathcal{F}} : \mathcal{D} \rightarrow 2^{N \times N} : D \mapsto \{(i_k,j_k);\ 1 \leq i_k \leq j_k \leq |D|\}$

with

- i_k *and* j_k *indicate the left and right boundaries of a text fragment;*
- (i_k,j_k) *is called a* prediction*;*
- $|D|$ *is the number of tokens in the document.*

An IE system provides us with a function $\widehat{\mathcal{F}}$ *for each type of information.*

Example 3 *A possible solution for* $\mathcal{F}_{loc}$ *is* $\widehat{\mathcal{F}_{loc}}(D) = \{(18, 25), (50, 50), (93, 101)\}$*. This function reflects the situation of a trained IE system returning annotations at positions (18,25), (50,50) and (93,101) as fillings for the slot "location" in document D.*

[1] If an IE system tries to solve different tasks at once, we still can evaluate each task separately.

2.3 One Best per Document

The *One Best per Document* (OBD) approach can be seen as filling a template, as was shown in Example 1. For evaluation of an IE system following the OBD approach, the exact location of the items is not important.

Note: $(x_1, \ldots, x_n) \sqsubseteq (y_1, \ldots, y_m)$ indicates that $(x_1, \ldots, x_n)$ is a subsequence of $(y_1, \ldots, y_m)$. That is, $\exists j \forall i = 1 \ldots n : x_i = y_{j+i}$ (thus $(x_1, \ldots, x_n)$ is a contiguous subsequence).

Definition 4 *A* One Best per Document Information Extraction task for a particular type of information *is a function*

$\mathcal{F} : \mathcal{D} \rightarrow \{$*set of sequences*$\} : D \mapsto \{(a_1, \ldots, a_k); (a_1, \ldots, a_k) \sqsubseteq D\}$. *We call* $(a_1, \ldots, a_k)$ *a* template slot filling*, or an* answer*.*

An OBD IE task *is a set of functions, one for each template slot.*

Example 4 *The OBD IE task for the item 'location' for the document D in Figure 1 would give* $\mathcal{F}_{loc}(D) = \{(\texttt{CMU}; ,; \texttt{Adamson}; \texttt{Wing}; (; \texttt{Baker}; \texttt{Hall};))\}$.

The OBD IE task for the item 'speaker' for the document D in Figure 1 would give $\mathcal{F}_{sp}(D) = \{(\texttt{Al}; \texttt{Roth}), (\texttt{Ido}; \texttt{Erev})\}$.

Note that in the first case ('location'), only 1 slot filling is mentioned, although there are multiple occurrences of this slot in the document. However, in the OBD approach, we don't mind which occurrence has been found. In the second case ('speaker'), both answers indicate a different speaker, as in the AO setting, but this time we are not interested in the exact positions of the slot fillings, we only care about the content of the fillings.

Definition 5 *A* solution for an OBD IE task for a particular type of information *is a function*

$\widehat{\mathcal{F}} : \mathcal{D} \rightarrow \{$*set of sequences*$\} : D \mapsto \{(p_1, \ldots, p_n); (p_1, \ldots, p_n) \sqsubseteq D\}$. *We call* $(p_1, \ldots, p_n)$ *a* template slot prediction*.*

An IE system provides us with a function $\widehat{\mathcal{F}}$ *for each type of information.*

Example 5 *A solution for the OBD IE task* $\mathcal{F}_{loc}$ *for the item 'location' for the document D in figure 1 is* $\widehat{\mathcal{F}_{loc}}(D) = \{(\texttt{Adamson}; \texttt{Wing}; (; \texttt{Baker}; \texttt{Hall};))\}$.

A solution for the OBD IE task $\mathcal{F}_{sp}$ *for the item 'speaker' for the document D in figure 1 is* $\widehat{\mathcal{F}_{sp}}(D) = \{(\texttt{Al}; \texttt{Roth}), (\texttt{Ido})\}$.

3 Evaluating Information Extraction Systems

3.1 Preliminaries

We evaluate each template slot/kind of information separately. If necessary, the overall scoring of an IE system can easily be computed by averaging over all results.

A typical way to evaluate an IE system is by using a confusion matrix. This is a well-known technique of counting results. Figure 3.1 shows a confusion matrix.

