Hybrid Statistical and Attentive Deep Neural Approach for Named Entity Recognition in Historical Newspapers

Ghaith Dekhili and Fatiha Sadat

University of Quebec in Montreal, 201 President Kennedy avenue, H2X 3Y7
Montreal, Quebec, Canada.
dekhili.ghaith@courrier.uqam.ca, sadat.fatiha@uqam.ca

Abstract. Neural networks-based models have proved their efficiency on Named Entities Recognition, one of the well-known NLP task. Besides, attention mechanism has become an integral part of compelling sequence modeling and transduction models on various tasks. This technique allows context representation in a sequence by taking into consideration neighboring words.
In this study, we propose an architecture that involves BiLSTM layers combined with a CRF layer and an attention layer in between. This was augmented with pre-trained contextualized word embeddings and dropout layers. Moreover, apart from using word representations, we use character-based representations, extracted by CNN layers, to capture morphological and orthographic information.
Our experiments show an improvement in the overall performance. We notice that our attentive neural model augmented with contextualized word embeddings gives higher scores compared to our baselines.
To the best of our knowledge, there is no study which combines the application of attention mechanism and contextualized word embeddings on NER and historical newspapers.

Keywords: Deep Neural Networks · Attention Mechanism · Contextualized Word Embeddings · Character Embeddings

1 Introduction

This work is done as part of the HIPE (Identifying Historical People, Places and other Entities) shared task, "organised as a CLEF 2020 evaluation Lab and dedicated to the evaluation of named entity processing on historical newspapers in French, German and English" [11]. The shared task is organized as part of "impresso Media Monitoring of the Past", a project focused on information extraction in historical newspapers. [1]

[1] https://impresso-project.ch/

Named Entity Recognition and Classification (NERC) is a sub-task of information extraction and Natural Language Processing (NLP). It consists on identifying certain textual objects such as names of persons, organizations and places.

Early NER systems were based on handcrafted rules, lexicons, orthographic features and external knowledge resources. This was followed by "feature engineering based NER systems" and machine learning [25]. Starting with [7], neural networks based systems with a minimum of feature engineering have become popular. Such models are interesting because they typically do not need domain specific resources like in earlier systems, and are thus qualified to be more domain independent. Many neural architectures have been introduced, most of them based on some form of Recurrent Neural Networks (RNN) over characters, sub-words and/or word embeddings [30].

NER systems based on knowledge do not need labeled data as they rely on lexicon resources and domain specific knowledge. These systems perform well in cases where the lexicon is exhaustive, but fail when the information does not exist in domain dictionaries [30]. A second drawback of these systems is that they require domain experts to construct and maintain the knowledge resources. Finally, these systems can be used only on domains and languages for which they were designed, because of the specific features they had learned during training [12].

Supervised machine learning models learn how to make predictions during training on couples of inputs and their expected outputs, and can be used in place of handcrafted rules [30].

NER task becomes more challenging when applied on historical and cultural heritage collections. On the one side, inputs can be extremely noisy, with errors which differ from the ones in tweet misspellings or speech transcription hesitations [22, 5, 28]. On the other side, the language that we have is mostly of earlier stage, "which renders usual external and internal evidences less effective (e.g., the usage of different naming conventions and presence of historical spelling variations)" [4, 3]. Finally, "archives and texts from the past are not as anglophone as in today's information society, making multilingual resources and processing capacities even more essential" [26, 11]. In this context, the objective of CLEF HIPE 2020 shared task is threefold:

> strengthening the robustness of existing approaches on non-standard inputs; enabling performance comparison of NE processing on historical texts; and in the long run, fostering efficient semantic indexing of historical documents in order to support scholarship on digital cultural heritage collections [11].

2 Related Work

Main NER approaches are based on computational linguistics and machine learning. [13] proposed *ProMiner* which is based on a dictionary of synonyms to identify genes and proteins mentions in text and link them to their corresponding

ids in the dictionary. [27] presented an approach based on dictionaries as well for NER in the medical domain. There are other well-known rules based NER systems such as LaSIE-II [15], NetOwl [17] and FASTUS [1]. These systems are mainly based on semantic and syntactic rules to recognize entities [20].

