Tree-Structured Named Entity Recognition on OCR Data: Analysis, Processing and Results

Marco Dinarelli, Sophie Rosset

LIMSI-CNRS
B.P. 133, 91403 Orsay Cedex, France
marcod@limsi.it, rosset@limsi.fr

Abstract

In this paper we deal with named entity detection on data acquired via OCR process on documents dating from 1890. The resulting corpus is very noisy. We perform an analysis to find possible strategies to overcome errors introduced by the OCR process. We propose a preprocessing procedure in three steps to clean data and correct, at least in part, OCR mistakes. The task is made even harder by the complex tree-structure of named entities annotated on data, we solve this problem however by adopting an effective named entity detection system we proposed in previous work. We evaluate our procedure for preprocessing OCR-ized data in two ways: in terms of perplexity and OOV rate of a language model on development and evaluation data, and in terms of the performance of the named entity detection system on the preprocessed data. The preprocessing procedure results to be effective, allowing to improve by a large margin the system we proposed for the official evaluation campaign on Old Press, and allowing to outperform also the best performing system of the evaluation campaign.

Keywords: Named Entity detection, old newspapers, automatic OCR correction

1. Introduction

In the last few years a lot of data have been acquired with Optical Character Recognition (OCR) techniques with the aim of giving easier access to historical documents via automatic content extraction systems. Unfortunately, OCR-acquired data contain many mistakes due to the OCR technology limitations. This makes development of typical content extraction systems, e.g. named entities or relations between entities, a very challenging task. Models more or less robust to noisy data are already available, nevertheless the level of noise in OCR-ized data is much higher than data used typically for these tasks. This is indeed reflected by systems performances (Claire Grover and Ball, 2008; Miller et al., 2000; Byrne, 2007).

In order to deal with such noisy data, the best solution is to analyse and pre-process them, so that to detect mistakes introduced by OCR process and to find a strategy to correct or, at least, overcome the errors.

While there is a vast literature on content extraction tasks like Named Entity Recognition (NER) (e.g. (Grishman and Sundheim, 1996), (Sekine and Nobata, 2004)), there is much less work on OCR-ized data in general, and on content extraction on OCR acquired data in particular.

In this paper we present an analysis of a corpus acquired via OCR process on French historical newspapers. Our work has been done on the data provided for the 2011 Quaero evaluation on Named Entity Recognition in Old Press (Galibert et al., 2012). The annotation made on this corpus followed the Extended Named Entity definition fully described in (Grouin et al., 2011; Rosset et al., 2011), with the difference that, for each entity realizing-surface containing OCR errors, a special attribute containing the correction is added to the annotation. For example:

```
<pers.ind correction="Le Moine">
  <name.last> LE Moibte. </name.last>
</pers.ind>
```

We propose a three steps procedure to correct or overcome, or at least reduce, mistakes introduced by OCR in a preprocessing step. The evaluation of corrections made on input data is overall evaluated with a measure of perplexity of the language model built on training data. Our analysis is similar to that of (Lopresti, 2008), however in such work only an analysis of the effect of OCR mistakes on results was performed, no strategy to overcome or correct mistakes was applied.

After the preprocessing, data are used to train a Named Entity Recognition system. The task is made harder also by the fact that the named entities annotated on OCR-ized data have a tree-structure (Grouin et al., 2011).

We present comparative results of our system different steps of preprocessing, as well as comparative results with the same system trained and evaluated on manually transcribed broadcast data (Dinarelli and Rosset, 2011b), so that to show the gain we achieve with each preprocessing step, and the overall gain. The final results on the evaluation test set outperform the best system of the official evaluation campaign.

The remainder of the paper is organized as follows: in the next section we describe the system used for tree-structured named entity detection, in section 3. we provide an analysis and a description of the corpus which are at the basis of the preprocessing procedure we propose for correcting OCR-ized data, such procedure is described in section 4.. In section 5. we describe and comment the experiments performed in order to evaluate our preprocessing procedure and the named entity detection system on OCR-ized data. We conclude the paper drawing some conclusions in section 6..

