EVALIEX – A Proposal for an Extended Evaluation Methodology for Information Extraction Systems

Christina Feilmayr[1], Birgit Pröll[1], Elisabeth Linsmayr[2]
[1] Johannes Kepler University Linz
Altenberger Straße 69, 4040 Linz, AUSTRIA
[2] NETCONOMY Software & Consulting
Hilmgasse 4, 8010 Graz, AUSTRIA
E-mail: cfeilmayr@faw.jku.at, bproell@faw.jku.at, elisabeth.linsmayr@netconomy.net

Abstract

Assessing the correctness of extracted data requires performance evaluation, which is accomplished by calculating quality metrics. The evaluation process must cope with the challenges posed by information extraction and natural language processing. In the previous work most of the existing methodologies have been shown that they support only traditional scoring metrics. Our research work addresses requirements, which arose during the development of three productive rule-based information extraction systems. The main contribution is twofold: First, we developed a proposal for an evaluation methodology that provides the flexibility and effectiveness needed for comprehensive performance measurement. The proposal extends state-of-the-art scoring metrics by measuring string and semantic similarities and by parameterization of metric scoring, and thus simulating with human judgment. Second, we implemented an IE evaluation tool named EVALIEX, which integrates these measurement concepts and provides an efficient user interface that supports evaluation control and the visualization of IE results. To guarantee domain independence, the tool additionally provides a *Generic Mapper for XML Instances* (GeMap) that maps domain-dependent XML files containing IE results to generic ones. Compared to other tools, it provides more flexible testing and better visualization of extraction results for the comparison of different (versions of) information extraction systems.

Keywords: Information Extraction Evaluation, Evaluation Methodology, Performance Measures

1. Introduction

A vast amount of online information appears in collections of unstructured text, which is the predominant medium for information exchange among people. The volume of available text resources requires techniques such as information extraction (IE) as a prerequisite for efficient location, retrieval, and management of relevant information. IE is commonly defined as extracting structured data from unstructured data as provided, for instance, in textual documents (Appelt & Israel, 1999; Cunningham, 1997). The facts to be extracted are also called *information extraction templates*. Each template usually consists of several slots in the form of attribute-value pairs.

Assessing the correctness of extracted data requires evaluating the performance of an IE system, which is accomplished by calculating quality metrics.

IE performance evaluation has a long history; most of the IE evaluation methodology was developed in the course of the Message Understanding Conference (MUC '87-'97) [1] (Chinor & Dungca, 1995; Douthat, 1998). The main contributions of MUC were, on the one hand, adapting the concepts of *precision* and *recall* (borrowed from information retrieval (IR)) to score filled templates and, on the other hand, providing a fully automated scoring software to measure IE performance. For a given test corpus, algorithms align templates that were filled automatically by the IE system, so-called *hypotheses*, with manually filled templates that establish the correct results or *references*. The corresponding slots are then classified as either *correct*, *partially correct* or *incorrect*.

Developing a knowledge-based or an automatically trainable IE system is time- and labor-intensive. The former requires extraction rules to be adapted continually, and the latter requires the features and parameters of learning methods to be selected and tested carefully. The challenge in this iterative engineering process is that extraction rules must be (i) sufficiently generic to extract the full extent of available information and (ii) sufficiently specific to extract relevant information according to a given specification. Therefore, continuous evaluation is indispensable. Even though MUC came up with the basic methodology and metrics of IE evaluation, the field still lacks standard datasets, evaluation procedures, and appropriate measures (Lavelli et al., 2008) which consider, for instance, certain peculiarities of natural language, such as semantic similarities. In summary, there are numerous open research questions, including: *How to evaluate the occurrence of various values for one slot? How to evaluate the similarity between reference and hypothesis beyond the correct/ partially correct/incorrect classification*? *How to select the best value with which to populate any given slot? How to evaluate complex (and accordingly nested) templates?* Existing IE evaluation tools lack transparency and flexibility

[1] http://www-nlpir.nist.gov/related_projects/muc/index.html, last visit: March, 5th 2012

in scoring and provide insufficient support in performing the evaluation.