Among machine learning applied techniques, we quote Hidden Markov Model (HMM), Maximum Entropy, decision trees, Support Vector Machines (SVM) and Conditional Random Fields (CRF). [18] proposed a CRF model and include morphological features, Part-Of-Speech (POS) Tags, and words sequences. [16] used a CRF too, and show that using Word2Vec pre-trained word embeddings improves NER models performances.

On the other hand, neural networks based models have proved their efficiency on NER tasks. Long Short-Term Memory (LSTM) [14] based neural networks have been widely used in different NLP applications thanks to their ability to detect long-term dependencies. These models showed good results compared to traditional approaches, even if they do not need dictionaries, Gazetteers or other additional information. [6] presented a hybrid model by combining Bidirectional LSTM (BiLSTM) and a Convolutional Neural Network (CNN). [19] introduced a neural model similar to [6], based on BiLSTM combined with a CRF. [23] used the attention mechanism to develop a model which takes advantage of sentence level and document level hierarchical contextual representations. [8] introduced *BERT*, acronym of Bidirectional Encoder Representations from Transformers, "which is a language model designed for pre-training deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers" [8]. The model obtained new state-of-the-art results on eleven natural language processing tasks. For more details on related work we refer the reader to [10].

3 Background on some Supervised ML Models

In this section we present some supervised machine learning models used in this research, such as LSTM operating model, BiLSTM which is the combination of two LSTMs, followed by a brief description of the CRF modelling model and its usefulness.

3.1 The Long Short-Term Memory model

LSTM is a RNN architecture used in the field of deep learning. "This powerful family of connectionist models can capture time dynamics via cycles in the graph" [14, 24].

RNNs take as input a vectors sequence $(x_1, x_2, ..., x_n)$ at time t and return the hidden state vectors sequence $(h_1, h_2, ..., h_n)$, which stocks information learned in actual and previous steps. "Although RNNs can, in theory, learn long dependencies, in practice they fail to do so and tend to be biased towards their most recent inputs in the sequence"[2]. Thanks to their memory cells, LSTMs are able

to resolve this point by capturing useful information from previous states[14]. A LSTM unit is updated at a time t using the following equations:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{1}$$

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{2}$$

$$\tilde{c_t} = tanh(W_c h_{t-1} + U_c x_t + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{4}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where:

- σ is the element-wise sigmoid function
- $\odot$ is the element-wise product
- x_t is the input vector at time t
- h_t is the hidden state (could be called output) vector stocking useful information at (and before) time t
- U_i , U_f , U_c , U_o are the weight matrices of different gates for input x_t
- W_i,W_f,W_c,W_o denote the weight matrices for hidden state h_t
- b_i, b_f, b_c, b_o are the bias vectors

[14, 24].

3.2 Bidirectional LSTM

A LSTM computes a hidden state vector $\overrightarrow{h_t}$ representative of the left context in a sentence at every step t [19]. To take advantage of information that we could get from treating the same sentence in reverse, [9] proposed the BiLSTM model. The idea is to use an another LSTM, to generate a second hidden state vector $\overleftarrow{h_t}$ representative of the right context in the sentence. Concatenating these two vectors leads to a representation ht=[$\overrightarrow{h_t}$;$\overleftarrow{h_t}$] of the word in its general context. The resulting representation is useful for numerous tagging applications [19].

3.3 CRF

"A very simple but surprisingly effective tagging model is to use the h_t's as features to make independent tagging decisions for each output y_t"[21]. In sequence labeling tasks, taking into consideration neighboring labels could be helpful while analyzing a given input sentence, like in some "grammar" rules where a noun more likely follows an adjective than a verb, this can be equivalent in NER to the fact that I-ORG cannot follow I-PERS [24].

Therefore, as in the research presented by [19], we model label sequence jointly using a CRF, instead of modeling them independently.

3.4 Extracting Character Features Using a CNN

CNN layers have become ubiquitous in many NLP tasks. As in [6], we use for each word a convolution and a max layer to extract a per-character feature and optionally the character type to obtain at the end a new character embedding with resulting features. A special token "PADDING" has been used on both sides of words to keep the same length of sequences. In this section we present a brief description of CNN applied to text [31].

