WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER

Simone Tedeschi[1], **Valentino Maiorca**[1], **Niccolò Campolungo**[2]
Francesco Cecconi[1] and **Roberto Navigli**[2]
[1]Babelscape, Italy
[2]Sapienza NLP Group, Sapienza University of Rome
{tedeschi, maiorca, cecconi}@babelscape.com,
campolungo@di.uniroma1.it, navigli@diag.uniroma1.it

Abstract

Multilingual Named Entity Recognition (NER) is a key intermediate task which is needed in many areas of NLP. In this paper, we address the well-known issue of data scarcity in NER, especially relevant when moving to a multilingual scenario, and go beyond current approaches to the creation of multilingual silver data for the task. We exploit the texts of Wikipedia and introduce a new methodology based on the effective combination of knowledge-based approaches and neural models, together with a novel domain adaptation technique, to produce high-quality training corpora for NER. We evaluate our datasets extensively on standard benchmarks for NER, yielding substantial improvements of up to 6 span-based F_1-score points over previous state-of-the-art systems for data creation.

1 Introduction

Named Entity Recognition (NER) is the task of identifying specific words as belonging to predefined semantic types, such as Person, Location, Organization, etc. (Nadeau and Sekine, 2007). NER is widely used in many downstream tasks, like question answering (Mollá et al., 2006), machine translation (Babych and Hartley, 2003), information retrieval (Petkova and Croft, 2007), text summarization (Aone et al., 1998), text understanding (Zhang et al., 2019; Cheng and Erk, 2019) and entity linking (Tedeschi et al., 2021), among others.

With recent advances in Natural Language Processing, and in particular with the advent of pretrained language models such as BERT (Devlin et al., 2019), once a sufficient amount of training data is available for the task of interest, fine-tuning is often employed to address the task successfully. Unfortunately, such training data are scarce and expensive to create, especially when labels are fine-grained and many languages have to be covered, as is the case for NER.

Various works have been put forward which address data paucity by aiming at automatically producing multilingual silver-standard training data for NER (Nothman et al., 2013; Al-Rfou et al., 2015; Tsai et al., 2016; Pan et al., 2017). Each of these leverages the link structure of Wikipedia to generate named entity annotations. However, this strategy has two drawbacks: only small portions of text in Wikipedia are linked, and mapping Wikipedia links to the corresponding NER classes is not trivial and introduces errors. Different methods have been investigated to cope with these problems, such as heuristics based on Wikipedia redirects, surface form token matching, and category-based rules.

Although we also rely on Wikipedia text and its hypertext organization, we depart from previous works in our exploration of new language-independent techniques for silver data creation for NER by providing a general approach based on an effective combination of knowledge-based techniques and neural models.

Our contributions are as follows:

1. We propose a novel technique which builds upon external knowledge bases and pretrained language models to produce high-quality annotations for multilingual NER;
2. We assess the quality of the corpora produced with an extensive evaluation and a statistical analysis, showing consistent improvements of up to 6 span-based F_1-score points on common benchmarks for NER against state-of-the-art alternative data production methods;
3. We present a novel approach for creating interpretable word embeddings;
4. Based on these embeddings, we introduce a domain adaptation algorithm which yields further performance gains on all test settings.

We release data and software at https://github.com/Babelscape/wikineural.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2521–2533
November 7–11, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Since the first shared task on NER organized by Grishman and Sundheim (1996), several tasks and human-annotated datasets have been proposed. The CoNLL-2002 and 2003 datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) were created in four different languages (Spanish, Dutch, English, and German) from newswire articles and focused on 4 entity types: PER (Person), ORG (Organization), LOC (Location), and MISC (Miscellaneous, i.e., all other entity types). Several other NER shared tasks were organized in the years which followed, covering further languages such as Indic (Rajeev Sangal and Singh, 2008) and Balto-Slavic languages (Piskorski et al., 2017).

Early NER systems were based on domain-specific features and rules, which require human engineering. Starting from (Collobert et al., 2011), neural NER architectures requiring minimal feature engineering have become enduringly popular (Li et al., 2018). Nevertheless, to fully benefit from these systems, large amounts of data for training are required. Although various NER datasets have been created, they have remained a scarce resource, available only for a narrow set of languages. Moreover, they have often been small in size, limited to a few domains, and genre-specific (e.g., news). For these reasons, over the last two decades, various works have been carried out to turn Wikipedia texts into multilingual NER training corpora.

Nothman et al. (2013) introduced WikiNER, a pipeline to automatically create multilingual training data for NER by exploiting the structure and the texts of Wikipedia. First, they classified each Wikipedia document into named entity types, training and evaluating on manually-labeled Wikipedia articles across 9 languages. Then, Wikipedia links were converted into labels by classifying the target articles into entity types (PER, ORG, LOC, MISC). Finally, heuristics based on redirects were applied to infer more named entity mentions. Interestingly, Nothman et al. (2013) showed that, when testing on manually-annotated Wikipedia sentences, models trained on gold-standard newswire datasets perform poorly compared to models trained on automatically-created Wikipedia corpora.

Similarly, Pan et al. (2017) proposed WikiANN, a language-independent framework to automatically extract name mentions from documents by leveraging Wikipedia markups. Specifically, they first classified English Wikipedia entries into certain entity types, and then they applied a cross-lingual entity transfer to propagate these labels to other languages.

Other works relied on Freebase (Bollacker et al., 2008), a sizeable collaborative graph database, either by using its association to English Wikipedia as a training feature (Tsai et al., 2016), or by mapping its attributes to entity types, in order to identify the NER classes (Al-Rfou et al., 2015). Moreover, Al-Rfou et al. (2015) also tried to overcome the issue of missing annotations of non-anchored mentions in Wikipedia by using a simple surface string-matching heuristic, and resampled the datasets they produced to reduce the high class imbalance.

In our work we follow this same direction but introduce several contributions based on a novel combination of knowledge-based and neural techniques that lead to considerable improvements. To the best of our knowledge, we are the first to exploit multilingual BERT's power in a silver data creation process for NER: we use it to independently i) distinguish named entities from concepts, ii) validate annotations and, iii) discover annotations. Further, to address the sparsity problem, rather than relying on often-noisy redirections, we exploit the synonymy information provided by a multilingual lexical knowledge base, i.e., BabelNet[1] (Navigli and Ponzetto, 2012; Navigli et al., 2021).