2. Tree-Structured Named Entity Recognition System

The system used in this work for tree-structured named entity recognition is described in details in (Dinarelli and Ros-

set, 2011b). In this section, we provide a short description for a matter of completeness and self-containment.
The tree-structured named entities annotated on data used in this work have been defined within the project Quaero[1] and they are described in (Grouin et al., 2011). Two examples of such entities are shown in figure 1 and 2, where words realizing entities have been removed to keep the figure readable.
Given their tree structure, the Named Entity Recognition task presented here cannot be modeled as sequence labelling. Intuitively, entity trees can be constructed adopting solutions for syntactic parsing. However, as mentioned in (Dinarelli and Rosset, 2011b) in relation to broadcast news transcriptions, an approach coming from syntactic parsing to perform named entity annotation in "one-shot" is not robust on OCR data neither. The solution we proposed, is a two-steps approach. The first one is designed to be robust to noisy data and is used to annotate the entity components, i.e. the basic entities annotated directly on words. While the second is used to parse complete entity trees and is based on a relatively simple model. Since OCR data are very noisy, the hardest part of the task is indeed to annotate components on words. On the other hand, since entity trees are relatively simple, at least much simpler than syntactic trees, once entity components have been annotated in a first step, for the second step, a complex model is not required, which would also make the processing slower. Taking all these issues into account, the two steps of our system for tree-structured named entity recognition are performed as follows:

1. A Conditional Random Fields model (Lafferty et al., 2001) annotates the entities components
2. A Probabilistic Context-Free Grammar (PCFG) together with a chart parsing algorithm (Johnson, 1998) builds complete entity trees upon entities components

An example of entity components is shown in figure 2, where components are the leaves of the tree: ***"val object loc.admin.town name time.modifier val kind name"***. The corresponding and complete sentence, where the words realizing entities have been highlighted in bold, is as follow:

> **90 personnes** toujours présentes **à Atambua** c'est là qu'**hier matin** ont été tués **3 employés du haut commissariat des Nations unies** aux réfugiés, le **HCR**[2]

Using the same example in figure 2, a schema of the two steps performed by our system for tree-structured named entity detection is depicted in figure 3.

3. OCR Acquired Data: Description and Analysis

The corpus used in this work is made of documents acquired with Optical Character Recognition (OCR) from a

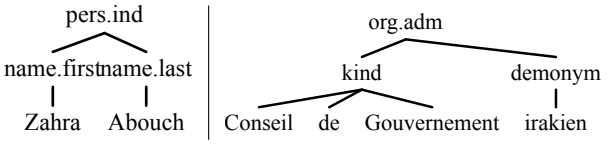


Figure 1: Examples of structured named entities defined within the Quaero project

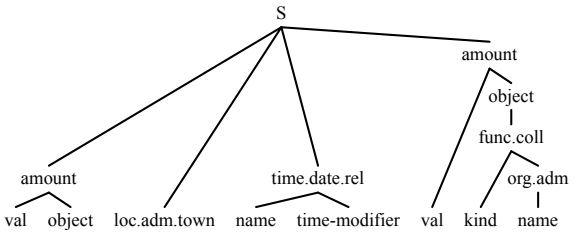


Figure 2: An example of named entity tree corresponding to entities of a whole sentence. Tree leaves, corresponding to sentence words have been removed to keep readability.

