Towards Automatic Creation of Annotations to Foster Development of Named Entity Recognizers

Emanuel Matos
IEETA, DETI, Universidade de Aveiro, Portugal

Mário Rodrigues
IEETA, ESTGA, Universidade de Aveiro, Portugal

Pedro Miguel
IEETA, DETI, Universidade de Aveiro, Portugal

António Teixeira
IEETA, DETI, Universidade de Aveiro, Portugal

Abstract

Named Entity Recognition (NER) is an essential step for many natural language processing tasks, including Information Extraction. Despite recent advances, particularly using deep learning techniques, the creation of accurate named entity recognizers continues a complex task, highly dependent on annotated data availability. To foster existence of NER systems for new domains it is crucial to obtain the required large volumes of annotated data with low or no manual labor. In this paper it is proposed a system to create the annotated data automatically, by resorting to a set of existing NERs and information sources (DBpedia). The approach was tested with documents of the Tourism domain. Distinct methods were applied for deciding the final named entities and respective tags. The results show that this approach can increase the confidence on annotations and/or augment the number of categories possible to annotate. This paper also presents examples of new NERs that can be rapidly created with the obtained annotated data. The annotated data, combined with the possibility to apply both the ensemble of NER systems and the new Gazetteer-based NERs to large corpora, create the necessary conditions to explore the recent neural deep learning state-of-art approaches to NER (ex: BERT) in domains with scarce or nonexistent data for training.

2012 ACM Subject Classification Information systems → Specialized information retrieval; Applied computing → Computers in other domains

Keywords and phrases Named Entity Recognition (NER), Automatic Annotation, Gazetteers, Tourism, Portuguese.

Digital Object Identifier 10.4230/OASIcs.SLATE.2021.1

1 Introduction

Named entities are single-word or multi-word expressions that refer to specific individuals, such as people and organizations, or denote other concrete information such as postal and e-mail addresses or dates. They are expressed using specific patterns (ex: addresses, dates, e-mails) or composed by a sequence of nouns referring a single entity as, for instance, "António Guterres" or "The Secretary-General of the United Nations" which, in 2021, refer to the same person.

Automatic recognition of these named entities is the objective of Named Entity Recognition (NER) [20], a key step used in several natural language processing (NLP) tasks. Its goal is to identify single-word or multi-word expressions in texts and classify them using a set of categories that usually include names of that usually include names of organizations and people, locations, and time references. Due to this need to both identify

and classify, some approaches split the task in two parts: first, detection of named entities; second, Named Entity Classification (NEC). The complete set of categories depends on the type of information to be detected. For instance, if you need to extract information about restaurants, the set of categories may include organizations (the name of the company), types of food, locations, service hours, among other.

The range of NLP tasks that benefit from NER include Information Extraction (IE) [29, 19], Business Intelligence [31], Forensics [2], Medicine [30] and question answering systems [6]. In general, detecting named entities prevents later steps of processing to break multi-word entities in segments that can lose the original meaning. In IE, the correct recognition of named entities is crucial, for example, to be able to detect in documents which entities are the subjects and objects. For example, a text: "Researcher Emanuel Matos at the University of Aveiro", a typical IE processing sequence is called entity detection (in this case "Emanuel Matos" and "Universidade de Aveiro"), classification (PERSON and ORGANIZATION, respectively), and extraction of relationship(s) between the identified entities.

Despite their potential and relevance for NLP, the performance of out-of-the-box NER systems is not always adequate for specific domains, preventing its wider adoption. Most systems provide methods to train and adapt them to concrete applications but this adaptation is usually limited to developers with in-depth knowledge of the area and/or having access to large amounts of annotated texts. To obtain or create annotated data that associates words or sequences of words to a type of entity is essential and in general a major problem. Another challenge is to train existing systems using these data. Not just expressing data in the format(s) accepted by existing systems but also doing so in a way non-developers feel comfortable with.

Aiming to mitigate the problems highlighted above, the following main objectives were adopted for the work presented in this paper:

1. Develop processes to simplify the creation of NER systems for new domains, starting by the creation of the needed annotated data;
2. Make NER deployment as easy as possible in order to used by non specialists, contributing to breaking existing usage barriers thus fostering wider adoption of such systems.

After this initial section establishing motivation, problem(s), relevance and main objectives, the rest of the paper is structured as follows: section 2 presents a brief view of current approaches to NER, NER systems for Portuguese, as well as information regarding sets of categories considered in previous work; section 3 presents the proposed solution, involving an ensemble of NERs; initial representative results are presented in section 4; section 5 concludes the paper with some discussion of the results and pointing to several lines of research for the future.

