A Bootstrapping Approach for Training a NER with Conditional Random Fields

Jorge Teixeira, Luís Sarmento, and Eugénio Oliveira

LIACC - FEUP/DEI & Labs Sapo UP
Porto, Portugal
{jft,las,eco}@fe.up.pt

Abstract. In this paper we present a bootstrapping approach for training a Named Entity Recognition (NER) system. Our method starts by annotating persons' names on a dataset of 50,000 news items. This is performed using a simple dictionary-based approach. Using such training set we build a classification model based on Conditional Random Fields (CRF). We then use the inferred classification model to perform additional annotations of the initial seed corpus, which is then used for training a new classification model. This cycle is repeated until the NER model stabilizes. We evaluate each of the bootstrapping iterations by calculating: (i) the precision and recall of the NER model in annotating a small gold-standard collection (HAREM); (ii) the precision and recall of the CRF bootstrapping annotation method over a small sample of news; and (iii) the correctness and the number of new names identified. Additionally, we compare the NER model with a dictionary-based approach, our baseline method. Results show that our bootstrapping approach stabilizes after 7 iterations, achieving high values of precision (83%) and recall (68%).

Keywords: Named Entity Recognition, Machine Learning, Conditional Random Fields, Natural Language Processing.

1 Introduction

There are currently many popular machine learning approaches for inferring Named Entity Recognition (NER) systems. Most of these techniques require a relatively large amount of text where entities have been annotated in context. However, annotating such corpora is difficult and expensive, and these factors usually limit both the *size* and the *recency* of such corpora. As a consequence, most available NER-annotated corpora are usually small and are composed of annotations made in text with several years old. From a practical point of view, this raises two problems. First, a small corpus may not be enough to allow inferring robust NER models, since only a relatively small number of contexts are present. Second, models inferred from old data may not be suitable to classify new data [8][7]. As we will show later, by training a classification model based on part of HAREM [10], a relatively small and old (from 1997) annotated NER

L. Antunes and H.S. Pinto (Eds.): EPIA 2011, LNAI 7026, pp. 664–678, 2011.

corpus, and testing the other part on the learned model, we show that modest precision values can be attained. Additionally, by testing the learned model on a dataset of recent news (from May 2011), we obtained even lower accuracy, meaning that the training corpus did not have enough new information to build a reliable model. We will present this data and results in Section 6.

The solution for both these problems would consist in constantly updating the annotated corpus with more recent examples (possibly substituting older annotations). The resulting corpus would become larger, and would contain recent text. But the amount of human effort involved in such task is simply too much for this strategy to become sustainable. Thus, we propose a bootstrapping approach to perform the annotation of entities in a large corpus, while simultaneously inferring a NER model.

We start with a large set of (non-annotated) news items and a dictionary of names that are very frequently found in news. We only consider names that have two or more words (e.g. "name surname"), which we assume to be unambiguously mentioned. Next, we annotate names in the set of news items by considering matches with entries in the dictionary. We then select the subset of sentences in which all the capitalized tokens are part of an annotated name, which can thus be considered *completely* annotated. This set of sentences will serve as the *seed* corpus.

In the second stage, we use the seed corpus to infer a conditional random field (CRF) model for performing name annotation. Such model is then run over the initial seed corpus to increase the number of (completely) annotated sentences. The resulting larger corpus is used to infer a new CRF model. This cycle is repeated until the model stabilizes. In the end, we expect to have a very large corpus of news annotated with high accuracy.

In each iteration, we evaluate three parameters. First, we evaluate the precision and recall of the inferred model in annotating a small gold-standard collection (HAREM) [10]. This allow us to check how robust our classification model is becoming, taking into account a standard (although relatively small and old) reference corpus. Second, we manually evaluate the precision and recall of the annotation over a small sample of news corpus from which we generated the news corpus. This allow us to estimate the accuracy of the annotation that we are producing for the entire news corpus. Finally, we manually check the correctness and the number of new names identified using the inferred model (i.e. not found in the initial dictionary) for assessing the speed at which the system converges to a stable NER model.

The remaining of the paper is organized as follows. In Section 2 we discuss some related work. In Section 3 we describe our Method and in Section 4 the Classification Model and Features Description. The Experimental Set-up will be presented in section 5, the Results obtained are described in Section 6 and its Analysis and Discussion are presented in Section 7. Finally, Conclusions and Future Work are presented in Section 8.

2 Related Work

The difficulty in obtaining manually annotated data for training NER systems has motivated researchers to look for alternative ways of generating annotated data, or for making the best possible use of unlabeled data.