The main contribution of our research work is twofold: (i) Based on longtime research and practical experience in IE, we identified concepts missing in traditional measures and devised an extended evaluation methodology that provides the flexibility and effectiveness needed for comprehensive performance measurement. (ii) We implemented an IE evaluation tool named EVALIEX, which integrates these measurement concepts and provides an efficient user interface that supports evaluation control and the visualization of IE results. To guarantee domain independence, the tool additionally provides a *Generic Mapper for XML Instances* (GeMap) (Linsmayr, 2010) that maps domain-dependent XML files containing IE results to generic ones.

The remainder of this paper is structured as follows: Section 2 presents the traditional approach to evaluating IE, evaluation metrics applied, and selected state-of-the-art evaluation tools. Section 3 describes the evaluation methodology in detail and its integration in EVALIEX. Section 4 discusses the validation of the IE evaluation methodology. Additionally, the applicability of EVALIEX is shown in three scenarios taken from IE applications in the areas of eRecruitment (*JobOlize*), eTourism (*TourIE*) and eManufacturing (*Marlies*) (Pröll et al., 2009). Finally, Section 5 concludes the paper with a critical reflection in terms of lessons learned and different aspects of future work.

2. State-of-the-Art Evaluation Measures & Methodologies

The focus of our research work is the evaluation of IE system performance (also termed *IE assessment*) (Cole et al., 2004), which enables the comparison of two (or more) alternative implementations, such as alternative IE systems or different versions of one IE system that are produced during system development or maintenance.

2.1 Performance Measures

Established criteria for performance evaluation are *precision* (P), (i.e., the number of slots filled *correctly* (C) divided by the *number of fills attempted* (M)) and *recall* (R) (i.e., the number of slots filled correctly divided by the *number of possible correct fills* (N) according to manual extraction). Precision (cf. 1) deals with *substitution* (S) and *insertion* (I) errors, while recall (cf. 2) deals with *substitution* and *deletion* (D) errors. The *F-measure* (F) considers all three types of errors and is defined as the weighted harmonic mean of precision and recall (cf. 3).

$$P = \frac{C}{M} = \frac{C}{C+S+I} \tag{1}$$

$$R = \frac{C}{N} = \frac{C}{C+S+D} \tag{2}$$

$$F = \frac{PR}{(1-\alpha)P + \alpha R}, 0 \le \alpha \le 1 \tag{3}$$

The error measure E corresponds to the F-measure (cf. 4).

$$E = 1 - F = \frac{S + (1-\alpha)D + \alpha I}{C + S + (1-\alpha)D + \alpha I}, 0 \le \alpha \le 1 \tag{4}$$

Also of interest is the *accuracy* of a system, which is expressed by the error rate ERR (cf. 5). The primary limitation of the F-measure is that it implicitly discounts the overall error rate, which makes a system appear to perform better than it actually does. In other words, no matter what weight is chosen for α (for combining P and R), the subtraction of weight from D and I relative to S is guaranteed. ERR removes the subtraction of weight from D and I by simply removing the α weights in (4) (Chinor & Dungca, 1995) (Makhoul et al., 1999).

$$ERR = \frac{S + D + I}{C + S + D + I} \tag{5}$$

The Slot Error Rate (SER, cf. 6) is the ratio of the total number of slot errors divided by the total number of slots in the reference, which is fixed for a given test. The insertion errors I are removed from the denominator in (5).

$$SER = \frac{S + D + I}{N} = \frac{S + D + I}{C + S + D} \tag{6}$$

The limitations of these performance measures are discussed in a number of research papers (De Sitter & Daelemans, 2003; Kim & Woodland, 2003; Makhoul et al., 1999). Furthermore, they do not consider natural language peculiarities (e.g., string similarity and semantic similarities, such as synonyms) and information extraction aspects (e.g., boundaries of extracted templates, nested templates). Therefore, the parameterization of these measures must be revised. Current evaluation tools implement only the "core" information extraction performance measures – recall, precision, and F-measure.

2.2 Evaluation Tools

Below we discuss three widely used IE evaluation tools. While *GATE* and *Ellagon* represent text-engineering frameworks, which include a proprietary evaluation tool, *ANNALIST* constitutes an independent evaluation tool.