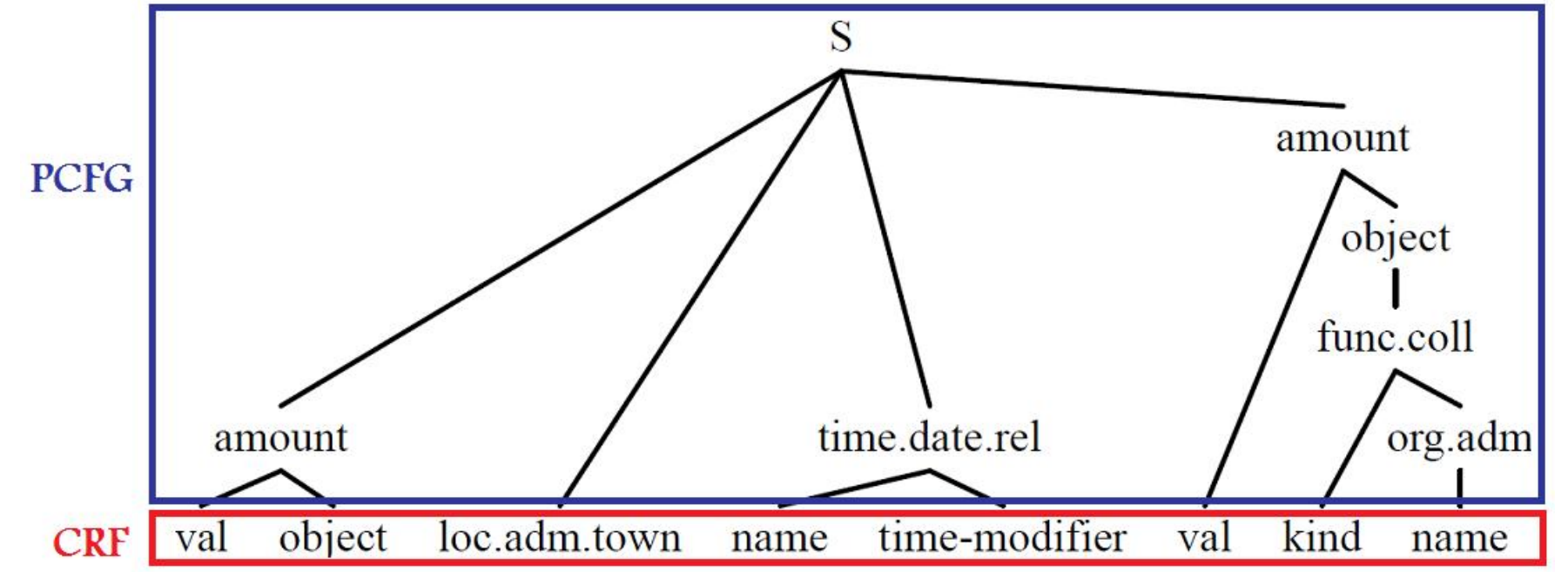
newspaper collection dating from 1890. Due to the nature of the OCR process, data acquired with this technique contain a lot of spurious tokens. Spurious tokens present inserted or substituted characters with respect to the corresponding correctly spelled token, as well as deleted characters.
In the training data provided for the evaluation, only words containing OCR mistakes and realizing entities are annotated with the corrected words, thus only these words can be used for an analysis of OCR mistakes. In order to perform the analysis, we aligned words mistaken by OCR process with their corrected version using an edit distance alignment. Using the alignment, we extracted errors introduced by the OCR process at character level. Such errors can be potentially insertion, deletion and substitution of any character, but they can be associated to two types:

- Word-character errors
- Noise errors

We call word-character errors those involving characters that can be used in words, like alpha-numeric characters, e.g. *Cbambors* instead of *Chambors*. We associate to this category also errors involving punctuation. We call noise errors those involving characters that typically cannot appear in words, e.g. *Cha'mbors* instead of *Chambors*.
Since words realizing entities are a small percentage of the total number of words (4326 corrections for 1,297,742 words in the training data, or 0.3%), but also of the number of words composing entities (roughly 2.5%), we performed also an analysis by hand of the other words. Although in this case we cannot extract errors since we don't have a correction reference, a quick look gave an idea of a possible direction to follow in order to deal with OCR errors. It was evident, indeed, that words truncated at the end of lines by the OCR process, do end with a dash character (-), which is a quite standard way to end a line and start a new one. Another important point is the fact that noise errors never involve punctuation, i.e. we never found something

[1] http://www.quaero.org

[2] 90 people still present in Atambua is where yesterday morning killed three employees of the United Nations High Commissioner for Refugees UNHCR.