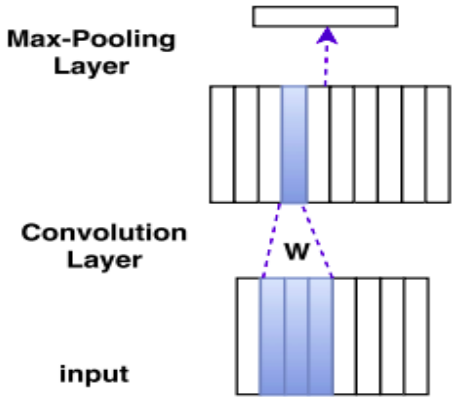

Fig. 1. Basic representation of CNN layers [31].

Input Layer An input sequence x with n elements, each one is represented by a d-dimensional vector, can be represented as a map of features of dimensionality d x n. The bottom of figure 1 shows the input layer as a rectangle with multiple columns [31].

Convolution Layer Convolution layer is employed to represent learning by sliding w-grams over an input sequence $(x_1, x_2, ..., x_n)$. We consider a vector c_i ϵ $\mathbb{R}^{wd}$ as the concatenated embeddings of w entries $(x_{i-w+1}, ..., x_i)$, where w is the filter width and $0 < i < s + w$. We pad embeddings of x_i , where $i < 1$ or $i > n$, with zeros. We then represent the *w-grams* $(x_{i-w+1}, ..., x_i)$ by a new vector p_i ϵ $\mathbb{R}^d$ using the convolution weights W ϵ $\mathbb{R}^{d^* wd}$:

$$p_i = tanh(W c_i + b) \tag{7}$$

where b ϵ $\mathbb{R}^d$ is the bias [31].

Maxpooling We use w-gram representations pi $(i = 1...s+w-1)$ to generate the input sequence x representation by applying maxpooling: $x_j = max(p_{1,j}, p_{2,j}, ...)$ where $(j = 1, ..., d)$ [31].

3.5 Attention mechanism

"The attention mechanism has become an integral part of compelling sequence modeling and transduction models in various tasks"[29]. This technique allows to represent the context in a sequence by taking into consideration neighboring words [29]. For more details on attention mechanism we refer the reader to [29].

4 Our Proposed Approach

This study aims to compare the effectiveness of our proposed attentive neural approach in recognizing and classifying NEs in historical newspapers with a comparison to a statistical model augmented with orthographic features and two other neural models.

4.1 Statistical approach

Statistical approaches based on CRF, SVM or Perceptron have proven good performances using only handcrafted features in many NLP tasks such as NERC [6]. In our work, we use CRFsuite[2] implementation of CRF provided by HIPE team. Among all CRFs implementations, CRFsuite is the fastest one for training the model and labeling data.

In our baseline we use orthographic basic spelling features extracted from words such as prefix and suffix, the casing of the initial character, and whether it is a digit.

4.2 Neural network approach

In this section we present our NER neural model followed by a brief description of used input embeddings and additional features.

Proposed NER Model As in [6, 19], we use in our architecture BiLSTM layers for the extraction of word-level features. These layers are followed by an attention layer. We also use a CRF layer on the top of our model, augmented with some features such as dropout layers. Figure 2 presents our proposed architecture. Apart from using word representations, we also use character representations to extract morphological and orthographic features. As shown in Figure 2, word embeddings are given to a BiLSTM. l_i and r_i represent the word i in its left and right contexts respectively. The concatenation of these two vectors represent the word's context c_i [19].

