NERC-fr: Supervised Named Entity Recognition for French

No Author Given

No Institute Given

Abstract. There is currently a lack of available language resources for French, especially for basic tasks such as Named Entity Recognition and Classification (NERC), which makes it difficult to build natural language processing systems for this language. This paper presents a supervised NERC model for French that has been trained and tested under a maximum entropy approach. The Apache OpenNLP libraries have also been extended, to support the required part-of-speech feature extraction component. The model achieves state of the art results for French, when compared to similar systems developed for other languages, and will be made publicly available.

1 Introduction

The Named Entity Recognition and Classification (NERC) task consists of detecting lexical units that refer to specific entities, in a sequence of words, and determining which kind of entity the unit refers to (e.g. person, location, organisation). NERC consists of two steps that can be approached either in sequence or in parallel. The first step is the detection of named entities in a text, while the second is the correct classification of the detected named entities, using a set of predefined categories.

Among the possible methods to determine and classify named-entities, we opted for a machine learning (ML) approach, as it provides language-independent core algorithms for the development of state of the art language processing modules. The system we present extends a supervised approach to NERC implemented in the Apache OpenNLP library[1].

Previous research has shown that part of speech (PoS) information improves the overall performance of NERC systems (Ekbal and Bandyopadhyay, 2008). Although OpenNLP integrates various feature extractors, it does not yet support PoS information extraction. We thus extended the OpenNLP library to include PoS feature extraction and integrated this information into our NERC system for French.

In general, there are fewer publicly available NLP tools for French than, for instance, English or Spanish. Our first goal was thus to build a state of art

[1] `http://opennlp.apache.org`

core NLP component for this language, and make it available to the research community.[2]

One of the first problems faced when building a NERC is the lack of labelled datasets. For our task, we used the ESTER corpus, a NE annotated corpus with 1.2 million words[3]. Section 3 provides a detailed description of the corpus.

This paper is organised as follows: Section 2 presents related work; section 3 describes the dataset used to train system; section 4 contains the description of the system; section 5 presents evaluation results; section 6 describes and evaluates the system's performance; finally, section 7 draws conclusions from this work and presents suggestions for future research.

2 Related Work

A considerable amount of work has been done in recent years on named entity recognition and classification. The main approaches to NERC can be categorised into Knowledge-based, Supervised, Semi-supervised and Unsupervised. Briefly, **Knowledge-based systems** were developed in the early stages of NERC research. Methods in this approach are essentially based on finite state machines and rule sets (see, e.g. (Appelt et al., 1995; Mikheev et al., 1999)). Research has continued along these lines, with e.g., (Budi and Bressan, 2003) reaching an F1 measure of approximately 70% for the recognition task. These tools are however costly to develop, as they require the development of knowledge-based resources which are difficult to port to other languages.

Supervised learning is currently the most widely used approach for the NERC task. Different techniques have been used such as Hidden Markov Models (Bikel et al., 1997), Decision Trees (Sekine, 1998) and Maximum Entropy Models (Borthwick, 1999). The main problem with Supervised Learning techniques is that a large amount of tagged data is needed to implement an effective system, and the accuracy of the models in a given domain is dependent on the training corpus.

With **Semi-supervised systems**, a first classifier is learned, which is then improved using unlabelled data. The most effective systems are based on linguistic features (Collins and Singer, 1999) or learn various types of NE simultaneously (Collins, 2002). Bootstrapping is also a popular method (Cucchiarelli and Velardi, 2001), where the output of existing NERC systems, or manually annotated seeds, are used to train the NERC model.

3 Datasets

To create the French NERC model, we used the ESTER corpus. This corpus is based on more than 1700 hours of Broadcast News data (from 6 French radio

[2] Both the NERC model for OpenNLP and the extensions made to the OpenNLP library will be made available here: `http://www.opener-project.org/`

[3] `http://catalog.elra.info/product_info.php?products_id=999`

channels), out of which 100 hours were manually transcribed. The corpus contains 1.2 millions of words for a vocabulary of 37,000 words and 74,082 named entities (15,152 unique NEs), labelled with a tagset of about 30 categories folded into 8 main types: persons (pers), locations (loc), organisations (org), geo-socio-political groups (gsp), amounts (amount), time (time), products (prod) and facilities (fac).

