ETER: a new metric for the evaluation of hierarchical named entity recognition

Mohamed Ameur Ben Jannet[α,β,γ,δ], **Martine Adda-Decker**[δ,α], **Olivier Galibert**[γ], **Juliette Kahn**[γ], **Sophie Rosset**[α]

[α]LIMSI–CNRS UPR 3251 [β]Université Paris-Sud [γ]LNE

[δ]LPP–CNRS UMR 7018, Université Sorbonne Nouvelle

{first.last}@limsi.fr

{first.last}@lne.fr

Abstract

This paper addresses the question of hierarchical named entity evaluation. In particular, we focus on metrics to deal with complex named entity structures as those introduced within the QUAERO project. The intended goal is to propose a smart way of evaluating partially correctly detected complex entities, beyond the scope of traditional metrics. None of the existing metrics are fully adequate to evaluate the proposed QUAERO task involving entity detection, classification and decomposition.

We are discussing the strong and weak points of the existing metrics. We then introduce a new metric, the Entity Tree Error Rate (ETER), to evaluate hierarchical and structured named entity detection, classification and decomposition. The ETER metric builds upon the commonly accepted SER metric, but it takes the complex entity structure into account by measuring errors not only at the slot (or complex entity) level but also at a basic (atomic) entity level. We are comparing our new metric to the standard one using first some examples and then a set of real data selected from the ETAPE evaluation results.

Keywords: hierarchical named entity; metrics; evaluation

1. Introduction

The successful development of information extraction technologies over time has been accompanied by an increase in the complexity of the task, including attempts to structure the information to be extracted. At the same time, evaluation metrics have been defined to handle this increasing complexity. Within the French OSEO QUAERO program, a new definition of structured and hierarchical named entities was proposed (Grouin et al., 2011) and subsequently used within the French ANR ETAPE project for its evaluation campaign on the Named Entity task (Galibert et al., 2014). In this work, we show that existing metrics are not well suited for such a task. We propose a new metric to take into account the potentially complex structure of named entities. This new metric is also able to take into account different application cases.

In the next section we present the different metrics proposed since the first named entity evaluation campaign following the different named entity definition. Then, the structured named entity task is presented in section 3. followed by the description of our proposed ETER metric (section 4.). The application of this new metric is presented in section 5. together with an detailed analysis. Finally, we conclude this paper in section 6..

2. Evolution of NE definition and metrics

The Named Entity Extraction tasks as introduced within the 6th MUC conference (Grishman and Sundheim, 1996; SAIC, 1998), consisted in detecting and annotating proper names, numeric values and time expressions in text documents. The very encouraging results achieved during these evaluations attracted the interest of several technological areas such as information extraction, understanding, indexation... As a consequence, the exact definition of named entity evolved in complexity from one conference to the next. Nowadays named entities (NE) cover a large panel of different expressions, and named entity recognition (NER) systems try to extract more and more predefined entity classes also including information about their mentions and relations. This entails an overall NER task complexification. Figures 1 and 2 illustrate an example of a hierarchically structured entity as defined within the QUAERO program. In parallel to that growing complexity, NER performance metrics had to evolve accordingly.

Precision (P) and recall (R) are the usual metrics used to measure the performance of information extraction systems. Precision gives the ratio of correct predictions within all the predictions, while Recall gives the ratio of correct predictions within all the entities to be found. Hence, precision measures the correctness of the system, while recall measures its coverage of the input data. In order to obtain a single value the F-measure has been defined as the harmonic mean between P and R and has been used to rank participants during many evaluation campaigns. However, the F-measure shows some limitations. First, as demonstrated in (Makhoul et al., 1999), when P and R are fused with an harmonic mean, deletion and insertion errors are lowered in importance in comparison to substitutions.

More importantly, the correctness evaluation for precision and recall evaluations are binary: either an hypothesis is correct or it isn't. But breaking down entity types into several sub-types and the use of structured entities give rise to different kinds of substitution errors (type substitutions, sub-type substitutions and boundary substitutions) and one may want to differently weight each kind of error case.