For example, Collins et al. [1] use seven very simple rules to perform the annotation of a seed news corpus. The rules are: "New York", "California" and "U.S." are locations; any name containing Mr. is a person; any name containing Incorporated is an organization; and I.B.M. and Microsoft are organizations. This is the only supervised information used. The approach proposed by the authors is to find a weighted combination of simple (weak) classifiers. The two classifiers are built iteratively: each iteration involves minimizing a continuously differential function which bounds the number of unlabeled examples (around 90,000) on which the two classifiers disagree. The authors used a dataset of approximately 1 million sentences extracted from New York Times and manually evaluated a sub-set of 1,000 examples, assigning one of the four available categories: location, person, organization or noise. The authors report that their system classified names with over 91% accuracy, which was obtained with almost no manual effort involved.

Valchos et al. [13] demonstrated that bootstrapping an entity recognizer for genes from automatically annotated text can be more effective than by using a fully supervised approach based on manually annotated biomedical text. Their system was based on an improvement of a bootstrapping method previously presented by Morgan et al. [6]. The authors started by creating a test set for evaluating the quality of the NER gene recognizer proposed. The test set contained 82 biomedical articles manually annotated, following some pre-determined guidelines and taking special attention for the context around the words to be annotated. The authors then used the previously annotated texts to automatically annotate abstracts based on pattern matching. The resulting corpus, which, contained approximately 117,000 annotated names (17,000 of them unique) was used to train an Hidden Markov Model (HMM) for performing gene NER. Evaluation on the test set achieved an F-score of 81%. The authors also presented three different approaches for improving the results achieved. The first one consists in using state-of-the-art gene dictionary to increase the number of names annotated in the articles. After reapplying their HMM system, they achieved lower F-score (78%), which lead the authors to stress the importance of using naturally occurring data as training material. For improving the results previously obtained, the authors remove all sentences from the training set that did not contain any entities. After retraining the models, the resulting F-score obtained decreased slightly (80%), mainly because the precision decreased considerably, since this strategy deprived the classifier from contexts that could help the resolution of erroneous cases. Lastly, the authors tried to filter the contexts used for substitution and the sentences that were excluded using the confidence values of the HMM system. Results obtained improved slightly (83%) indicating that this was the best approach proposed.

Our work is similar to the one presented by Valchos et al. [13], since we also start with a dictionary of names to perform the seed annotation. However, tackling name recognition in news is a more dynamic problem, since new persons' names may "appear" everyday in news streams, including foreign ones for which no dictionary information may be (even partially) available. Also, in contrast with other works, namely Collins et al. [1] , we iteratively re-annotate our initial corpus using the models that we infer. The bootstrapping cycle has no predefined number of iterations, and runs until it reaches stability. This strategy allows our system to deal with an open set of names.

Regarding the impact of using relatively old data to train NER system, the study of Mota and Grishman [7] is one of the most relevant ones. The authors tested the performance of their NER system on a news corpus that spans for 8 years. Their NER tagger was trained and tested on distinct time segments of the news corpus. The main result was that the performance of the tagger clearly decreased as the the time gap between the training data and the test data became larger.

As far as we know, there has not been much work in trying to automatically rebalance a reference corpus with more up to date material. In this work, we also try to tackle this dimension of the problem.

3 Method

3.1 Initial Data

Our initial data is a corpus of news items, $\mathcal{C}^{news}$, and a list of names, $\mathcal{N}^{initial}$. The $\mathcal{C}^{news}$ corpus is composed of 50,000 news items extracted from Portuguese online newswires between the end of April 2011 and the middle of May 2011. Each news item contains a title and a body, and both parts are subject to identification of named entities. On total, this dataset contains approximately 400,000 sentences. The dictionary of names, $\mathcal{N}^{initial}$, is a list of 2,450 persons' names that are frequently found on news, and includes both Portuguese and international names. This list was compiled by scanning a collection of approximately 500,000 news and extracting all sequences of capitalized words that could be found in a context that is very correlated with names of people. The context used was "[Capitalized Word Sequence], [ergonym], ", where ergonym is a word normally included in a job description. Such pattern is frequently used on news to introduce people relevant to the news piece (e.g. "[Nicholas Sarkozy], [president]..."). We only considered capitalized word sequences that were identified more than 3 times on the entire collection, so only 2,450 persons' names where obtained. Although this is a relatively small number, past studies ([5] and [13]) have proven that a small but yet well-known and naturally occurring list of names is more advantageous than large gazetteers of low-frequency names.

