# Transfer Learning for Named Entity Recognition in Historical Corpora

Konstantin Todorov*[0000−0002−7445−4676] and
Giovanni Colavizza*[0000−0002−9806−084X]

*University of Amsterdam, The Netherlands
kztodorov@outlook.com
g.colavizza@uva.nl

**Abstract.** We report on our participation to the 2020 CLEF HIPE shared task as team Ehrmama, focusing on bundle 3: Named Entity Recognition and Classification (NERC) on coarse and fine-grained tags. Motivated by an interest to assess the added value of transfer learning for NERC on historical corpora, we propose an architecture made of two components: (i) a modular embedding layer where we combine newly trained and pre-trained embeddings, and (ii) a task-specific Bi-LSTM-CRF layer. We find that character-level embeddings, BERT, and a document-level data split are the most important factors in improving our results. We also find that using in-domain FastText embeddings and a single-task as opposed to multi-task approach yields minor gains. Our results confirm that pre-trained language models can be beneficial for NERC on low-resourced historical corpora.

**Keywords:** NERC · BERT · Bi-LSTM-CRF · Transfer learning.

## 1 Introduction

The advent of contextual language models such as Bidirectional Encoder Representations from Transformers (BERT) [3] has furthered the adoption of transfer learning in Natural Language Processing (NLP). Transfer learning aims at transferring knowledge from a general-purpose source task to a specialised target task [11,13]. The specialised target task is often linguistically under-resourced (e.g., small data or lack of linguistic resources) [2]. Transfer learning further allows saving computation resources by training once and applying the same model widely with little or no further adaptation [15].

The increasing abundance of historical text corpora offers a compelling opportunity to apply transfer learning. Historical texts pose a set of challenges to the NLP and Digital Humanities (DH) communities, of which the most general and pressing are [12,4]: a) noisy inputs, for example due to Optical/Handwritten

Character Recognition (OCR/HTR) errors; b) linguistic change over time; c) language variety, in the absence of mainstream languages such as English. These challenges are not unique to historical texts, but they come to the forefront when dealing with them.

We participated in the CLEF HIPE as team Ehrmama, focusing on bundle 3: NERC coarse and NERC fine-grained [5], conducted over English, French and German languages. This task bundle focuses on the recognition of named entities in six different tag types, namely coarse- and fine-grained and their metonymic senses, as well as components and nested entities of depth one. Our general goal is to assess if and how transfer learning using modern-day language models can help with tasks on OCRed historical corpora. To this end, we propose a model composed of a general purpose embedding layer which allows to combine character, sub-word and word-level embeddings in a modular way, equipped with a state-of-the-art NERC-specific layer. We then explore the use of newly-trained and pre-trained embeddings in isolation and in combination. Our code is publicly available[1].

## 2 Method

Our proposed architecture is composed of two parts: an embedding layer and a task-specific layer, in this case for NERC. An illustration is given in Figure 1.

Several embeddings can be combined into a modular embedding layer which we use to represent input text. We broadly distinguish between (i) *pre-trained (transferred) embeddings* and (ii) *newly trained embeddings*. While pre-trained embeddings can either be fine-tuned or frozen during learning, newly trained representations are learned from scratch. Furthermore, embeddings can be applied at different input granularities, including: (i) *character-*, (ii) *sub-word-* and (iii) *word-level*. These are also modular, and can be used in combination.

The task-specific layer is a Bidirectional Long Short-Term Memory, using Conditional Random Field (Bi-LSTM-CRF) as proposed by [9], with the additional removal of the `tanh` non-linearity after the LSTM. As a sanity test, we applied our model on the modern-day CoNLL-2003 dataset [14], achieving results comparable to current state of the art.

### 2.1 Empirical setup

**Embedding layer** containing four different embedding modules.

*Character embeddings* consist of an embedding layer, followed by a bidirectional LSTM. The embedding layer's size and the LSTM hidden size are both hyper-parameters with values ranging from 16 to 128 and 16 to 256 respectively. We use a character-level custom vocabularies for each language built from the training and validation data sets.

[1] https://github.com/ktodorov/eval-historical-texts.

