

Where Does It End? Long Named Entity Recognition for Propaganda Detection and Beyond

Piotr Przybyła^{1,2,*}, Konrad Kaczyński^{2,3}

¹LaSTUS Lab, TALN Group, Universitat Pompeu Fabra, Barcelona, Spain

²Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

³Faculty of Economic Sciences, University of Warsaw, Poland

Abstract

Propaganda detection is usually defined and solved as a Named Entity Recognition (NER) task. However, the instances of propaganda techniques (text spans) are usually much longer than typical NER entities (e.g. person or location names) and can include dozens of words. In this work, we investigate how the extensive span lengths affect the recognition of propaganda, showing that the task difficulty indeed increases with the span length. We systematically evaluate several common approaches to the task, measuring how well they recover the length distribution of true spans. We also propose a new solution, including an adaptive convolution layer that facilitates sharing information between distant words. Our approach allows to improve length preservation without sacrificing overall performance.

Keywords

propaganda detection, named entity recognition, long named entities

1. Introduction

The rise of the many challenges collectively known as *misinformation* has prompted researchers working on Natural Language Processing (NLP) to propose several tasks aimed at assessing the reliability of online text. This includes detecting social media bots [1], non-credible news articles [2] or other hyper-partisan content [3]. Propaganda detection is based on similar inspiration, but differs from the other tasks, since it involves pinpointing specific usages of manipulative techniques on the word level. In practice, it means that instead of a general *fake news* label, an end user might see certain text passages highlighted and categorised with respect to the type of manipulation they involve. This helps the user to understand why certain text should be considered unreliable. Interpretability matters in misinformation context, where it has been shown to impact the users' trust in credibility labels [4].

For example, in the following header, the underlined text was annotated as *Flag-waving* by the human annotators:


“Hungary PM Viktor Orban Vows to ‘Fight Those Who Want to Change the Christian Identity of Europe’”.

NLP-MisInfo 2023: SEPLN 2023 Workshop on NLP applied to Misinformation, held as part of SEPLN 2023: 39th International Conference of the Spanish Society for Natural Language Processing, September 26th, 2023, Jaen, Spain

✉ piotr.przybyla@upf.edu (P. Przybyła); konrad.kaczynski@ipipan.waw.pl (K. Kaczyński)

🆔 0000-0001-9043-6817 (P. Przybyła); 0000-0002-1819-5649 (K. Kaczyński)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

To approach this problem, the proposed solutions build on the previous work on Named Entity Recognition (NER), which involves finding spans in text that belong to certain categories. However, the most popular NER solutions were developed for recognising relatively short names of entities, such as persons, geographical entities or biomedical concepts. The instances of propaganda techniques are typically much longer and may cover many words, as in the example above, or even multiple sentences. As we show later, this makes these very long spans rarely correctly recognised by generic NER approaches. In this study, we make the following contributions:

- We systematically evaluate the current NER approaches in the propaganda detection, both in terms of recognition accuracy and how well the span length distribution of the predicted entities matches the gold standard.
- We propose a new model architecture, called *LoNER* (Long Named Entity Recognition), including a neural layer based on context-sensitive convolution operations, which improves length preservation while maintaining overall accuracy.

The code of our solution is openly available¹.

2. Related work

Work on defining propaganda. Propaganda can be defined as a planned and systematically applied suggestion, expressed largely in symbols and linguistic stimuli for the purpose of controlling attitudes and behaviour of individuals towards a predetermined mode of conduct [5, 6]. Propaganda can be a tool of *disinformation*, which intends to deceive the public to believe in false claims, but it can be used sway opinions and behaviours towards generally beneficial goals [7]. The word *propaganda* originated between the sixteenth and nineteenth century and referred to the spreading of religious doctrine of the catholic church [8], but modern propaganda gained momentum during World War I [9].

One of the first works defining propaganda as usage of specific persuasive techniques was that of Lee and Lee [10], listing seven distinct propaganda techniques: *Name calling*, *Glittering Generalities*, *Transfer*, *Testimonial*, *Plain Folks*, *Card stacking* and *Band wagon*. Lazarsfeld and Merton [11] added two more general categories: one-dimensional classification of symbols and personification stereotype. Later taxonomies brought further propaganda techniques. Brown [12] put forward eight broad categories, including *Use of Stereotypes*, *Substitution of Names*, *Selection*, *Repetition*, *Assertion* or *Appeal to Authority*. Smith [13] mentioned *Symbolic Fiction*, *Multiple Standards*, *Historical Reconstruction* and *Asymmetrical Definition*. The most extensive list of propaganda techniques is produced by Weston [14] who proposes 24 rules of making a successful argument.