To overcome the pointed out F-measure limitations, alternative error metrics have been proposed to measure system performances. Error-based metrics try to estimate the cost of an error with respect to an end-user or a larger application context. A decrease in error rate then corresponds to

an overall performance improvement.
Initially, the ERR error metric (Error per response) has been introduced during MUC-5 (Chinchor and Sundheim, 1993). It is defined as:

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

where:

- S is the number of substitutions
- I is the number of insertions (false alarms)
- D is the number of deletions (misses)
- C is the number of corrects items

The main problem with ERR is that putting I in the denominator makes it vary with the system hypothesis, thus making systems comparisons harder. (Makhoul et al., 1999) proposed the now well-known Slot Error Rate:

$$SER = \frac{S + D + I}{C + D + S} = \frac{S + D + I}{R}$$

where R is the number of entities to detect in the reference. The SER works quite well for the simple detection and classification tasks with the possibility of varying the substitution cost of an error depending on criteria considered relevant for the application. Some tasks however go further and require classifying entity mentions and/or relations. In this case, the SER metric is not usable anymore. That was the case in the ACE 2004 (Automatic Content Extraction) evaluation of NE extraction which introduced a new metric called the EDT value (Doddington et al., 2004). This metric sums values for system-detected entities with the values being defined beforehand:

$$EDT\_value_{sys} = \sum_i value\_of\_sys\_entity_i$$

The definition of the values is outside the scope of this paper but takes into account all the required information. That metric evolved into LEDR (ACE, 2008) for ACE-2008 where the values were normalized by the sum of the values of the reference data:

$$LEDR\_value_{sys} = \frac{\sum_i value\_of\_sys\_token_i}{\sum_j value\_of\_ref\_token_j}$$

Within the QUAERO project a variant of NE annotation methodology has been experimented with (Grouin et al., 2011). In that framework, entities are represented by a two-level structure. In a first step, the whole entity is marked and typed. In a second step, the word span is broken down into contiguous components with a specific type each. The components allow to refine categorizations according to applicative needs. Hence, an entity is not limited to a (tag, span) pair but may grow to a complex structure that can be represented as a tree, with the main entity type at the root and its components as leaves. Figure 1 presents such a tree. In the QUAERO context, we experimented with a semi-direct implementation of SER evaluation (Galibert et al., 2011), considering root types and components as independent entities. However, this approach gave unsatisfying results.

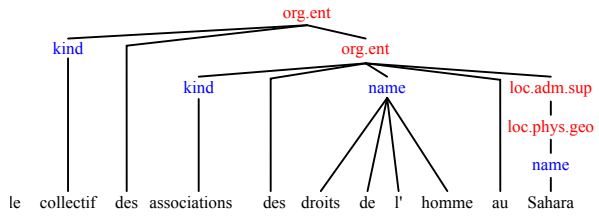

Figure 1: Example of complex tree entities with roots (types/sub-types) in red and leaves (components) in blue.

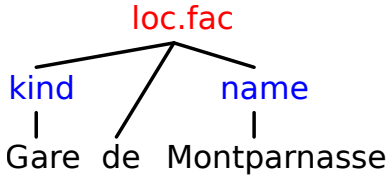

Figure 2: Example of simple tree with roots (types/sub-types) in red and leaves (components) in blue.

One may think that our evaluation task is close to evaluating parsers such as in a tree-bank task and that we may apply similar evaluation schemes. Though, we consider that these two evaluation cases are different for two main reasons. Firstly, in the tree bank evaluation the response space is not unique and the first step consists in generating all possible responses (Carroll et al., 1998), while in the case of named entity extraction a unique annotation model is imposed. Secondly, we consider that an error-based measure is more suited to the NE entity task whereas for parsers evaluation metrics are based on precision and recall.
In the next section, we present an alternative way of evaluating NE hypotheses with the ambition of better reflecting the quality of the systems.