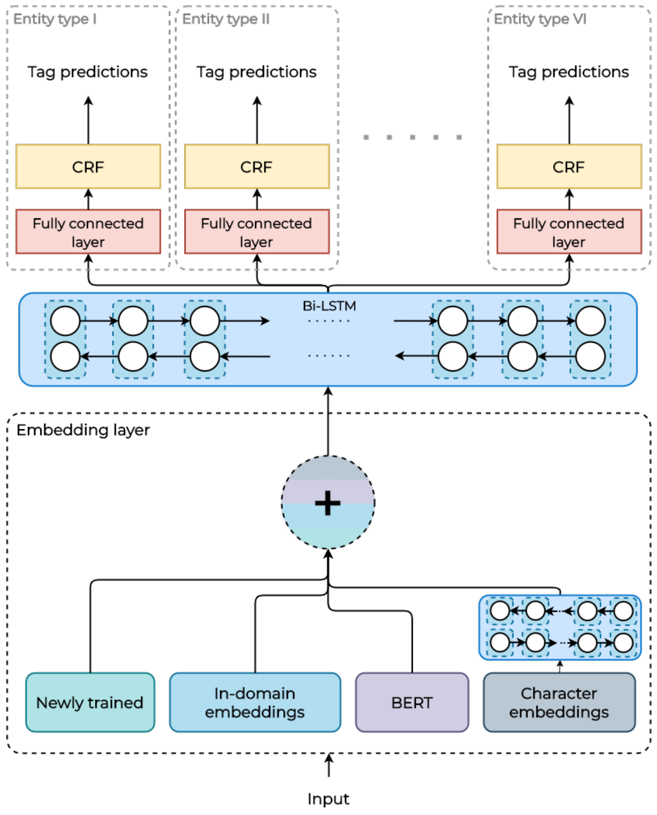

Fig. 1: NERC multi-task model architecture. Our single-task architecture is identical and only contains a fully connected layer and CRF for one entity type.

*BERT* embeddings work on sub-word level. We use *bert-base-multilingual-cased* for French, *bert-base-german-cased* for German, and *bert-base-cased* for English, relying on the HuggingFace Transformers library [16]. This brings the specific limitation of only working with sequences of 512 tokens in maximum length. As our text sequences are usually longer, we implement a sliding-window splitting of input sequences before passing them through BERT. While splitting, we keep the first and last 5 tokens of each chunk as overlap among sequential chunks. After embedding each chunk, we then reconstruct the full input sequences by averaging the embeddings of the overlapping characters.

*Newly trained* embeddings work on sub-word level and their weights are randomly initialised and learned during training. We use the same vocabulary as with BERT. The size of these embeddings is a hyper-parameter and ranges between 64 and 512.

*In-domain pre-trained embeddings* provided by the task organisers are used for feature extraction only (frozen). These embeddings have size of 300 and work at the sub-word level. This model uses the `FastText` library [7].

After testing different alternatives, we found that the simplest and fastest way to combine these embeddings is by concatenating them, resulting in concatenated sub-word embeddings of a size equal to the sum of the embedding sizes of all enabled modules.

**Task-specific layer** based on a Bi-LSTM-CRF [9].

*The Bi-LSTM-CRF* uses the concatenated sub-word embeddings as its input, and then merges the output to word level by taking the mean. Finally, the resulting representation is pushed to a fully connected layer which then outputs tag probabilities for each token. We tested concatenating embeddings before or after the Bi-LSTM, or not merging at all, and found that our approach performs best, also in accordance with previous findings [13]. A Conditional Random Field (CRF) [8] is eventually used over the produced tag probabilities to decode the final tag predictions.

*A multi-task approach* is our primary setup. We introduce additional output heads, one for each of the different entity types that the task aims to predict. The final two layers of the model, namely the fully connected layer and CRF, are specific to each entity tag type, while the rest of the architecture is shared. The individual losses for each task are summed during backpropagation. We compare using single vs multi-task approach in what follows.

**Additional resources** We use the Annotated Corpus for Named Entity Recognition built on top of Groningen Meaning Bank (GMB) [1][2]. This dataset is annotated specifically for training NER classifiers, and contains most of the coarse grained tag types which occur in the English dataset provided by organisers. We consolidate some tags with the same meaning but different labels ourselves. The dataset contains in total 1,354,149 tokens of which 85% are labelled as `O` originally. We convert the tag types that are not part of this challenge to `O` as well, resulting in total of 94.92% tokens having `O` literal tags.

We used an NVidia GeForce 1080Ti GPU with 11GB GDDR5X memory for our experiments.

### 2.2 Model fitting

In this section, we discuss the remaining pre-processing or hyper-parameter choices which we assessed empirically.

[2] https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus [accessed 2020-07-16].

**Pre-processing** The input data is organised into *documents*, and each document is split into multiple *segments* where usually one segment corresponds to one line in the original historical source. The input can thus be split into segments or into documents. Using segments leads to much faster convergence, while document splitting usually yields better results in our experiments. We further analyse the importance of splitting by introducing a *multi-segment* option which combines more than one consecutive segment. We perform a hard split and pick the maximum length of one multi-segment sequence to be the maximum length allowed by the HuggingFace Transformers library. We do this to avoid any unwanted noise. At document level we overcome this limitation by splitting documents using a sliding window approach where the first and last 5 tokens for each split are overlapping with the previous and next splits respectively. We perform the cutting before extracting features through BERT after which we concatenate the representations back. We take the average values for the overlapping tokens. Finally, we replace all numbers with zeros, including such that contain more than one digit. Besides, we do not lowercase, nor do we remove any punctuation or other characters.

**Fine-tuning vs. freezing** There are two possibilities when using pre-trained models: keep them frozen or fine-tune further more. Fine-tuning the model lets us introduce two additional configuration options. The first one is related to when to start fine-tuning. This is most often performed at the beginning of the training process and until convergence. We try a second approach where firstly the full model with frozen pre-trained weights converges. After, the pre-trained weights are fine-tuned. This is something that we also investigate but find no difference between the two approaches. We therefore fine-tune from the start in the reported experiments with fine-tuning enabled.