2 Related Work

As many other natural language processing tasks, Named Entity Recognition (NER) can be achieved either using rule based or data driven (machine learning) approaches. Some approaches are better adjusted to a specific scenario depending on the entities to detect. To detect entities with well-defined and mandatory surface patterns, such as e-mail

addresses, postal codes, telephone numbers, among other, rule-based approaches allow to code compact patterns that are quite effective to detect (almost) all entities. To detect entities whose patterns may differ in format (capital letters) without impeding human recognition of the entity, such as names of people, streets, organizations, etc., data-based approaches are best in handling these more flexible patterns, we can recognize more easily the surrounding contexts checking the highest frequency at which these entities are likely to appear. In software applications, the named entity task often uses a mix of both approaches to detect entities.

2.1 Current Approaches to NER

Considering the mentioned two main types, the most relevant approaches are:

Rule-based NER – covering both systems based in patterns and in lists (the so-called Gazetteers).

Surface Patterns – Surface patterns NER are usually implemented using regular expressions, or regex for shorthand, which are sequences of characters that specify search patterns. They are easy to implement but are quite sensitive to errors as they can break patterns and thus cause detection failures. Also, improvements come with increasing complexity as rules can interact, and their order can be meaningful, which makes more difficult to manage the rule set [9]. One of the main challenges of creating handcrafted rules is that it can be very time consuming to compile a comprehensive set of rules when target entities do not have well-defined and mandatory surface patterns. It can also be difficult to port the solution to other application contexts. For entities with well-defined surface patterns this approach is often easy to implement and provides reliable results.

Gazetteers – A gazetteer is a geographical dictionary or directory. In NLP context, the term gazetteer was further extended and now means a list of items that often include organizations, people names, alongside geographical entities such as cities or landmarks. Approaches using gazetteers can be as simple as matching candidate portions of text against the lists and having the decision just based on the existence in the list. More sophisticated methods of using gazetteers include using them as triggers in which a keyword can be used to find an entity (for example, Ms. can be used to identify that the next statement is a person), This approach is easy and can get rather good results. Unfortunately, the creation and maintenance of the lists can be a hard and tedious process and it also has problems with ambiguity (ex: gate can be an object or a name of a person depending on the context).

NERs using Machine learning – Machine learning methods are more flexible to adapt to distinct contexts provided that exists enough data about the target context. Diverse machine learning methods have been applied to NER. They can be categorized in three main branches that have distinct needs of training data: (1) supervised learning, (2) unsupervised learning and (3) reinforcement learning .

The supervised learning methods use a training set (a corpus) that was already manually labeled by experts. The unsupervised learning method consumes an untrained data set and extract patterns from it, contrary to the supervised method this one doesn't need a labeled training set. The reinforcement learning method uses agents to learn

polices [12] that can be used to label an untrained data set. These agents are trained using a reward system. Machine learning methods that were successfully applied to NER over the years include:

Hidden Markov Model (HMM) – HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobservable, hidden, states. There is another process, visible, whose behavior depends on the underlying hidden process. The goal is to learn about the hidden process by observing the visible one. NER is considered as a classification problem where the named entity class is the hidden part and the textual form the visible one. The goal is to decide which word is part of some name or not part of any name, and only one label can be assigned to a word in each context. Therefore, the model assigns to every word either one of the desired classes or a label representing none of the desired classes.

Support Vector Machines (SVM) – SVM are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Support vectors are data points that are closer to the decision boundaries that help classify the data points. In NER, SVM classifies each word, using its context, to one of the classes that represent region information and named entity's semantic class.

Conditional Random Field (CRF) – In CRF, the prediction implements dependencies between predictions. In NER, a linear chain CRF the decision if a word belongs to a named entity class, or not, depends only the word itself and on the information of the previous word. The CRF approach utilizes conditional probability to train untrained data using a trained data set.

Recursive Neural networks – Most neural-based models for NER are based on some sort of Long Short Term Memory (LSTM). LSTM are recursive neural networks in which the hidden layer updates are replaced by purpose-built memory cells. As a result, they may be better at finding and exploiting long range dependencies in data [16, 18]. Bidirectional LSTM are amongst the best performers and in these, word and character embeddings are passed through a left-to-right LSTM and a right-to-left LSTM. The outputs are combined to produce a single output layer. In the simplest method, this layer can then be directly passed onto a softmax that creates a probability distribution over all NER tags, and the most likely tag is chosen.

Transfer learning – Transfer learning reuses pre-trained models in order to perform a different task. It's very popular as it makes possible training deep neural networks with small amounts of data. In NER it was used, for example, to develop NERs for novel types of entities [28].

Transformers – Transformers [34], introduced in 2017, are a deep learning model based on the attention mechanism designed to handle sequential input data, such as natural language. Unlike RNNs, Transformers do not require data to be processed in order allowing much more parallelization and, because of that, training with huge datasets. This created the conditions for the development of pre-trained systems such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) [24]. Transformers demonstrated their superior efficiency in the recognition of named entities and in a variety of other classification tasks. A variety of state-of-the-art NER systems were developed adopting BERT for different domains and languages (ex: [32]).