[2] `http://www.chokkan.org/software/crfsuite/`

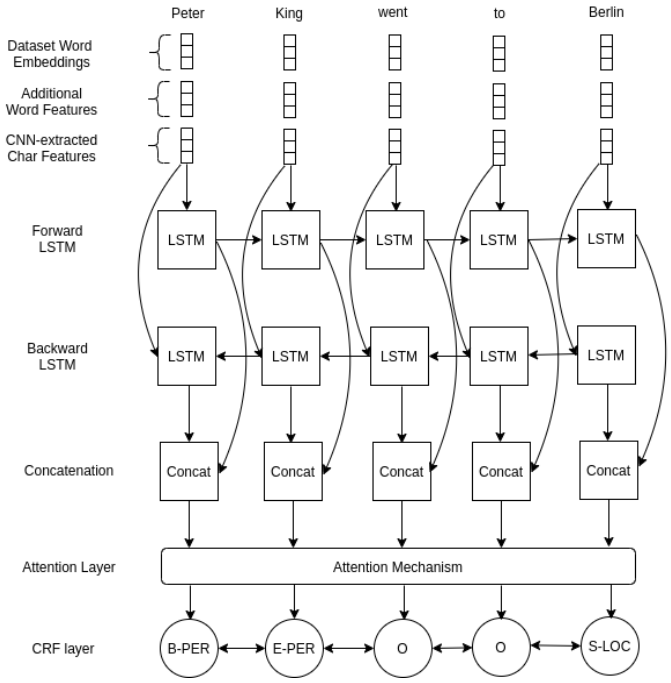

Fig. 2. The main architecture of our NER system using BiLSTM, attention and CRF layers [6, 19].

Input Embeddings The input layers of our model are vector representations of words. "Learning independent representations for word types from the limited NER training data is a difficult problem: there are simply too many parameters to reliably estimate" [19]. In our study, we use pre-trained contextualized word embeddings to initialize our look-up table and to enrich our training dataset. In our experiments, we use indomain Flair embeddings provided by HIPE organizers. "These embeddings were computed with a context of 250 characters, 1 hidden layer of size 2048, and a dropout of 0.1. Input was normalized with lowercasing, replacement of digits by 0, everything else was kept as in the original text" [11]. Extracting character-level representations allows us to take advantage of features related to the domain in hand. Following [6], we use a CNN layer to represent each word based on its characters. We initialize a lookup table randomly with values between -0.5 and 0.5 to generate a character representation of 25 dimensions. The character set is formed by all characters present in the dataset, with PADDING and UNKNOWN tokens, used for the CNN and all other characters respectively. Figure 3 [6] presents an example where we give the word "Picasso" characters embeddings to a CNN.

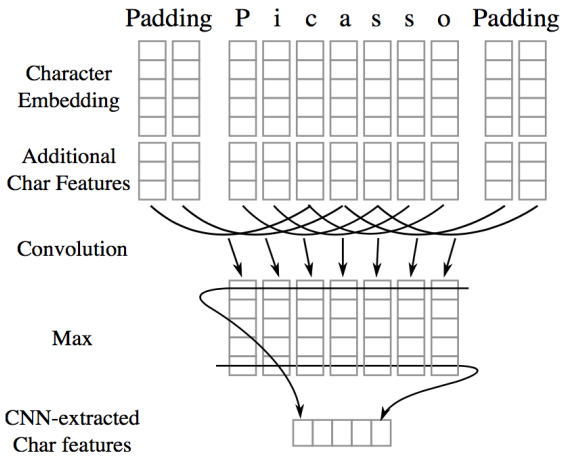

Fig. 3. An architecture using character embeddings of the word "Picasso" in CNN [6].

Additional features As information related to capitalisation has been removed during word embeddings' map construction. We used a separate look-up table to add this feature with the following options: allCaps (the word is in capital letters), upperInitial (only the first letter is capitalized), lowercase (the word is lower cased) and mixedCaps (capital and small letters are mixed) [7, 6]. In our work, we used additional character-based features as well, by using a look-up table, to generate a vector which represent the character's type (uppercase, lowercase, punctuation or other).

5 Experiments and Evaluations

In this section we present our experiments and the results obtained with different models.

5.1 Task Description

The CLEF HIPE 2020 shared task includes two NE processing tasks with sub-tasks of different level of difficulty. In our work we participate in NERC coarse-grained sub-task of the NERC task. This task includes the recognition and classification of entity mentions according to high-level entity types. In our case the types used for annotations are: LOC, ORG, PERS, PROD and TIME.