**Manually crafted features** Following previous work [6], we assess the importance of manually crafted features. We use `AllLower`, `AllUpper`, `IsTitle`, `FirstLetterUpper`, `FirstLetterNotUpper`, `IsNumeric` and `NoAlphaNumeric` as extra morphological features. When including these features in the model, we do not get significant improvements.

**Weighting** As it is common with NERC tasks, most of the ground truth is composed of *outside* or `O` tags. In our case, these make up for approximately 94.92%, 95.95%, and 96.5% of the total tokens for English, French and German languages respectively. To counteract tag imbalance, we test a *weighted loss* which we plug into the CRF layer, giving more weight to tags predicted as outside ones but are in fact part of entities, and less on tokens which are predicted as inside an entity but are actually outside. This weighted loss does not prove to be beneficial.

**Hyper-parameters** We assess Adam and AdamW[10] optimizers. For the *learning rate* we see that higher values benefit the model more. We pick a default value of 1e−8 for *weight decay* for all optimizers. The final hyper-parameter configurations that we use are summarised in Table 1.

Table 1: Hyper-parameter configurations. *Configuration I* is used for `Base`. *Configuration II* is used for `Base + CE + BERT` and `Base + CE + BERT - newly`. *Configuration III* is used for all remaining setups.

| **Hyper-parameters** | **Configuration I** | **Configuration II** | **Configuration III** |
|---|---|---|---|
| **RNN hidden size** | 512 | 256 | 512 |
| **RNN directionality** | bi-directional | bi-directional | bi-directional |
| **RNN dropout** | 0.5/0.8 | 0.5/0.8 | 0.5/0.8 |
| **Newly trained embeddings size** | 64 | 64 | 64 |
| **Character embeddings size** | – | 16 | 16 |
| **Character embeddings RNN hidden size** | – | 32 | 32 |
| **Replace numbers during pre-processing** | yes | yes | yes |
| **Weighted loss usage** | no | no | no |
| **Optimizer** | AdamW | AdamW | AdamW |
| **Learning rate** | 1e−2 | 1e−2 | 1e−2 |
| **Fine-tune learning rate** | 1e−4 | 1e−4 | 1e−4 |

## 3 Results

We report results for the three languages part of the task, namely French, German and English, using the official test set v1.3[3]. In addition to that, we report results using *multi-segment* and *document* split types for French and German and *segment* split type for English, since our English training data lacks the document level.

All results are reported in the two scoring approaches used in the challenge — *fuzzy* and *strict*. As a reminder, fuzzy scoring works in a relaxed way, allowing fuzzy boundary matching of entities. That is if an entity is only partially recognised, e.g., if 4 out of total of 6 tokens are recognised correctly, this is still considered a successful recognition. Conversely, strict matching requires all tokens to match with exact boundary matching — in previous example this would require 6 out of 6 total tokens to be predicted correctly. For each scoring approach, we provide *precision* (P), *recall* (R) and *F-score* (F), all reported as micro and calculated using the original scorer, used in the competition[4]. We report the baseline model provided by organisers for reference, reminding the reader that the baseline model always uses a document level split. We also report the baseline model results on our English data.

We order the different configurations for all languages following our ablation studies, which primarily focus on assessing the impact of transfer learning. We start with the simplest `Base` model which is only using newly-trained sub-word

[3] https://github.com/impresso/CLEF-HIPE-2020.
[4] https://github.com/impresso/CLEF-HIPE-2020-scorer.

embeddings and no pre-trained information of any type. Then we continue by adding Character Embeddings (CE) which use Bi-LSTM (`+CE`). Due to the significant improvements observed by adding character embeddings, we keep them enabled in all of our next reported setups. We further report results that were achieved by adding firstly the (frozen) FastText embeddings provided by organisers (`+FT`), then (frozen) BERT embeddings (`+BERT`), and finally both. Whenever BERT is enabled, we also report runs where we disable newly trained embeddings (`-newly`). Eventually, we report three different setups where we unfreeze BERT and fine-tune them on the task at hand. Due to the long sequence lengths when working on document level, we are unable to perform fine-tuning of BERT at the document level. We therefore report the results of fine-tuning BERT only using multi-segment split. All models use the multi-task approach, except for one single-task run, which has all available embeddings enabled (single).

We start reporting results for French, in Table 2. Firstly, adding character-level embeddings and BERT consistently improves results. Better results overall are obtained with a single-task approach and using all available embeddings, including newly trained ones. A document level split, following this configuration, perform best across the board. We also see that most of our configurations struggle on tasks with sparser annotations such as Metonymic and Nested. Furthermore, fine-tuning BERT does not seem to improve results. Results for German, shown in Table 3, are consistent with those for French. It is worth noting that our models struggle even more on the German Metonymic and Nested tasks. For nested tags, we are not able to be predictive at all, specifically on the multi-segment level.