3.2 Bootstrapping Cycle

The bootstrapping cycle is summarized in Figure 1. In the first run of the bootstrapping cycle (identified in Figure 1 by Iteration 0), we automatically annotate $\mathcal{C}^{news}$ following a simple dictionary-based approach, using the 2,450 entries stored in $\mathcal{N}^{initial}$. This annotation is performed using the following rules:

1. Exact matches starting by the longest name string from $\mathcal{N}^{initial}$ towards the shortest;
2. Soft matches between $n_i \in \mathcal{N}^{initial}$ on $\mathcal{C}^{news}$, which will allow us to include parts of names in common to both the $n_i \in \mathcal{N}^{initial}$ and $\mathcal{C}^{news}$ (e.g. we consider "Obama" as a soft match of "Barack Obama");

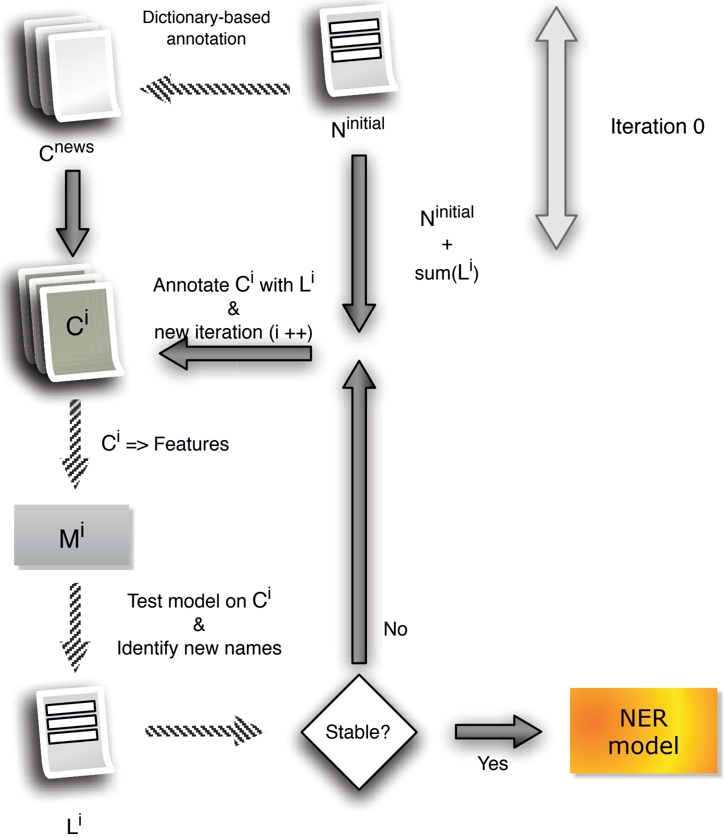
Dictionary-based annotation
Cnews
Ninitial
Iteration 0
Ninitial + sum(Li)
Annotate Ci with Li & new iteration (i ++)
Ci
Ci => Features
Mi
Test model on Ci & Identify new names
No
Stable?
Yes
NER model
Li

Fig. 1. Bootstrapping method

By following these rules, we were able to automatically annotate $\mathcal{C}^{news}$ and end-up with an annotated news corpus $\mathcal{C}^0$ with 57,642 persons' names, from which 50,514 were annotated in the body b_i of the news and 7,128 from the title t_i. We then used $\mathcal{C}^0$ to learn a classification model based on CRFs. We start by describing each example in the annotated corpus using a rich set of features $\mathcal{F}$, explained in section 4.2. Then, we infer a model $\mathcal{M}^0$. This model will then be applied on our previously used corpus $\mathcal{C}^0$ and we will create a list of the newly identified names, $\mathcal{L}^0$. With this list, together with the initial list of names $\mathcal{N}^{initial}$, we will be able to re-annotate the news corpus $\mathcal{C}^0$ and obtain a new annotated corpus, $\mathcal{C}^1$. The re-annotation process is based on the annotation rules described above.

At this point we will start a new iteration i of the bootstrapping process. This process will finish as soon as the system achieves a stable state.

4 Classification Model and Feature Description

4.1 Conditional Random Fields Models

Although our bootstrapping strategy does not directly depend on the classification algorithms used, we opted for Conditional Random Fields. CRFs are undirected statistical graphic models, and McCallum et al. [4] have shown that are well suited for sequence analysis, particularly on named entity recognition on newswire data.