3 WikiNEuRal

We now describe our approach to producing multilingual silver-standard training data for Named Entity Recognition. A graphical representation of the steps that characterize the WikiNEuRal annotation pipeline is depicted in Figure 1.

3.1 Preprocessing Wikipedia

We clean up the text of Wikipedia articles by removing the sections with the 10 most frequent titles, which usually list related resources (e.g., bibliography, references, see also). We also remove other elements that tend to introduce noise, such as lists, tables, templates, formulas, etc.

The remaining elements are Wikilinks, which provide potential entity mentions, and may show up either with links only (e.g., [[apartment]]) or with both link and surface form (e.g., [[apartment|flats]]). We opt to discard all occurrences of this latter because such Wikilinks might

[1] https://babelnet.org

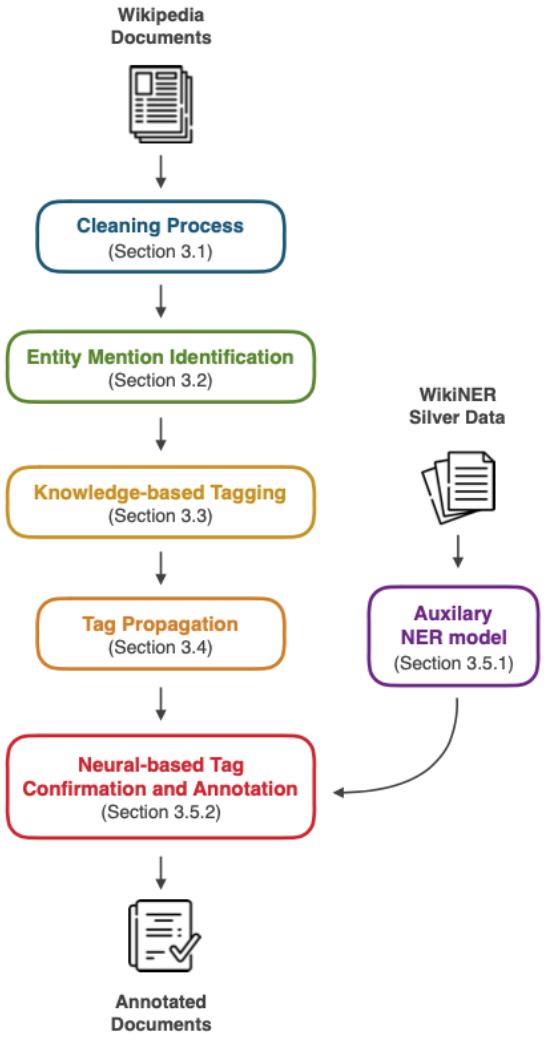

Figure 1: WikiNEuRal annotation pipeline.

introduce errors (e.g., in [[Royal Dutch Shell|oil industry company]], the surface text *oil industry company* would be tagged as ORG).

3.2 Identifying Entity Mentions in Wikipedia

Although Wikilinks offer valuable information, not all links point to a Named Entity (e.g., see [[apartment]] above, which denotes a concept, instead). We therefore exploit the one-to-one correspondence between Wikipedia articles and BabelNet synsets to classify each synset into either an abstract concept (C) or a named entity (NE), which we will refer to as its *type*. Although BabelNet offers this information for most of its synsets, upon manual inspection we find that many of these *type annotations* are noisy[2], so we opt to train a model to perform binary classification, effectively replacing the annotations provided by BabelNet.

The dataset for this task is constructed exploiting the multilingual nature of BabelNet: given a synset s and a language L, we have access to a set of glosses $G_L(s)$. For every such gloss $g \in G_L(s)$ we generate a sample with the type as label, and provide the string $I(s, g, L) =$ [CLS] $l_L(s) \mid g$ [SEP] as input, where $l_L(s)$ is the main lemma for synset s in language L or, if it does not exist, a special token [NOLEMMA], shared among languages. We rank BabelNet synsets according to the number of glosses they contain and take the top 500k synsets to build our training and validation splits (450k and 50k respectively). As this system represents only one step of the entire WikiNEuRal pipeline, we measure its quality in vivo (see Section 7), as a test set generated with the same distribution would yield inconclusive results.

To tag each BabelNet synset (and therefore each Wikipedia article) as either a concept or a named entity, we follow Devlin et al. (2019) and use a simple yet efficient Sequence Classification architecture, using Multilingual BERT as encoder and a linear layer to perform binary classification on top of the [CLS] token. The model is trained on the inputs generated by the function I applied to the aforementioned 500k synsets. Lastly, in order to classify any particular synset, we gather all its possible inputs (i.e., we take into account all of its glosses in the considered languages $\hat{L}$[3]) $I(s) = \{I(s, L, g) | L \in \hat{L},\ g \in G_L(s)\}$ and label them independently using the trained model. Then, we select, for synset s, the label with the maximum average confidence[4] – with this label becoming the new *type* for synset s and, therefore, for its linked Wikipedia page, regardless of the language. We use this label to discard links to Wikipedia articles corresponding to concept types, as we are only interested in Named Entities.

3.3 Tagging Named Entity Links Through Synsets

We now aim at providing each named entity link in a Wikipedia article with a common NER class (PER, ORG, LOC, MISC). Once again, we exploit the one-to-one linkage between Wikipedia articles and BabelNet synsets and classify all synsets in the

[2]Based on the majority case of the title occurrence within its Wikipedia article.

[3]We only take glosses in $\hat{L} =$ {English, Italian, German, Spanish, French}, as they are the best supported languages between BabelNet and Multilingual BERT.

[4]We ignore predictions for which the averaged confidence falls below 0.6.

nominal taxonomy of BabelNet. To achieve this, we start by selecting and annotating 200 synsets to cover as many high-order concepts of the WordNet (Miller, 1995) nominal taxonomy[5] as possible. Then, to descend the WordNet taxonomy, a Breadth-First Search (BFS) algorithm on hyponymy relationships is employed to resolve multi-parent collisions, resulting in an expanded set of 40,000 high-quality annotations. Finally, to annotate all the remaining BabelNet synsets, we again apply a BFS (with max depth set to 2) following hypernymy relationships until one of the 40k synsets in the expanded set is reached. The annotation of the first hypernymous synset reached through BFS (if any) is inherited. This procedure yields a classification of more than 7.5 million synsets corresponding to named entities.