3. QUAERO Named Entities

The recently introduced QUAERO NE annotation guidelines (Grouin et al., 2011; Rosset et al., 2011) propose a new kind of named entities which are both hierarchical and compositional. These two features provide the advantage of a generic annotation easily adaptable to a new application domain with minimal changes. At the same time, the resulting complex structures make the task evaluation harder. Two kind of labels are used:

- NE-type labels: used to classify detected named entity, they constitute the first level of annotation and provide information about type and sub-type (nodes in red in the different figures). The taxonomy includes 7 types and 32 sub-types. The seven top-level NE-types are persons, locations, organizations, products, functions, amounts and temporal expressions. The NE-sub-types are used to more precisely specify the NE-type. For example, the *time* NE-type denotes a temporal expression. With an additional sub-type *time.date* it indicates more specifically a day or more, and *time.date.abs* indicates an absolute date.

- Component labels: used to annotate semantically in-

teresting elements inside a detected named entity. They are never a full named entity of their own, but only part of one. They constitute a second level of annotation (nodes in blue in the different figures). There are NE-type specific component labels and component labels which can be found within any NE-type.

Entity annotation thus makes use of two level annotations, a top level with NE-type annotations (in red) on the full words spans of the named entity, and a bottom level which may decompose the entity into sub-spans corresponding to components. A small number of words may not be annotated at the bottom level, mainly determinants. Figures 1 and 2 show examples of QUAERO annotations.

4. Proposed error measure: ETER

As discussed in section 2., F-measure and MUC-type SER metrics are not best suited for evaluating a named entity detection task where complex, agglutinated named entities are frequent. A better suited metric should be able to take into account that:

- entities, with one main type for the full span, tend to be structured, subdivided into one or more components and/or smaller entities;
- entity types are hierarchical with types and sub-types. NE-type comparison must take that structure into account;
- the same entity may be given multiple NE-types in case of metonymy.

The metric should be easily adaptable to account for more or less levels of complexity as a function of the evaluation setup and/or applicative requirements.

4.1. General structure

The proposed metric builds upon the standard SER methodology by adding options to take structuring into account. Three steps are needed to compute the metric:

1. match reference and hypothesis entity trees;
2. for each matched tree pair, associate subtrees/components between reference and hypothesis;
3. compute error counts for all matched tree, subtree and component pair.

The matching or association steps are commonly called the *alignment*. The final error rate is computed as the total number of errors divided by the number of slots, e.g. the number of entities. The slots may represent either types, subtypes or components, depending on the chosen option.

4.2. Alignment

The first step of any comparison-based metric consists in associating the slots of the reference with those of the hypothesis. The standard SER metric evaluates associated tags independently of their types, giving rise to alignments such as those shown in Figure 3. Such an alignment, while correct at first sight, lacks in precision. Associating the *name* component labels together will make the evaluation consider that the *func.ind* entity is fully correct, although the types of the components are not fully specified (missing detection of 2 location entities of location NE-type and administration/nation *loc.adm.nat* subtypes . To reveal such errors (missing information), we need to apply a two-level alignment, where the full (first level) entities are aligned together, and then within aligned entity pairs the components are aligned together.
Such an alignment is given in Figure 4, showing that the two *loc.adm.nat* components are considered as missing. In addition, the two *name* tags are now considered as substitutions of these components, which is the expected result when taking structure into account.
When aligning components of a matched tree pair, different matching options need to be explored. The chosen alignment is the one with the best (lowest) score. The computation of the score is described in the next subsection.

4.3. Entity Tree Error Rate

The proposed *ETER* metric or *Entity Tree Error Rate* metric follows the usual SER in computing a total number of errors and dividing it by the number of slots, here the number of entities. The total set of errors can be divided into three parts: insertion, deletion and substitution errors for paired entities. Usually, insertion and deletion costs are fixed to 1. The ETER metric can then be written as follows:

$$ETER = \frac{I + D + \sum_{(e_r,e_h)} E(e_r, e_h)}{N_E}$$

with:

- I: number of Insertions (spurious entities) - entities in the hypothesis that are not associated with an entity of the reference;
- D: number of Deletions (missing entities) - entities in the reference that are not associated with an entity of the hypothesis;
- (e_r, e_h): aligned pair of reference and hypothesis entities;
- $E(r, h)$: error computed on the reference/hypothesis entitity pair (which can be zero);
- N_E: number of entities in the reference.