PCFG
S
amount
object
func.coll
org.adm
amount
time.date.rel
CRF
val
object
loc.adm.town
name
time-modifier
val
kind
name

tokens are actually just noise, i.e. they are only made of non-alphanumeric symbols.

4.2. Re-tokenization of words

Re-tokenization of words consists in removing noise errors, mentioned in previous section, and separating words from punctuation characters. Note that in removing noise errors there is no risk of removing punctuation since, as mentioned earlier, noise errors never involve punctuation. For the same reason, separating punctuation from words should be beneficial for tokenization, since it creates two new tokens that are actually correct, e.g. like in *le,château* ("the,castle") which becomes *le , château* ("the , castle"), although a comma in that position is likely to be an OCR error.
As an example of noise error correction, from *Cha'mbors* we remove the ' character to have *Chambors*. Beyond the simplicity of the example, it seems reasonable that applying this kind of processing is safe, since non-alphanumeric characters are not statistically likely to appear in tokens made of alphanumeric characters.

4.3. Correction of OCR Errors

Correction of errors introduced by the OCR process is made exploiting the reference correction provided for entity realizing-surface and a manual correction of the most occurring OOV words with respect to a given French dictionary.
Concerning the exploitation of the reference correction, given an annotation of an entity like:

```
<pers.ind correction="Le Moine">
  <name.last> LE Moibte. </name.last>
</pers.ind>
```

and ignoring for the moment the mistake of the case in the article `LE`, which is not really an issue, we `Moibte` with `Moine` at the character level with and edit distance alignment. This allows us to such as *substitution of **n** with **bt***. In order to perform correction, we apply the opposite operation with respect to the error. Using the same example, in order to correct *Moibte*, we apply a substitution of ***bt*** with ***n***. Since when applying corrections to unseen data we don't really know if the word contains an error or not, we take into account also the context of the mistakes, in particular we use the previous and the following characters. This information is used to construct error patterns and to apply the corresponding corrections to words matching the same pattern. Thus, whenever we found a word presenting the pattern *i**bt**e*, we correct such pattern with *i**n**e*, ingnoring character case.
Concerning the manual correction of OOV words with respect to a French dictionary, we extracted the word dictionary of the whole training data (all 231 documents, see section 3.), sorted by decreasing order of number of occurrences. From such dictionary we selected all OOV words, which thus were also sorted by decreasing number of occurrences. The list being very large ($19,696$ entries), and containing many tokens occurring only once or twice, considering also time constraints, we manually corrected the top 300 entries. This resulted in correcting almost half of the total number of OOV words, i.e. the top 300 most occurring OOV words cover almost half of the total occurrences of OOV words.
Aligning the original mistaken OOV words with their manually-corrected version at character level, we extracted patterns for correction in the same way as described above. All the correction patterns were then applied to unseen data in order to correct OCR mistakes.

5. Evaluation

In this section we provide an evaluation of our preprocessing procedure and of the Named Entity Recognition system on OCR-ized data, at each step of the preprocessing procedure:

1. **baseline** Evaluation on the original unprocessed data
2. **reseg.** Evaluation on the data after re-segmentation of sentences
3. **retok.** Evaluation on the data after re-tokenization
4. **reseg.+retok.** Evaluation on data after performing both re-segmentation and re-tokenization
5. **correct** Evaluation on data after performing all the preprocessing steps, resegmentation, retokenization and correction of OCR mistakes

Taking into account the characteristics of our data, in particular the sparseness of the annotation of named entities, as well as the small percentage of reference corrections of entity realizing-surface with respect to the total number of entities, we performed the evaluation of our procedure in terms of the perplexity of a language model on the development and evaluation data, and in terms of the performance of the Named Entity Recognition system. These two evaluations are described in the two following sections.