Work on detecting propaganda. When propaganda detection was first defined as an NLP task, it was tackled mostly on the article level [15, 16]. More recently, research on fine-grained (i.e. sentence- or word-level) propaganda detection has been gaining traction. It was presented as a shared task at NLP4IF workshop [17] with two subtasks: fragment-level classification of propaganda techniques used in a given span and a sentence-level binary classification task

¹<https://github.com/piotrmp/loner>

(propaganda present or not). Another shared task (Detection of Propaganda Techniques in News Articles) was organised at Semeval 2020 workshop [18], where it was formulated as a pipeline task consisting of span identification subtask (i.e. spotting propaganda fragments in plain-text documents) and a technique classification subtask (i.e. determining the propaganda technique in a given span). Finally, in the shared-task on Detection of Persuasion Techniques in Texts and Image at Semeval 2021 [19], the participants could choose from three subtasks: multi-label classification task (given textual content of a meme, identify which techniques are used in it), multilabel sequence tagging task (identify pairs of techniques and spans they cover), and a multimodal, multi-label classification task (using both textual and pictorial content of a meme, identify which techniques are used in it). Transformer-based [20] architectures dominated all three shared-tasks. Best results on multilabel sequence tagging tasks were obtained with designs consisting of pre-trained models such as BERT [21], RoBERTA [22], or ELMo [23] combined with additional classification layers and fine-tuned within various scenarios: NER, multi-task learning, question-answering, etc. Task participants often experimented with different loss function designs and data augmentation methods in order to alleviate the problem of imbalance or sparseness of the data.

Note that even though our work uses data from one of the shared tasks (see section 3.1), the results are not directly comparable with those of the tasks’s participants. This is because we have no access to the test set and use a different success measure, focused on entity length preservation (see section 7.2). However, our goal is not to establish the new state of the art in propaganda recognition, but rather use the problem as an opportunity to explore the issue of long entities in NER.

NER with long entities. Despite the abundance of previous work on NER, it appears that the vast majority of research is still focused on very short entities. This study in general and the LoNER method in particular are focused on the problem of NER for very long entities, for which, as shown in section 4.1, propaganda detection is a fitting example. While the relevant shared tasks have attracted numerous interesting solutions, none of them have explicitly focused on the problem of entity length [17, 18, 19]. Similarly, the very long entities have been present, but not taken into account, in some other NER studies involving medical entities [24] and job requirements [25]. The only one study that evaluated NER performance for longer entities we are aware of is that of Li et al. [26], but they analyse entities of up to 6 words. The propaganda instances are even longer as they can span multiple sentences, justifying a special approach. To the best of our knowledge, our work is the first to investigate the impact of such extensive span length on NER performance.

3. Propaganda detection task

The input of the propaganda detection task is a short text (sentence, paragraph or document) in natural language (English in our case). Since we are expressing the problem through the NER framework, the output is a list of entities. Each entity corresponds to the usage of a certain propaganda technique and is described by a *span*, i.e. a continuous section of text defined through character offsets, and a *category*, i.e. one of the pre-defined propaganda techniques.