According to Lafferty et al. [3] and McCallum et al. [4], let $o = \{o_1, o_2, ..., o_n\}$ be a sequence of words from a text with length s. Let $\mathcal{S}$ be a set of states in a finite state machine, each of which is associated with a label $l \in \mathcal{L}$ (e.g.: name, job, etc.). Let $s = \{s_1, s_2, ..., s_n\}$ be a sequence of states that corresponds to the labels assigned to words in the input sequence o. Linear chain CRFs define the conditional probability of a state sequence given an input sequence to be:

$$P(s|o) = \frac{1}{Z_o} exp \left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(s_{i-1}, s_i, o, i) \right) \tag{1}$$

where Z_o is a normalization factor of all state sequences, $f_j(s_{i-1}, s_i, o, i)$ is one of the m functions that describes a feature, and λ_j is a learned weight for each such feature function. For this work we only use binary feature functions, a first order Markov independence assumption. A feature function may be defined, for example, to have value 0 in most cases, and have value 1 if and only if state s_{i-1} is state #1 (this state may has, for example, label *verb*) and state s_i is state #2 (for example a state that have label *article*). Intuitively, the learned feature weight λ_j, for each feature f_j, should be positive for features correlated with the target label, negative for features anti-correlated with the label, and near zero for relatively uninformative features, as described by [12]. CRFs are described in more detail by [3].

We used CRF++ (version .054)[1], a customizable implementation of CRFs for segmentation/labeling of sequential data, and we set to 50 the maximum number of iterations of the algorithm. On one hand, the convergence becomes extremely slow for large sets of data, such as the one we are using; and on the other hand 50 iterations are enough for the algorithm to converge in our scenario. We also specify a template that will be used by the CRF++ algorithm

[1] Available at: `http://crfpp.sourceforge.net/`

to learn the model. We opted for using a simple and straightforward template that only describes each of the tokens (usually a word, but may also include punctuation), their positions and their features within a sliding window of size 5. However, templates allow us to make different combinations of each token, its position and its features along with the other tokens from the sliding window. After several tests we conclude that the gains achieved by changing the templates description were very low, thus we used the simplest template approach.

4.2 Features Description

The quality and robustness of the NER model obtained greatly depends on the set of features used to describe the examples [9]. In our case, we decided to use *word-level* features and a window of 2 tokens to the left and to the right of the focus word. Table 1 presents groups of features used:

Table 1. Set of features used for the annotation of $\mathcal{C}^{news}$

	Features	Examples
$\mathcal{F}_{cap}$	Capitalized word	*John* or *Sophie*
$\mathcal{F}_{acr}$	Acronym	*NATO* or *USA*
$\mathcal{F}_{lng}$	Word Length	"musician" - *8*
$\mathcal{F}_{end}$	End of sentence	
$\mathcal{F}_{syn}$	Syntactic Cat.	"said" - *verb*
$\mathcal{F}_{sem}$	Semantic Cat.	"journalist" - *job*
$\mathcal{F}_{names}$	Names of people	*Barack Obama*

For the first group of features from Table 1, "Capitalized Word", "Acronym" and "Word Length", we developed simple and straightforward methods that fit these features. Regarding the "End of sentence" features, we used a tokenizer, developed by Laboreiro et al. [2]. This tool is based on a classification approach and is focused on the Portuguese language. After tokenizing the text, we apply a set of regular expression in order to split sentences and correctly identify the end of the sentences. For the "Syntactic Category" and "Semantic Category" features, we used LSP (Léxico Semântico do Português). LSP is a lexicon developed for the Portuguese Language, able to perform syntactic (and for some words a semantic) analysis of words. This allows us to add, for example, the semantic category "[nationality]" to the word "american" or even the semantic category "[communication verb]" to the word "say". The last set of features, $\mathcal{F}_{names}$ represent a list of names extracted from a Portuguese gazetteer developed by Sarmento et al. [11]. REPENTINO is a gazetteer for the Portuguese language that stores names under nearly 100 categories and subcategories. For this work, we are only interested in names of people, which are identified by the category *HUM* (human), subcategory *EN_SER* (human being entity). The task of extracting names from REPENTINO is thus straightforward and consists simply on building a list of all entities tagged on REPENTINO with the previous described category and subcategory.

Preliminary studies that we have conducted led to the conclusion that the best performance obtained by the trained models for NER tasks is by using all the 7 features together. Thus, we will describe training examples with all the features described in Table 1.

5 Experimental Set-Up

We are interested in: (i) proving that the age of the corpus has an important effect on the performance of NER systems; and (ii) evaluating our bootstrapping method in two different perspectives: by measuring the quality of the CRFs models created at each iteration and by evaluating the performance of our method in annotating a news corpus.

