Identifying named entity from researcher curricula

Rodrigo Gonçalves, Carina F. Dorneles

[1]Departamento de Informática e Estatística - INE
Universidade Federal de Santa Catarina - UFSC/Florianópolis

rodrigo.g@ufsc.br, carina.dorneles@ufsc.br

Abstract. *NER (Named Entity Recognition) is an essential task in recognizing real-world entities scattered in a document. The task has been beneficial for detecting people, institutions, and places. In a researcher's curriculum repository, a NER process can be beneficial for understanding the associated context of a given document. For example, it could be possible to identify which persons/institutions are present in a given researcher's curriculum. This process is fundamental to identifying experts to work on a project or collaboration among researchers. In this paper, we evaluate entity extraction methods' effectiveness for identifying entities from scientific publications, including vocabulary-based and model-based methods. We describe an analysis of existing NER tools while proposing a procedure to apply NER identification over curricula from the Brazilian Lattes Curricula platform.*

1. Introduction

Named Entities Recognition (NER) is a sub-task of information extraction that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, time, date, and money [Yadav and Bethard 2018, Jurafsky and Martin 2018, Angeli et al. 2015]. Traditionally, a Named Entity (NE) is an entity associated with a set of rigid expressions that identify the same real-world entity. However, the definition was loosened to include other entities, such as date. For example, "June" may refer to the 6th Month of any year, while "June 2019" refers specifically to the 6th Month of 2019 (a rigid designator).

The term "NER" was coined around 1996 and has been widely used in the NLP (Natural Language Processing) area. NERs are located and classified in a process called NERC - *Named Entity Recognition and Classification*. Earlier systems required hand-crafted rules, but nowadays, most systems use machine learning techniques [Nadeau and Sekine 2007]. Most systems developed for NERC focus on the English language. Some current tools, such as spaCy (`https://spacy.io/`) have NERC models for several languages. Besides language, the domain associated with the texts where NERC is applied plays an important role. More promising results are obtained with models trained explicitly for a given domain. The reason for that is that text and document-related features are used for NERC model training [Nadeau and Sekine 2007].

Since NERC models are trained based on ML techniques, the three basic approaches can be used: supervised, semi-supervised, and unsupervised learning. In the *supervised* approach, a training corpus of documents is manually tagged and used to train the Model. Among the techniques used in *supervised* approaches, we can cite [Yadav and Bethard 2018]: *Hiden Markov Models* - each word (either of interest or not)

receives a tag, and the models find the most likely sequences of tags of a given list of words; *Maximum Entropy-based Model* - a discriminative model that, given a previous classification/training data, tries to maximize the entropy of a given text NER classification, trying to match the training data level; *SVM Based Models* - SVM models work on the idea that a plane can be established between positive and negative examples of NEs; thus, it can be trained to identify and classify NEs; *CRF Based Models* - Conditional Random Fields are based on pattern recognition that, using training data, is capable of establishing a model to determine the probability of a given sequence of elements to be a desired class of NE.

An interesting application for NERC is to extract and use the NEs present in documents to contextualize them. For example, given a person's curriculum, listing all professional activities, one may be interested in finding which activities are related to a specific aspect. This aspect may be a given university or a particular technology - even another person cited in these activities. Context can help, for example, to locate a person adequate for teaching a given undergraduate class or participating in a research project.

In this work, we analyze the application of some NERC tools aiming to extract NER from researchers' curricula. Four tools were elected: spaCy, OpenNLP, Google Natural Language API (`https://cloud.google.com/natural-language/docs/analyzing-entities`) and the Google AutoML API (`https://cloud.google.com/natural-language/automl/entity-analysis/docs/`).

2. NERC tools

In this work, we focused on four tools for NERC: **spaCy**, which is a Natural Language Processing library that includes NERC features - it provides NER support for several languages: English, German, Spanish, Portuguese, French, Italian, and Dutch, and there are trained models available for those languages, ready to use; **OpenNLP**, which is a Natural Language Processing library developed by the Apache Foundation and, among several components, includes a NER for person's names, locations, and organizations; **Google Natural Language API** that is an API provided by Google for online NER, and that can identify several kinds of NE (such as Person, Location, Work of Art, Common (Misc), etc.); and **Google AutoML API** that allows training new NER models, and that can identify any category of NER as long as enough training material is provided, and currently, it only supports the English language.

3. Lattes Case Study

As introduced earlier, NEs can help contextualize information, such as information in a given researcher's curricula. Our work aims at developing a case study applying NER over a set of researchers' curricula. Our main issues are: (i) if existing NERC tools can be applied over researcher curricula; and (ii) if they require additional training to provide acceptable results, is it within a reasonable effort to justify the usage? Based on the initial results, our study case included establishing a bootstrap-based approach to train and optimize NERC application over researcher curricula.