2.2 Examples of Entities

Examples of entities considered over the years are presented in Table 1. The number of entities was quite reduced in initial datasets, only 4 in CONLL 2003. Despite increase the set continues to be limited, in most cases, to no more than 10.

Table 1 Entities integrating a representative selection of datasets.

Dataset	N	Entities
CADEC	5	Adverse Drug Reaction (ADR), Disease, Drug, Finding, Symptom
CONLL 2003	4	LOC (location), ORG (organization), PER (person), MISC (miscellaneous)
i2b2 Challenges	16	Username, City, Patient, ZIP, Doctor, Country, Hospital, Profession, Phone, State, Street, Medical Record, Date, Organization, Age, IDnum
MITRestaurant	8	Amenity, Cuisine, Dish, Hours, Location, Price, Rating, Restaurant Name
MUC-6	6	PER (person), LOC (location), ORG (organization), PCT (percentage), MON (month), DAT (date)
NIST-IEER	10	MUC + TIM (time), DUR (duration), CAR (cardinality), MEA (measure)
GUM	11	Person, Place, Organization, Quantity, Time, Event, Abstract, Substance, Object, Animal, Plant
NAACL 2019	6	Organization, Person, Location, Geopolitical, Facility, Vehicles
re3d	10	Document Reference, Location, Military Platform, Money, Nationality, Organisation, Person, Quantity, Temporal, Weapon

2.3 NER systems for Portuguese

Several systems can detect and tag Named Entities in texts in Portuguese, being particularly relevant the following:

Linguakit [11] - A Natural Language Processing tool containing several NLP modules, developed by ProLNat@GE Group[1], CiTIUS, University of Santiago de Compostela. It can process 4 languages: Portuguese, English, Spanish, and Galician. One of its modules is a NER tagger that classifies entities into four classes: person, organization, local or miscellaneous. The system employs lists of known entities (gazetteers) and a set of rules that allow disambiguating entities that appear in more than one list (which can be, for example, person or place).

FreeLing [23] - An open-source language analysis that, besides NER, it also implements tokenization, MSD-tagging, syntactic parsing, and lemmatization. The supported languages are Catalan, English, Galician, Italian, Portuguese, and Welsh.

NLPyPort [10] - A Python development focused on Portuguese based on NLTK. Adopting pre-existing resources and their adaptations, it provided better performance than the existing Python alternatives in tokenization, PoS Labeling, stemming and NER.

StanfordNLP [27] - Also Python based, it contains useful tools to: convert a string containing natural language text into lists of phrases and words; perform morphological analysis; obtain syntactic dependence structure (for more than 70 languages, using the Universal Dependencies formalism); perform constituent analysis and co-reference

[1] http://gramatica.usc.es/pln/

resolution, detect, and classify Named Entities. In addition, it can call the CoreNLP Java package.

AllenNLP [13] - A multilingual deep learning Python library for NLP developed by the Allen Institute for Artificial Intelligence, one of the leading research organizations of Artificial Intelligence. Using AllenNLP to develop a model is much easier than building a model by PyTorch from scratch. Not only it provides easier development but also supports the management of the experiments and its evaluation after development. AllenNLP has the feature to focus on research development. More specifically, it is possible to prototype the model quickly and makes easier to manage the experiments with a lot of different parameters. In Allennlp, NER predictions are based in pretrained models. There are many types of pretrained [2].

Representative recent examples of NER for Portuguese are briefly presented in Table 2.

Table 2 Recent representative Work in NER for Portuguese.

Ref.	Language	Domain	Technics
[25]	Multilingual (inc. Portuguese)	Webpages, Newspaper, Several genres	HMM, CRF
[22]	Brazilian Portuguese	Legal	LSTM-CRF
[26]	Portuguese	General	CRF+LG
[21]	European Portuguese	Clinical	BiLSTM-CRF
[8]	European Portuguese	Sensitive Data	Rule-based, CRF, Random Fields and BiLSTM
[32]	Portuguese	HAREM Golden collection	BERT, CRF

In [25], the authors applied multiple techniques to different datasets to create an "out-of-box" comparisons. Results revealed Stanford CoreNLP [F-measure=56.10%] as the best system and NTLK [F-measure=30.97%] as the worst.

The LeNER-Br system [22], presented in 2018, was developed for Brazilian legal documents The Paramopama training data set was used to train LSTM-CRF models with F1 scores of 97.04% and 88.82% for Legislation and judicial entities. According to the authors, the results showed the feasibility of the NERs for judicial applications.