For each extracted entity, we have to evaluate if it is correct (and thus a *true positive*) or not (and thus a *false positive*). The *false negative* in the matrix is the number of entities that should have been extracted, but haven't. In IE applications, the *true negative* is usually not used.

All usually reported measures (recall, precision,...) can be computed from the confusion matrix.

In order to fill the matrix, we have to compare $\mathcal{F}(D)$ and $\widehat{\mathcal{F}}(D)$ for each document D in the test set. It is obvious that the way how we fill the matrix, and thus the way we decide if an extracted

		prediction	
		+	-
answer	+	true positive	false negative
	-	false positive	true negative

Figure 2: Confusion matrix: the columns indicate the decision of the system, the rows indicate the correct decisions.

entity ("prediction") is correct or not, is crucial in the computation of the scores. D. Freitag gives three different possibilities [2]:

- *exact rule*: a prediction is only correct, if it is exactly equal to an answer. Thus, if 'Al Roth' is the given answer, 'dr. Al Roth' would not be correct.
- *contain rule*: a prediction is correct, if it contains an answer, plus possibly a few extra neighboring tokens. Thus, if 'Al Roth' is the given answer, 'dr. Al Roth' would be correct as well.
- *overlap rule*: a prediction is correct, if it contains a part of a correct instance, plus possibly some extra neighboring tokens. Thus, if 'dr. Al Roth' would be the given answer, 'Al Roth' would be correct as well.

What rule is preferable, depends on the situation at hand. If the goal of the IE system is to fill a database on which queries can be asked, one wants to make sure that whatever gets into the database is exact. On the other hand, if someone has as task to point out the job title in each document in a set of job advertisements, a system that gives a solution overlapping with the correct answer for each document will be more helpful than one that returns the exact answer for only a small portion of the documents.

In the following two subsections, we define the above mentioned rules formally in the AO and the OBD settings.

3.2 All Occurrences

We first define a few intermediate operators:

Definition 6 *Let $1 \leq i \leq j \leq N$, $1 \leq i_k \leq j_k \leq N$, and let e, m be positive integers.*

(a) $(i_k, j_k) \equiv_E (i, j) \Leftrightarrow i = i_k$ *and* $j = j_k$

(b) $(i_k, j_k) \equiv_{C(e)} (i, j) \Leftrightarrow [i, j] \subseteq [i_k, j_k]$ *and* $(j_k - i_k) - (j - i) \leq e$

(c) Define first:

- $ml = max(0, i_k - i)$*, the number of missing tokens at the left;*
- $mr = max(0, j - j_k)$*, the number of missing tokens at the right;*
- $el = max(0, i - i_k)$*, the number of extra tokens at the left;*
- $er = max(0, j_k - j)$*, the number of extra tokens at the right.*

then, $(i_k, j_k) \equiv_{O(e,m)} (i,j) \Leftrightarrow [i_k, j_k] \cap [i,j] \neq \{\}$*, and* $el + er \leq e$ *(thus there are at most* e *extra tokens), and* $ml + mr \leq m$ *(thus there are at most* m *missing tokens).*

The first rule indicates when 2 tuples are exactly the same. The second rule defines when a tuple contains another tuple, plus at most e neighboring tokens. Figure 3 illustrates the $\equiv_{O(e,m)}$-operator.

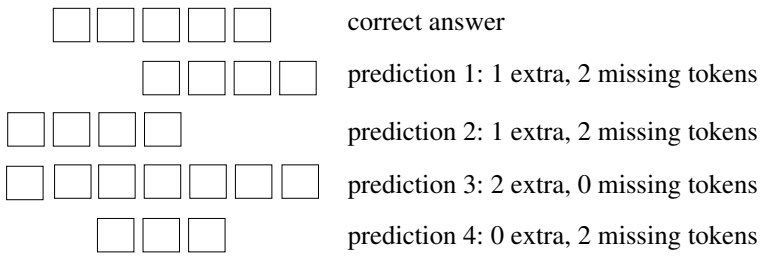

Figure 3: Illustration of the $\equiv_{O(e,m)}$-operator.