Category	Count			Length			Coverage		
	Train.	Dev.	Test	Train.	Dev.	Test	Train.	Dev.	Test
AtA	89	30	25	120.56	106.27	123.96	0.70%	0.66%	0.74%
AtFP	197	61	41	94.79	76.80	61.22	1.22%	0.97%	0.60%
BRaH	50	12	13	92.80	101.42	87.85	0.30%	0.25%	0.27%
BaWF	85	29	11	102.44	71.90	128.00	0.57%	0.43%	0.33%
CO	135	44	35	121.10	117.27	125.23	1.07%	1.07%	1.04%
D	279	119	84	108.95	118.66	83.39	1.99%	2.92%	1.66%
EM	312	104	74	41.71	46.18	43.57	0.85%	0.99%	0.77%
FW	170	59	48	53.86	68.17	60.35	0.60%	0.83%	0.69%
LL	1369	398	379	23.08	23.73	20.20	2.07%	1.95%	1.82%
NCL	668	245	198	26.00	27.87	21.13	1.14%	1.41%	0.99%
R	466	103	145	16.70	16.78	12.22	0.51%	0.36%	0.42%
S	121	20	28	25.64	24.30	22.89	0.20%	0.10%	0.15%
TTC	51	23	18	30.10	33.48	42.39	0.10%	0.16%	0.18%
WSMRH	74	33	24	96.88	98.48	88.46	0.47%	0.67%	0.50%
All	4066	1280	1123	44.32	48.27	38.12	11.82%	12.67%	10.16%

Table 1

The number of entities, average length (in characters) and total coverage of text for each of the propaganda category. See the main text for an explanation of the technique abbreviations.

3.1. Annotated data

We base our experiments on the PTC Corpus from Semeval 2020 Task 11 [18], which is the largest publicly available dataset annotated with propaganda categories on the token-level. Because sequence tagging-based approaches do not allow overlapping spans, we need to disregard some of the entities. This process is implemented in a way that prioritises preservation of entities (1) belonging to rarer categories and (2) having less overlap. In the end, around 10% of entities are removed in the process.

From the original corpus, only the training and development subsets are publicly available, since the test set was used to perform evaluation within the shared task. For the purpose of our experiments, we aggregate the available subsets and randomly re-split them, assigning documents into training (60%), development (20%) and test (20%) subsets.

The corpus contains annotations of the the following 14 propaganda techniques:

- Appeal to Authority (AtA),
- Appeal to Fear, Prejudice (AtFP),
- Bandwagon, Reductio ad Hitlerum (BRaH),
- Black and White Fallacy (BaWF),
- Causal Oversimplification (CO),
- Doubt (D),
- Exaggeration, Minimisation (EM),
- Flag-Waving (FW),
- Loaded Language (LL)
- Name-Calling, Labelling (NCL),

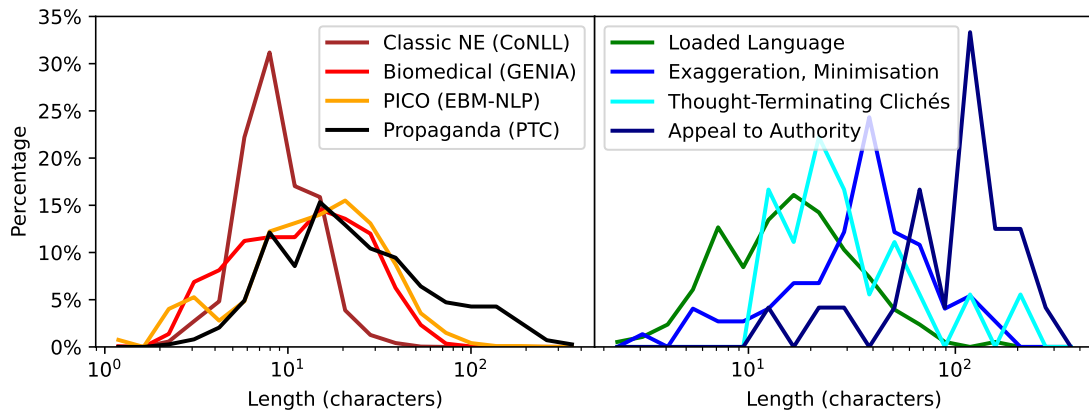


Figure 1: The distribution of span lengths in four NER tasks (left) and four propaganda techniques (right), computed as percentage of entities in 20 equal-width bins (in logarithmic space).

- Repetition (R),
- Slogans (S),
- Thought-Terminating Clichés (TTC),
- Whataboutism, Straw Men, Red Herring (WSMRH).

For detailed statistics of the number of entities, average length and coverage of each technique in the corpus, see table 1.

The categories differ significantly in terms of number of instances: the most common one (*Loaded Language*) has 1369 instances in the training subset, while the rarest one (*Bandwagon*, *Reductio ad Hitlerum*) occurs just 50 times. The differences in span length are less stark, though still significant: *Appeal to Authority* has on average over 120 characters, while *Repetition* less than 17. The former may include a subordinate clause (*Leading experts in the field agree that ...*), while the latter can strengthen impact by repeating a single word.