5.2 Training data

"The shared task corpus is composed of digitized and OCRed articles originating from Swiss, Luxembourgish and American historical newspaper collections and

selected on a diachronic basis"[11].[3] Table 1 shows an overview of the French corpus statistics.

Datasets	#docs	#tokens	#mentions	%noisy
Train	158	129,925	7885	-
Dev	43	29,571	1938	-
Test	43	32,035	1802	12.15
All	244	191,531	11,625	-

Table 1. Overview of French Corpus Statistics[11].

5.3 Training and implementation details

As in [6] we use in our experiments the IOB tagging scheme which stands for Begin, Inside and Outside. This schema allows us to mark the position of the word in the named entity. We implement our model using Keras library with Tensorflow as a backend. As in [6], we initialize LSTM states with zero vectors. Except for the character and word embeddings whose initializations have been described previously, we initialize all lookup tables randomly. We train our model with mini-batchs using *nadam* optimization algorithm. As in [19] we use a single layer for both forward and backward LSTMs and we apply dropout layers to make our model learn from word and character features. Furthermore, applying dropout was effective in reducing overfitting and improving our model's performance.

5.4 Evaluation Measures

> NERC task in CLEF HIPE 2020 shared task is evaluated in terms of Precision, Recall and F-measure (F1). Evaluation is done at entity level according to two metrics: micro average, with the consideration of all TP, FP, and FN [4] over all documents, and macro average, with the average of document's micro figures. NERC benefits from strict and fuzzy evaluation regimes. For NERC, the strict regime corresponds to exact boundary matching and the fuzzy to overlapping boundaries [11].

For more details on evaluation metrics we refer the reader to [11].

[3] From the Swiss National Library, the Luxembourgish National Library, and the Library of Congress (Chronicling America project), respectively. Original collections correspond to 4 Swiss and Luxembourgish titles, and a dozen for English.

[4] True positive, False positive, False negative

5.5 Models Evaluation

In this section we evaluate different used models which have been already described above. In table 2 we cite main differences between models.

Models	Statistical	Orth. features	Neural	Cont. WE	Att. mech.
Model 1	✓	✓	✗	✗	✗
Model 2	✗	✗	✓	✗	✗
Model 3	✗	✗	✓	✗	✓
Our model	✗	✗	✓	✓	✓

Table 2. Different studied models.

Results Tables 3, 4, 5 and 6 show a comparison of results obtained with different studied models.

Discussion According to the results presented in tables 3 and 4, we notice on the one hand that the use of in domain contextualized word embeddings and attention mechanism lead to a higher F-measure for LOC, PROD and TIME entities compared to all other models in both fuzzy and strict regimes. On the other hand the statistical model augmented with orthographic features performs better in both ORG and PERS entities, this could be explained by the importance of syntactic information provided by these features and the large portion of information that they encode which are essential in the NERC task.

Now if we consider metonymic sense, according to table 5, all neural models perform better than the statistical model augmented with orthographic features in both regimes, and our model has higher scores than the two other neural models, except model 3 which has higher F-measure in strict regime. Moreover, even if other models have higher precision, our model showed higher recall which lead to a higher F-measure. We are convinced by the fact that "actively tackling the problem of OCR noise and hyphenation issues helps to achieve better recall"[11]. These results show that neural models especially our proposed model, where we use contextualized word embeddings and attention mechanism, perform far better than the statistical model on all entities when it is about metonymic sense.

Now if we consider table 6, we notice that model 2 and model 3 perform better than the statistical model and barely better than our proposed model on the ORG entity, which shows that these models were more able to generalize on test data in this stage.

All these improvements prove the efficiency of our neural model architecture and of different features used in training, especially contextualized word embeddings trained on large quantities of raw data and character embeddings extracted from specific domain dataset. Therefore, our neural model is able to

extract necessary knowledge from training data, without using handcrafted features.

An important aspect of the CLEF HIPE 2020 shared task corpus, and for historical newspaper data in general, is the noise generated by OCR. As reported in [11], noisy mentions affect remarkably the model's performance: "little noise as 0.1 severely hurts the system's ability to predict an entity and may cut its performance by half"[11]. In our study, we do not report results obtained on the dev set as in the final step, after using dev set to fine tune our model's parameters, we used train and dev sets for training. However, we would like to confirm the degradation of our model's performance, caused in part by the fact that "11 % of all mentions in test set contain OCR mistakes"[11].