For completeness, we report results for English in Table 4, limited to the Literal coarse task. For a better comparison, we provide results from two baseline models: i) results from the organisers and ii) results training the baseline model on the English dataset we use. Our models are mostly not able to perform beyond the provided baseline. This is likely due to the training data we use.

We clarify that most of these results were obtained after the task submission deadline. For the deadline, and as reported in [5], we submitted three different runs for German and French and two for English. For German and French we submitted one run using multi-task learning and document-level splitting; another run using multi-task learning and multi-segment splitting; for English we had one run using multi-task learning and segment splitting; finally, we submitted one run for all languages where we used literal tag types from two single-task learning runs. All of our submitted runs had all modules enabled.

Table 2: NERC, French. The best result per table and column is given in bold, the second best result is underlined

(a) multi-segment level, coarse grained entity type

| Configuration | Literal coarse | | | | | | Metonymic coarse | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .825 | .721 | .769 | .693 | .606 | .646 | .541 | .179 | .268 | .541 | .179 | .268 |
| Base | .776 | .69 | .73 | .618 | .55 | .582 | .5 | .424 | .459 | .495 | .42 | .454 |
| Base + CE | .806 | .739 | .771 | .649 | .594 | .62 | .552 | .379 | .45 | .545 | .375 | .444 |
| Base + CE + FT | .789 | .78 | .784 | .65 | .642 | .646 | .481 | .339 | .398 | .468 | .33 | .387 |
| Base + CE + BERT | **.886** | .801 | .841 | **.782** | .707 | .743 | .424 | .397 | .41 | .41 | .384 | .396 |
| Base + CE + BERT – newly | .859 | .818 | .838 | .719 | .685 | .702 | .417 | .384 | .4 | .417 | .384 | .4 |
| Base + CE + FT + BERT | .866 | .836 | .851 | .767 | .739 | .753 | .664 | .362 | .468 | .656 | .357 | .462 |
| Base + CE + FT + BERT – newly | .864 | **.848** | **.856** | .765 | **.751** | **.758** | **.766** | .321 | .453 | **.766** | .321 | .453 |
| Base + CE + FT + BERT (single) | .872 | .835 | .853 | .769 | .737 | .753 | .036 | .069 | .000 | .036 | .069 | .000 |
| + Fine-tuning (unfreezing) BERT | | | | | | | | | | | | |
| Base + CE + BERT | .876 | .824 | .849 | .775 | .729 | .751 | .442 | .375 | .406 | .432 | .366 | .396 |
| Base + CE + BERT – newly | .877 | .804 | .839 | .775 | .711 | .742 | .754 | .384 | .509 | .754 | .384 | .509 |
| Base + CE + FT + BERT | .857 | .836 | .846 | .759 | .741 | .75 | .551 | .482 | .514 | .541 | .473 | .505 |
| Base + CE + FT + BERT – newly | .845 | .838 | .842 | .742 | .737 | .74 | .659 | **.5** | **.569** | .659 | **.5** | **.569** |

(b) multi-segment level, fine grained entity type

| Configuration | Literal fine | | | | | | Metonymic fine | | | | | | Component | | | | | | Nested | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .838 | .693 | .758 | .644 | .533 | .583 | .564 | .196 | .291 | .538 | .187 | .278 | .799 | .531 | .638 | .733 | .487 | .585 | .267 | .049 | .082 | .267 | .049 | .082 |
| Base | .8 | .67 | .729 | .548 | .459 | .499 | .476 | **.451** | .463 | .472 | **.446** | .459 | .774 | .531 | .630 | .692 | .475 | .563 | .383 | .140 | .205 | .333 | .122 | .179 |
| Base + CE | .825 | .708 | .762 | .562 | .482 | .519 | .594 | .366 | .453 | .594 | .366 | .453 | .779 | .556 | .649 | .720 | .514 | .600 | **.5** | .067 | .118 | .364 | .049 | .086 |
| Base + CE + FT | .801 | .763 | .781 | .568 | .541 | .554 | .567 | .228 | .325 | .533 | .214 | .306 | .762 | .598 | .67 | .682 | .535 | .600 | .425 | **.207** | **.279** | **.375** | **.183** | **.246** |
| Base + CE + BERT | **.889** | .781 | .831 | .658 | .578 | .616 | .532 | .366 | .434 | .519 | .357 | .423 | .803 | .579 | .673 | .715 | .515 | .599 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT – newly | .865 | .748 | .802 | .613 | .53 | .568 | .54 | .241 | .333 | .54 | .241 | .333 | .821 | .504 | .625 | .732 | .449 | .557 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .866 | .818 | .842 | .672 | .634 | .653 | .702 | .263 | .383 | .643 | .241 | .351 | .804 | .563 | .662 | .712 | .499 | .587 | .357 | .03 | .056 | .143 | .012 | .022 |
| Base + CE + FT + BERT – newly | .873 | .82 | **.846** | .672 | .631 | .651 | **.771** | .241 | .367 | **.743** | .232 | .354 | **.842** | .546 | .663 | **.774** | .503 | .61 | .393 | .067 | .115 | .286 | .049 | .083 |
| Base + CE + FT + BERT (single) | .868 | .818 | .842 | .676 | .636 | .655 | .538 | .442 | .485 | .533 | .438 | **.48** | .752 | **.677** | **.713** | .659 | **.594** | **.625** | .000 | .000 | .000 | .000 | .000 | .000 |
| + Fine-tuning (unfreezing) BERT | | | | | | | | | | | | | | | | | | | | | | | | |
| Base + CE + BERT | .877 | .806 | .840 | .654 | .600 | .626 | .434 | .379 | .405 | .429 | .375 | .400 | .77 | .598 | .673 | .673 | .523 | .588 | .267 | .049 | .082 | .133 | .024 | .041 |
| Base + CE + BERT – newly | .885 | .782 | .83 | .672 | .593 | .63 | .739 | .29 | .417 | .705 | .277 | .397 | .818 | .524 | .639 | .745 | .477 | .582 | .107 | .018 | .031 | .071 | .012 | .021 |
| Base + CE + FT + BERT | .871 | .814 | .842 | **.687** | .642 | **.664** | .568 | .411 | .477 | .543 | .393 | .456 | .741 | .672 | .705 | .648 | .587 | .616 | .232 | .159 | .188 | .179 | .122 | .145 |
| Base + CE + FT + BERT – newly | .852 | **.837** | .845 | .663 | **.652** | .658 | .681 | .420 | **.519** | .609 | .375 | .464 | .785 | .626 | .697 | .701 | .559 | .622 | .333 | .183 | .236 | .244 | .134 | .173 |