The metric then relies on the per-association error computation. We split the errors into two parts, one linked to the main entity determination (e.g. root), and one to the components:

$$E(r, h) = (1 - \alpha)Er(r, h) + \alpha Ec(r, h), \alpha \in [0..1]$$

where $Er(r, h)$ and $Ec(r, h)$ correspond to the number of errors in root nodes and in components respectively; α is a parameter allowing to select the importance between entity classification and entity decomposition. By default we consider that $\alpha = 0.5$ is appropriate for the global named entity detection, classification, decomposition task It gives an equal weight to entity classification and entity decomposition irrespective of the number of type and component slots. Nevertheless, evaluators may modify the α weight according to their specific needs. This weight option is helpful to enable the checking of system weaknesses (e.g. by switching from entity classification to decomposition). It can also be useful if one of the two sub-tasks (entity classification vs decomposition) is more important than the other one in

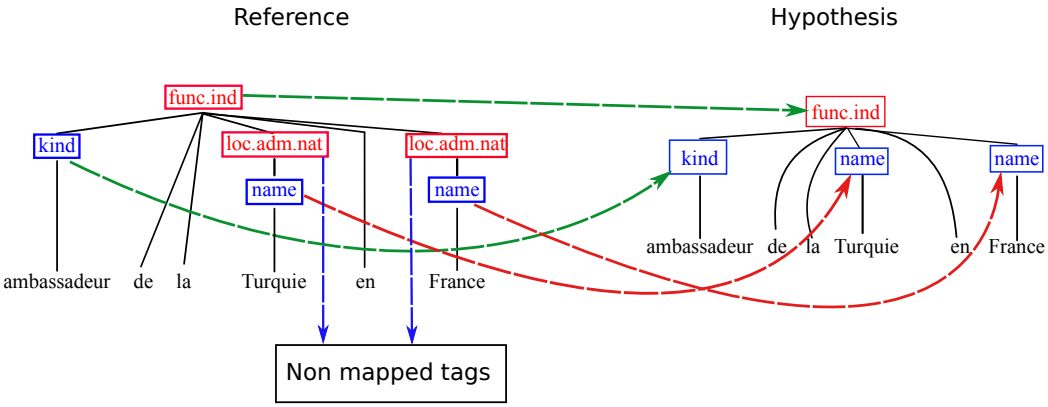

Figure 3: Example of tag based alignment.

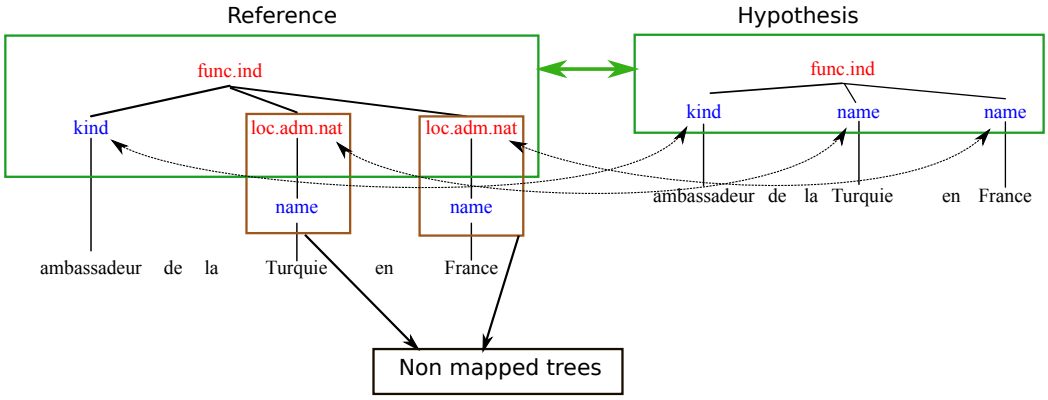

Figure 4: Example of a two-level NE tree alignment.

a given applicative context and we want to assign it a higher weight.

The root, or main entity, error score reports whether the type or types (in case of metonymy) are correct, and whether the span (boundaries) is correct. A span error costs 0.25 point, a type error up to 0.5 point. This gives a maximum error for the root entity of 0.75, acknowledging that detecting the presence of an entity is always better than completely missing it, where the cost would be 1.