Note that the combined effect of number and length of entities means that a category can achieve high coverage (and thus impact on evaluation) through a large number of short entities (e.g. *Loaded Language* or *Name-Calling, Labelling*) or through fewer long entities (e.g. *Doubt* or *Appeal to Fear, Prejudice*). Recognising the latter type is definitely more challenging, as it means learning complex structures (multi-word phrases) from few training example. This is also reflected by the mismatch of coverage in the different subsets; e.g. *Doubt* covers 2.92% of development text, but only 1.66% of the test text.

4. Propaganda techniques in the corpus

4.1. Span length

The extensive length of propaganda techniques makes the problem stand out when compared to other NER tasks. The average entity length in the test set is 38.12 characters (6.24 words).

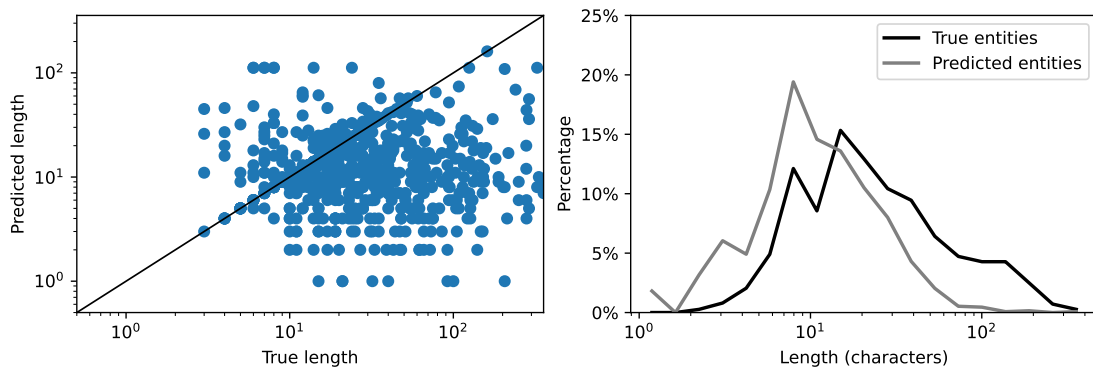


Figure 2: Span lengths (in characters) of the predicted and gold-standard (true) entities, shown as a scatterplot for the matching entities (left) and overall distribution (right).

While there are some one-word instances (e.g. in *Slogans* or *Repetition*), longer ones dominate, with the maximum of 69 words (412 characters).

In figure 1 (left), we compare the length distribution of propaganda entities to previously published NER corpora: ConLL-2003, including general-domain names of persons, locations etc. [27]; GENIA/BioNLP, including biomedical concepts, such as genes, proteins, etc. [28] and EMB-NLP [24], including PICO elements (further description in section 7.4). The average span length for these are, respectively, 9.45, 15.35 and 18.79 characters. As mentioned previously, EMB-NLP has the longest entities, but they are still less than half of the propaganda techniques, which average at 38.12 characters. Note the significant portion of very long entities (over 100 characters), absent from the other datasets. Figure 1 (right) demonstrates the internal diversity of propaganda techniques by showing an analogous comparison between four of the techniques.

It is expected that extensive span length affects the NER performance. While the full evaluation results are presented in section 8, here we broadly demonstrate the problem by showing how the baseline model (BERT raw, see more in section 5) performs on our data. In figure 2 (left), each point corresponds to a matching between a true (gold standard) and a predicted entity. For the purpose of this visualisations, two entities are considered to match if their spans overlap, irrespective of the categories. The X and Y coordinates correspond to the character length of the true and predicted entity, respectively. We can clearly see that the number of entities shorter than expected (below the $Y=X$ line) is much larger than of those longer than expected (above the line). This is especially noticeable for very long entities (true length over 50 characters). Moreover, the set of predictions includes numerous entities including 1 or 2 characters, which obviously have no counterparts in the gold standard.

Figure 2 (right) shows the distribution of lengths of the predicted and gold-standard entities, regardless of their matching status. As expected, the entities predicted by the model are noticeably shorter. In 7 we show how to quantify the distribution mismatch visible in this plot and use it as an evaluation measure called *entity length discrepancy*.