GATE[2] **(General Architecture for Text Engineering)** (Cunningham et al., 2010) provides a GNU-licensed open source software and offers an infrastructure for developing and deploying software components that process human language. A core plug-in of GATE termed *Annotation Diff* tool provides the measurement, evaluation, and benchmarking of IE systems. The *Annotation Diff* tool

[2] http://www.gate.ac.uk, last visit: March 5th 2012

compares two annotation sets within a document – one annotated by the system (*hypothesis*) and one annotated by hand (*reference*). Precision, recall, and F-measure are computed for each annotation type. All measures can be calculated according to three different criteria - strict, lenient, and average.

- *Strict measure* considers all partially correct responses to be incorrect.
- *Lenient measure* considers all partially correct responses to be correct.
- *Average measure* assigns half weight to partially correct responses (i.e., takes the average of strict and lenient).

After performing the comparison, the results are visualized in the *Annotation Diff* window. Additionally, GATE offers a *Corpus Quality Assurance (QA)* tool, which enables comparison of several documents (corpus) and annotation types. Like the *Corpus QA* tool, the corpus benchmark tool permits evaluation across a whole corpus. Unlike *Corpus QA*, it achieves this by using matched corpora and not by comparing annotation sets within a corpus. It supports tracking of the system performance over time.

Ellagon (Petasis et al., 2002) is a multi-lingual, cross-platform, general-purpose text-engineering environment. Ellagon provides a generic framework in which external components can be embedded; it provides the infrastructure for:

- Managing, storing, and exchanging textual data.
- Creating, embedding, and managing linguistic processing components; facilitating communication between different linguistic components by defining an API.
- Visualizing and comparing textual data.

The *Collection Comparison Tool* of Ellagon evaluates the divergence of two corpora, their annotations, and attributes. In addition, the measurement tool calculates precision, recall, and F-measure. Detailed information about the evaluation can be gathered from *Ellagon's log tool.*

ANNALIST (Annotation Alignment and Scoring Tool) (Demetriou et al., 2008) is an evaluation system that is easily extensible and configurable for different domains, annotation tasks, and input formats for both system developers and system users. ANNALIST offers a variety of options that allow it to be used as a "black box" system via the input of annotations in XML format. ANNALIST provides an *Alignment Module*, which is accessed to provide mappings between key and response annotations. The *Scoring Module* takes input from the *Alignment Module* and produces scores according to the metrics chosen for evaluation, i.e., precision, recall, and F-measure. Finally, this data is passed to, and formatted by, the *Output Module*. During the pairing and alignment of annotations, an annotation can be classified as correct, partially correct, incorrect, missing, or spurious. ANNALIST's matching criteria for entities include options for strict matching or inclusiveness, which allows for partial (substring) matching. ANNALIST produces two kinds of report: a *scores report* providing the scores of both individual documents and the collection as a whole and an *alignment report* providing alignment and scoring details of all annotations during the evaluation.

All tools mentioned are useful not only for providing a final performance measure, but also for aiding system development by tracking progress and evaluating the impact of changes as they are made. A summary of each system (*GATE*, *Ellagon,* and *ANNALIST)* and a comparison to EVALIEX are presented in Section 4.

The main reason that led to the development of a proposal for a new evaluation methodology and the tool EVALIEX was the inadequacy of existing tools in fulfilling particular requirements for the evaluation of information extraction. These include, for instance, determining the significance and the weights of slots, setting a positive or negative scoring of an error, measuring string similarity, and taking synonyms and possible multiple fillers for a single slot into account. EVALIEX satisfies these requirements, as shown in Section 4.

3. Evaluation Methodology for EVALIEX

Existing IE evaluation approaches are unsatisfactory and cannot meet the mentioned challenges. IE system developers and users require an efficient, accurate, domain-independent, and user-friendly evaluation tool.

3.1 Requirements for an Evaluation Methodology

Based on the considerations stated, the following requirements for an extended IE evaluation methodology were identified:

Standardized schema. A standardized evaluation methodology requires a standardized format for extracted information and a common approach to scoring the differences between hypothesis and reference templates (Lavelli et al., 2004). These requirements lead to the need for (i) a standardized format for the reference and hypothesis templates (an XML schema termed EVALIEX schema), (ii) an import function, and (iii) a mapping of reference and hypothesis templates to the standardized EVALIEX schema. The EVALIEX schema enables import of data in heterogeneous description languages by using a preprocessing parser offered by *GeMap*. Furthermore, it can handle nested templates.