3.4 Named Entity Tag Propagation

As a result of the above steps, for each Wikipedia article we know which anchored strings correspond to named entities and which NER classes they belong to. As anticipated in Section 1, one of the major drawbacks in using Wikipedia texts is that only small portions of text are anchored. This includes the fact that, according to the Wikipedia guidelines, only the first mention to the article has to be linked. Since every anchored string a corresponds to a BabelNet synset s, we gather the synonyms in s (including the anchored text itself) in the language of the article, and employ an exact matching heuristic to propagate the named entity tag of a to any occurrence of synonyms of s throughout the whole article. For instance, "Apple" is among the synonyms of "Apple Inc.", hence all its occurrences within a text in which the latter link occurs are tagged as ORG, leading to denser annotations.

3.5 Confirming and Augmenting Annotations

We are now left with two potential weaknesses: first, even though aiming for high precision, our annotations might still include some mistakes; second, our tagged sentences might still contain unannotated entities due to unlinked or unmatched entity mentions. To address both issues, we first introduce our NER classifier, which we will use to perform tagging with our annotated data, then we apply it to confirm our annotations and augment the sentences with additional tags.

[5]We initially restrict to WordNet synsets as they are manually curated.

3.5.1 Our NER model

Our NER model is a variant of the BERT-based neural model of Mueller et al. (2020): following recent literature, rather than representing a word with the first contextualized subword representation as provided by multilingual BERT, we take the mean of its subwords (Ács et al., 2021). The resulting vectors are passed through a multi-layer sentence-level BiLSTM network, whose logits are then fed into a CRF model (Lafferty et al., 2001), trained to maximize the log-likelihood of the span-based gold label sequences.

3.5.2 Improving Precision and Recall

To address the above-mentioned weaknesses, we train our NER model with the WikiNER dataset (cf. Section 2) and use it to confirm and augment our annotations. More formally, for every sentence x composed of n tokens $x_1, \ldots, x_n$, we compare the annotation (i.e., a named entity tag) y_i produced by our approach for each token x_i with the one produced by the neural model $\hat{y}_i$, and keep the sentence if i) there is at least one annotation $y_i \neq O$, and ii) every $y_i \neq O$ has the same annotation of the corresponding $\hat{y}_i$. This results in an improved precision of our annotations, as they are confirmed through an ensemble approach. Finally, we output as annotations for sentence x the labels produced by the neural model $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_n]$, therefore accounting for previously undiscovered entities and improving recall.

4 Domain Adaptation

Thanks to the use of Wikipedia, our automatically-created datasets cover a wide range of domains. However, in many cases tests are performed on a limited set of domains. To address this issue and enable the production of domain-fitting NER datasets, here we introduce our methodology for performing domain adaptation. This consists of a domain extraction technique which, later combined with the approach presented in Section 3, enables the creation of domain-adapted training data for NER systems when given domain-specific texts.

4.1 Category selection

We first select a general subset of Wikipedia categories, under the assumption that they reflect all domains. Let us start by considering the directed category graph $G = (C, E)$ of (English) Wikipedia, where a node $c \in C$ represents a Wikipedia category and $(c_1, c_2) \in E$ is an edge representing

that c_1 is a parent category of c_2. First, since $|C| \approx 1.6$M (as of September 2020), we need to find a subset $\hat{C}$ of C such that the coverage of knowledge fields is maximized, whereas the word vector dimensionality is minimized. We compute $\hat{C}$ by taking all nodes in G with depth from the root node ≤ 2, yielding a total of ~ 1200 categories.

Since the majority of Wikipedia articles do not belong to any of the selected $\hat{C}$ categories, we need to compute a distribution over $\hat{C}$ for every category $c \in C \setminus \hat{C}$. We follow the intuition that the number of random walks from c to $\hat{c} \in \hat{C}$ is proportional to $P(\hat{c}|c)$; hence, we compute:

$$P(\hat{c}|c) = \frac{1}{k} \cdot \sum_{i=1}^{k} RW(c, \hat{c})$$

where k is the number of random walks performed and $RW(c, \hat{c}) = 1$ if a random walk starting from node c and walking only to parent nodes ends up on category $\hat{c}$.

Now, given an article w and its associated set of categories C_w,

$$P(\hat{c}|w) = 1 - \prod_{c \in C_w} (1 - P(\hat{c}|c)) \; \forall \; \hat{c} \in \hat{C}$$

can be interpreted as the probability of associating $\hat{c}$ with one of the categories of w; we discard this association in the case that $P(\hat{c}|w) < \sigma$.[6] The next natural step is to prune $\hat{C}$ so as to further reduce categories that are either too general or too specific. For all $\hat{c}$, we compute the unconditioned probability

$$P(\hat{c}) = \frac{1}{|W|} \cdot \sum_{w \in W} P(\hat{c}|w),$$

where W is the set of all Wikipedia articles, compute the median value m_v of all $P(\hat{c}) \; \forall \; \hat{c} \in \hat{C}$ and take the 600 elements of $\hat{C}$ closest to m_v, yielding the final set of supercategories $\mathcal{S} \subset \hat{C}$, which cover the general knowledge encoded in Wikipedia in a concise way.

4.2 Domain embedding computation and domain extraction

We now use the above category selection to produce both interpretable and domain-aware embeddings, which we then use to select the best-fitting Wikipedia articles for producing our Named Entity tagged dataset. Let us now consider all Wikipedia articles W, a token t occurring in any of its articles, a supercategory $s \in \mathcal{S}$, the set of articles W_s associated with s, and a function f which takes as input a token t' and a collection D of Wikipedia articles, and returns the number of the occurrences of t' within the documents of D; we compute the relevance score of token t in supercategory s as:

$$P(t|s) = \frac{P(s|t)P(t)}{P(s)}$$

where:

$$P(t) = \frac{f(t, W)}{\sum_{t' \in W} f(t', W)}$$
$$P(s) = \frac{\sum_{t \in W_s} f(t, W_s)}{\sum_{t' \in W} f(t', W)}$$
$$P(s|t) = \frac{f(t, W_s)}{f(t, W)}.$$

By repeating the above computation for every token $t \in W$ (excluding stopwords) and every supercategory $s \in \mathcal{S}$, we obtain a large matrix $\mathbb{E}^{n \times m}$,[7] i.e., our category embeddings for the selected language.