We divided the data into 3 sets: a training set (77.8% of the total corpus), a development set (9.7%) and a test set (12.5%). There is a 6 month time difference in the occurrence between data in the training/development set and in the test set: the training set contains Broadcast News ranging from 2002 to December 2003 and the test corpus was recorded in October 2004. The test set also contained data collected from 2 news radio channels which were not among the sources of the training data. One of the main characteristics of the ESTER corpus is the size of the NE tagset and the high ambiguity rate amongst NE categories (e.g. administrative regions and geographical locations): 40% of the matched corpus entities in the training corpus, and 32% of the unmatched ones, are ambiguous (Favre et al., 2005).

3.1 Corpus preprocessing

In order to use the ESTER corpus to train and test the models, we first converted it to the OpenNLP format. The two tagsets being different, we reduced the 30 named entity categories defined in ESTER corpus into the 6 categories needed for the task, along the lines described as follows:

- The geo-social-political tags were divided into three subcategories: *gsp.pers*, *gsp.loc* and *gsp.org*. These subcategories were then placed under *person*, *location* and *organisation*, respectively.
- The *amount* category, and its *amount.cur* subcategory, were categorised under *money*.
- *product* named entities were not used in our system.
- *person*, *location*, *organization*, *time* and *date* types were maintained as is.

The corpus was also formatted to allow processing within OpenNLP: each sentence was placed on a separate line, sentence initial words were capitalised, and all sentences were tokenised.

4 System Description

The system developed for the experiment described in this paper implements a supervised approach. We used a maximum entropy framework and a classifier that detects each NE candidate given certain features. The system's properties are described in the next sub-sections.

4.1 OpenNLP library extension

To build the NERC model, we used the Apache OpenNLP library, which is a machine learning based toolkit for natural language processing. It supports the most common NLP tasks, and in particular provides a maximum entropy-based framework for NERC. As previously mentioned, (Ekbal and Bandyopadhyay, 2008) demonstrated the usefulness of PoS information for NERC, when training models based on support vector machines. For that reason, we decided to extend the OpenNLP library for PoS feature extraction.

To extract PoS information, we trained a PoS tagger using annotated data in the French Treebank (Abeillé et al., 2003). The tagger is MaxEnt-based and was developed using OpenNLP's functionality.

4.2 Estimating parameters

Although OpenNLP provides default parameters for feature selection, we estimated optimal parameters in order to achieve better results. Parameters were tested along the lines given as follows:

1. **Sentence Boundaries**: the impact of beginning and end of sentence features was measured.
2. **Neighbouring tokens**: contextual features were extracted to measure the impact of neighbouring tokens. We defined a unit w as a token, a token pattern and token class, estimating optimal values for the i and j parameters within different w_{-i}, w, w_{+j} windows.
3. **Bigram window**: we extracted bigrams of neighbouring words, also estimating optimal window parameters for bigram sequences.
4. **Prefixes and suffixes window**: for any word w, we extracted i prefixes and suffixes on the left, and j prefixes and suffixes on the right. The maximum prefix and suffix char length was set to 4.
5. ***charngram* length parameter**: *charngram* concerns the features covering the minimum and maximum length of entities character-level *ngrams*.
6. **PoS window**: the system extracts features from the neighbouring PoS tags of the w-th word. We will provide results with and without this step in what follows.
7. **Cutoff**: cutoff specifies the minimal number of times a feature must be seen to be selected.
8. **Number of iterations**: this part estimates the number of iterations for the Generalised Iterative Scaling (GIS) procedure.

4.3 Named Entity Categories

To create our NERC model, we opted to recognise and classify six different named entity categories: **person**, **location**, **organisation**, **date**, **time** and **money**.

5 Experiments

To evaluate the performance of the NERC model at different stages, we used the development corpus included in the ESTER corpus and the three standard measures of *precision*, *recall* and F_1.