Evaluation process transparency. Existing evaluation tools conceal their scoring process. Thus, to provide greater transparency, a visualization of various error types is proposed. It focuses primarily on the examination of missing slots (*deletion D*). In addition, the tool should enable its users to parameterize the performance metrics.

For flexible and effective scoring, the ability to set the following parameter options is necessary:

1. Significance of slots (necessary or optional)
2. Weighting of slots w_{Slot} (0..1)
3. Scoring of error (positive or negative, e.g., useful and correct additional insertions should be evaluated as positive)
4. Weighting of F-measure (α)
5. Specifying the boundaries of information extraction templates.
6. Selecting the approach to extracting slot fillers (see below).

Multiple possible fillers for a single slot can influence the performance. Lavelli et al. (Lavelli et al., 2004, 2008) identified three different filler variants – OAS, OAOD, and OADS.

Using the first variant, *One Answer per Slot (OAS),* means that the names "JKU Linz" and "Johannes Kepler University Linz" are considered to be one correct answer. The second filler variant, *One Answer per Occurrence in the Document (OAOD),* establishes that each individual appearance of a string must be extracted from the document, and each occurrence of "JKU Linz" would be counted separately. The third variant *One Answer per Different String (OADS)* means that two separate occurrences of "JKU Linz" are considered to be one answer, but "Johannes Kepler University Linz" is yet another one.

Support in reference template construction. The construction of manually coded reference templates is a time-consuming process. Therefore, users need support in constructing these templates in the corresponding common EVALIEX schema. The reference construction requires displaying the original website or document, in which the user defines the reference templates, their attributes, and values.

3.2 Design Issues of the Evaluation Methodology underlying EVALIEX

The comparison and scoring of extracted information templates (hypotheses) offered by an IE system are based on three main steps: (i) correct mapping of a hypothesis to its reference template, (ii) identification of error type (insertion, deletion, substitution) and calculation of performance metrics, and (iii) adaptation of performance metrics by user-defined parameters (string and semantic similarities).

3.2.1 Mapping Hypothesis to Reference Templates

For the correct mapping of a hypothesis template to its reference template, an adapted version of the *General Greedy Mapping Algorithm* (Douthat, 1999) is used. This algorithm must be extended in the following ways: First, it must consider nested templates and multiple slot values. Second, it must take the significance of slots into account. When there are multiple fillers for a slot, the selected approach (OAS, OAOD, OADS) is considered. Error computation and the subsequent performance measurement result from the hypothesis/reference mapping at each hierarchical level (*template, slot,* and *value levels*).

3.2.2 Calculating the Scoring Metrics

EVALIEX allows flexible parameterization of performance measures. As mentioned in the requirements for evaluation methodologies, users and developers of an evaluation system need an error weighting for the error's positive or negative scoring. Therefore, the existing performance measures – precision, recall, F-measure, error measure, error rate, and slot error rate – must be adapted. The definitions of the performance measures described in Section 2 were modified by multiplying each type of error with an individual weight (w_I, w_D, and w_S).

For a positive or negative scoring of the error type, the following approach was implemented:

- *Positive Scoring* is equivalent to subtracting weight from the error type. In EVALIEX it is realized by a negative parameter (negative floating point number).
- *Negative Scoring* is equivalent to adding weight to the error type. EVALIEX requires a positive floating point number as a parameter (> 1).
- *Without Scoring*; in this case the error type is not weighted in the performance measures (parameter = 1).

With regard to the flexibility of parameterization, the performance measures are adapted (cf. 7-10) as follows:

$$P = \frac{C_g}{C + S + I} \tag{7}$$

$$R = \frac{C_g}{C + S + D} \tag{8}$$

$$ERR = \frac{S_g * w_S + D_g * w_D + I_g * w_I}{C + S + D + I} \tag{9}$$

$$SER = \frac{S_g * w_S + D_g * w_D + I_g * w_I}{C + S + D} \tag{10}$$

The F-measure requires the parameter α; its most commonly used value is $\alpha = 0.5$.