The procedure for exploiting the above-mentioned embeddings in order to extract the main categories from a corpus of documents is formally described in Algorithm 1. The algorithm's core is a hierarchical aggregation of the probability distribution of tokens over categories: first, it averages the token-level distributions M to obtain a document-level distribution Z. Then it proceeds by taking the main categories C across the whole set of document distributions. Finally, only categories that appear at least δ times are considered as categories of that corpus. These extracted categories will be used to select the Wikipedia pages for silver-standard training data production which best fit the input document domains.

5 Experimental Setup

5.1 Reference model

We use the NER model introduced in Section 3.5.1 to compare our produced dataset's impact against competitors. All models are trained with early stopping set with a patience parameter of 10; we use Adam (Kingma and Ba, 2015) with learning rate fixed at 10^{-3} and a cross-entropy loss criterion. The full list of hyperparameter values is shown

[6] A manually-tuned threshold, set at $3 \cdot 10^{-4}$.

[7] n is the size of the Wikipedia vocabulary, $m = |\mathcal{S}| = 600$.

Algorithm 1 Domain extraction procedure

Inputs: Corpus D, Category embeddings $\mathbb{E}^{n \times m}$, Counting function f
Parameters: Threshold δ, Token-level top-k k_t, Document-level top-k k_d
Output: Main categories in D

$$\gamma(\mathbf{x}, k) \rightarrow [\hat{x}_1, \cdots, \hat{x}_n], \hat{x}_i = \begin{cases} 1 & \text{if } \hat{x}_i \in \text{top}_k(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}$$

$C \in \mathbb{N}^m \mid c_i = 0 \; \forall i \in [1, m]$
for $d \in D$ **do**
  $\mathbf{v} \in \mathbb{N}^n, \mathbf{v}_i = f(w_i, d) \; \forall w_i \in \mathbb{E}$
  $M \leftarrow \mathbf{v}^T \cdot \mathbb{E}$
  $R \in \mathbb{R}^{n \times m}, R_i = M_i^T \cdot \gamma(M_i, k_t)$
  $Z \in \mathbb{R}^m, Z_i = \frac{1}{n} \cdot \sum_{i=1}^{n} (R^T)_i \; \forall i \in [1, m]$
  $C \leftarrow C + \gamma(Z, k_d)$
end for
return $\{i \mid C_i \geq \delta \; \forall i \in [1, |\mathcal{S}|]\}$

in Table 1. We repeat each training on 10 different seeds, fixed across experiments, and report the mean and standard deviation of their span F_1 score; we compare experiments by means of Student's t-test (Student, 1908). Further details about the hyperparameter search, training times and hardware infrastructure are provided in Appendix A.

5.2 Training Data

We train our reference model with four different silver-standard datasets:

- **WikiNEuRal**: the dataset created using the methodology described in Section 3 from Wikipedia[8]. It covers 9 languages: Dutch, English, French, German, Italian, Polish, Portuguese, Russian and Spanish. Data statistics are shown in Table 2.

- **WikiNEuRal+DA**: We apply our domain adaptation technique (Section 4) to filter the Wikipedia articles used to create our training data and fit them to the domains of the test data. To this end, we use the CoNLL and OntoNotes test sets[9] where, except for the Spanish CoNLL test set, this kind of document split is provided. We perform domain extraction as described in Algorithm 1 with parameters $\delta = 5$, $k_t = 50$, $k_d = 5$ to the test set documents; thus, we provide WikiNEuRal with articles whose domains, i.e., categories, match the ones extracted for the targeted corpus. Statistics are shown in Table 2.

- **WikiNER** (Nothman et al., 2013): the current best-performing approach for NER silver data creation. It covers the same languages as WikiNEuRal.

- **WikiANN**[10] (Pan et al., 2017): a multilingual NER dataset consisting of Wikipedia articles annotated in 282 languages.

Hyperparameter name	Value
number of Bi-LSTM layers	2
LSTM hidden size	512
batch size	128
learning rate	0.001
dropout	0.5
gradient clipping	1.0
adam β_1	0.9
adam β_2	0.999
adam ϵ	1e-8

Table 1: Hyperparameter values of the reference model used for our experiments.

We also train our reference model for every available, manually-annotated gold-standard training set from the **CoNLL-2002** NER Shared Task (Tjong Kim Sang, 2002) for Spanish and Dutch, the **CoNLL-2003** NER Shared Task (Tjong Kim Sang and De Meulder, 2003) for English and German, and the **OntoNotes 5.0** dataset for English. All silver- and gold-standard datasets are tagged with the four standard entity types (PER, ORG, LOC, MISC), except for WikiANN which does not contain the MISC label.

5.3 Test Data

We use five different test sets in our experiments:

- **CoNLL-2002** NER Shared Task dataset (Tjong Kim Sang, 2002): a popular collection of newswire articles for Spanish and Dutch.

- **CoNLL-2003** NER Shared Task dataset (Tjong Kim Sang and De Meulder, 2003): a

[8] July 2020 snapshot for all languages, sampling random articles.

[9] We strongly emphasize that we do not use anything from the test sets except their raw text.

[10] The version used corresponds to the balanced train, dev, and test splits of Rahimi et al. (2019), which supports 176 of the 282 languages from the original WikiANN corpus, available at https://huggingface.co/datasets/wikiann.

WikiNEuRal	Articles	Sentences	Tokens	Avg. length	Avg. NEs	PER	ORG	LOC	MISC	OTHER
English	50k	116k	2.73M	23.53	1.67	51k	31k	67k	45k	2.40M
Spanish	50k	95k	2.33M	24.46	1.61	43k	17k	68k	25k	2.04M
Dutch	65k	107k	1.91M	17.91	1.43	46k	22k	61k	24k	1.64M
German	50k	124k	2.19M	17.66	1.42	60k	32k	59k	25k	1.87M
Russian	105k	123k	2.39M	19.49	1.47	40k	26k	89k	25k	2.13M
Italian	50k	111k	2.99M	26.85	1.90	67k	22k	97k	26k	2.62M
French	50k	127k	3.24M	25.47	1.80	76k	25k	101k	29k	2.83M
Polish	105k	141k	2.29M	16.21	1.65	59k	34k	118k	22k	1.91M
Portuguese	80k	106k	2.53M	23.99	1.88	44k	17k	112k	25k	2.20M
English DA (CoNLL)	20k	29k	759k	21.41	1.55	12k	23k	6k	3k	0.54M
Dutch DA (CoNLL)	25k	34k	598k	17.69	1.44	17k	8k	18k	6k	0.51M
German DA (CoNLL)	20k	41k	706k	17.29	1.37	17k	12k	23k	3k	0.61M
English DA (OntoNotes)	35k	48k	1.18M	24.31	1.70	20k	13k	38k	12k	1.02M

Table 2: Statistics on the produced data on a fixed number of articles. "Avg. length" is the average sentence length and "Avg. NEs" is the average number of named entities per sentence. DA stands for Domain Adaptation.

well-known collection of newswire articles for English and German taken from the Reuters Corpus and the ECI Multilingual Text Corpus, respectively.