As described in section 4, we made use of the following features to improve our NERC model: default parameters, sentence boundaries, neighbouring tokens, bigram window, prefixes and suffixes window, *charngram* length parameter, PoS window, cutoff and number of iterations. Table 1 shows the results with and without the PoS feature.[4]

Steps	Without PoS			With PoS		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
1. Default parameters	90,47	82,87	86,5	90,47	82,87	86,5
2. Sentence boundaries	90,47	82,87	86,5	90,47	82,87	86,5
3. Neighbouring tokens	90,8	83,35	86,91	90,8	83,35	86,91
4. Bigram window	90,91	83,68	87,14	90,91	83,68	87,14
5. Prefixes and suffixes window	91,2	85,32	88,16	91,2	85,32	88,16
6. charngram	91,54	85,09	88,2	91,54	85,09	88,2
7. Pos window	-	-	-	91,35	84,6	87,85
8. Cutoff	91,54	85,09	88,2	91,35	84,6	87,85
9. Iterations	91,45	85,23	88,23	91,5	85,39	88,34

Table 1. Best performing features, with and without PoS

The best results were obtained with the parameters below:

- **Sentence Boundaries**: the best sentence boundaries features were sentence start indicators.
- **Neighbouring tokens**: the best window scores for tokens, token classes and token patterns were 2-1, 1-2 and 0-0, respectively.
- **Bigram window**: the best results were achieved with a bigram window of 1-0.
- **Prefixes and suffixes window**: for prefixes and suffixes, the best window sizes were 1-0 and 1-1, respectively.
- ***charngram* length parameter**: *charngram* best length was 6.
- **PoS window**: 0-3 was the best PoS window size.
- **Cutoff**: both with and without PoS features, the optimal cutoff was 5.
- **Number of iterations**: without PoS features 380 iterations were needed to achieve the best scores; including PoS features, 260 iterations were needed.

The last step involved estimating the optimal number of iterations for the Generalised Iterative Scaling procedure. As shown in figure 1, the performance reaches a high after only 140 iterations, both with and without the inclusion of the PoS feature. In the latter case, results improved until 380 iterations were

[4] *Default parameters* denotes the default OpenNLP parameters.

reached; when including the PoS feature, best results were achieved after 260 iterations.

As shown in table 1, the best performances in terms of F_1-*measure* were 88.34% and 88.23%, with and without PoS features, respectively.

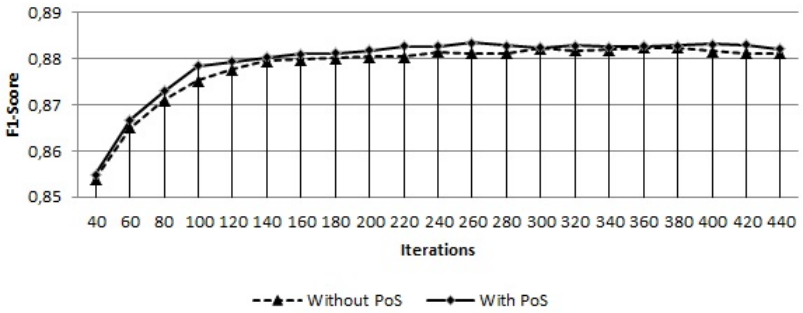

Fig. 1. Model performances per iteration

Table 2 shows the results obtained on the ESTER corpus development and test datasets. The 6 months difference and sources for the data that made up both sets, described in section 3, might be a reason for the differences in scores that were observed.

Dataset	Without PoS			With PoS		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
Development	91.45	85.23	88.23	91.5	85.39	88.34
Test	86.15	75.69	80.59	86.2	75.85	80.69

Table 2. Performance results on the ESTER corpus test and dev sets

Table 3 presents the evaluation results for each entity category in the test set. Although the overall detection was good, the low recall value for the *money* entity category has a negative impact on the final results.