(c) document level, coarse grained entity type

| Configuration | Literal coarse | | | | | | Metonymic coarse | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .825 | .721 | .769 | .693 | .606 | .646 | .541 | .179 | .268 | .541 | .179 | .268 |
| Base | .812 | .686 | .743 | .671 | .566 | .614 | .444 | .536 | .486 | .444 | .536 | .486 |
| Base + CE | .802 | .762 | .782 | .658 | .625 | .641 | .575 | .272 | .370 | .566 | .268 | .364 |
| Base + CE + FT | .815 | .737 | .774 | .673 | .608 | .639 | .510 | .469 | .488 | .505 | .464 | .484 |
| Base + CE + BERT | .871 | .831 | .851 | .779 | .743 | .760 | .684 | .232 | .347 | .684 | .232 | .347 |
| Base + CE + BERT – newly | **.890** | .828 | .858 | .788 | .733 | .759 | .564 | .277 | .371 | .545 | .268 | .359 |
| Base + CE + FT + BERT | .872 | .828 | .849 | .772 | .733 | .752 | .433 | **.696** | **.534** | .428 | **.688** | **.527** |
| Base + CE + FT + BERT – newly | .869 | **.872** | .871 | .78 | **.782** | .781 | **.755** | .357 | .485 | **.755** | .357 | .485 |
| Base + CE + FT + BERT (single) | **.89** | .856 | **.873** | **.807** | .776 | **.791** | .699 | .424 | .528 | .691 | .420 | .522 |

(d) document level, fine grained entity type

| Configuration | Literal fine | | | | | | Metonymic fine | | | | | | Component | | | | | | Nested | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .838 | .693 | .758 | .644 | .533 | .583 | .564 | .196 | .291 | .538 | .187 | .278 | .799 | .531 | .638 | **.733** | .487 | .585 | .267 | .049 | .082 | .267 | .049 | .082 |
| Base | .822 | .672 | .739 | .594 | .486 | .534 | .446 | **.513** | .477 | .419 | **.482** | .448 | .738 | .6 | .662 | .657 | .534 | .589 | **.512** | .250 | .336 | .350 | .171 | .230 |
| Base + CE | .809 | .752 | .78 | .586 | .546 | .565 | .521 | .223 | .313 | .521 | .223 | .313 | .743 | .618 | .675 | .65 | .541 | .59 | .35 | .171 | .23 | .275 | .134 | .180 |
| Base + CE + FT | .811 | .722 | .764 | .599 | .534 | .565 | .54 | .362 | .433 | .507 | .339 | .406 | .759 | .603 | .672 | .684 | .544 | .606 | .453 | .177 | .254 | **.406** | .159 | .228 |
| Base + CE + BERT | .885 | .799 | .84 | .696 | .629 | .661 | .654 | .304 | .415 | .654 | .304 | .415 | .719 | .686 | .702 | .625 | .596 | .610 | .304 | .104 | .155 | .250 | .085 | .127 |
| Base + CE + BERT – newly | **.896** | .790 | .840 | .675 | .595 | .633 | .568 | .223 | .321 | .568 | .223 | .321 | **.808** | .603 | .690 | .696 | .520 | .595 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .883 | .800 | .839 | .717 | .649 | .682 | **.741** | .371 | .494 | .679 | .339 | .452 | .794 | .631 | .703 | .715 | .568 | .633 | .341 | .183 | .238 | .318 | .171 | .222 |
| Base + CE + FT + BERT – newly | .881 | .841 | .861 | .703 | .671 | .687 | .705 | .384 | **.497** | .689 | .375 | **.486** | .792 | .644 | .71 | .704 | .572 | .631 | .233 | .043 | .072 | .067 | .012 | .021 |
| Base + CE + FT + BERT (single) | .882 | **.853** | **.867** | **.729** | **.704** | **.716** | **.741** | .357 | .482 | **.741** | .357 | .482 | .734 | **.726** | **.73** | .650 | **.642** | **.646** | .438 | **.299** | **.355** | .393 | **.268** | **.319** |