The NE-type computation is a little complex due to the existence of metonymy. We first define an elementary type comparison error:

$$Et_1(t_1, t_2) = \begin{cases} 0.5 & \text{NE} - \text{type error} \\ 0.25 & \text{NE} - \text{subtype error} \\ 0 & \text{if the types are fully identical} \end{cases}$$

In absence of metonymy (both in reference and hypothesis), a basic comparison can be used. We can see that a complete misclassification gives a maximum error of 0.5, while detecting a correct NE-type but failing at a lower subtype level will be only half that.

If only one side has a metonymy, we use the mean between a class error (0.5) and the best elementary comparison, choosing between the two possible pairs. If both have metonymy, we use the mean of both basic comparisons, matching the type the best possible way:

$$\begin{aligned} Et(\{r_1\}, \{h_1\}) &= Et_1(r_1, h_1) \\ Et(\{r_1, r_2\}, \{h_1\}) &= \min(\frac{0.5 + Et_1(r_1, h_1)}{2}, \\ &\quad \frac{0.5 + Et_1(r_2, h_1)}{2}) \\ Et(\{r_1\}, \{h_1, h_2\}) &= \min(\frac{0.5 + Et_1(r_1, h_1)}{2}, \\ &\quad \frac{0.5 + Et_1(r_1, h_2)}{2}) \\ Et(\{r_1, r_2\}, \{h_1, h_2\}) &= \min(\frac{Et_1(r_1, h_1) + Et_1(r_2, h_2)}{2}, \\ &\quad \frac{Et_1(r_1, h_2) + Et_1(r_2, h_1)}{2}) \end{aligned}$$

The component, or decomposition error score, is a local SER on the components themselves of the second level alignment:

$$Ec(r, h) = \frac{Ic(r, h) + Dc(r, h) + \sum_{(c_r, c_h)} Ec_1(c_r, c_h)}{Nc(r)}$$

where:

- $Ic(r, h)$: number of inserted components;
- $Dc(r, h)$: number of missed components;

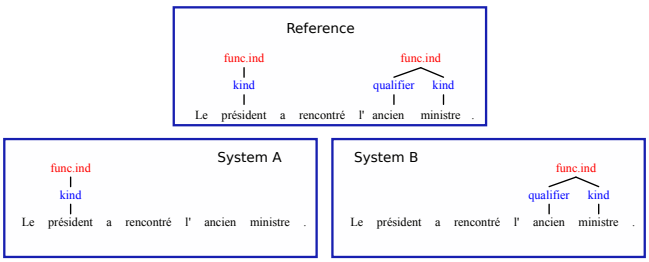

Figure 5: Two system outputs for the same reference (*The president meets the former minister*), with different SER (60% and 40%) and identical ETER (50%).

- (c_r, c_h): associated pair of reference and hypothesis components;
- $Ec_1(r, h)$: error computed on the reference/hypothesis component association (which can be zero);
- $Nc(r)$: number of components in the reference entity;

Finally the component error is computed as:

$$Ec_1(c_r, c_h) = \begin{cases} 0.5 & \text{if the components are different} \\ 0 & \text{otherwise} \end{cases}$$
$$\begin{cases} 0.25 & \text{if the boundaries are different} \\ 0 & \text{otherwise} \end{cases}$$

This is identical to the root types, without considering the subtypes or the metonymy concepts.
The proposed method allows us to compute a root entity error rate and a component decomposition error rate which may be linearly combined to get the final global score.