5.1 Measuring the Effect of Age in the Training NER Models

Mota and Grishman [7] had shown that there is a significant effect of the age of an annotated corpus on a NER tagger, and we are interested on evaluating this effect. For that, we will start by using 80% of HAREM annotated corpus, $\mathcal{C}_{train}^{HAREM}$, as our training corpus and the remaining 20% (with the annotations removed) as the test corpus, $\mathcal{C}_{test}^{HAREM}$. Then, for our baseline NER method, we create a dictionary of names from the training corpus, and annotate the test corpus by simply performing string matching operations. The quality of the annotated test set will allow us to calculate a performance measure for our baseline.

For the CRF method, we will train a CRF model with the $\mathcal{C}_{train}^{HAREM}$ and then test this model on $\mathcal{C}_{test}^{HAREM}$. Similar to the previous case, we will measure the performance of the CRF method based on the results of the annotation of the test set. By applying these evaluation methods, we want to prove that HAREM corpus is small and thus insufficient to be used as a model for NER. Then, we will use the same training set - $\mathcal{C}_{train}^{HAREM}$ - but this time the test set will be a small set of 1,000 recent news items, $\mathcal{C}_{test}^{news}$, extracted from the web in May 2011. We apply both the baseline NER model and the CRF NER model on this test set and evaluate the annotations automatically produced. Following the idea of Mota et al [7], with this test we intend to show that the performance of NER systems trained with a corpus that is chronologically distant (14 years) from the test corpus is seriously affected by the age factor. This should reinforce our motivation for proposing the bootstrapping technique we described before. Tests performed over the gold-standard corpus (HAREM) are totally automatic, as we have access to the complete annotated dataset. On the other hand, evaluation tests performed on the test set of recent news are manual, and consist of manually annotating a random sample of 50 different news items extracted from $\mathcal{C}_{test}^{news}$.

5.2 Evaluating the Bootstrapping Process

To measure the performance of our bootstrapping method and its evolution in each iteration, we will calculate the precision and recall of the inferred bootstrapping CRF model in annotating a small gold-standard collection (HAREM), in order to test the robustness of our NER model taking into account a gold-standard corpus (HAREM). Also, we will manually evaluate the precision and recall of the annotation process over a random sample of 20 news items extracted from the automatically annotated set of news. This will allow us to estimate the accuracy of the NER system on annotating a news corpus.

Our experiments will be performed considering the following empirically set conditions:

- The CRF threshold was empirically set to 0.6, so that the system will only assign a new name to the list of new names if its precision value obtained from the CRF bootstrapping model is higher than 0.6.
- The system will only assign a new name to the list of new names if it occurs at least 4 times on the entire test set, thus avoiding incorrect rare names that may introduce noise to the bootstrapping system.
- Persons' names with only one word (this means that the context words were not identified by the NER model as persons' names, or do not exist) will only be considered as valid new names, and thus added to the list of new names, if the precision value obtained by the CRF model is greater than 0.9. (e.g.: "Obama" or "Sócrates").

6 Results

6.1 Results on Evaluating NER by Training with HAREM Dataset

Results obtained for both the baseline NER model and the CRF NER model are presented in Table 2. Both methods were trained with $\mathcal{C}_{train}^{HAREM}$. This allows us to directly compare results obtained by each of them.

Regarding the dictionary-based method (see Table 2 - Dictionary Training Method), one can see that the precision is 1 for both test sets, as the annotation method consists only of string matches. Also, F1-measures obtained are relatively low (54% when tested with $\mathcal{C}_{test}^{HAREM}$ and 21% when testing with $\mathcal{C}_{test}^{news}$) and decrease when we test the model with recent news items. For the results obtained using the CRF NER model (see CRF Training Method on Table 2), the F1-measure values are considerably higher when compared to the dictionary-based method. Additionally, the F1-measure also decreases when the model is tested with the subset of 1,000 recent news, $\mathcal{C}_{test}^{news}$.

Table 2. Results for baseline NER model and the CRF NER model

Training Method	Testset	Precision	Recall	F1-measure
Dictionary	$\mathcal{C}_{test}^{HAREM}$	1.00	0.37	**0.54**
Dictionary	$\mathcal{C}_{test}^{news}$	1.00	0.12	**0.21**
CRFs	$\mathcal{C}_{test}^{HAREM}$	0.93	0.82	**0.87**
CRFs	$\mathcal{C}_{test}^{news}$	0.94	0.40	**0.55**

6.2 Results for the Bootstrapping Method

As far as the bootstrapping method is concerned, we performed two different evaluations, as described in section 5.2. Both evaluations were performed on the bootstrapping CRF model. This model was built from the news corpus $\mathcal{C}^{news}$ (composed by 50,000 news) and the initial set of names $\mathcal{N}^{initial}$ (containing 2,450 names frequently found on news). Results for the automatic evaluation, performed on the gold-standard corpus (HAREM), are presented in Table 3 (precision $\mathcal{P}$, recall $\mathcal{R}$ and F1-measure $\mathcal{F}1$).