Table 3: NERC, German. The best result per table and column is given in bold, the second best result is underlined

(a) multi-segment level, coarse grained entity type

| Configuration | Literal coarse | | | | | | Metonymic coarse | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .79 | .464 | .585 | .643 | .378 | .476 | **.814** | .297 | .435 | **.814** | .297 | .435 |
| Base | .698 | .526 | .6 | .535 | .404 | .46 | .559 | **.602** | **.58** | .551 | .593 | .571 |
| Base + CE | .685 | .605 | .642 | .535 | .473 | .502 | .588 | .568 | .578 | .588 | .568 | **.578** |
| Base + CE + FT | .691 | .554 | .615 | .528 | .424 | .47 | .534 | **.602** | .566 | .534 | **.602** | .566 |
| Base + CE + BERT | .801 | .675 | .733 | .596 | .502 | .545 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT – newly | .759 | .706 | .732 | .582 | .541 | .561 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .784 | .724 | .753 | .639 | .589 | .613 | .598 | .542 | .569 | .598 | .542 | .569 |
| Base + CE + FT + BERT – newly | **.84** | .64 | .726 | .696 | .53 | .602 | .696 | .466 | .558 | .696 | .466 | .558 |
| Base + CE + FT + BERT (single) | .827 | **.731** | **.776** | **.708** | **.625** | **.664** | .492 | .53 | .51 | .472 | .508 | .49 |
| + Fine-tuning (unfreezing) BERT | | | | | | | | | | | | |
| Base + CE + BERT | .756 | .718 | .737 | .546 | .519 | .532 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT – newly | .752 | .718 | .734 | .56 | .534 | .547 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .738 | .678 | .707 | .575 | .528 | .551 | .562 | .500 | .529 | .543 | .483 | .511 |
| Base + CE + FT + BERT – newly | .802 | .689 | .741 | .658 | .565 | .608 | .621 | .521 | .567 | .616 | .517 | .562 |

(b) multi-segment level, fine grained entity type

| Configuration | Literal fine | | | | | | Metonymic fine | | | | | | Component | | | | | | Nested | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .792 | .419 | .548 | **.641** | .339 | .444 | **.805** | .28 | .415 | **.805** | .28 | .415 | .783 | .34 | .474 | .727 | .316 | .44 | **.333** | **.014** | **.026** | **.333** | **.014** | **.026** |
| Base | .723 | .517 | .602 | .479 | .343 | .4 | .593 | .593 | .593 | .585 | .585 | .585 | .589 | .298 | .396 | .486 | .246 | .327 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE | .704 | .585 | .639 | .466 | .388 | .424 | .667 | .559 | **.608** | .667 | .559 | **.608** | .589 | .432 | .498 | .506 | .371 | .428 | .25 | **.014** | **.026** | .000 | .000 | .000 |
| Base + CE + FT | .706 | .521 | .6 | .478 | .353 | .406 | .538 | **.602** | .568 | .538 | **.602** | .568 | .654 | .266 | .378 | .571 | .232 | .33 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT | .773 | .693 | .731 | .348 | .312 | .329 | .000 | .000 | .000 | .000 | .000 | .000 | .562 | .222 | .318 | .382 | .151 | .216 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT – newly | .800 | .647 | .716 | .358 | .289 | .320 | .000 | .000 | .000 | .000 | .000 | .000 | .455 | .480 | .467 | .310 | .327 | .318 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .800 | .626 | .703 | .515 | .403 | .452 | .581 | .547 | .563 | .568 | .534 | .55 | .670 | .471 | .553 | .525 | .369 | .433 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT – newly | **.816** | .639 | .717 | .551 | .432 | .484 | .627 | .542 | .582 | .627 | .542 | .582 | .533 | .227 | .319 | .397 | .169 | .237 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT (single) | .776 | .569 | .656 | .477 | .35 | .403 | .000 | .000 | .000 | .000 | .000 | .000 | **.841** | .423 | **.563** | **.751** | .378 | **.503** | .000 | .000 | .000 | .000 | .000 | .000 |
| + Fine-tuning (unfreezing) BERT | | | | | | | | | | | | | | | | | | | | | | | | |
| Base + CE + BERT | .759 | **.703** | .73 | .311 | .288 | .299 | .000 | .000 | .000 | .000 | .000 | .000 | .418 | .295 | .346 | .276 | .195 | .229 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + BERT – newly | .758 | .696 | .726 | .29 | .267 | .278 | .000 | .000 | .000 | .000 | .000 | .000 | .399 | .328 | .36 | .239 | .197 | .216 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .736 | .687 | .711 | .433 | .405 | .418 | .524 | .551 | .537 | .508 | .534 | .521 | .474 | .508 | .49 | .338 | .362 | .349 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT – newly | .801 | .685 | **.738** | .548 | **.469** | **.506** | .691 | .475 | .563 | .691 | .475 | .563 | .58 | **.51** | .543 | .472 | **.415** | .442 | .000 | .000 | .000 | .000 | .000 | .000 |