The different error types (w_I, w_D, w_S) and the template slots (w_{Slot}) are weighted in the enumerators to ensure a constant denominator for the metric scoring. The individual error types C_g, S_g, D_g, and I_g are weighted sums in the hypothesis templates.

3.2.3 Semantic Similarity and String Similarity

Final considerations are the semantic and string similarities between the reference and hypothesis templates. There are many ways to provide a more effective and flexible mapping of the templates, for instance, by incorporating algorithms that measure string similarity (Jaccard algorithm (Camacho & Salhi, 2006; Cohen et al., 2003), Levenshtein

algorithm[3]) or by using thesauri to determine homonyms and synonyms.

Semantic Similarity Score. EVALIEX provides the possibility to define lists of synonyms for each string in the reference template. In the mapping process the system checks whether there is a corresponding list entry and returns a correct annotation of the hypothesis template slot.

String Similarity Score. EVALIEX provides a parameter to determine the template boundaries, which influences the mapping rules, and in addition it calculates the Jaccard-coefficient to measure string similarity. DeSitter et al. (De Sitter et al., 2003, 2004) and Freitag (Freitag, 1998) described rules for determining the correctness of an extracted template, which is crucial in computing the scores. There are three different approaches for defining flexible template boundaries: *exact, contains* and *overlap*.

The *exact* rule says that the hypothesis is correct when it is identical to the reference. The *contain* rule establishes that the hypothesis is correct when it contains the reference and up to *e* additional neighboring tokens. The *overlap* rule says that a hypothesis template is correct when it contains any part of a correct instance and some extra (*e..extra, m..missing*) neighboring tokens. In the following cases, the *overlap* rule classifies a hypothesis template as correct:

- String of the hypothesis contains the reference string plus e neighboring tokens.
- Reference contains hypothesis, and the hypothesis misses a maximum of m neighboring tokens.
- The right part of the hypothesis and the left part of the reference overlap, and the hypothesis misses a maximum of m neighboring tokens at the right boundary and has a maximum of e additional neighboring tokens at the left boundary.
- The left boundary of the hypothesis overlaps with the right boundary of the reference, and the hypothesis misses a maximum of m neighboring tokens on the left side and has a maximum of e additional neighboring tokens on the right side.

Combining mapping rules and calculating the Jaccard-coefficient results in a more precise string similarity measure. Additionally, the parameters *e* (extra token), *m* (missing token), and the setting that defines whether *m* and *e* correspond to one or more tokens or several characters provide maximum flexibility and efficiency in the evaluation of string similarity.

Measuring the similarity of numeric values (e.g., the level of a job skill) is application-oriented. In EVALIEX the difference between the numeric values is calculated according to

$$1 - \left| \frac{Value_{Reference}}{100} - \frac{Value_{Hypothesis}}{100} \right| \qquad (11)$$

[3] http://www.levenshtein.net/, last visited: March, 5th 2012

3.3 Implementation Details of EVALIEX

The system architecture of EVALIEX is illustrated in Figure 1. Input is provided in the form of two XML documents; (i) the hypothesis and (ii) the corresponding schema. Subsequently, *GeMap* (1) produces the mapping of the hypothesis and the standardized EVALIEX schema to provide a hypothesis XML schema that can be read and processed by EVALIEX. After data input and before scoring, an XML document for the reference data (2) must be created (usually by a domain expert). In the graphical user interface of EVALIEX the filler variant (OAS, OAOD, OADS), the mapping rule (exact, contains, overlap), its parameters *m* and *e*, and a list of synonyms can be specified. In the evaluation module (3), the scoring of metrics is influenced by (i) individual slot weighting, (ii) the parameter(s) of the metrics, and (iii) the selected similarity measure. Each option is available in the evaluation table of the EVALIEX GUI. The data is finally passed to and formatted by the results module (4), which is used for presenting the results.