5. Comparative analysis of performance mesures

In this section, we illustrate the proposed metric using some examples before presenting results using the French ETAPE data.
In the example of Figure 5, the reference has two entities, the first one, *président*, involving one *kind* component and the second one (*ancien ministre*) involving two components (*qualifier* and *kind*). We want to compare two NER systems called A and B, where A annotates correctly the first entity but deletes the second one, while system B deletes the first entity and correctly annotates the second one.
Applying the basic SER metric to our tree-based entities, system A has two correct slots and three deleted ones, while system B has three correct slots and two deleted ones. This gives an error rate of $\frac{3}{5} = 60\%$ for system A and $\frac{2}{5} = 40\%$ for system B. However, from a task point of view, both systems missed one entity, and when focusing only on named entity detection, there may be no reason to consider one entity more important than the other one. ETER, evaluating at the entity level, gives one deletion and one correct slot for both systems, and an identical error rate of $\frac{1}{2} = 50\%$. That is in line with considering that each system did half the job.
A second example stems from the previous example in figures 3 and 4 involving nested entities. In the reference annotation, *ambassadeur de la Turquie en France* has type *func.ind* (function), and that entity is decomposed into three components, a *kind* (*ambassadeur*) and two locations (*Turquie* and *France*). Recursively, *Turquie* and *France* are two entities, each one having a *name* as component. The hypothesis has one entity with three components, the *kind* and two *name*. Alignment results for both the SER (shown in figure 3) and the ETER (shown in Figure 4) metrics are discussed in section 4.1.
Using these alignements we can easily compute the results for the two metrics. In the SER point of view, there are four correct slots and two deletions, giving a final error rate of $\frac{2}{6} = 33\%$. The ETER case is slighly more complex, with two deleted entities and one with a small decomposition error of $E_T = 0.16$. This gives a global error rate of $\frac{2+0.16}{3} = 72\%$.
From our point of view, losing two out of three entities justifies an error rate of at least two-thirds, or 67%. And given that the remaining entity is not even perfect, climbing to 72% makes sense. The SER score on the other hand is abnormally low. We can see that ETER takes into account the role each label plays in the whole annotation interpretation, and respects the dependency relationship between components and root entity.
To illustrate the use of ETER on more realistic data, in the next section, we compare the error measures provided by SER and ETER for the named entity detection task of the ETAPE evaluation. The task consisted in extracting, categorizing and decomposing a large number of named entities in accordance with the guidelines defined during the QUAERO project.

5.1. The ETAPE test corpus

The ETAPE test-corpus (see (Galibert et al., 2014) for a complete description) consists of 8h20 of radio and TV program including planned and spontaneous speech. Table 1 briefly describes the data available. All the shows were manually transcribed and annotated in named entities according to the QUAERO guidelines (Rosset et al., 2011).

	ETAPE test	%
Number of entities	5954	40.8
Numr of components	8627	59.2
Total	14581	

Table 1: Description of the test corpus of the ETAPE evaluation.

5.2. Interpretation of the ETAPE evaluation results

Table 2 shows the results in terms of SER and ETER obtained by ten systems having participate in the ETAPE evaluation, on the test data.
One can notice that error rates are different depending on the metric used and, more important, that the ranking of the systems changes. That shows that the two metrics have different behaviors on real data.
We selected among the participants of the ETAPE evaluation three NER systems which used different approaches and ended up showing different behaviors:

System	SER		ETER	
	Score	Rank	Score	Rank
NER-1	35.4	1	34.4	1
NER-2	38.0	4	35.5	2
NER-3	51.4	8	50.0	7
NER-4	36.4	2	40.4	5
NER-5	37.3	3	38.7	4
NER-6	39.2	5	37.9	3
NER-7	50.0	7	52.6	9
NER-8	56.4	9	50.1	8
NER-9	44.6	6	42.8	6
NER-10	85.6	10	81.4	10

Table 2: Comparison between SER and ETER ($\alpha = 0.5$) on the ETAPE test data.

	D	I	S	SER	ETER
NER-4 types	1925	155	541	39,5%	
NER-4 components	2304	310	587	33.7%	
NER-4 global task	4229	465	1128	36,4%	40%
NER-5 types	1476	543	888	41%	
NER-5 components	2092	525	589	33.7%	
NER-5 global task	3568	1068	1477	37.3%	38.6%
NER-8 types	1511	290	818	37%	
NER-8 components	4295	266	2908	69.7%	
NER-8 global task	5806	556	3726	56.4%	50%

Table 3: Comparison between SER and ETER (α= 0.5), Results for ETAPE test.

- NER-4: this system, described in (Raymond, 2013), ignores the structure and considers that all labels are independent.
- NER-5: this system, described in (Dinarelli and Rosset, 2012), builds entities trees in two steps, starting with a CRF model to detect and annotate components, followed by a PCFG-based parser to rebuild the whole trees.
- NER-8: this system, described in (Hatmi et al., 2013), doesn't try to detect components, every detected entity has only one *name* component embedded.