(c) document level, coarse grained entity type

| Configuration | Literal coarse | | | | | | Metonymic coarse | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .79 | .464 | .585 | .643 | .378 | .476 | **.814** | .297 | .435 | **.814** | .297 | .435 |
| Base | .678 | .552 | .609 | .519 | .422 | .465 | .571 | .581 | .576 | .567 | .576 | .571 |
| Base + CE | .688 | .573 | .626 | .548 | .456 | .498 | .618 | .576 | .596 | .618 | .576 | .596 |
| Base + CE + FT | .706 | .548 | .617 | .549 | .426 | .48 | .725 | .492 | .586 | .725 | .492 | .586 |
| Base + CE + BERT | .763 | .752 | .758 | .642 | .632 | .637 | .714 | .508 | .594 | .714 | .508 | .594 |
| Base + CE + BERT – newly | .805 | .654 | .722 | .641 | .52 | .574 | .433 | .517 | .471 | .426 | .508 | .463 |
| Base + CE + FT + BERT | .767 | **.765** | .766 | .647 | .645 | .646 | .622 | **.627** | **.624** | .622 | **.627** | **.624** |
| Base + CE + FT + BERT – newly | .799 | .726 | .761 | .671 | .609 | .639 | .696 | .542 | .610 | .696 | .542 | .610 |
| Base + CE + FT + BERT (single) | **.86** | .738 | **.795** | **.753** | **.647** | **.696** | .709 | .517 | .598 | .709 | .517 | .598 |

(d) document level, fine grained entity type

| Configuration | Literal fine | | | | | | Metonymic fine | | | | | | Component | | | | | | Nested | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| Baseline | .792 | .419 | .548 | .641 | .339 | .444 | **.805** | .28 | .415 | **.805** | .28 | .415 | .783 | .34 | .474 | **.727** | .316 | .44 | .333 | .014 | .026 | .333 | .014 | .026 |
| Base | .69 | .53 | .599 | .448 | .344 | .389 | .586 | .606 | .596 | .582 | .602 | .592 | .592 | .394 | .474 | .491 | .327 | .393 | .312 | .068 | .112 | .250 | .055 | .09 |
| Base + CE | .706 | .555 | .622 | .483 | .380 | .426 | .67 | .534 | .594 | .67 | .534 | .594 | .683 | .447 | .54 | .589 | .385 | .466 | .154 | .027 | .047 | .077 | .014 | .023 |
| Base + CE + FT | .726 | .53 | .613 | .527 | .384 | .445 | .766 | .500 | .605 | .766 | .500 | .605 | .722 | .332 | .455 | .636 | .292 | .401 | **.5** | **.082** | **.141** | **.5** | **.082** | **.141** |
| Base + CE + BERT | .782 | .734 | .757 | .571 | .536 | .553 | .75 | .508 | .606 | .75 | .508 | .606 | .7 | .500 | .583 | .623 | .445 | .52 | .333 | .027 | .051 | .333 | .027 | .051 |
| Base + CE + BERT – newly | .806 | .594 | .684 | .496 | .365 | .421 | .500 | .508 | .504 | .500 | .508 | .504 | .565 | .09 | .156 | .42 | .067 | .116 | .000 | .000 | .000 | .000 | .000 | .000 |
| Base + CE + FT + BERT | .791 | **.763** | .777 | .594 | .574 | .584 | .649 | **.610** | **.629** | .649 | **.610** | **.629** | .703 | .582 | .637 | .585 | .485 | .53 | .250 | .014 | .026 | .250 | .014 | .026 |
| Base + CE + FT + BERT – newly | **.84** | .679 | .751 | .615 | .497 | .55 | .744 | .517 | .610 | .744 | .517 | .610 | **.792** | .397 | .529 | .699 | .35 | .467 | .250 | .007 | .013 | .000 | .000 | .000 |
| Base + CE + FT + BERT (single) | .839 | .743 | **.788** | **.669** | **.593** | **.629** | .667 | .525 | .588 | .645 | .508 | .569 | .718 | **.588** | **.647** | .632 | **.517** | **.569** | .000 | .000 | .000 | .000 | .000 | .000 |

Table 4: English, segment split (The best result per table and column is given in bold, the second best result is underlined.)