- **WikiGold** (Balasuriya et al., 2009): a small set of English Wikipedia articles manually annotated with CoNLL named entity classes.
- **OntoNotes 5.0** (Pradhan et al., 2012): this includes texts from five different text genres: broadcast conversation, broadcast news, magazine, newswire, and web data. We use it as an additional test set for English.
- **BSNLP-2017** (Piskorski et al., 2017): this consists of articles in various Slavic languages and we use it to evaluate Russian and Polish performances. Two test sets are provided: one contains articles about a specific politician, the other one about the European Commission.

All the datasets employ the CoNLL entity types (PER, ORG, LOC and MISC), except OntoNotes, which is annotated with 18 fine-grained entity types, which we manually map to the CoNLL tag set. Further details about how the OntoNotes classes are mapped to the CoNLL ones are provided in Appendix C. For CoNLL, OntoNotes, and BSNLP, which are often used to benchmark NER, we take the official splits for validation and test sets. For WikiGold, which is much smaller, we use the full dataset as test material. Following the literature, we evaluate performances by means of the F_1 score, i.e., the harmonic mean between Precision and Recall, using the official conlleval script. We convert all datasets to the popular BIO format.

6 Results

6.1 Multilingual Evaluation

We assess the quality of the WikiNEuRal datasets extensively, comparing the performances obtained training the model presented in Section 3.5.1 both on WikiNEuRal and on the other datasets listed in Section 5.2. The results are reported in Table 3. We observe consistent improvements of WikiNEuRal over the WikiNER and WikiANN alternatives on almost all tested languages and datasets. In particular, on the CoNLL test sets, we notice an average improvement, computed over the scores on the four languages, of 21.4 and 2.3 F_1 points over WikiANN and WikiNER, respectively. Three out of four results are also statistically significant. Morever, in the remaining test sets (i.e., WikiGold, OntoNotes and BSNLP), our approach achieves results which are better than the results of the two competitors, again in a statistically significant way.

Finally, we also show how WikiNEuRal-based models perform 8.2 F_1 points better than CoNLL-trained models on the WikiGold test set, and almost 1 point better when tested on neutral test sets, namely corpora from sources different from both WikiNEuRal and CoNLL training sets (i.e., OntoNotes). Similarly, WikiNEuRal-based models perform 10.6 F_1 points better than OntoNotes-trained models on the WikiGold test set, and almost 4 points better when tested on neutral test sets, i.e., CoNLL.

6.2 Silver- and Gold-Standard Data Aggregation

In order to further demonstrate the quality of the data produced, we aggregate WikiNEuRal with

Training set \ Test set	CoNLL				WikiGold	OntoNotes	*BSNLP*	
	English	Spanish	Dutch	German	English	English	Russian	Polish
WikiANN	56.85 ± 2.18	53.55 ± 3.10	55.76 ± 2.19	44.39 ± 0.95	57.05 ± 2.76	36.43 ± 4.54	51.85 ± 1.80	53.50 ± 1.63
WikiNER	73.05 ± 1.20	75.07 ± 0.96	74.75 ± 0.59	64.03 ± 1.86	81.98 ± 0.28	71.16 ± 0.72	65.99 ± 0.94	62.31 ± 0.95
WikiNEuRal	76.94 ± 0.75	**77.87 ± 0.85**	77.40 ± 0.57	64.02 ± 0.54	**82.42 ± 0.33****	71.98 ± 0.55*	**66.50 ± 0.67**	**62.44 ± 1.00**
WikiNEuRal DA	**79.07 ± 0.51**	-	**79.07 ± 0.52**	**68.33 ± 0.46**	-	**74.38 ± 0.30**	-	-
CoNLL	90.07 ± 0.33	**86.78 ± 0.44**	89.48 ± 0.69	77.57 ± 0.51	74.22 ± 0.45	71.03 ± 0.47	-	-
+ WikiANN	88.58 ± 0.22	86.66 ± 0.37	85.08 ± 0.49	74.94 ± 0.36	68.93 ± 0.58	67.72 ± 0.49	-	-
+ WikiNER	89.28 ± 0.27	85.80 ± 0.45	85.88 ± 0.51	74.10 ± 0.20	82.08 ± 0.48	72.10 ± 0.36	-	-
+ WikiNEuRal	89.95 ± 0.20	86.49 ± 0.51	89.24 ± 0.23	77.97 ± 0.46	**82.83 ± 0.34**	73.78 ± 0.18	-	-
+ WikiNEuRal DA	**90.14 ± 0.28**	-	**89.50 ± 0.39**	**78.78 ± 0.59**	-	**75.11 ± 0.22**	-	-
OntoNotes	73.39 ± 0.60	-	-	-	71.59 ± 0.42	**89.39 ± 0.39**	-	-
+ WikiANN	72.80 ± 0.82	-	-	-	69.00 ± 1.04	88.30 ± 0.71	-	-
+ WikiNER	75.31 ± 0.51	-	-	-	**82.21 ± 0.35**	87.15 ± 0.25	-	-
+ WikiNEuRal	77.19 ± 0.48	-	-	-	82.04 ± 0.34	87.90 ± 0.71	-	-
+ WikiNEuRal DA	**89.21 ± 0.36**	-	-	-	-	88.77 ± 0.18	-	-

Table 3: Span-based micro F_1 scores on common NER benchmarks. DA stands for Domain Adaptation. Statistical significance is computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance scores are computed between a system and its next best scoring competitor (e.g., WikiNEuRal vs WikiNER, or WikiNEuRal DA vs WikiNEuRal). Further results are provided in Appendix B.