Table 3 shows the results obtained by these three specific systems. We distinguish in this table four different error measures: SER for type slots, SER for component slots, SER for the global task and ETER with $\alpha = 0.5$ for the global task. As we can see, for the global task the performance measured by both metrics (SER and ETER) is different.

For the NER-8 system, ETER shows a lower error rate than SER, while for the other two systems (NER-5, NER-4) ETER is higher than SER. If we look at the performance of the three NER systems on types and components separately, we notice that NER-8 has a lower SER on types than on components, and that in the global task it obtained an ETER score lower than the SER one. On the other hand, we can see that NER-4 and NER-5 systems have a lower error rate on components than on types, and that both systems obtained on the global task an ETER score bigger than the SER one.

We have here a direct impact of the way SER is computed. Going back to the formula:

$$SER = \frac{total\ number\ of\ slot\ errors}{total\ number\ of\ slots\ in\ reference}$$
$$= \frac{\overbrace{\sum_{i=0}^{NbT} type\ error}^{Classification\ errors} + \overbrace{\sum_{j=0}^{NbC} component\ error}^{Decomposition\ errors}}{total\ number\ of\ slots\ in\ reference}$$

where:

- NbT: Number of type slots in hypothesis.
- NbC: Number of components slots in hypothesis.

And $NbC \approx 1.5 \times NbT$ for real data. As a consequence, the use of a slot-based metric such as the SER tends to give more importance to the components because of their higher count. Systems which perform better in classification than in decomposition end up penalized. In the case of ETER the comparative weight of these two aspects is set explicitely through the α parameter, with the default value of 0.5 ensuring a equal weighting.

5.3. Alpha parameter and system performance comparison

The default α value of 0.5 is the most natural to evaluate the global task, but changing its value allows to try to understand the strengths and weaknesses of the systems. Figure 6 illustrates the performance measurement variation of the three NER systems in function of α.

- $\alpha = 0$: evaluation of performances in entity detection and classification, decomposition is not taked into account;
- $\alpha = 0.5$: the most appropriate for the overall task, it give an equal weight to entity classification and entity decomposition;
- $\alpha = 1$: evaluation of performances in entity detection and decomposition, classification is not taked into account.

As we can see figure 6, for $\alpha = 0$ the best score is reached by NER-8. It means that it has the best performance in entity detection and classification. However, its performances decrease rapidly when α increases, which shows that this NER system has a bad performance in decomposition, ending up with a very high ETER when $\alpha = 1$. This confirms the system description indicating that decomposition was in practice not taken into account.

We can also notice that NER-5 and NER-4 obtain almost the same ETER when $\alpha = 0$, which means that they have

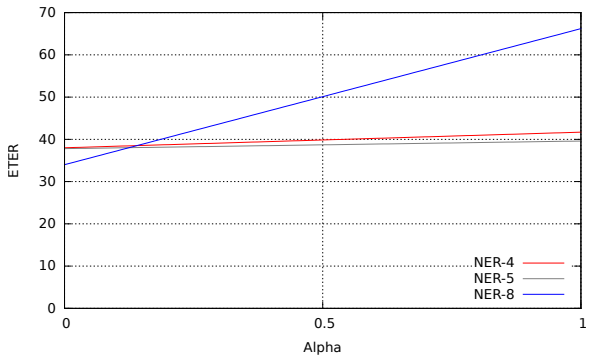

Figure 6: Variation of ETER depending on the component-root entity importance ratio α.

	Type substitutions	Sub-type substitutions	Boundaries substitutions
NER-1	371	137	341
NER-2	307	119	485
NER-3	167	56	260

Table 4: Distribution of types, sub-types and boundaries substitution errors when using ETER.

similar performance in entity detection and classification. However, when $\alpha = 1$ NER-5 obtains a lower ETER pointing that this NER system performs better in entity detection and decomposition. But the SER results shown in table 3 suggest that NER-4 and NER-5 have exactly the same performances in decomposition (SER = 33.7%). We explain the ETER vs. SER difference as reflecting the fact that the NER-4 system did not take the structure itself into account and handled all annotations independently.