5. NER approaches evaluated

In this section we briefly describe the existing solutions to the NER task, which are evaluated in our experiments, focusing on the features that may affect their performance in case of long entities. We start with the most frequently used approach (sequence tagging using language models) and discuss its variants in section 5.1 before showing the less popular models (BiLSTM-CRF and span prediction) in section 5.2.

5.1. Sequence tagging using language models

The most popular framework for approaching the NER tasks is through *sequence tagging*. It involves creating a *label* for each token of the text, determining whether the token is included in a span of an entity, and if so, what category does this entity belong to (section 5.1.1). These token labels can be generated by a two-step model: (1) representing each token in a multi-dimensional space using a pretrained language model, such as BERT, and (2) predicting the token label based on this representation (section 5.1.2).

5.1.1. Span encoding

Span encoding determines how information about entities present in a given sequence (e.g. a sentence) is translated to token-level labels. In the most straightforward approach (*raw category labels*), each token included in the span of an entity is assigned a label equal to the category of this entity (e.g. *AtA*), while tokens not covered by entities are assigned a special *out* label (*O*). The extension of this scheme, known as BIO (begin-inside-out), was first adopted by Ramshaw and Marcus [29] and has since become a standard technique. It differentiates between tokens that are the first in a given entity (e.g. *B-AtA*) from the subsequent ones (e.g. *I-AtA*). Several more elaborate schemes exist, e.g. IOBES with label types for the last token in an entity (*E*) and single-token entities (*S*), used in biomedical NER [30]. There is no definite answer as for which of these variants is most effective. Comparative studies on CoNLL-03 and MUC7 datasets produced mixed results, with Ratinov and Roth [31] and Cho et al. [32] proving more complex schemes to score higher, and Konkol and Konopík [33] showing the opposite.

Given that the more elaborate span encoding schemes allow for richer representation of multi-token entities, we expect that this choice may influence the recognition performance for our long-entity task.

5.1.2. Label prediction

Sequence labelling can be seen as a classification problem, where the correct label is predicted for each token in the sentence. This is however a *contextual* classification, since the label at position i depends not only on the token at position i , but also on the neighbouring ones. The label prediction using neural networks is performed in two steps.

Firstly, each i -th token is represented as an embedding vector h_i of length e . While the embeddings can be static, from a method such as *word2vec*, the contextual classification goal is better served through contextual (and trainable) embeddings generated by a pretrained language

model. In our experiments we use BERT [21] in the Base variant, which means each *wordpiece* token is represented through a vector of length $e = 768$.

Secondly, each hidden representation vector h_i needs to be converted to a k -dimensional vector s_i , where the score $s_{i,j}$ reflects the likelihood that the i -th token should be assigned the j -th label (out of k). This prediction layer could be implemented as a dense layer with $e \times k$ coefficients, and followed by a softmax layer to create label probabilities $p_{i,j}$ such that $\sum_j p_{i,j} = 1$.

Often more sophisticated approaches to the label prediction are used to make this operation context-sensitive. They include CRF (analogous to BiLSTM-CRF, see section 5.2.1), convolutional layers or recurrent ones, such as LSTM.

5.2. Other approaches

5.2.1. BiLSTM-CRF

BiLSTM-CRF is a model proposed by Lample et al. [34], which is similar to the approach described above, but using a bi-directional LSTM layer [35] instead of the pretrained language model. Specifically, the hidden representation h_i of the i -th token is created by concatenating the output of one LSTM processing forward and another one operating in reverse. The input of these LSTM layers consists of GloVe embeddings [36] and character-level LSTM output.

The subsequent CRF layer computes the score of a given sequence of predictions $\mathbf{y} = y_1, \dots, y_n$ by taking into account the token-level scores $s_{i,j}$ (obtained from the preceding layer) and transition scores $t_{v,w}$ (trainable as weights). the CRF layer can encode the information on the entity length through the values of $t_{v,v}$, indicating the likelihood that the tokens with category v follow each other. For example, in our dataset we would expect $t_{AtA,AtA}$ to be higher than $t_{LL,LL}$.