Table 3. Automatic Evaluation of the performance of the bootstrapping method on HAREM (gold-standard corpus)

Iteration	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{P}$	0.89	0.88	0.90	0.88	0.91	0.90	0.90	0.86	0.86	0.89	0.90	0.88
$\mathcal{R}$	0.32	0.36	0.45	0.36	0.41	0.44	0.47	0.49	0.48	0.48	0.56	0.45
$\mathcal{F}1$	**0.47**	**0.51**	**0.60**	**0.51**	**0.56**	**0.59**	**0.62**	**0.62**	**0.62**	**0.62**	**0.69**	**0.60**

From these results one can see that the bootstrapping system consistently increases the F1- measure of the NER system along the iterations. Also, after 7 iterations the NER system stabilizes, as the F1-measure obtained for subsequent iterations is mostly constant (62%).

Results obtained for the manual evaluation of the bootstrapping method, performed on a small random subset of recent news, are presented in Table 4.

Table 4. Manual evaluation of the performance of the CRF models trained using a bootstrapping approach

Iteration	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{P}$	0.78	0.78	0.74	0.88	0.82	0.78	0.83	0.81	0.77	0.77	0.76	0.78
$\mathcal{R}$	0.42	0.61	0.50	0.53	0.53	0.61	0.68	0.66	0.65	0.66	0.64	0.68
$\mathcal{F}1$	**0.55**	**0.68**	**0.60**	**0.66**	**0.64**	**0.68**	**0.75**	**0.73**	**0.71**	**0.71**	**0.70**	**0.73**

From Table 4, one can see that the F1-measure after each bootstrapping iteration grows sustainedly until iteration 7 supported by a near constant growth of recall, despite small fluctuations in precision. From iteration 8 onwards both recall and precision start oscillating resulting in a set of F1 values that oscillate between 0.70 and 0.73. However, the maximum value of F1 is reached at iteration 7.

Additionally, we evaluate the new names identified on each iteration of the bootstrapping method (built from $\mathcal{C}^{news}$). Results are presented in Table 5 and include both the number of new names identified as well as its correctness, measured by the precision measure, of the new names identified.

Table 5. Manual evaluation of the new names identified

Iteration	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{P}$	0.90	0.90	1.00	0.95	0.85	1.00	0.95	1.00	1.00	0.80	0.85	0.95
#new names	1,165	500	159	374	28	40	52	101	203	94	52	29

From results presented in Table 5 one can see that the precision values are equal or higher than 85% for the majority of the bootstrapping iterations. Also, one can see that the number of new names identified on each iteration is decreasing from iteration 1 to iteration 7. After iteration 7 we observe small variations of the number of new names. However, the global tendency is a decrease of the number of new names identified.

7 Analysis and Discussion

Table 2 shows that HAREM is not adequate to be used on an up to date NER system, when considering its age. Let us compare results obtained by using $\mathcal{C}_{test}^{HAREM}$ as test set, against $\mathcal{C}_{test}^{news}$. Both tests use the same training set, $\mathcal{C}_{train}^{HAREM}$. For the first case, this represents a chronologically similar test set, when compared to the training set. On the other hand, the second test set, represents a chronologically distant dataset (about 14 years old of difference). In the first case, we obtained a F1-measure of 54%. However, on the second case, F1-measure decreases to 21%. This means that using an old corpus to build a NER model is less efficient when it is applied to new, and chronologically distant, data.

Still observing results from Table 2, it is interesting to compare results obtained by using the baseline method, a straightforward dictionary-based approach, against the ones obtained by using CRF model. As one can see, for both test sets, the F1-measure obtained when using CRFs method is always significantly higher than using the dictionary based approach. From these results we may conclude that: (i) both NER and NER CRF models suffer from the effect of the training set age; (ii) CRFs seems to be more robust to the age effect when compared with the NER model, which is based on a dictionary of names that quickly gets out of date.

We used two different strategies for evaluating the bootstrapping approach we propose. Table 3 shows the results obtained from the automatic evaluation of the performance of the bootstrapping method on HAREM, the gold-standard corpus. From this results one can see that the bootstrapping system stabilizes after 7 iterations, as the F1-measure obtained for subsequent iterations is always constant. These results allows us to say that our bootstrapping approach is robust for the NER task proposed.