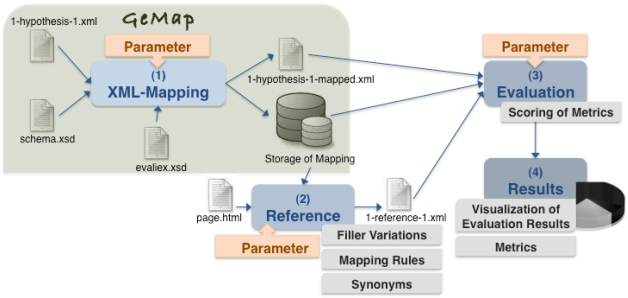

Figure 1: System Architecture of EVALIEX

4. Evaluation of EVALIEX

EVALIEX was used to evaluate the following three IE applications, which address different domains and IE tasks: *JobOlize*[4] deals with the extraction of job offers from Web pages, *TourIE*[5] addresses the extraction of relevant data from heterogeneously designed accommodation Web sites in the tourism domain, and *Marlies*[6] aims to extract the machines and production techniques as they are advertised on manufacturers' Web sites.

In the first part of this section, we highlight some of the differences in the scores produced by the EVALIEX evaluation methodology. The second part of the section provides a comparison of EVALIEX with the evaluation tools described in Section 2.

[4] www.faw.jku.at/index.php?id=55&PROJECT_ID=32&no_cache=1

[5] www.faw.jku.at/index.php?id=55&PROJECT_ID=34&no_cache=1

[6] www.faw.jku.at/index.php?id=55&PROJECT_ID=106&no_cache=1 last visit: March, 5th 2012

4.1 Evaluation Scenarios

Below we present examples of the projects *JobOlize*, *TourIE,* and *Marlies* and their evaluation results scored by EVALIEX. The evaluation scenarios show the scoring according to the EVALIEX methodology.

Scoring string similarity. The following examples show the differences between scorings of string similarity on the basis of tokens and characters.

Hypothesis	Reference	Parameter	Metrics
H Höfler HELIX	*Höfler HELIX 400*	base: token, overlap, e = 1, m=1	F = 1.0 ERR = 0.0 SER = 0.0
Java developer	*Java developer*	base: token, exact, threshold (th) = 0.8	F = 1.0 ERR = 0.0 SER = 0.0
Beach-Hotel SanMarino	*Hotel SanMarino*	base: char, contains, e = 4	F = 1.0 ERR = 0.0 SER = 0.0
Beach-Hotel SanMarino	*Hotel SanMarino*	base: char, exact	F = 0.0 ERR = 1.0 SER = 1.0
43-01-888 61 64	*0043-01-888 61 64*	base: char, overlap, m = 2	F = 1.0 ERR = 0.0 SER = 0.0

Scoring similarity of numeric values. The scoring of numeric values is based on calculating the difference between the hypothesis and the reference value. The following example describes the skill level required for a job. The threshold influences the correctness of the hypothesis.

Hypothesis	Reference	Parameter	Metrics
Level = 80	*Level = 100*	th = 0.8	F = 1.0 ERR = 0.0 SER = 0.0
Level = 80	*Level = 100*	th = 0.9	F = 0.0 ERR = 1.0 SER = 1.0

Scoring semantic similarity. Additional synonym lists influence the correctness of the hypothesis.

Hypothesis	Reference	Parameter	Metrics
private company limited by shares	*Ltd.*	list of synonyms not available	F = 0.0 ERR=1.0 SER=1.0
private company limited by shares	*Ltd.*	list of synonyms available, contains reference	F = 1.0 ERR=0.0 SER=0.0
private company limited by shares	*Ltd.*	list of synonyms available, does not contain reference	F = 0.0 ERR=1.0 SER=1.0

Filling variations and scoring. The following examples illustrate the effect of different approaches to slot filling.

Hypothesis	Reference	Parameter	Metrics
developer	*developer (f/m), developer*	OAS	F = 1.0 ERR = 0.0 SER = 0.0
developer	*developer (f/m), developer*	OAOD	F = 0.67 ERR = 0.33 SER = 0.5

Considering nested templates. The final example illustrates the evaluation of nested templates. The first template "address" incorporates the "country" template (= nested template). The first evaluation, which takes the country template into account, results in better scoring metrics than the second one, which evaluates the country template separately.