| Configuration | Literal coarse | | | | | |
|---|---|---|---|---|---|---|
| | Fuzzy | | | Strict | | |
| | P | R | F | P | R | F |
| Baseline (organisers) | **.736** | .454 | **.562** | **.531** | **.327** | **.405** |
| Baseline (ours) | .377 | **.612** | .466 | .190 | <u>.31</u> | .236 |
| Base | .32 | .444 | .372 | .143 | .198 | .166 |
| Base + CE | .315 | .576 | .407 | .132 | .241 | .17 |
| Base + CE + FT | .261 | <u>.611</u> | .366 | .106 | .247 | .148 |
| Base + CE + BERT | .442 | .442 | .442 | .174 | .174 | .174 |
| Base + CE + BERT – newly | .419 | .568 | <u>.482</u> | .195 | .265 | .225 |
| Base + CE + FT + BERT | .391 | .506 | .441 | .191 | .247 | .216 |
| Base + CE + FT + BERT – newly | <u>.457</u> | .455 | .456 | <u>.237</u> | .236 | .237 |
| Base + CE + FT + BERT (single) | .396 | .508 | .445 | .22 | .283 | <u>.248</u> |
| + Fine-tuning (unfreezing) BERT | | | | | | |
| Base + CE + BERT | .374 | .566 | .450 | .118 | .178 | .142 |
| Base + CE + BERT – newly | .442 | .509 | .473 | .143 | .165 | .153 |
| Base + CE + FT + BERT | .403 | .530 | .458 | .171 | .225 | .194 |
| Base + CE + FT + BERT – newly | .399 | .518 | .451 | .187 | .243 | .211 |

## 4 Conclusion

Our model achieves its best performance on French followed by German, while we do not consider our English results to be yet useful for comparison, given the limitations of the dataset we used. A few results clearly emerge:

– Each embedding module contributes to the overall performance on most sub-tasks and evaluation metrics. Character-level and BERT embeddings are particularly important for performance, while in-domain FastText embeddings seem to help in particular for tags other than literal.

– Fine-tuning pre-trained embeddings in general does not improve performance, despite requiring more computation resources.

– A single-task approach performs better than multi-task in general, even if the differences are often minor. It must be noted that, with our setup, six single-task runs require 2.5 times more time on average to converge than one multi-task run using a document split. Instead, six single-task runs using a multi-segment split are as fast as one multi-task run. Comparing one-to-one, a single-task run is on average twice as fast.

– A document-level split of the data is in general better than a multi-segment split, highlighting how larger windows of context are helpful with our model.

When compared to the results of the other task participants [5], our model performs well in general, and notably well on the NERC-Fine sub-task where we achieve best performance on several evaluation metrics and in particular a high precision.

## References

1. Bos, J., Basile, V., Evang, K., Venhuizen, N.J., Bjerva, J.: The Groningen Meaning Bank. In: Handbook of linguistic annotation, pp. 463–496. Springer (2017)
2. Chronopoulou, A., Baziotis, C., Potamianos, A.: An embarrassingly simple approach for transfer learning from pretrained language models. 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). pp. 2089–2095 (2019)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423
4. Ehrmann, M., Colavizza, G., Rochat, Y., Kaplan, F.: Diachronic evaluation of ner systems on old newspapers (2016), https://infoscience.epfl.ch/record/221391
5. Ehrmann, M., Romanello, M., Flückiger, A., Clematide, S.: Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Working Notes. Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum. CEUR-WS (2020)
6. Ghaddar, A., Langlais, P.: Robust lexical features for improved neural network named-entity recognition. In: Proceedings of the 27th International Conference on Computational Linguistics. p. 1896–1907. Association for Computational Linguistics (Aug 2018), https://www.aclweb.org/anthology/C18-1161
7. Joulin, A., Grave, É., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431 (2017)
8. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. p. 282–289. ICML '01, Morgan Kaufmann Publishers Inc. (Jun 2001)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. p. 260–270. Association for Computational Linguistics (Jun 2016). https://doi.org/10.18653/v1/N16-1030
10. Loshchilov, I., Hutter, F.: Fixing Weight Decay Regularization in Adam (Feb 2018), https://openreview.net/forum?id=rk6qdGgCZ
11. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (Oct 2010). https://doi.org/10.1109/TKDE.2009.191

12. Piotrowski, M.: Natural language processing for historical texts. Synthesis Lectures on Human Language Technologies **5**(2), 1–157 (Sep 2012). https://doi.org/10.2200/S00436ED1V01Y201207HLT017
13. Ruder, S.: Neural Transfer Learning for Natural Language Processing p. 329 (Feb 2019)
14. Sang, E.T.K., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2003, Edmonton, Canada. pp. 142–145. Morgan Kaufman Publishers (2003)
15. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to Fine-Tune BERT for Text Classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) Chinese Computational Linguistics. p. 194–206. Lecture Notes in Computer Science, Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-32381-3_16
16. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs] (Feb 2020), `http://arxiv.org/abs/1910.03771`, arXiv: 1910.03771