6 Discussion

For languages such as Spanish or English, many resources are available for NERC. For French, few systems are readily available due to the absence of publicly available entity annotated datasets which could be used to create NERC supervised models. The two NLP tools described below perform NER for French and were used to compare systems performance:

Categories	Without PoS			With PoS		
	Precision	*Recall*	F_1-*score*	*Precision*	*Recall*	F_1-*score*
Location	88.58	86.04	87.29	88.56	85.84	87.18
Person	88.71	80.5	84.41	88.59	80.84	84.54
Time	89.54	81.07	85.09	89.61	81.66	85.45
Date	84.01	74.48	78.96	84.03	74.59	79.03
Organization	76.53	54.99	64	76.79	55.42	64.38
Money	70.31	24.59	36.44	74.19	25.14	37.55
Total	86.16	75.69	80.59	86.20	75.85	80.69

Table 3. Evaluation results per entity category

Unitex CasEN[5] uses lexical resources and local description of patterns, transducers which act on text insertions, deletions and substitutions. The CasEN tool requires a certain knowledge of the language, as the NERC component is based on regular expressions which would need to be adapted for other languages.

LingPipe[6] is a toolkit for text processing which has been used for French in HLT 2005 (Favre et al., 2005). In this paper, an automatic speech recognition system (ASR) for NERC tasks is presented, which uses a text-based NERC model trained with the LingPipe tool. To train and evaluate their NERC model, the ESTER corpus was used, which allows for a direct comparison with our system. LingPipe comes with a license that makes the code freely available for research purposes, with additional constraints for its integration in a commercial product. The NERC component we have developed will be distributed under the Apache License v.2 and will thus be useful for both research and industrial applications.

Table 4 shows the performance of our model against CasEN and Lingpipe. As can be seen, our model showed better performance overall, although our results are close to those obtained with LingPipe.

Systems	*Precision*	*Recall*	F_1-*score*
CasEN	68.86	40.63	50.99
LingPipe	-	-	79.00
NERC-fr	86.2	75.85	80.69

Table 4. NERC tools comparison

7 Conclusion and future work

In this paper, we presented a module for Named Entity Recognition and Classification in French. This tool has been developed using OpenNLP, extended to extract PoS information features. The component was used to train several

[5] `http://tln.li.univ-tours.fr/Tln_CasEN.html`
[6] `http://alias-i.com/lingpipe/`

NERC models, using the ESTER corpus. The optimal model built under our approach showed slightly better performance than comparable tools, without requiring the development of language-dependent resources beyond annotated corpora. Despite the simplicity of the approach, the system showed better performance than the CasEN rule-based system and slightly better results than a supervised system developed with LingPipe.

When compared to NERC in other languages, the results for French are lower on average, in terms of precision and recall, which is due to the existence of larger amounts of training corpora for these languages.

In future work, we plan to extend the system with additional linguistic features, for example integrating syntactic information and evaluate their impact on the performance of the NERC module we described in this paper.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for french. In *Treebanks*, pages 165–187. Springer.

Douglas E Appelt, Jerry R Hobbs, John Bear, David Israel, Megumi Kameyama, David Martin, Karen Myers, and Mabry Tyson. 1995. Sri international fastus system: Muc-6 test results and analysis. In *Proceedings of the 6th conference on Message understanding*, pages 237–248. Association for Computational Linguistics.

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: A high performance learning name-finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, Washington DC.

A. Borthwick. 1999. A maximum entropy approach to named entity recognition. In *Ph.D. thesis, New York University.*

Indra Budi and Stéphane Bressan. 2003. Association rules mining for name entity recognition.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the EMNLP'02.*

Alessandro Cucchiarelli and Paola Velardi. 2001. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131.

Asif Ekbal and Sivaji Bandyopadhyay. 2008. Named entity recognition using support vector machine: A language independent approach. *International Journal of Computer Systems Science & Engineering*, 4(2).

Benoît Favre, Frédéric Béchet, and Pascal Nocéra. 2005. Robust named entity extraction from large spoken archives. In *Proceedings of HLT-EMNLP*, pages 491–498. Association for Computational Linguistics.

A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the 9th EACL*, pages 1–8.

Satoshi Sekine. 1998. Nyu: Description of the japanese NE system used for met-2. In *Proc. Message Understanding Conference.*