5.1. Evaluation of the Procedure for Error Correction

In order to evaluate the different preprocessing procedure in terms of perplexity of language model, we extracted text, without entities, from the training, development and test data sets at each steps of the preprocessing procedure. When evaluating on the test set, the whole training corpus was actually used (training + development). In order to avoid confusion, we indicate with `TRN` and `train` the data used against the development and the test data sets, respectively. A language model was then trained on the TRN and train data.
The language model used to evaluate the corrections is trained with the SRI language modeling toolkit (Stolcke, 2002). The evaluation metric is the perplexity of the language model on a given data set. Given a sentence $W = w_1, ..., w_N$, the perplexity of a stochastic language model, represented as a probability distribution p, is defined as:

$$PPL(W) = 2^{H(W)}, H(W) = -\sum_{i=1}^{N} p(w_i) \cdot log(p(w_i)) \quad (1)$$

	DEV		TEST	
PPL on **Quaero-BN**	**162.8**	3,508	**162.8**	3,508
Corrections	PPL	OOV rate (units)	PPL	OOV rate (units)
Baseline	369.069	9.04% (31,741)	378.117	9.59% (33,013)
+ reseg.	355.764	8.86% (30,212)	363.360	9.39% (31,436)
+ retok.	177.052	5.29% (21,621)	178.018	5.43% (23,140)
+ reseg.+retok.	165.053	5.16% (20,639)	165.884	5.31% (22,030)
+ correct	**164.769**	5.16% (20,601)	**165.451**	5.30% (22,002)

Table 3: Evaluation of the procedure for correcting mistakes on OCR data, compared with manually transcribed broadcast data (ESTER2)

$H(W)$ is the entropy of W under the distribution p of the language model. The perplexity on a whole data set is computed by simply considering the data as a stream of tokens. The perplexity of a language model represents, given a word as input, the uncertainty of the model in choosing a word that can follow the given one. Since it is directly related to the entropy, the perplexity reflects effectively how well a sentence fits the model, and then it can be used as evaluation metric for our purposes.

The results of the evaluation of our correction procedure are presented in table 3 in terms of perplexity (PPL) and out-of-vocabulary rate, reporting also OOV words numbers in parenthesis. In order to underline the effect of the correction procedure, we show the comparison with manually transcribed broadcast news data (Quaero-BN), annotated with the same kind of named entity as OCR data used in this work and used for the 2011 Quaero Named Entity Recognition evaluation described in (Galibert et al., 2011; Dinarelli and Rosset, 2011b).

In table 3, the perplexity and OOV are given in the first line (**Quaero-BN**). As we can see, applying our correction procedure, we decreased the perplexity of the language model on the test data from roughly 378 to 165, which is roughly the same perplexity obtained for manually transcribed data (162.8). We can see also that, although all preprocessing steps contribute to decrease systematically the perplexity and the OOV rate, the most impacting preprocessing step is always the re-tokenization. This is due to the fact that OCR process introduces several different spurious tokens due to noise errors, mentioned in section 3., which cause a multiplication of token types. For instance, if noise errors like *', ., !* or *"* at the beginning and at the end of a token, alternatively, this can create potentially 16 new tokens. Since re-tokenization corrects this type of mistakes, it reduces drastically the number of tokens. As we can see also from Table 3, the less effective preprocessing step is always the OCR errors correction. As mentioned previously, words annotated with entities are a small percentage of the total number of words, and in turns, entities provided with reference correction for the realizing-surface are also a small percentage. Same rational holds for the manually corrected words. At the same time the correction patterns we apply, may present generalization problems, i.e. a given OCR error may occur in different contexts than those defined in the correction patterns described in sub-section 4.3.. As a consequence, the corrections have a small impact on the whole corpus.