Table 4 presents results obtained for the manual evaluation of our bootstrapping method on a set of recent news. These results are coherent (similarly behave) with those achieved for the automatic evaluation of our method with HAREM. One can see that for this evaluation scenario the bootstrapping method also stabilizes after 7 iterations. Interestingly, the F1-measure obtained for iteration 7

(75%) is considerably higher that the one obtained from the automatic evaluation on HAREM (62%). As described in subsection 3.1, the news dataset $\mathcal{C}^{news}$ used for training is recent (from April to May 2011). The chronological distance between the training and testing datasets is very small for this scenario. On the contrary, this distance is considerably higher (14 years old) when comparing it to HAREM test set, used on the automatic evaluation. This clearly shows the effect age of the training set on NER systems.

Also, we manually evaluate the number and correctness of the new names identified by our bootstrapping methods. Results in Table 5 show that this number tends to decrease from iteration to iteration. This is an expected behavior, and ideally this number would tend to zero, meaning that the model would not be able to identify more names. However, from a practical point of view, the gain of new names identified for iterations 8 and more compared with the global accuracy of the system becomes insignificant. Performing a simple error analysis on the new names identified, we found two different types of errors, presented in Table 6.

Table 6. Error analysis on the new names identified on each bootstrapping iteration

Error type	Error description	Example
$\mathcal{E}_{ne}$	Wrong type of named entity	"*General Motors* comment on the crisis"
$\mathcal{E}_{conj}$	Missed name conjunction	"Jorge Nuno Pinto *da* Costa"

The first type of errors, $\mathcal{E}_{ne}$, happens when the named entity (on the example from Table 6, an organization) occurs on a context that is misleading. Considering the example phrase "Barack Obama comment on the crisis", in this case the context around the named entity (Barack Obama) is exactly the same as in the erroneous one. However, while the incorrectly annotated name is a name of an organization, the name of the example is a person's name. This type of error may be reduced if we use additional information as lists of names of organizations. Additionally, when the bootstrapping CRF model is not able to identify the conjunctions in the middle of a persons' name, we are in the presence of errors of type $\mathcal{E}_{conj}$. This error is only common in long names (four or more words) because the context is too broad and the sliding window may not be sufficiently large to capture all the relevant context around the name, and thus correctly identify the boundaries of the name.

In Figure 2 we present a comparative study of the performance of the baseline and the bootstrapping method, measured in terms of the F1-measure. We compare four different methods:

- *Baseline dictionary on news*: We built a model based on a dictionary of names from the training set $\mathcal{C}^{HAREM}_{train}$ and test this model $\mathcal{C}^{news}_{test}$ on a set of 1,000 recent news.
- *Baseline CRF on news*: We built a model based on the same dictionary of names from the previous case and apply it on $\mathcal{C}^{HAREM}_{train}$. This model was then tested on $\mathcal{C}^{news}_{test}$.

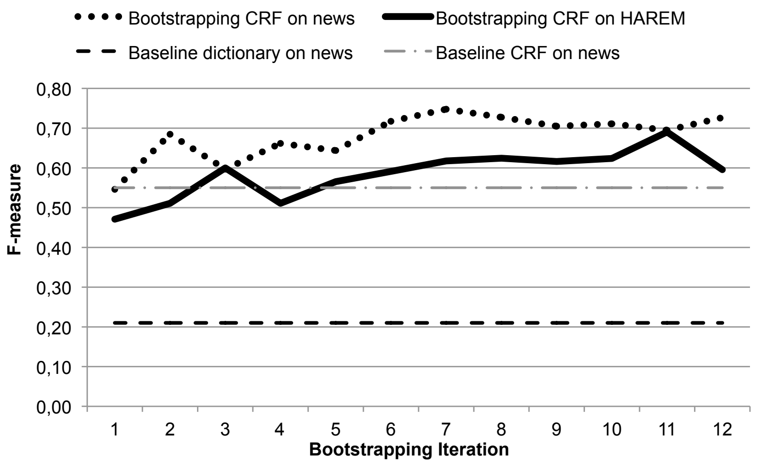

Fig. 2. Comparison of the annotation methods

- *Bootstrapping CRF on HAREM*: We built a bootstrapping CRF model based on the news dataset $\mathcal{C}^{news}$ (50,000 news) and a dictionary of names of people frequently mentioned on news. We automatically tested this model on the gold-standard corpus HAREM.
- *Bootstrapping CRF on news*: We use the same training corpus, but this time tested it on $\mathcal{C}^{news}_{test}$.