Hypothesis	Reference	Parameter	Metrics
Mainstr 31 NY 10019 USA	*Mainstr 31 NY 10019 USA*	address, allowed values = 1	F = 0.67 ERR = 0.48 SER = 0.95
Bakerstr 12 NY 10019 USA			
Mainstr 31 NY 10019 USA	*Mainstr 31 NY 10019 USA*	Address, allowed values = 1	F = 0.33 ERR = 0.73 SER = 1.45
Bakerstr 12 NY 10019 USA			

4.2 Comparison with other Evaluation Tools

Table 1 presents a comparison of EVALIEX with these widely used evaluation tools: *GATE's Annotation Diff Tool*, *Ellagon,* and *ANNALIST*. The following criteria were considered for the evaluation:

- *Scoring base*: What is the system's input ?
- *Comparison of extraction results and comparison of offset*: How are the extracted results compared?
- *Selection of templates*: Is it possible to select individual templates in the evaluation process?
- *Nested templates*: Is it possible to use nested templates

	GATE	Ellagon	ANNALIST	EVALIEX
Scoring base	corpus, 2 documents	corpus	corpus, documents	corpus, documents
Comparison of extraction results	offset, attributes	offset, attributes	offset, values	values
Comparison of offset	exact, overlap	exact, overlap	exact, overlap	-
Selection of templates	all, individual	individual, multiple	all	all, multiple, individual
Nested templates	no	no	no	yes
Semantic similarity	-	-	-	synonyms
String similarity	-	-	-	Jaccard-coefficient
Filler variation	-	-	-	OAS, OAOD, OADS
Mapping rules	exact, contains, overlap	-	exact, contains, overlap	exact, contains, overlap
Allowed fillers	-	-	-	yes
Parameter: weighting of measures	F-measure	offset	F-measure, substitution error	F-measure, templates, error type, Jaccard-coefficient
Parameter: weighting of template	-	-	-	yes
Evaluation metrics	recall, precision, F-measure	recall, precision, F-measure	recall, precision, F-measure	recall, precision, F-measure, E, ERR, SER
Interface	GUI	GUI	Command line	GUI
Export	HTML	-	TXT	HTML

Table 1: Comparison of Selected Evaluation Systems.

in the hypothesis template?

- *Semantic and string similarity*: Are there any measures/possibilities available to consider string and semantic similarity?
- *Filler variation*: Is it possible to select the approach to extracting slot fillers (OAS, OAOD, OADS)?
- *Mapping rules*: How are the hypothesis and the reference templates mapped? Which rules are implemented in the various systems?
- *Allowed fillers*: Is it possible to constrain the slot fillers permitted?
- *Parameter weighting of measures, parameter weighting of slot significance*: Is it possible to influence the slot significance?
- *Evaluation metrics*: Which scoring metrics are implemented (e.g., F-measure, precision, SER)
- *Interface*: Which interface is available (e.g., GUI)?
- *Export*: Which kind of export is available? In which format is the output stored (e.g., HTML, TXT)?

5. Lessons Learned & Future Work

The work reported in this paper aims to provide a methodology for evaluating information extraction results that is flexible and effective and enables a fair and reliable comparison of rules and machine-learned models used in IE systems. In order to achieve this goal, first a methodology based on long-term practical experience in IE was devised, and second a tool that incorporates the standardized IE methodology was developed.

The overall goal – *creating a domain-independent methodology for measuring the performance of an IE system* – is very challenging. It involves (i) the study of relevant literature in the area of scoring metrics, (ii) a comprehensive analysis of state-of-the-art evaluation tools, and (iii) an anlaysis of the requirements based on past experience and current IE projects. The work comprises the adaptation and extension of existing scoring metrics, a reimplementation of published (cf. general greedy algorithm) and an integration of existing algorithms.

Developing such a methodology and a tool that builds upon it requires continuous revision and refinement, which forms an inherent part of our future research work.

Some issues that arose while standardizing the IE evaluation process and that constitute, in part, future work are:

Determining boundaries. The rules for determining the boundaries of hypothesis and reference templates must be refined. In the current EVALIEX version the user can either select a token or a character as the basis for the comparison. In the mapping both variants are often required, for instance, when evaluating the extracted names of an accommodation. The following example illustrates the current shortcoming:

Hypothesis	Reference	Parameter	Result
Beach-Hotel	*Hotel*	base: token	not correct
Beach-Hotel	*Hotel*	base: ***char***, m = 4	***correct***
Hotel Hofer	*Hofer*	base: ***token***, m = 1	***correct***

Versioning. The current version of EVALIEX provides versioning only in a very simple form. Different versions of

reference-hypothesis comparisons with different parameterizations can be displayed in the visualization tab of the tool. More comprehensive version tracking is needed to provide, for example, (i) deeper insight into errors (e.g., to identify constantly missing slot values and (ii) more statistics on errors or improvements achieved.