We can also notice that NER-4 and NER-5 have obtained a different SER on types only but show the same performance in entity detection and classification (ETER with $\alpha = 0$). In order to explain this observation, we need to have a look at the errors of each system. As we can see in table 3 NER-5 makes less deletions than NER-4 but more substitutions and insertions.

Substitutions errors are processed differently by each metric, due to the different weighting of the error. In the SER case, classification errors cost 0.5 and boundaries errors cost 0.5. In the ETER case a top-level type error costs 0.5, a sub-type error 0.25 and a boundary error 0.25. Table 4 shows that the NER-5 substitutions tend to be of the less costly kinds (sub-types, boudaries), making them less expensive than the deletions of NER-4. The penalty choices are a matter of taste, but in any case it seems to make sense that detecting the presence of an entity, even if a little wrong on boundaries or classification, is better than not detecting it at all.

All those results confirm that our proposed metric is a better fit to the QUAERO named entity structure detection and classification task than the currently used SER metric. It gives the possibility to take into account the structure of the entities and the annotation scheme and enable a better analysis of NER system behaviours.

6. Conclusion

This paper highlights the problems that have arisen during the named entity evaluation campaign as a consequence of a new complex named entity annotation scheme during the QUAERO project. The proposed annotation allows not only for named entity classification but also for its decomposition (or structuring). Strong and weak points of previously used metrics were pointed out and we proposed the new ETER metric as an alternative. This metric builds upon the common SER metric but it aims at taking the complex structure into account by measuring errors at the entity level instead of the slot level. Within the adopted methodology, a root entity error rate and a component decomposition error rate are linearly combined for the final score.

The relative importance of entity classification and entity decomposition errors can be selected by the evaluators according to their evaluation context or the expected use of the system via an α parameter.

Examples and tests on real data show that the slot-based metric is not well suited to evaluate such a complex named entity task and that the proposed ETER metric allows for a better interpretation of the NER performance when dealing with complex tree entities.

7. Acknowledgements

This work was partially realized within the framework of the project VERA (adVanced ERror Analysis for speech recognition) funded by ANR Blanc - ANR 12 BS02 006 04 and the CIFRE grant No 2012/0771. This work was also supported by the French Investissements d'Avenir - Labex EFL program (ANR-10-LABX-0083).

8. References

ACE. (2008). Evaluation plan (ace08). pages 1–3.

Carroll, J., Briscoe, T., and Sanfilippo, A. (1998). Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pages 447–454.

Chinchor, N. and Sundheim, B. (1993). Muc-5 evaluation metrics. In *MUC*, pages 69–78.

Dinarelli, M. and Rosset, S. (2012). Tree representations in probabilistic models for extended named entities detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–184. Association for Computational Linguistics.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*. Citeseer.

Galibert, O., Rosset, S., Grouin, C., Zweigenbaum, P., and Quintard, L. (2011). Structured and extended named entity evaluation in automatic speech transcriptions. In *Proc of IJCNLP*, Chiang Mai, Thailand.

Galibert, O., Leixa, J., Adda, G., Choukri, K., and Gravier, G. (2014). The etape speech processing evaluation. In *Proc of LREC*, Reykjavik, Iceland. ELRA.

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference - 6: A brief history. In *Proc. of COLING*, pages 466–471.

Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O., and Quintard, L. (2011). Proposal for an Extension of Traditional Named Entities: From Guidelines to Evaluation, an Overview. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 92–100, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hatmi, M., Jacquin, C., Morin, E., Meigner, S., et al. (2013). Named entity recognition in speech transcripts following an extended taxonomy. In *Workshop on Speech, Language and Audio in Multimedia (SLAM)*.

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). Performance measures for information extraction. In *Proc. of DARPA Broadcast News Workshop*, pages 249–252.

Raymond, C. (2013). Robust tree-structured named entities recognition from speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8475–8479. IEEE.

Rosset, S., Grouin, C., and Zweigenbaum, P. (2011). Entités Nommées Structurées: guide d'annotation Quaero. LIMSI–CNRS, Orsay, France.

SAIC. (1998). Proceedings of the seventh message understanding conference (MUC-7).