Aiming to recognize named entities in many textual genres, including genres that differ from those for which you were trained, Pirovani and coworkers [26], in 2019, adopted a hybrid technique combining Conditional Random Fields with a Local Grammar (CRF+LG), that they adapted to various textual genres in Portuguese, according to the task of Recognition of Named Entities in Portugal at IberLEF 2019.

In 2019, Lopes and coworkers [21], addressed NER for Clinical data in Portuguese with BiLSTMs and word embeddings, the state-of-the-art model at the time obtaining F1-scores slightly higher than 80% and equivalent results for both Precision and Recall. The data set was pre-processed by NLPPort [10] and processed by BiLSTM-CRF and CRF for comparison. BiLSTM was superior in all comparisons for "In-Domain" models.

[2] In https://github.com/allenai/allennlp-hub/blob/master/allennlp_hub/pretrained/allennlp_pretrained.py some pretrained models can be found.

In a work published in 2020, NER was applied to sensitive data Discovery in Portuguese [8], being used in the process of protecting sensitive data. A component was developed to extract and classify sensitive data, from unstructured text information in European Portuguese combining several techniques (lexical rules, machine learning algorithms and neural networks). The rule-based approaches were used for a set of specific classes (ex: NumIdentificacaoCivil). For the other classes of entities, CRF, Random Fields and BiLSTM were used. The datasets used for training and testing were HAREM Golden Collection, SIGARRA News Corpus and DataSense NER Corpus. This validation was carried out with the project's stakeholders. Although the global results with the use of lexicon-based models were inferior to the current state of the art, for the TEMPO and VALOR entities the results were superior to those obtained with other methodologies.

A first use of BERT in NER for Portuguese appeared in 2020 [32]. In this work, Portuguese BERT models were trained and a BERT-CRF architecture was employed, combining transfer capabilities of BERT with the structured CRF forecasts. Pre-training of BERT used brWac corpus, which contains 2.68 billion tokens from 3.53 million documents and is the largest Portuguese open corpus to date. Training of the NER model was done with First HAREM. Tests with MiniHAREM dataset surpassed the previous state art (BiLSTM-CRF+FlairBBP), despite being trained with much less data.

From the selected representatives of recent developments of NER for Portuguese it is clear that: (1) the target domains are quite diverse, being different for all the selected references; (2) the set of techniques applied is also diverse, often being adopted Machine Learning methods and tools, including more recent ones such as LSTM and BERT; (3) NER for Portuguese continues to be a relevant and active area, with developments aligned with state-of-the-art evolution; (4) there are signs of expansion of application areas/domains.

3 Proposed solution - NER without annotated data using an Ensemble of NERs

To provide annotated corpora for development of new NER systems we propose to explore the combined use of several existing NER systems and resources capable of providing information regarding entities and their types, for example DBpedia [3, 4].

3.1 Overview of the system

The complete system derives directly of the main idea of the proposal - explore an ensemble of NERs - and consists of 4 phases, as represented in Fig. 1, that will be described in the following sections.

3.2 Phase 1: Obtaining documents for the domain

For an initial proof-of-concept related to tourism were selected. A set of sources were selected manually, and the text scrapped using the Scrapy library for retrieving documents and BeautifulSoup library for getting document data. One of the main sources selected was Wikivoyage [35]. Based on the tool's ease of use concept, the manual extraction option, without advanced features, was implemented to adjust expectations regarding the development of the work and evaluate the capture properties of possible entities without advanced techniques. The option for Tourism domain resulted from our previous work

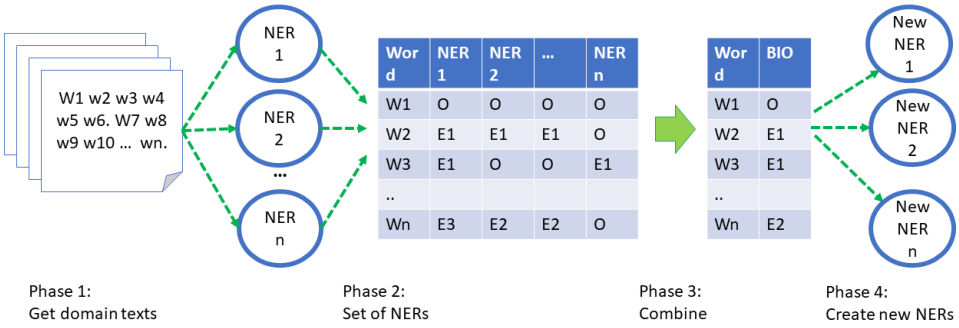

Figure 1 Overview of the proposed process for automatic tagging of named entities by using and Ensemble of NERs, showing its 4 phases.

in this domain [33], selected by the high potential of automatic information extraction to provide relevant information to several domain stakeholders (e.g., hotel managers).