	Evaluation on **DEV**	
Corrections	**SER**	**F1**
Baseline	43.1%	63.3%
+ reseg.	42.6%	64.0%
+ retok.	**41.6%**	64.1%
+ reseg.+retok.	42.2%	**64.2%**
+ correct	42.3%	**64.2%**

Table 4: Evaluation of the Named Entity Recognition system on the OCR-ized development data set at the different preprocessing steps

	Evaluation on **TEST**	
Corrections	**SER**	**F1**
Baseline	44.0%	62.3%
+ reseg.	43.8%	63.0%
+ retok.	**42.0%**	**63.5%**
+ reseg.+retok.	42.7%	63.4%
+ correct	43.0%	63.2%

Table 5: Evaluation of the Named Entity Recognition system on the OCR-ized evaluation data set at the different preprocessing steps

5.2. Evaluation of the Named Entity Recognition System

In this section we provide an evaluation of our NER system on the OCR-ized data, at all steps of the preprocessing procedure, as in previous section, and in terms of *Slot Error Rate* (SER) (Makhoul et al., 1999) and the traditional *F1-measure*. Slot Error Rate has a similar definition of word error rate for ASR systems, with the difference that substitution errors are split in three types: i) correct entity type with wrong segmentation; ii) wrong entity type with correct segmentation; iii) wrong entity type with wrong segmentation; here, i) and ii) are given half points, while iii), as well as insertion and deletion errors, are given full points. The total number of errors is divided by the total number of reference constituents.

Results obtained with our Named Entity Detection system, applying also our preprocessing procedure, on development and evaluation data are shown in table 4 and 5, respectively. As we can see again, the preprocessing procedure systematically yields improvements of results over the baseline system, where no preprocessing is performed on data. However in this experiments improvements are not incremental at each step, except for the `F1-measure` on the development data set (`DEV`). In particular the gains in terms of `F1-measure` on the `DEV` set are rather similar for the different preprocessing steps, with a total gain over the baseline of 0.9 points, obtained with re-segmentation and re-tokenization of data (`reseg.+retok.`), and with all the preprocessing steps (`correct`). The `F1-measure` on the evaluation set (`TEST`) tells roughly the same story, with similar gains for all preprocessing steps, with a total best gain of 1.2 points, obtained again with re-tokenization alone (`retok.`). (`reseg.+retok.`).

Interpretation of results in terms of Slot Error Rate is slightly different, and it is the same for `DEV` and `TEST`. The best performing preprocessing step is the re-tokenization alone, with a gain of 1.5 points on `DEV` and 2.0 on `TEST`. The other preprocessing steps, although they still yield improvements over the baseline, applied together are less performant than retokenization alone.

	Official Evaluation on **TEST**
System	**SER**
P1	60.3%
LIMSI	50.0%
P2	44.3%
+retok.	**42.0%**

Table 6: Evaluation of the Named Entity Recognition systems at the official evaluation campaign

Even with a deep analysis of the results, we did not find out any reasonable cause for the worsening of results obtained with the additional preprocessing steps. It may be because improvements with all preprocessing steps are quite similar with respect to the baseline. Beyond the fact that, without a corrected reference text to be used for validating the correction procedure, it is not easy to establish if a given strategy yields more benefit with corrections than noise, this outcome reflects also the fact that the percentage of words annotated with named entities is relatively small (always around 16%). Thus it is normal that the preprocessing step with larger impact on SER results is also the one with larger impact on perplexity and OOV rate on text, as it affect a much wider set of words.

In table 6 we provide also a comparison of our best performing system with those participating in the official 2011 evaluation campaign on old press data (Galibert et al., 2012). *LIMSI* is the system we proposed for the evaluation campaign, which was the same as described in this work, with the difference that it was not integrating the preprocessing procedure and that our new baseline better integrates semantic information (which explains the difference between this baseline and the official results). The other participants are indicated with *P1* and *P2*. Their systems are based on deep syntactic analysis (*P1*), and on a cascade of log-linear models (*P2*), respectively. As we can see, the system proposed here, integrating the preprocessing procedure, is far better than the one used for the official evaluation campaign (42.0% vs. 50.0% SER, respectively). Most importantly, the system proposed in this work outperforms the best system of the evaluation campaign.