It is obvious from the definitions that $\equiv_{C(0)}$ is equal to $\equiv_E$, and that $\equiv_{O(e,0)}$ is equal to $\equiv_{C(e)}$.

We are now able to define the three operators. A prediction p belongs to the set of answers, and thus is correct, if and only if the prediction is "equal" in one of the above senses to an answer.

Definition 7 *Let* $p = (i_k, j_k)$ *be a prediction for document* D*.*

$p\ in_E\ \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (i,j) \in \mathcal{F}(D) : (i_k, j_k) \equiv_E (i,j)$

$p\ in_{C(e)}\ \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (i,j) \in \mathcal{F}(D) : (i_k, j_k) \equiv_{C(e)} (i,j)$

$p\ in_{O(e,m)}\ \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (i,j) \in \mathcal{F}(D) : (i_k, j_k) \equiv_{O(e,m)} (i,j)$

We are now ready to fill the confusion matrix. Figure 4 states the algorithm that fills the matrix.

Note that in the case that $\mathcal{F}(D) = \{\}$ and $\widehat{\mathcal{F}}(D) = \{\}$, this is not counted as a true positive, nor as a false positive or false negative.

Example 6 illustrates the algorithm.

Example 6 $\mathcal{F}(D) = \{(18,25), (93,100)\}$

$\widehat{\mathcal{F}}(D) = \{(20,25), (93,101), (50,52)\}$

Confusion matrix, using in_E*:*

		prediction	
		+	-
answer	+	*0*	*2*
	-	*3*	

Confusion matrix, using $in_{C(1)}$*:*

		prediction	
		+	-
answer	+	*1*	*1*
	-	*2*	

Confusion matrix, using $in_{O(1,2)}$*:*

Given: an in_θ-operator as defined. θ is one of E, $C(e)$ or $O(e,m)$.
for each document D in the testset:

- if $\mathcal{F}(D) = \{\}$ and $\widehat{\mathcal{F}}(D) \neq \{\}$:

  (The IE system did some predictions, while the information needed was not present in the document, thus all predictions must be false.)

  add $\#\widehat{\mathcal{F}}(D)$ to the number of false positives.

- else

  - for each prediction $p \in \widehat{\mathcal{F}}(D)$:
    if p in_θ $\mathcal{F}(D)$: add 1 to the number of true positives
    else add 1 to the number of false positives.
  - for each answer $a \in \mathcal{F}(D)$, such that $\neg\exists p \in \widehat{\mathcal{F}}(D) : p \equiv_\theta a$

    (the answer a is not covered by the predictions made by the system)

    add 1 to the number of false negatives.

Figure 4: Filling the confusion matrix

		prediction	
		+	-
answer	+	*2*	*0*
	-	*1*	

3.3 One Best per Document

As in the AO-setting, we first define a few intermediate operators:

Definition 8 *Let $(p_1, \ldots, p_n)$ and $(a_1, \ldots, a_k)$ be sequences of tokens.*

(a) $(p_1, \ldots, p_n) \equiv_E (a_1, \ldots, a_k) \Leftrightarrow n = m \wedge \forall i = 1 \ldots n : p_i = a_i$

(b) $(p_1, \ldots, p_n) \equiv_{C(e)} (a_1, \ldots, a_k) \Leftrightarrow k \leq n \leq k + e \wedge (a_1, \ldots, a_k) \sqsubseteq (p_1, \ldots, p_n)$

(c) $(p_1, \ldots, p_n) \equiv_{O(e,m)} (a_1, \ldots, a_k)$ *if and only if one of the following four cases applies:*

1. *The answer is included in the prediction, and at most e tokens in the prediction are extra w.r.t. the answer; that is:*

$$(p_1, \ldots, p_n) \equiv_{C(e)} (a_1, \ldots, a_k)$$

2. *The prediction is included in the answer, and at most m tokens in the answer are missing in the prediction; that is:*

$$(a_1, \ldots, a_k) \equiv_{C(m)} (p_1, \ldots, p_n)$$

3. *The right side of the prediction overlaps with the left side of the answer, and the prediction misses at most m tokens (on the right) of the answer, and has at most e tokens extra (on the left side) w.r.t. the answer; that is:*

$$\exists i \geq n-k: \quad a_1 = p_{i+1}, a_2 = p_{i+2}, \ldots, a_{n-i} = p_n, \\ \text{and } i \leq e \text{ and } k-(n-i) \leq m$$