Visualization of different versions. EVALIEX provides only one chart for visualizing the scored metrics. For an upcoming version, more detailed forms of visualization are planned, for instance, visualization of various tests and evaluations over a longer period, which represents the effectiveness of changes in manually coded rules.

6. References

Appelt, D., Israel, D. (1999). Introduction to Information Extraction Technology. A Tutorial Prepared for IJCAI-99, SRI International.

Camacho, H., Salhi. A. (2006). A String Metric Based on a one-to-one Greedy Matching Algorithm, In Research in Computing Science, volume 19, pp. 171–182.

Chinor, N, Dungca, G. (1995). Four Scorers and Seven Years Ago: The Scoring Method for MUC-6, In Proceedings of MUC-6 Conference, Columbia, MD, pp. 33–38.

Cole, R., Mariani, J., Uszkoreit, H., Varile, G.B., Zaenen, A., Zampolli, A., Zue, V. (1998). Survey of the State-of-the-Art in Human Language Technology, Cambridge University Press and Giardini.

Cohen, W.W., Ravikumar, P., Fienberg, S. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of the IJCAI Workshop on Information Integration on the Web, pp. 73–78.

Cunningham, H (1997). Information Extraction a User Guide. Technical report, Research Memo CS-97-02. Institute for Language, Speech and Hearing (ILASH), University of Sheffield, UK.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M., Saggion, H., Petrak, J., Li, Y., Peters, W. (2010). Developing Language Processing Components with GATE Version 6 (a User Guide), Technical Report, The University of Sheffield University.

De Sitter, A., Calders, T., Daelemans, W. (2004). A Formal Framework for Evaluation of Information Extraction, Technical report, No. 2004-4, In University of Antwerp, Dept. of Mathematics and Computer Science, 2004.

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).

Demetriou, G., Gaizauskas, R., Sun, H., Roberts, A. (2008). ANNALIST: ANNotation ALIgnment and Scoring Tool. In Proceedings of of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco.

Douthat, A.L. (1998). The Message Understanding Conference Scoring Software User's Manual, In the 7th Message Understanding Conference.

Douthat, A.L. (1999). HUB-4 "Templette" Task Scoring Procedure, Version 0.3, In Hub-4 Broadcast News February 19, 1999. Science Applications International Corporation.

Freitag, D. (1998). Machine Learning for Information Extraction in Informal Domains. PhD thesis, Carnegie Mellon University.

Kim, J.H., Woodland, P. (2003). A Combined Punctuation Generation and Speech Recognition System and Its Performance Enhancement Using Prosody, In: Speech Communication, vol. 41, no. 4, pp. 563–577.

Lavelli, A., Califf, M.E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L. (2004). IE Evaluation: Criticisms and Recommendations. In AAAI-2004 Workshop on Adaptive Text Extraction and Mining.

Lavelli, A., Califf, M.E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., Ireson, N. (2008). Evaluation of Machine Learning-based Information Extraction Algorithms: Criticisms and Recommendations. In Language Resources and Evaluation, volume 4.

Linsmayr, E. (2010). GeMap – a Generic Mapper for XML Instances, Technical Report, Institute of Application Oriented Knowledge Processing (FAW), Johannes Kepler University Linz.

Makhoul, J., Kubala, F., Schwartz, R., Weischedel. R. (1999). Performance Measures for Information Extraction, In Proceedings of the DARPA Workshop on Broadcast News Understanding, pp. 37-40.

Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I., Spyropoulos, C.D. (2002). Ellagon: A New Text Engineering Platform. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC), Las Palmas, Canary Islands, pp. 72-78.

Pröll B., Feilmayr, C., Buttinger, C,. Guttenbrunner, M., Parzer, S. (2009). Web Information Extraction, in: "Hagenberg Research" Ed. Bruno Buchberger et al. Chapter 7: "Information and Semantics in Databases and on the Web".