Version	CoNLL	WikiGold	OntoNotes
WikiNEuRal DA	**79.07 ± 0.51**	-	**74.38 ± 0.30**
- Domain Adaptation	76.94 ± 0.75	**82.42 ± 0.33****	71.98 ± 0.55
- Concept vs NE	76.24 ± 0.35	81.66 ± 0.22	71.69 ± 0.24
- NE Augmentation	68.34 ± 0.64**	75.83 ± 0.36	62.84 ± 0.40
- NE Confirmation	64.60 ± 0.56**	70.98 ± 0.42	58.57 ± 0.21
- NE Discrimination	59.19 ± 0.63	64.46 ± 1.21	52.76 ± 1.28
- Tag Propagation	57.15 ± 1.53	63.36 ± 1.55	51.77 ± 1.25

Table 4: Span-based micro F_1 scores of WikiNEuRal versions on the three English CoNLL, WikiGold, and OntoNotes test sets. Statistical significance is computed using Student's t-test: ** stands for $p < 0.01$, underline stands for $p < 0.001$. Statistical significance is expressed with respect to the row immediately below.

manually-created datasets in the corresponding languages. Once again, as shown in Table 3, there is a general improvement when comparing models trained on WikiNEuRal and CoNLL against their concatenated counterpart: on average[11], the two datasets alone achieve a span F_1 score of 75.1 and 81.5, respectively, while their concatenation attains 83.4 F_1. Similar results can be observed when comparing models trained on WikiNEuRal and OntoNotes against their concatenated counterpart: on average, WikiNEuRal alone achieves a span F_1 score of 77.1 and OntoNotes alone reaches 78.1, while their concatenation attains 82.4 F_1.

Our analysis shows that, in real-world cases where gold training data are available but they do not match the target test set in terms of textual genre or domains covered (e.g., only manually-annotated news articles are available to train a user's system, but the user wants to test it on web documents), WikiNEuRal can be beneficial for handling domain generalization. This is the case when we test CoNLL+WikiNEuRal on OntoNotes or OntoNotes+WikiNEuRal on CoNLL, getting scores which are much higher than the ones obtained with the two gold-standard datasets alone. This shows how the domain coverage of datasets matters even with manually crafted training data, since CoNLL and OntoNotes have different text genres and mismatched topics. WikiNEuRal boosts the domain coverage regardless of the starting data and, therefore, helps to cope with this problem.

6.3 Results on Domain Adaptation

The results reported in Table 3 show that Domain Adaptation consistently improves performances over already state-of-the-art results, while requiring much less training data compared to the standard WikiNEuRal version (see Table 2). On average, domain-adapted datasets attain a 2.6, 1.3 and 6.4 span F_1 improvement over WikiNEuRal, CoNLL+WikiNEuRal and OntoNotes+WikiNEuRal, respectively. This means that the domain-adapted datasets are strongly biased towards the domains targeted by the adaptation technique, as expected.

7 Ablation Study

In order to show the effectiveness of the steps detailed in Section 3, we disassembled our NER data creation pipeline. We conducted these experiments on the English WikiNEuRal corpus; results are shown in Table 4. We first removed *Domain Adaptation* (Section 4), whose benefits have already

[11] Computed on all datasets for which results for the three alternatives are available.

been discussed in Section 6.3. Second, we removed the *Concept vs. Named Entity* module of Section 3.2 and simply relied on the entity typing provided by BabelNet. The result is a drop in performances (second and third rows of the Table), confirming the need for this kind of validation. Then, we removed the *named entity augmentation* module presented in Section 3.5, i.e., named entities that are neither anchored in Wikipedia articles nor identified as synonyms of anchored ones, are no longer caught by the neural model. This removal causes a reduction of annotations, which results in an average decrease – over the three test sets – of 7.53 F_1-score points. Subsequently, we also removed the *named entity confirmation* module, which used the BERT-based model to corroborate the annotations produced by the knowledge-based approach (Section 3.3). These annotations were obtained through an automatic approach, and our intuition suggested using a neural model to discard potentially imprecise sentences, which is confirmed by the further average decrease in performances of 4.29 points when removing it. At this stage, the neural model is only employed as a discriminator, i.e. it just outputs *NE* or *not NE* for each token. Hence, for each sentence, if there are tokens annotated as PER, ORG, LOC or MISC by the knowledge-based approach, but labeled as *not NE* by the model, the sentence is discarded. The removal of this *named entity discrimination* block again results in a worsening of performances by 5.91 points, on average. Finally, we also ablated the tag propagations of Section 3.4, i.e., we just left the tags associated with preexisting links in the article. Both tag propagation methods were used to increase the density of annotations, crucial for obtaining high-quality annotated sentences: in fact, their exclusion leads to a further deterioration in performance.

We can summarize this ablation study by pointing out an average gap of more than 21 F_1-score points between the final WikiNEuRal version detailed in Section 3 and the basic one, which only uses strings anchored in Wikipedia articles.

For completeness, we also constructed a baseline version of the WikiNEuRal dataset using just the neural model employed in Section 3.5.2 to annotate Wikipedia articles. The system trained on the resulting dataset achieved 69.46 $\pm$ 0.50 on CoNLL, 77.46 $\pm$ 0.43 on WikiGold and 64.53 $\pm$ 0.41 on OntoNotes, showing how the combination of neural and knowledge-based approaches adopted by the final version of WikiNEuRal is essential in order to achieve higher performances.

8 Conclusion

We presented WikiNEuRal, an automatic, language-independent approach for generating labeled datasets for NER in multiple languages. While we follow other silver-data creation approaches and exploit the hyperlinked texts of Wikipedia articles, we depart from past works in three fundamental aspects which integrate knowledge-based and neural approaches: i) we automatically type tags by utilizing the structure of a multilingual lexical-semantic knowledge base, BabelNet, ii) we exploit neural BERT-based models to discern entity from non-entity tags and iii) as a complementary approach to confirm and augment sentences with entity tags, iv) we put forward a domain adaptation technique which can produce NER training data for arbitrary domains.