3.3 Phase 2: Application of an Ensemble of NERs

From the NERs capable of processing texts in Portuguese, a subset was selected covering the main types. As representative of rule based and gazetteer approaches was selected Linguakit [11]. Allen NLP [13] was selected as a representative of a machine learning approach. As 3rd NER of the ensemble a system based in DBpedia was selected.

For integration of Linguakit in the system a simple Python wrapper script was created to invoke the Perl implementation and load and process its output.

A Python script was also created for the NER based in Allen NLP, but in this case using allennlp Python library [1] `Predictor` class. Publicly available modules were used [3].

The DBpedia-based NER was developed by the authors in Python using the SPARQL-Wrapper library [17] to make SPARQL queries to DBpedia endpoint. The overall process is presented as pseudo-code in Algorithm 1.

All 3 NERs save their output as CSV files to allow access to all intermediate processing results and simple communication with the following phases.

3.4 Phase 3: Combination of the outputs of the Ensemble of NERs

The Phase 3 goal is to create new tag candidates from the tags produced by the 3 NERs. For the initial proof-of-concept version, two decision strategies were implemented: **Winner Take All** (WTA) which assigns to each word the most common tag, augmenting the confidence in the tags; **Entity detection** (ENTITY), that tags words as only Entity or not, producing annotated data useful to train entity detectors (not including classification of the entity).

[3] https://storage.googleapis.com/allennlp-public-models/ner-model-2020.02.10.tar.gz

Algorithm 1 DBpedia-based NER

```
input  : text file, maxl =max length of word sequences
output: two column dataframe with word and tag
read cache from file;
/* read text, tokenize and create dataframe with 2 columns (1st with
   words, 2nd with tag initialized to "O") and words list          */
dataframe , words_list ← init_from_file(filename);
for l ← 1 to maxl do
    for pos ← 0 to len(words_list) − l + 1 do
        word_seq ← words_list [pos ]+ .. + words_list [pos +l-1] ;
        dbpedia_result =query_DBpedia (word_seq) ;
        selected_result = postprocess (dbpedia_result) ;
        tag =get_super_class (selected_result) ;
        add tag to dataframe [pos ];
        add (word_seq, tag) to cache;
    end
end
save dataframe to CSV file;
save cache to JSON file;
```

3.5 Phase 4: Exploration of the results

The obtained annotated data from previous phase can be explored in several ways: analyzed in terms of agreement among NERs; evaluated against manually annotated data; Named Entities annotated extracted and used to create Gazetteer-based NERs; used to supervised training of data-driven machine learning NERs.

4 Results

In this section are reported the initial results obtained with the proposed system, starting with representative examples of obtained annotations based on combining the output of the 3 NERs.

4.1 Examples of obtained annotations

Illustrations of the obtained results with the ENTITY and WTA strategies for a text example are presented in Fig. 2.

The results obtained with a text example are presented in Fig. 2 for the strategies ENTITY and WTA. By combining the entities marked by the 3 annotators (at bottom right) it is possible to detect entities such as "Centro de Portugal" or "Península Ibérica". WTA strategy achieves a lower recall caused by Linguakit and Allen NERs (almost) only detecting PERSON and LOCATION, with large disagreements among them. AllenNLP seems to favor PERSON and classifies, for instance, “Praça Barão da Batalha” as PERSON while Linguakit correctly classifies it as LOCATION. The major contributor for higher

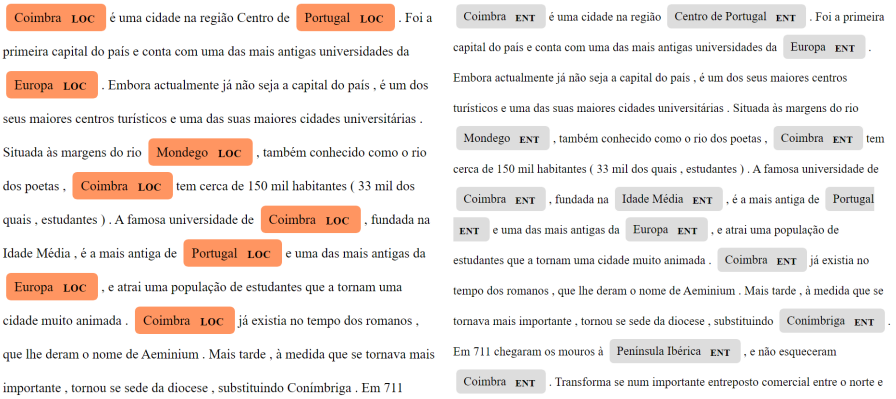

Figure 2 NER Tagging of text from https://pt.wikivoyage.org/wiki/Coimbra. WTA results at left, ENTITY results at right.

recall using the ENTITY strategy is the DBPedia-based NER, frequently contributing with the tags necessary to mark text segments as ENTITY.