4. *This case is the mirror image of case 3; the left side of the prediction overlaps with the right side of the answer, and the prediction misses at most m tokens (on the left) of the answer, and has at most e tokens extra (on the right side) w.r.t. the answer; that is:*

$$\exists i \geq k-n: \quad a_{i+1} = p_1, a_{i+2} = p_2, \ldots, a_k = p_{k-i}, \\ \text{and } i \leq m \text{ and } n-(k-i) \leq e$$

The first rule indicates when two sequences are exactly equal. The second one contains 2 parts: the first part defines that there are at most e extra tokens, and no missing tokens, the second part requires that $(t'_k)_k$ is a subsequence of $(t_n)_n$, as defined in Section 2.3. Figure 5 illustrates the third rule.

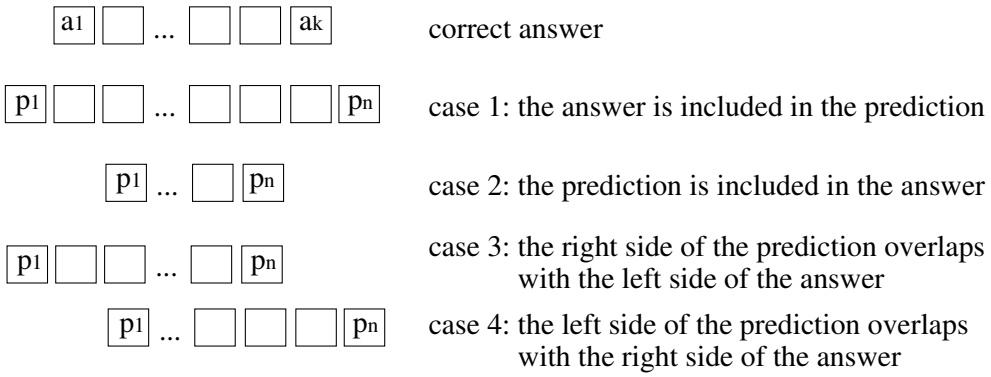

Figure 5: Illustration of the $\equiv_{O(e,m)}$-operator in the OBD-setting.

Again, it is obvious from the definitions that $\equiv_{C(0)}$ is equal to $\equiv_E$, and that $\equiv_{O(e,0)}$ is equal to $\equiv_{C(e)}$.

We are now able to define the three operators. A prediction p belongs to the set of answers, and thus is correct, if and only if the prediction is "equal" in one of the above senses to an answer.

Definition 9 *Let $(p_1, \ldots, p_n)$ be a prediction for document D.*

$$(p_1, \ldots, p_n) \; in_E \; \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (a_1, \ldots, a_k) \in \mathcal{F}(D) : (p_1, \ldots, p_n) \equiv_E (a_1, \ldots, a_k)$$
$$(p_1, \ldots, p_n) \; in_{C(e)} \; \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (a_1, \ldots, a_k) \in \mathcal{F}(D) : (p_1, \ldots, p_n) \equiv_{C(e)} (a_1, \ldots, a_k)$$
$$(p_1, \ldots, p_n) \; in_{O(e,m)} \; \mathcal{F}(D) \quad \Leftrightarrow \quad \exists (a_1, \ldots, a_k) \in \mathcal{F}(D) : (p_1, \ldots, p_n) \equiv_{O(e,m)} (a_1, \ldots, a_k)$$

We can now use the algorithm in Figure 4 again to fill the confusion matrix.

Example 7 illustrates the algorithm.

Example 7 $\mathcal{F}(D) = \{(\texttt{Al}; \texttt{Roth}), (\texttt{Ido}; \texttt{Erev})\}$

$\widehat{\mathcal{F}}(D) = \{(\texttt{Al}; \texttt{Roth}), (\texttt{Ido}), (\texttt{Cristina}; \texttt{Bicchieri})\}$

Confusion matrix, using in_E:

		prediction	
		+	-
answer	+	*1*	*1*
	-	*2*	

Confusion matrix, using $in_{O(1,2)}$*:*

		prediction	
		+	-
answer	+	*2*	*0*
	-	*1*	

4 Evaluation measures

Since we evaluate each type of information in isolation with a separate confusion matrix, we are able to calculate different evaluation measures for each slot. In practice, it has been found that the difficulty of the IE task depends on the field that has to be recognized, e.g., an email address is easier to recognize than a job title. Thus, it is certainly useful to report evaluation measures for each of the tasks (i.e. speaker, start-time, end-time and location) separately. One can obtain an overall scoring of an IE system by computing the (weighted) average over all results.