Table 3. Our models results for NERC-Coarse in French, considering literal sense of entities (micro average).

	Model 1						Model 2						Model 3						Our model					
	Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LOC	.853	.797	.824	.778	.727	.752	808	.879	**.842**	.728	.793	**.759**	.777	.833	.804	.704	.754	.728	.871	.837	**.854**	.798	.767	**.782**
ORG	.596	.454	**.515**	.586	.446	**.507**	.482	.408	.442	.427	.362	.392	.5	.438	.467	.43	.377	.402	.488	.454	.47	.405	.377	.39
PERS	.851	.753	**.799**	.624	.552	**.586**	.722	.681	.701	.553	.522	.537	.722	.707	.714	.53	.52	.525	.752	.622	.68	.484	.4	.438
PROD	.565	.426	486	.457	.344	.393	.55	.361	.436	.525	.344	**.416**	.524	.361	.427	.476	.328	.388	.617	.475	**.537**	.553	.426	**.481**
TIME	.805	.623	.702	.488	.377	.426	.865	.849	**.857**	.519	.509	**.514**	.833	.755	**.792**	.542	.491	**.515**	.841	.698	**.763**	.568	.472	**.515**
ALL	.824	.736	**.777**	.698	.623	**.659**	.755	.758	.757	.644	.646	.645	.736	.741	.738	.621	.625	.623	.796	.72	.756	.66	.598	.627

Table 4. Our models results for NERC-Coarse in French, considering literal sense of entities (macro average).

	Model 1						Model 2						Model 3						Our model					
	Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LOC	.843	.789	.815	.776	.72	.748	.799	.864	**.821**	.723	.782	.742	.777	.828	.787	.708	.757	.717	.847	.828	**.823**	.775	.757	**.752**
ORG	.598	.501	**.629**	.565	.468	**.589**	.455	.371	.457	.382	.301	.379	.343	.353	.38	.298	.288	.325	.444	.511	.526	.335	.375	.393
PERS	.849	.745	**.809**	.622	.551	**.596**	.745	.704	.74	.597	.568	.595	.717	.7	.706	.563	.55	.554	.75	.609	.68	.502	.414	.462
PROD	.582	.564	.621	.47	.477	.519	.518	.352	.43	.514	.349	.426	.411	.255	.362	.366	.228	.322	.546	.539	**.677**	.5	.489	**.614**
TIME	.848	.671	.836	.569	.458	.571	.883	.906	**.928**	.575	.586	.602	.85	.83	**.868**	.537	.548	.566	.87	.751	**.884**	.691	.582	**.689**
ALL	.851	.734	**.779**	.72	.622	**.661**	.78	.767	.771	.672	.657	.662	.755	.745	.746	.651	.639	.641	.796	.712	.747	.669	.592	.624

Table 5. Our models results for NERC-Coarse in French, considering metonymic sense of entities (micro average).

	Model 1						Model 2						Model 3						Our model					
	Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ORG	.625	.18	.28	.625	.18	.28	.494	.351	**.411**	.494	.351	**.411**	.565	.351	**.433**	.565	.351	**.433**	.468	.468	**.468**	.423	.423	**.423**
TIME	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALL	.625	.179	.278	.625	.179	.278	.494	.348	**.408**	.494	.348	**.408**	.565	.348	**.431**	.565	.348	**.431**	.468	.464	**.466**	.423	.42	**.422**

Table 6. Our models results for NERC-Coarse in French, considering metonymic sense of entities (macro average).