We finally showed, through an extensive evaluation, that WikiNEuRal can be used to train competitive NER systems, providing substantial performance improvements over previous state-of-the-art approaches for silver-data creation. As future work, we plan to extend our study to produce named entity tags for a larger set of classes and languages.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

This work was also supported by the PerLIR project (Personal Linguistic resources in Information Retrieval) funded by the MIUR Progetti di ricerca di Rilevante Interesse Nazionale programme (PRIN 2017).

References

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. POLYGLOT-NER: massive

multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 586–594. SIAM.

Chinatsu Aone, Mary Okurowski, and James Gorlinsky. 1998. Trainable, scalable summarization using robust nlp and machine learning. pages 62–66.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Pengxiang Cheng and Katrin Erk. 2019. Attending to entities for better text understanding. *CoRR*, abs/1911.04361.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2018. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

David Mueller, Nicholas Andrews, and Mark Dredze. 2020. Sources of transfer in multilingual named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8093–8104, Online. Association for Computational Linguistics.

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

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217 – 250.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Desislava Petkova and W. Bruce Croft. 2007. Proximity-based document representation for named entity retrieval. In *Proc. of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 731–740. Association for Computing Machinery.

Jakub Piskorski, Lidia Pivovarova, Jan Šnajder, Josef Steinberger, and Roman Yangarber. 2017. The first

cross-lingual challenge on recognition, normalization, and matching of named entities in Slavic languages. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 76–85, Valencia, Spain. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Dipti Misra Sharma Rajeev Sangal and Anil Kumar Singh, editors. 2008. *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. Asian Federation of Natural Language Processing.

Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021. Named Entity Recognition for Entity Linking: What works and what's next. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, Punta Cana, Dominican Republic.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 219–228, Berlin, Germany. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Reproducibility Details

Hardware Infrastructure All model training was carried out on an NVIDIA GeForce RTX 3090 and 2080 Ti architecture. It required $\sim$45 s/epoch on the CoNLL and WikiANN datasets, whereas it required $\sim$6 min/epoch on the WikiNER and WikiNEuRal datasets.

Hyperparameter Tuning We performed hyperparameter selection on the English CoNLL dataset for the following hyperparameter values: lr = {0.0001, 0.001, 0.005}, average of the last k BERT layers with k = {1,4,6}, dropout = {0.2, 0.5, 0.7}, RNN hidden size = {128, 256, 512, 768} and 3 different random seeds. The combination of all the allowed values of the considered hyperparameters led to 324 independent configurations. The results of the grid-search applied on the above listed parameters are shown in Figure 2. Each curve corresponds to a model configuration: light curves correspond to high-performing models, whereas dark curves correspond to low-performing models. The best value for the learning rate is 0.001. Similarly, averaging the last k=4 layers of the BERT architecture is better than using the last k=6, and much better than using only the last layer. Regarding dropout, we found no evidence to make us prefer one value over another, so we decided to set it to the most commonly-used value, i.e., 0.5. Finally, the best values for the RNN hidden size are 768 and 512. On average, they reached similar scores but the 512 alternative was more stable (lower standard deviation) and faster. Hence, since the aim of this model is simply to allow comparisons between different datasets, we decided to use 512. Other hyperparameters were set to standard values used in the literature.

Tools/Technologies To ensure reproducibility of our work we relied on different libraries:

- *Transformers*[12] to seamlessly switch between different transformer architectures;
- *PyTorch Lightning* [13] as framework to ensure reusability of our code.
- *Hydra*[14] to obtain dynamic run configurations and sweeps.

[12]https://huggingface.co/transformers/
[13]https://www.pytorchlightning.ai
[14]https://hydra.cc

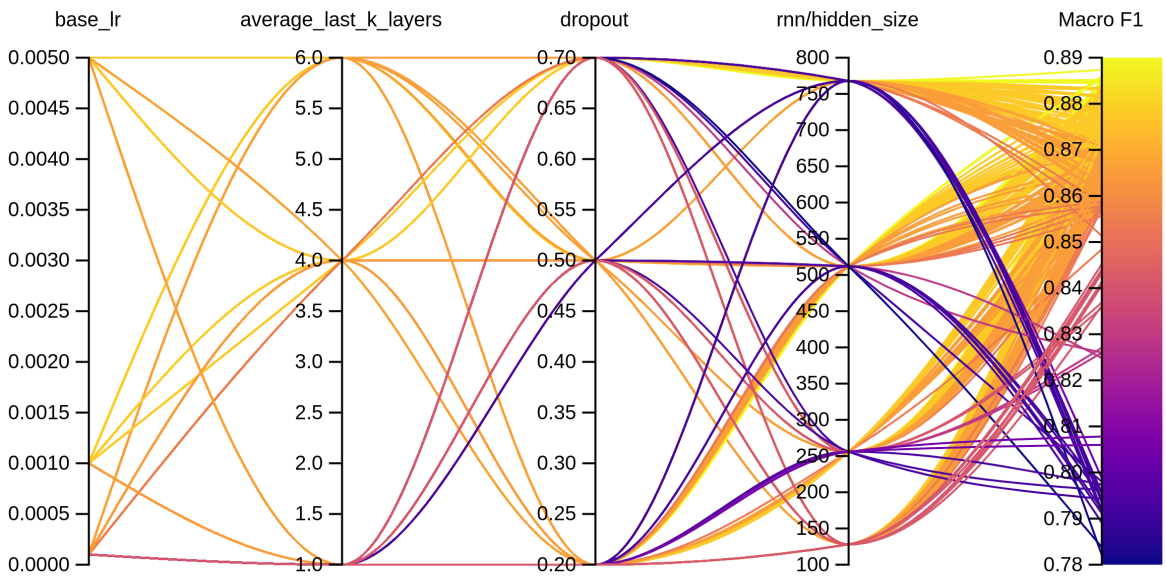

Figure 2: Results of the grid-search on the following 4 parameters: learning rate, number of BERT layers to average, dropout and RNN hidden size. Each curve corresponds to a different model configuration.

- *Weights and Biases*[15] as experiment logger, to gain useful insights and comparisons between different model runs (Figure 2 is an example);

B Additional Results

In Table 5 we show additional results about the comparison between WikiNEuRal, WikiNER and WikiANN, using the token-level macro F_1 score.