Given a person's curriculum, describing his/her profile and, if included, all professional activities (previous and current), one may be interested in finding which activities are related to a particular aspect/entity. This aspect may be a given university or a specific

Table	Column	Type	Description
lattes_ner	id	integer	Primary key (sequencial)
	lattes_id	string	Lattes platform curricula id
	pt	text	Lattes curricula extracted data
	ner_pt_manual	text	Lattes curricula extracted data, manually tagged
	iteraction	integer	Training process iteraction to use the manually tagged data
lattes_ner_iteraction	id	integer	Primary key (sequencial)
	lattes_id	string	Lattes platform curricula id
	ner_pt	text	Lattes curricula extracted data, manually tagged
	iteraction	integer	Training process iteraction in which this record was generated

Table 1. Support database tables

technology - even if there is a citation to another person in the activities. Such information can help, for example, to locate a person adequate for teaching a given undergraduate class or participating in a research project.

The curricula used for the tests were obtained from the *Lattes Platform* [1]. As a Lattes curriculum is composed of several sections, we analyzed these sections and elected those that could be considered for analysis and NERC. They include a profile summary (a brief description of the author's studies and research, written by himself), academic training, professional experience, and research projects. We discarded a large part of the curriculum that refers to academic production (articles, participation in events, etc.) since they do not provide relevant NEs according to our intent.

3.1. Data preparation

To prepare the data, first, we limited the number of curricula to be analyzed. Since there is no golden standard/test data to be used as a reference, there was a need to manually tag the data to train and calculate the NERC tool output quality. Thus, we elected 50 (fifty) curricula from Informatics and Statistics Department at UFSC. Although they are from the same general area, they work on several subjects and have distinguished backgrounds.

Over these 50 curricula, we analyzed the available data and decided to use only data in Portuguese. Most researchers keep only their Portuguese curricula up-to-date, or their English counterparts are incomplete. We extracted their profile description from the curricula, a free text that allows several lines and, in real terms, unlimited data. The extracted data was stored in a relational database, which is composed of two tables, `latter_ner` and `lattes_ner_iteraction`. Their schemas are described in Table 1. Table `lattes_ner` contains the extracted data from researcher curricula and table `lattes_ner_iteraction` contains the data generated using the NERC tool.

3.2. Training process

To the best of our knowledge, there is no standard to test the NERC tool application over curricula data in Portuguese, so we could not directly evaluate the quality of the existing tool. We had to define a procedure that allowed testing and training NERC tools while evaluating if their results were satisfactory. At the same time, we could not consider manually tagging large amounts of data. Thus, we developed a method to accomplish our needs based on the *semi-supervised* learning idea used by some NERC works.

[1] http://lattes.cnpq.br/web/plataforma-lattes

Our method is based on the concept of *bootstrap* data and works as follows: (i) the NERC tool is applied over the original data, generating the tagged curricula data. Such data is stored in the `lattes_ner_iteration` table, setting the `iteration` field as one; (ii) a sample of the tagged data is randomly selected, and the tagged results are reviewed and manually corrected. The data is stored in the `lattes_ner` table, in the field `ner_pt_manual` and the iteration field is set to one as well; (iii) the NERC is reapplied over the original data, but now the manually tagged data is fed as training data before this process. The resulting tagged data is saved in the `lattes_ner _iteration` table, with the iteration field set to two; (iv) the manually labeling is applied again, now on other sample data that has not been manually tagged previously. It is stored in the same form as described earlier, only with the iteration value of two; (v) a new round of training and classifying is executed, but now using all available training data (from iterations 1 and 2); and (vi) this process can be repeated for more iterations if needed.

3.3. Evaluation process

In each iteration, new data is generated and manually fixed. A breaking point is defined to indicate that new training does not provide significant improvement in the NERC process. However, to evaluate if the additional training data is improving the results and define the breaking point, we defined the following evaluation process: (i) to execute at least two iterations recorded in the database (to allow the evaluation); (ii) to compare the output of a given iteration with the manually tagged data from the next iteration, for a given researcher curriculum. Thus, if the curricula of researcher A were manually tagged in iteration 2, we compared the output of iteration 1 and iteration 2 with the manually tagged data. Thus we can compare each output with the *golden standard* in terms of missing, correct, and incorrect NEs. Using recall and precision metrics, we analyze the quality of each iteration output and determine when there is no need to train more, either due to minor improvement or over-fitting.

4. Experimental Evaluation

In this section, we describe our preliminary test process. Considering the tools described in Section 2, given our data being in Portuguese, we elected the spaCy tool, since it has a Portuguese language model, for our preliminary tests. In further tests, additional tools will be tested as well. spaCy model, although built over news data, provides a better starting point than a blank or English-based model.

From the set of 50 curricula, we used 10% for training data in each iteration and proceeded with three iterations, i.e., we elected 15 curricula, divided into three sets, as training/test data (depending on the iteration). In the first iteration, all 15 curricula were used as test data and none as training data. In iteration 2, five curricula were used as training data and ten as test data. In the last iteration, 15 curricula were used as training data and five as test data. The curricula were picked randomly among the 50 curricula.