6. Conclusions

In this paper we presented an analysis of mistakes found in a corpus acquired with OCR technology. We proposed a three-steps procedure to correct such mistakes. The procedure was evaluated in terms of perplexity of a language model built on training data against development and evaluation data. The procedure is effective since it allows to achieve, on OCR-ized data, a perplexity comparable to the one obtained on manually transcribed broadcast news data. Even more, using the preprocessing procedure, our named entity recognition system improves by a large margin results obtained during the official evaluation campaign on Old Press, and outperforms the best system of the evaluation campaign.

Given the variety of possible entity trees reconstructed with our parsing models, the approach proposed in this paper could be extended by reranking entity trees reconstructed starting from the n-best annotation generated by the CRF model, used to annotate components on words. Such reranking approaches have been used successfully in (Dinarelli et al., 2009b), (Dinarelli et al., 2009a) and in (Dinarelli and Rosset, 2011a).

Acknowledgments

This work was realized as part of the Quaero Program, funded by Oseo, French State agency for innovation.

7. References

Kate Byrne. 2007. Nested Named Entity Recognition in historical archive text. In *Proceedings of the first IEEE International Conference on Semantic Computing (ICSC 2007)*, Irvine, California.

Richard Tobin Claire Grover, Sharon Givon and Julian Ball. 2008. Named entity recognition for digitised historical texts. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

Marco Dinarelli and Sophie Rosset. 2011a. Hypotheses selection criteria in a reranking framework for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, pages 1104–1115, Edinburgh, U.K., Jul.

Marco Dinarelli and Sophie Rosset. 2011b. Models cascade for tree-structured named entity detection. In *Proceedings of International Joint Conference of Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, November.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2009a. Re-ranking models based on small training data for spoken language understanding. In *Conference of Empirical Methods for Natural Language Processing*, pages 11–18, Singapore, August.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2009b. Re-ranking models for spoken language understanding. In *Conference of the European Chapter of the Association of Computational Linguistics*, pages 202–210, Athens,Greece, April.

Olivier Galibert, Sophie Rosset, Cyril Grouin, Pierre Zweigenbaum, and Ludovic Quintard. 2011. Structured and Extended Named Entity Evaluation in Automatic Speech Transcriptions. In *IJCNLP Proc*.

Olivier Galibert, Sophie Rosset, Cyril Grouin, Pierre Zweigenbaum, and Ludovic Quintard. 2012. Extended named entity annotation on ocrized documents: From corpus constitution to evaluation campaign. In *Proceedings of the Eighth conference on International Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

Olivier Galibert. 2009. *Approches et méthodologies pour la réponse automatique à des questions adaptées à un cadre interactif en domaine ouvert*. Ph.D. thesis, Université Paris Sud, Orsay.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cyril Grouin, Sophie Rosset, Pierre Zweigenbaum, Karen Fort, Olivier Galibert, and Ludovic Quintard. 2011. Proposal for an extension or traditional named entities: From guidelines to evaluation, an overview. In *Proceedings of the Linguistic Annotation Workshop (LAW)*.

Mark Johnson. 1998. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24:613–632.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, Williamstown, MA, USA, June.

Daniel Lopresti. 2008. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, AND '08, pages 9–16, New York, USA. ACM.

John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. 1999. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252.

David Miller, Sean Boisen, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. 2000. Named entity extraction from noisy input : Speech and ocr. In *Proceedings of the sixth conference on Applied natural language processing*, pages 316–324.

Sophie Rosset, Olivier Galibert, Guillaume Bernard, Eric Bilinski, and Gilles Adda. 2008. The LIMSI participation to the QAst track. In *CLEF 2008 workshop*, Aarhus, Denmark, September.

Sophie Rosset, Cyril Grouin, and Pierre Zweigenbaum, 2011. *Entités Nommées Structurées : guide d'annotation Quaero*. LIMSI–CNRS, Orsay, France. `http://www.quaero.org/media/files/bibliographie/quaero-guide-annotation-2011.pdf`.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Proceedings of LREC*.

A. Stolcke. 2002. Srilm: an extensible language modeling toolkit. In *Proceedings of SLP2002*, Denver, USA.