Multilingual Evalutation Also with this metric we notice consistent improvements of WikiNEuRal over the WikiNER and WikiANN alternatives on all tested languages and datasets. In particular, on the CoNLL test sets, we observe an average improvement, computed over the scores on the four languages, of 13.7 and 4.0 F_1 points over WikiANN and WikiNER, respectively. All the four results are extremely statistically significant. Morever, in the other test sets (i.e., WikiGold, OntoNotes and BSNLP), our approach again achieved better results in comparison to those obtained by the two competitors. Finally, we also show how WikiNEuRal-based models perform 6.8 F_1 points better than CoNLL-trained models on the WikiGold test set, and almost 1 point better when tested on neutral test sets, namely corpora from sources different from both WikiNEuRal and CoNLL training sets (i.e., OntoNotes). Similarly, WikiNEuRal-based models perform 7.2 F_1 points better than Ontonotes-trained models on the WikiGold test set, and almost 1.6 points better when tested on neutral test sets, i.e., CoNLL.

Silver- and Gold-Standard Data Aggregation Again, we aggregate WikiNEuRal with manually-created datasets in the corresponding languages, showing how the combination of gold-standard and our silver-standard training data can achieve results that are even higher than the ones achieved with gold-standard training data alone. In particular, on average[16], the WikiNEuRal and CoNLL datasets alone achieve a span F_1 score of 76.9 and 82.4, respectively, while their concatenation attains 83.8 F_1. In a similar way, on average, WikiNEuRal and OntoNotes datasets alone achieve a span F_1 score of 79.0 and 80.9, respectively, while their concatenation attains 83.9 F_1. Hence, the concatenated models show a stronger consistency across genres, as demonstrated by the better results on all tested datasets.

Domain Adaptation Our Domain Adaptation (DA) strategy consistently improves performances over already state-of-the-art results, while requiring much less training data. On average, domain-adapted datasets attain 2.0, 1.3 and 5.4 macro F_1 improvements over WikiNEuRal, CoNLL+WikiNEuRal and OntoNotes + WikiNEuRal, respectively.

[15] https://wandb.ai

[16] Computed on all datasets for which results are available for the three alternatives.

Training set \ Test set	*CoNLL*				*WikiGold*	*OntoNotes*	*BSNLP*	
	English	Spanish	Dutch	German	English	English	Russian	Polish
WikiANN	63.90 ± 1.86	63.18 ± 2.80	64.32 ± 1.38	55.49 ± 0.97	68.31 ± 2.14	53.58 ± 3.24	54.00 ± 1.57	60.55 ± 1.45
WikiNER	71.15 ± 0.76	72.90 ± 0.97	75.90 ± 0.63	65.72 ± 1.03	84.30 ± 0.29	74.83 ± 0.76	59.71 ± 1.20	64.87 ± 2.00
WikiNEuRal	77.02 ± 0.66	**77.98 ± 0.49**	79.20 ± 0.40	67.50 ± 0.31	**84.69 ± 0.25***	75.25 ± 0.44	**61.08 ± 0.74***	**66.00 ± 1.33**
WikiNEuRal DA	**78.22 ± 0.57****	-	**81.27 ± 0.94**	**69.69 ± 0.57**	-	**77.58 ± 0.36**	-	-
CoNLL	88.77 ± 0.41	**88.66 ± 0.37***	88.23 ± 0.78	76.77 ± 0.44	77.90 ± 0.73	74.38 ± 0.43	-	-
+ WikiANN	86.46 ± 0.52	87.69 ± 0.37	84.07 ± 0.97	71.41 ± 0.81	67.55 ± 0.52	69.18 ± 0.42	-	-
+ WikiNER	87.56 ± 0.48	87.19 ± 0.50	84.37 ± 0.87	72.14 ± 0.37	84.15 ± 0.44	75.30 ± 0.38	-	-
+ WikiNEuRal	88.38 ± 0.40	88.02 ± 0.44	87.97 ± 0.38	76.98 ± 0.68	**84.35 ± 0.44**	77.08 ± 0.17	-	-
+ WikiNEuRal DA	**88.91 ± 0.40**	-	**88.51 ± 0.54**	**77.40 ± 0.52**	-	**78.56 ± 0.29**	-	-
OntoNotes	75.45 ± 0.55	-	-	-	77.47 ± 0.22	**89.80 ± 0.39**	-	-
+ WikiANN	70.82 ± 1.17	-	-	-	71.03 ± 1.50	87.84 ± 1.38	-	-
+ WikiNER	76.13 ± 0.87	-	-	-	**84.81 ± 0.29**	87.48 ± 0.23	-	-
+ WikiNEuRal	78.46 ± 0.88	-	-	-	84.63 ± 0.22	88.47 ± 0.25	-	-
+ WikiNEuRal DA	**88.44 ± 0.41**	-	-	-	-	89.22 ± 0.18	-	-

Table 5: Token-level macro F_1 scores on common NER benchmarks. DA stands for Domain Adaptation. Statistical significance was computed using Student's t-test: * stands for $p < 0.05$, ** stands for $p < 0.01$, underline stands for $p < 0.001$.

C OntoNotes-to-CoNLL Class Mapping

To better explain the mapping, we first report the 18 OntoNotes classes with their meanings: PERSON (people, including fictional characters), ORG (companies, agencies, institutions, etc.), GPE (countries, cities, states), LOC (non-GPE locations, mountain ranges, bodies of water), FAC (buildings, airports, highways, bridges, etc.), PRODUCT (objects, vehicles, foods, etc., but not services), EVENT (named hurricanes, battles, wars, sports events, etc.), WORK_OF_ART (titles of books, songs, etc.), LAW (named documents made into laws), LANGUAGE (any named language), NORP (nationalities or religious or political groups), DATE (absolute or relative dates or periods), TIME (times smaller than a day), PERCENT (percentages), MONEY (monetary values, including the unit), QUANTITY (measurements, as of weight or distance), ORDINAL ("first", "second", etc.), CARDINAL (numerals that do not fall under another type).

The above classes were converted to the five standard CoNLL-03 NER classes by analyzing how elements belonging to these classes were annotated in the CoNLL dataset. Specifically, we followed the mapping reported below: PERSON → PER, ORG → ORG, GPE → LOC, LOC → LOC, FAC → LOC, PRODUCT → MISC, EVENT → MISC, WORK_OF_ART → MISC, LAW → O, LANGUAGE → MISC, NORP → MISC, DATA → O, TIME → O, PERCENT → O, MONEY → O, QUANTITY → O, ORDINAL → O, CARDINAL → O.