	Model 1						Model 2						Model 3						Our model					
	Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict			Fuzzy			Strict		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ORG	.565	.278	.448	.565	.278	.448	.362	.42	**.551**	.362	.42	**.551**	.477	.437	**.592**	.477	.437	**.492**	.34	.36	**.458**	.32	.342	.431
TIME	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALL	.565	.278	.448	.565	.278	.448	.362	.0.42	**.551**	.362	.42	**.551**	.477	.436	**.592**	.477	.436	**.592**	.34	.36	**.457**	.32	.341	.431

6 Conclusion

In this paper, we presented a hybrid approach for NERC applied on historical newspapers. In our experiments, we used orthographic features related to words syntax. Besides, we used word and character embeddings, which allow us to detect morphological and orthographic features related to a specific domain. Our experiments show an improvement in the overall performance. We notice that our attentive neural model augmented with contextualized word embeddings performs better compared to our baselines overall. To the best of our knowledge, there is no study which combines the application of attention mechanism and contextualized word embeddings in NERC for historical newspapers domain.

As a future work, we aim to investigate the usefulness of adding additional features in the hybrid architecture and the use of external resources such as ontologies and other knowledge and common sense bases. Applying multi-task learning will be part of our future work, as well. Moreover, it would be relevant to apply explainability techniques on the neural network models in order to better explain and analyze the results.

Bibliography

[1] D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson. SRI International FASTUS systemMUC-6 test results and analysis. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, pages 237 – 248, 1995.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[3] M. Bollmann. A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019.

[4] L. Borin, D. Kokkinakis, and L.-J. Olsson. Naming the past: Named entity and Animacy recognition in 19th century Swedish literature. In *Proceedings of the Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)*., pages 1–8. Association for Computational Linguistics, 2007.

[5] G. Chiron, A. Doucet, M. Coustaty, M. Visani, and J. Moreux. Impact of ocr errors on the use of digital libraries: Towards a better access to information. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 249–252. IEEE Press, 2017.

[6] J. P. C. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4(1):357–370, 2015.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(1):2493–2537, 2011.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[9] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith. Transition-based dependency parsing with stack long short-term memory. volume 1, page 334 – 343, 2015.

[10] M. Ehrmann, M. Romanello, A. Flückiger, and S. Clematide. Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers. In L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, editors, *CLEF 2020 Working Notes. Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum*. CEUR-WS, 2020.

[11] M. Ehrmann, M. Romanello, A. Flückiger, and S. Clematide. Overview of CLEF HIPE 2020: Named Entity Recognition and Linking on Historical Newspapers. In A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, and N. Ferro,

editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020)*, volume 12260 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2020.

[12] A. Goyal, V. Gupta, and M. Kumar. Recent named entity recognition and classification techniques: A systematic review. *Computer Science Review*, 29(1):21–43, 2018.

[13] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(1):S14 – S14, 2005.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[15] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. Morgan, 1998.

[16] M. Joshi, E. Hart, M. Vogel, and J.-D. Ruvini. Distributed word representations improve NER for e-commerce. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 160–167, Colorado, 2015. Association for Computational Linguistics.

[17] G. R. Krupka and K. Hausman. IsoQuest Inc.: Description of the NetOwl^{TM} extractor system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, pages 21 – 28, 1998.

[18] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[19] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT 2016*, page 260–270, 2016.

[20] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *CoRR*, 2018.

[21] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[22] E. Linhares Pontes, A. Hamdi, N. Sidere, and A. Doucet. Impact of ocr quality on named entity linking. In A. Jatowt, A. Maeda, and S. Y. Syn, editors, *Digital Libraries at the Crossroads of Digital Information for the Future*, pages 102–115. Springer International Publishing, 2019.

[23] Y. Luo, F. Xiao, and H. Zhao. Hierarchical contextualized representation for named entity recognition. *CoRR*, abs/1911.02257, 2019.

[24] X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1064–1074. Association for Computational Linguistics, 2016.

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

[26] C. Neudecker and A. Antonacopoulos. Making europe's historical newspapers searchable. *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 405–410, 2016.

[27] A. P. Quimbaya, A. S. Múnera, R. A. G. Rivera, J. C. D. Rodríguez, O. M. M. Velandia, A. A. G. Peña, and C. Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100(1):55 – 61, 2016.

[28] D. A. Smith and R. Cordell. A Research Agenda for Historical and Multilingual Optical Character Recognition. Tech. rep. 2018.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[30] V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158. Association for Computational Linguistics, 2018.

[31] W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of CNN and RNN for natural language processing. *CoRR*, abs/1702.01923, 2017.