From the results obtained for the bootstrapping method (bootstrapping CRF on HAREM *versus* bootstrapping CRF on news) with both test sets, one can see they are comparable. Both results show an evolution of the F1-measure from the first to the seventh bootstrapping iteration, where the method stabilizes. Also, one can see that when training this model with recent news, testing it with an old dataset (HAREM) (see bootstrapping CRF on HAREM) does not decrease its performance (F1-measure) more than 10% when comparing it to $\mathcal{C}^{news}_{test}$ (see bootstrapping CRF on 1,000 recent news). Additionally, by comparing our baseline method based on a dictionary approach and the CRF bootstrapping method, both tested on $\mathcal{C}^{news}_{test}$, one can see that the performance achieved by the CRF bootstrapping method is much higher (73%) than that obtained with the dictionary-based method (21%). This proves, as expected, that the CRF bootstrapping method largely outperforms the dictionary-based one. Finally, considering the results obtained for the bootstrapping CRF on news and the baseline CRF on news, we are directly comparing methods trained with datasets of different sizes and age, but tested with the same data set, $\mathcal{C}^{news}_{test}$. In the first case, we used a training set with 50,000 recent news items, while in the second one, the training test was build from a gold-standard corpus, HAREM, with 14 years old. From these results one may see that the performance achieved by the baseline method is considerably lower (55%) than that obtained for the bootstrapping method, with a F1-measure of 73%. This result let us conclude that the CRF bootstrapping method, without any human effort, clearly outperforms the CRF baseline one, obtained using an - unfortunately old - human annotated corpus.

8 Conclusions and Future Work

We presented a bootstrapping approach for training a Named Entity Recognition (NER) system. We start by automatically annotating a news corpus of 50,000 news with a list of names of persons, with a dictionary-based approach. Then we built a CRF model that was tested on the previously annotated dataset, and we identified new names. These new names, together with the initial list of names, were used to re-annotate the news corpus and train a new model. This process was repeated until the system stabilized.

We were able to prove that typical gold-standard NER corpus (as HAREM) are not suitable for training NER systems for tagging recent texts, since they might not be sufficiently large and up to date. Also, we proved that our bootstrapping approach achieved a higher performance than when using CRFs trained with a limited dataset. Results have shown that our system stabilized after 7 iterations, which we consider a fast convergence, and with relatively high values of precision (83%) and recall (68%), corresponding to a F1-measure of 75%. Finally, using the CRF bootstrapping method we created a large annotated corpus of 50,000 news without any human effort and with a performance that clearly outperformed both the dictionary-based and CRF model approaches.

For future work, we may consider using sliding windows with different sizes, as this may help reducing errors found on the new names identified. Additionally, using lists of semantic categories (lists of jobs, list of organizations, etc.) could be helpful for the NER system to identify other named entities based on the context. Finally, one can think of experimenting and comparing different classification algorithms for this bootstrapping approach.

Acknowledgments. This work was partially supported by Labs Sapo UP from Portugal Telecom.

References

1. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 189–196 (1999)
2. Laboreiro, G., Sarmento, L., Teixeira, J., Oliveira, E.: Tokenizing Micro-Blogging Messages using a Text Classification Approach, AND 2010 - ACM, pp. 81–87 (2010)
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Machine Learning - International Workshop, Citeseer, pp. 282–289 (2001)
4. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 188–191. Association for Computational Linguistics (2003)

5. Mikheev, A., Moens, M., Grover, C.: Named Entity Recognition without Gazetteers. In: Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics, pp. 1–8. Association for Computational Linguistics (1999)
6. Morgan, A.a., Hirschman, L., Colosimo, M., Yeh, A.S., Colombe, J.B.: Gene name identification and normalization using a model organism database. Journal of Biomedical Informatics 37(6), 396–410 (2004)
7. Mota, C., Grishman, R.: Is this NE tagger getting old? In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), pp. 1196–1202 (2008)
8. Mota, C., Grishman, R.: Updating a name tagger using contemporary unlabeled data. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers on - ACL-IJCNLP 2009, 353 (August 2009)
9. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes 30(1), 3–26 (2007)
10. Santos, D., Seco, N., Cardoso, N., Vilela, R.: Harem: An advanced ner evaluation contest for portuguese. In: Odjik, Tapias, D. (eds.) Proceedings of LREC 2006, Genoa, pp. 22–28 (2006)
11. Sarmento, L., Pinto, A., Cabral, L.: REPENTINO A Wide-Scope Gazetteer for Entity Recognition in Portuguese. In: Computational Processing of the Portuguese Language pp. 31–40 (2006)
12. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA 2004, p. 104 (2004)
13. Vlachos, A., Gasperin, C.: Bootstrapping and evaluating named entity recognition in the biomedical domain. In: Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology, pp. 138–145. Association for Computational Linguistics, Morristown (2006)