4.2 NERs output analysis

The outputs of the 3 NERs were compared with the outputs of the Ensemble of NERs (WTA and ENTITY) for a better understanding on their behavior, and thus complementing the qualitative analysis of the annotation results. The analysis started by looking both confusion matrices and computing a simple similarity measure (Jaccard similarity calculated not considering the cases where "O" BIO tag was assigned by the NERs being compared). Examples of confusion matrices obtained for the same text used in Fig. 2 are presented in Fig. 3, and the results for Jaccard similarity in a set of 39 texts are presented in Table 3.

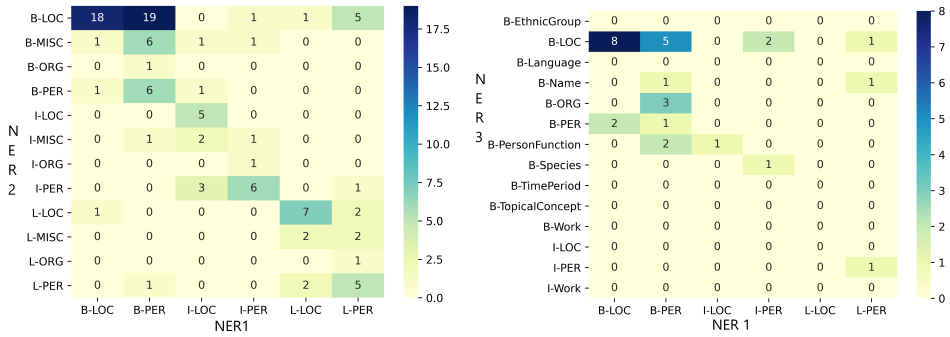

Figure 3 Confusion matrices for the tags assigned by the 3 NERs to the text https://pt.wikivoyage.org/wiki/Amarante: At left, AllenNLP (NER1) vs Linguakit (NER2), right, AllenNLP vs DBpedia NER (NER3).

The confusion matrices and the Jaccard similarity show that: (1) the 3 systems produce quite dissimilar results, being the higher Jaccard similarity only 0.422 (for Linguakit versus DBPedia-based NERs); (2) many of the entities marked as LOCATION by Linguakit and DBPedia are tagged as PERSON by AllenNLP; (3) DBPedia NER detects a richer set of entities than the other two; (4) WTA tags are more similar to the DBPedia-based NER

Table 3 Jaccard similarity among NER tags for a set of 30 Wikivoyage texts. Individual results for a sample of the texts is presented as well the average (M) and standard deviation for the complete set of texts (at the bottom of the table).

	NER1 NER2	NER3	ENT	WTA	NER2 NER3	ENT	WTA	NER3 ENT	WTA	WTA ENT
Coimbra	0.223	0.180	0.613	0.429	0.387	0.709	0.705	0.256	0.690	0.397
Luanda	0.480	0.696	0.667	0.804	0.517	0.753	0.830	0.333	0.792	0.580
Roma	0.318	0.667	0.705	0.667	1.000	0.795	0.733	0.034	1.000	0.341
M	0.346	0.336	0.632	0.579	0.422	0.631	0.649	0.431	0.750	0.512
STD	0.201	0.214	0.105	0.162	0.219	0.193	0.146	0.253	0.170	0.140

(M=0.75), demonstrating its usefulness; (5) the maximum Jaccard similarity for ENTITY is just 0.63, for the comparisons with AllenNLP and Linguakit NERs, which can be interpreted as the contribution of the several NERs to ENTITY annotations.

4.3 New NERs

As stressed before, to train data-driven NERs such as the Hybrid LSTM-CRF are necessary substantial amounts of tagged data. To contribute to the existence of the needed data it is crucial to tag as much text as possible and as fast as possible. For that, two new NER taggers were created to profit from the combined results of the ensemble of NERs: one is a fast implementation of a Gazetteer and the other one resorts to Finite State Transducers (FSTs).

The first, a **simple Gazetteer NER**, was developed by the authors using Bloom Filters [5] to store information about the entities. Separate filters were used for each number of words in an entity, i.e., single word entities in one filter, 2-word entities in another, etc.

The second annotator, an FST-based NER, was implemented using the library `pynini` [14] and adapting one of the examples in [15]. Briefly, the process consists of:

1. Construct transducers which insert the left and right tags.
2. Build a substitution transducer by concatenating these transducers to the left and right of the deterministic Finite State Automata (FSA) we are trying to match. Its definition needs: the left context for the substitution; the right context; and an FSA representing the alphabet of characters we expect to find in the input.
3. Create the substitution transducer which passes through any part of a string which does not match.
4. Apply to a string. The simplest way to do so is to compose a string, apply it on the transducer, and then convert the resulting path back into a string with:

```
output = pynini.compose(input_string, rewrite).string()
```

All word sequences marked as ENTITY were extracted from a corpus and used as lists. As an initial experiment, the corpus consisted of the 36 texts from Wikivoyage used to calculate the Jaccard similarities. The process resulted in a list with 379 entries, 176 with 1 word, 83 of 2 words, 59 of 3 words, etc. Both Gazetteer-based NERs were tested in texts

outside the corpus used to extract the list. Fig. 4 presents the results obtained for a text sample.