5.2.2. Span prediction

Finally, we include one NER solution based on span prediction to check how this new approach performs in our task. We choose the recently-published *SpanNER* [37].

In span prediction each entity is represented individually, rather than as a sequence of token labels. This means that every possible span $r_{i,i+l} = [i, i + 1, \dots, i + l]$ in a sentence is assigned a label $y_{i,i+l}$. In SpanNER, the label is predicted based on the hidden representation (computed using BiLSTM) from the first and the last token in the span (h_i and h_{i+l}) and the learnable span length $(l + 1)$ embedding.

6. LoNER

Our approach, called *Long Named Entity Recognition* (LoNER), is designed to improve the NER process in such a way that promotes entities matching those seen in training in terms of span length. To understand the overall idea behind LoNER, consider a system output, where a single token x_i is recognised as belonging to a certain category C , while the rest of the tokens in the sentence $x_{i \neq i}$ is predicted to contain no entities (0). However, if C is a category characterised by long spans (e.g. *Causal Oversimplification*), such output is extremely unlikely to be correct. It would be better to either extend the entity span to neighbouring tokens or remove it altogether. How can we communicate this intention to the model?

6.1. Adaptive convolution

The most straightforward approach is the *convolution* operation. If we convolve the class probabilities vector with another vector (e.g. resembling the Gaussian distribution), the large probability of C in x_i will be shared at x_{i-1} , x_{i+1} , x_{i+2} etc. The challenge here is that a fixed convolution kernel will not fit every situation. For example, the convolution size should depend on whether a token at the centre has a chance of representing a short-span entity: some do (e.g. emotive words indicative of *Loaded language*), while others do not (e.g. function words taking meaning only within long phrases). Similarly, some tokens should be associated with convolution obtaining maximum *after* its own position (e.g. words typical for a beginning of a phrase), while maximum *before* would suit others (e.g. words used to end a phrase).

For the reasons outlined above, we propose an *adaptive convolution* layer that convolves the scores $s_{i,j}$ (indicating likelihood that token i belongs to category j) with a Gaussian kernel, whose mean μ_i and standard deviation σ_i depend on the hidden representation of the token h_i :

$$s_{i,j}^* = \sum_{\iota=1}^n s_{\iota,j} \times \exp\left(-\frac{1}{2} \left(\frac{(\iota-i) - \mu_i}{\sigma_i}\right)^2\right)$$

We can see that the score at position i can be affected by scores at another position ι , though this influence wanes as the distance between the positions $(\iota - i)$ grows. Note that the Gaussian is scaled to peak at 1.0, so that $s_{i,j}^* \approx s_{i,j}$ when μ_i is close to 0 and σ_i is high. The values of standard deviation and mean are computed from the hidden representation of the associated tokens through a dense linear layer:

$$\mu_i = h_i \cdot \beta_\mu + \alpha_\mu, \quad \sigma_i = h_i \cdot \beta_\sigma + \alpha_\sigma$$

6.2. LoNER architecture

Here we describe how the adaptive convolution is integrated in the general architecture of our solution; see figure 3 for a diagram. The BERT pretrained model [21] is used to generate a hidden representation of each token (h_i). We use the BERT Base (uncased) model, which means that these vectors have length of 768. The hidden representation is fed to a dense layer in order to generate 15 label scores $s_{i,j}$ (for 14 categories and 0). In parallel, the same representation is used to compute token-wise kernel parameters μ_i and σ_i , which are then used in an adaptive convolution as described above to produce convoluted scores $s_{i,j}^*$. Additionally, we employ a residual connection [38] to accelerate the learning, adding the original and convoluted score matrices. Finally, a softmax operation is performed to obtain label likelihoods $p_{i,j}$ such that $\sum_j p_{i,j} = 1$. For each token, the label with the highest value is selected and used to construct the continuous spans with the associated categories.

The weights of the dense layer producing the kernel parameters are initialised so that $\mu_i = 0$ and $\sigma_i = 1$ are output, corresponding to a very mild smoothing effect. The dense layer producing the scores is initialised randomly. All of the weights in the model (including BERT) are trainable.

To reduce overfitting, dropout [39] is applied to the hidden representations returned from BERT. The dropout layer is designed in such a way that each h_i vector can be hidden (replaced with zeros) with probability 0.2. This forces the model to make predictions when some of the words in a sentence are unknown, learning to draw clues from their context.