The most used evaluation measures in IE are *recall*, *precision* and *F-measure*. Recall and precision can be calculated from the confusion matrix [5]:

Definition 10

$$recall = \frac{true\,positive}{true\,positive + false\,negative}$$

$$precision = \frac{true\,positive}{true\,positive + false\,positive}$$

Recall indicates how many of the items that should have been found, are effectively extracted. Precision indicates how many of the extracted items are correct. Usually there is a trade off of recall against precision. Therefor, often an average accuracy is reported in the form of the F_β-measure [5]. This is a harmonic mean which in its most general form provides a weight to determine the bias towards recall or precision. More commonly it is used with equal weighting for precision and recall, as in Definition 11.

Definition 11

$$F = \frac{2 \times p \times r}{p + r}$$

[7] discusses other possibilities for averaging over recall and precision, but those are to our knowledge not used in IE.

Often only the F-measure is reported as the evaluation measure of an IE system. If the same weighting for recall and precision is used in calculating the F-measure, this gives an indication of which system is the better one. However, often it may be important to know the individual

recall and precision scores of a system to be able to fully compare different systems. When one system has a recall of 10% and a precision of 90%, this will obtain the same F-measure as a system which obtains a recall of 90% and a precision of 10%, even though both systems are very different. Differences on how a system scores w.r.t. recall and precision will be unnoticed when reporting only F-measure.

Other measures that are used in text categorization applications, are *fallout* and *overlap* [4].

Definition 12

$$fallout = \frac{false\,positive}{false\,positive + true\,negative}$$

$$overlap = \frac{true\,positive}{true\,positive + false\,positive + false\,negative}$$

Since we don't count *true negative* in IE, fallout is not useful in this domain. Overlap may be used, but it is not clear what kind of extra information this measure gives.

5 Conclusions

This paper deals with a formal framework for evaluation of Information Extraction. We gave a formal definition of IE and defined how evaluation can be done, based on the notion of a confusion matrix. We gave a short discussion of the evaluation measures most commonly used in IE.

This paper does not deal with some other important problems concerning evaluation and comparison of IE:

- How to describe the corpus used for training and testing, and how to make sure that in comparing the results of different IE systems, the exact same corpus is used.
- Which train-test strategy is used?

To be able to compare results of different IE systems, it should not only be clear how the confusion matrix was built (as we describe in this paper), but these two questions should be adequately answered as well.

References

[1] M. E. Califf and R. J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. In *Journal of Machine Learning Research 4*.

[2] D. Freitag. Machine learning for information extraction in informal domains. In *Phd thesis, Carnegie Mellon University, Pittsburgh PA.*, 1998.

[3] D. Freitag and A. McCallum. Information extraction using hmms and shrinkage. In *Proc. AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

[4] D. Lewis. Evaluating Text Categorization. In *Proceedings of the Speech and Natural Language Workshop.* Asilomar, Feb. 1991.

[5] C. Manning, H. Schutze. Foundations of Statistical Natural Language Processing. *Cambridge, MA: MIT Press.* 1999.

[6] A. McCallum, W. Cohen. Information Extraction from the World Wide Web. *Tutorial at NIPS 2002, Vancouver BC, Canada*, Dec. 9, 2002.

[7] D. M. W. Powers. Recall & Precision versus The Bookmaker. In *Joint International Conference on Cognitive Science*, July 2003.

[8] RISE A repository of online information sources used in information extraction tasks. http://www.isi.edu/info-agents/RISE/index.html. University of Southern California, Information Sciences Institute.

[9] J. Zavrel and W. Daelemans. Feature-rich memory-based classification for shallow nlp and information extraction. In *Text Mining. Theoretical aspects and applications. Springer LCNS series.*, 2003.