Table 2 contains the experiments results. The columns have the following semantics: (i)**Total NEs** - the total number of NEs in the test data used; (ii) Correctly found - how many of the NEs in the generated tagging in the iteration were correct;(iii) **Missing** - the number of NEs that were present in the training data but not in the generated data; (iv) **Incorrectly found** - NEs proposed by the automated tagging that are incorrect;(v)

Iteration	#NERs	Correct	Missing	Incorrect	Precision	Recall	F-score
1	322	80	242	263	**23.32%**	**24.84%**	**0.24**
2	198	129	69	50	**72.07%**	**65.15%**	**0.68**
3	116	87	29	24	**78.38%**	**75.00%**	**0.76**

Table 2. Experiment results with spaCy

Iteration	#NERs	Correct	Missing	Incorrect	Precision	Recall	F-score
1	322	141	181	312	**31.13%**	**43.79%**	**0.36**
2	198	49	149	88	**35.77%**	**24.75%**	**0.29**
3	116	36	80	56	**39.13%**	**31.03%**	**0.35**

Table 3. Experiment results with OpenNLP

Precision - Calculated as (Correctly found / (Correctly found + Incorrectly found)); (vi) **Recall** - Calculated as (Correctly found / (Missing + Correctly found)), and; (vi)**F-score** - Calculated based on the Precision and Recall.

As we can see in the initial results, the spaCy tool, using a starting point in its Portuguese dictionary, could not provide good results for NERC in curricula data. However, once trained, it has shown considerable improvements, where, with at most 20% training data, it was capable of approaching a recall of 75% with a 78% precision, thus having an F-score of 0.76. We were also able to execute the same training and testing procedure with OpenNLP. It does not contain, per-default, a Portuguese language model, but we were able to adapt an existing training data for our purposes. This data, originally from the HAREM project [2], was adapted to the OpenNLP format by André Pires [3] as part of his Master dissertation. With the initial training model, we included the additional training data from our data. Unlike spaCy, the whole model had to be retrained - we were not able to directly extend an existing model. Table 3 show our results.

As shown by the initial results, the OpenNLP tool, using as a starting point in the HAREM dataset, could not provide good results for NERC in curricula data. Even with training data, it didn't show considerable improvements. On the contrary - although precision has steadily increased, recall and the F-score fluctuated in the three iterations. We consider that these results are due to the limited Portuguese Language model used. With a larger training model, there is a possibility of having results at least near those obtained with spaCy.

4.1. Additional tests

In order to extend our analysis to other tools besides free ones, we tested the Google Natural Language API. Since this API does not allow training, we executed a single interaction with the prepared curricula, obtaining the result shown in Table 5, for curricula in Portuguese. To demonstrate how tools that do not natively support the Portuguese language could be used, we translated the researcher curricula to English using the Google Translation API[4]. Over the translated curricula we applied a manual tagging process similar to

[2]https://www.linguateca.pt/aval_conjunta/HAREM/harem_ing.html
[3]https://github.com/arop/ner-re-pt/wiki/OpenNLP
[4]https://cloud.google.com/translate/docs/

Iteration	#NERs	Correct	Missing	Incorrect	Precision	Recall	F-score
1	313	118	195	585	**16.78%**	**37.69%**	**0.23**

Table 4. Experiment results with Google Natural Language API

Iteration	#NERs	Correct	Missing	Incorrect	Precision	Recall	F-score
1	113	76	28	66	**66.67%**	**67.26%**	**0.66**

Table 5. Experiment results with Google AutoML

that applied for the curricula in Portuguese. The resulting data was used as input for the Google AutoML API to train a model to identify NEs in researcher curricula.

Google AutoML requires an extensive number of samples to train (at least a hundred), so we considered in this test only two kinds of NE: Organization (ORG) and Miscelaneous (MISC). Person (PER) and Location (LOC) were ignored due to their reduced number in the sample data. Table 5 shows the trained model results, as returned from the Google AutoML API itself. The AutoML model achieved results near, but still worse than the spaCy. spaCy is also more pratical since it can extend and existing model, instead of training a new model as in AutoML. Training a new model is a time consuming and tedious activity. On AutoML benefit we highlight that it allows training new kinds of NEs, while spaCy is limited to the categories defined in its base model.

5. Conclusion and future work

In this work, we provided a brief analysis of the application of NERC in the context of researcher curricula. Due to data scarcity to train the NERC models, we proposed and validated a method, based on the *bootstrap* concept, to iteratively train and test the NERC models with proportionally reduced manual efforts. Using this method, we analyzed and classified curricula from a sample set of 50 researchers, obtaining results of 78% precision and 75% recall using the spaCy tool. The developed code and test data used are available[5]. Based on this starting point, future work includes testing with larger data sets and building a reference test set that can be used to train other NER tools.

References

Angeli, G., Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Assoc. for Comput. Linguistics (ACL)*.

Jurafsky, D. and Martin, J. H. (2018). *Speech and Language Processing (2rd Edition - draft)*. Upper Saddle River, NJ, USA.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Yadav, V. and Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proc. of the 27th International Conf. on Comput. Linguistics*.

[5]`https://github.com/keitarobr`