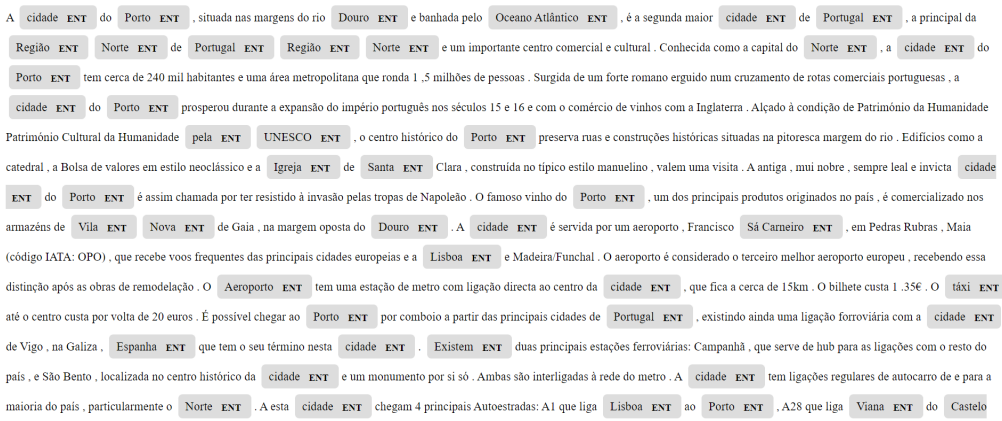

Figure 4 Example of text (about Porto) annotated with the Gazetteer-based NER created with lists derived from the automatic annotations obtained with the proposed system.

The Figure shows clearly that the Gazetteer-based NER could recognize several named entities. A major problem is clear, the approach is not capable of handling continuous or close parts of an entity (ex: "cidade do Porto") and fails to recognize several entities (ex: "Vigo").

5 Conclusion

Aiming at making possible and simple creation of new NERs for domains without annotated data available, the problem of automatic annotation of Named Entities is addressed in this paper, being proposed the combined used of an ensemble of NER systems. The emphasis was placed on creating annotated data based on several NERs. A first proof-of-concept of the proposed method was implemented with 2 existing NER systems (AllenNLP and Linguakit) and a DBpedia-based NER developed by the authors.

The main contributions of this paper are: (1) the method based on an ensemble of NERs and combination of their outputs; (2) start of a new annotated corpus for experiments in NER for Tourism domain; (3) fast new Gazetteer-based NERs.

The initial results revealed that combining the outputs of the NERs ensemble has potential to both extend the set of entities (limited in systems as AllenNLP) and contribute to reduce the difficulties distinguishing some entities (ex. AllenNLP and Linguakit present high degree of disagreement for PERSON and LOCATION).

With the combination strategy providing a 1-class annotation, designated as ENTITY, the proposed system could detect an interesting set of Named Entities in texts related to Tourism that were used to create initial versions of new Gazetteer-based NERs. The other combination strategy, WTA, is useful to create higher confidence annotations for a set of entities, such as those handled by Linguakit.

5.1 Future work

As the presented work is both a first step and an initial proof-of-concept, the future work is rich and covering distinct lines of research, the most relevant being:

- **Improve the NERs integrating the Ensemble:** Improve the NERs integrating the ensemble: Particularly improve the performance of DBpedia-based NER, both in augmenting its speed (ex: with a local Virtuoso server) and improving the decision of Tags to keep from the DBpedia query results.
- **Addition of NERs to the Ensemble:** Add other NERs to the ensemble: For example, NERs specialized in using other sources of information (ex: YAGO and Wikipedia) and ontologies.
- **Use tagged data to train NERs:** train systems based on BERT [7] for instance, both for ENTITY/NO-ENTITY tagging and to classify with tags adequate to the domain.
- **Improve the process of NERs output combination**.
- **Test the system in new domains**.
- **Integrate the newly obtained NERs in an Information Extraction pipeline**.

References

1. Allen NLP - An Apache 2.0 NLP research library, built on PyTorch, for developing state-of-the-art deep learning models on a wide variety of linguistic tasks. URL: https://github.com/allenai/allennlp.
2. Flora Amato, Giovanni Cozzolino, Vincenzo Moscato, and Francesco Moscato. Analyse digital forensic evidences through a semantic-based methodology and NLP techniques. *Future Generation Computer Systems*, 98:297–307, 2019.
3. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 2007.
4. Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165, 2009.
5. Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.
6. Hiral Desai, Mohammed Firdos Alam Sheikh, and Satyendra K Sharma. Multi-purposed question answer generator with natural language processing. In *Emerging Trends in Expert Applications and Security*, pages 139–145. Springer, 2019.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
8. Mariana Dias, João Boné, João C Ferreira, Ricardo Ribeiro, and Rui Maia. Named entity recognition for sensitive data discovery in portuguese. *Applied Sciences*, 10(7):2303, 2020.
9. Tobias Ek, Camilla Kirkegaard, Håkan Jonsson, and Pierre Nugues. Named entity recognition for short text messages. *Procedia - Social and Behavioral Sciences*, 27:178–187, 2011.
10. João Ferreira, Hugo Gonçalo Oliveira, and Ricardo Rodrigues. Improving NLTK for processing portuguese. In *8th Symposium on Languages, Applications and Technologies (SLATE)*, 2019.
11. Pablo Gamallo and Marcos Garcia. Linguakit: uma ferramenta multilingue para a análise linguística e a extração de informação. *Linguamática*, 9(1):19–28, 2017.
12. Pablo Gamallo, Marcos Garcia, César Piñeiro, Rodrigo Martínez-Castaño, and Juan Pichel. Linguakit: A big data-based multilingual tool for linguistic analysis and information extraction. 10 2018. doi:10.1109/SNAMS.2018.8554689.

13 Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform, 2018. arXiv:1803.07640.

14 Kyle Gorman. Pynini: A python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, 2016.

15 Kyle Gorman and Richard Sproat. How to get superior text processing in python with pynini, o'reilly ideas blog, 2016. accessed 22/04/2021. URL: https://www.oreilly.com/content/how-to-get-superior-text-processing-in-python-with-pynini/.

16 Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

17 Ivan Herman, Sergio Fernández, Carlos Tejo Alonso, and Alexey Zakhlestin. Sparql endpoint interface to python. URL: https://sparqlwrapper.readthedocs.io/en/latest/main.html.

18 Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

19 Daniel Jurafsky and James H. Martin. Information extraction. In *Speech and Language Processing (3rd ed. draft)*, chapter 17. 2020.

20 Daniel Jurafsky and James H. Martin. Sequence labeling for parts of speech and named entities. In *Speech and Language Processing (3rd ed. draft)*, chapter 8. 2020.

21 Fábio Lopes, César Teixeira, and Hugo Gonçalo Oliveira. Contributions to clinical named entity recognition in portuguese. In *Proc. 18th BioNLP Workshop and Shared Task*, 2019.

22 Pedro H. Luz de Araujo, Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In *PROPOR*, LNCS. Springer, 2018.

23 Lluís Padró. Analizadores Multilingües en Freeling. *Linguamática*, 3(2):13–20, 2011. URL: https://linguamatica.com/index.php/linguamatica/article/view/115.

24 A. Patel and A.U. Arasanipalai. *Applied Natural Language Processing in the Enterprise: Teaching Machines to Read, Write, and Understand*. O'Reilly Media, Incorporated, 2021.

25 André Pires, José Devezas, and Sérgio Nunes. Benchmarking named entity recognition tools for portuguese. *Proceedings of the Ninth INForum: Simpósio de Informática*, pages 111–121, 2017.

26 Juliana PC Pirovani, James Alves, Marcos Spalenza, Wesley Silva, Cristiano da Silveira Colombo, and Elias Oliveira. Adapting NER (CRF+ LG) for many textual genres. In *IberLEF@ SEPLN*, pages 421–433, 2019.

27 Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. Universal dependency parsing from scratch. In *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, 2018.

28 Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, and Timothy Baldwin. Named entity recognition for novel types by transfer learning, 2016. arXiv:1610.09914.

29 Mário Rodrigues and António Teixeira. *Advanced applications of natural language processing for performing information extraction*. Springer, 2015.

30 Antonio Moreno Sandoval, Julia Díaz, Leonardo Campillos Llanos, and Teófilo Redondo. Biomedical term extraction: NLP techniques in computational medicine. *IJIMAI*, 5(4), 2019.

31 K. Sintoris and K. Vergidis. Extracting business process models using natural language processing (nlp) techniques. In *Proc. Conf, on Business Informatics (CBI)*, pages 135–139, 2017.

32 Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Portuguese Named Entity Recognition using BERT-CRF, 2020. arXiv:1909.10649.

33 António Teixeira, Pedro Miguel, Mário Rodrigues, José Casimiro Pereira, and Marlene Amorim. From web to persons - providing useful information on hotels combining information extraction and natural language generation. In *Proc. IberSpeech*, Lisbon, November 2016.

34 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. arXiv:1706.03762.

474 35 Wikivoyage. URL: https://pt.wikivoyage.org/.