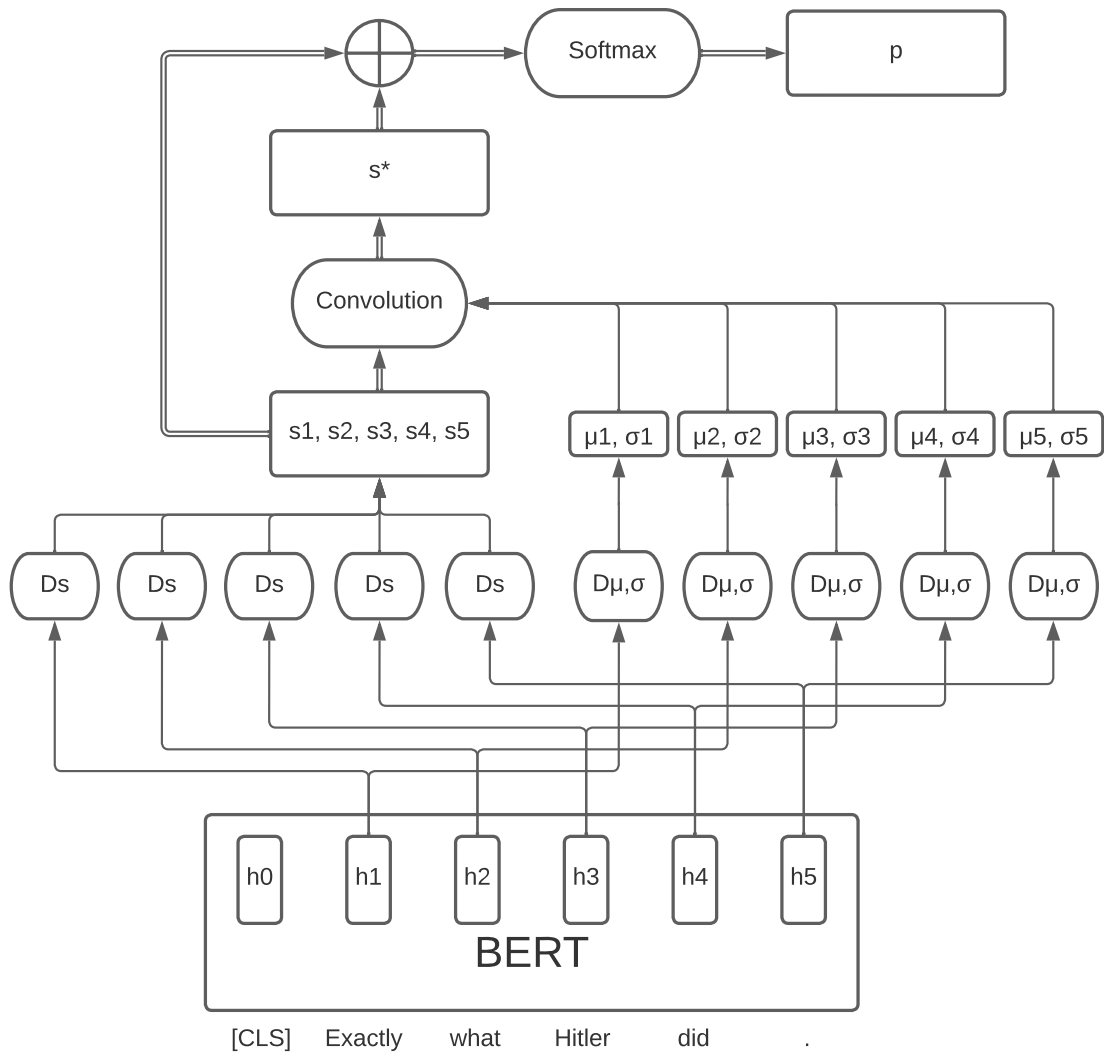


Figure 3: The LoNER architecture

7. Evaluation

We train a group of models selected to represent different paradigms (section 7.1) and check their performance, both using traditional measures and newly introduced *entity length discrepancy* (section 7.2). Section 7.3 contains details regarding the parameters of the trained models. We also include an additional evaluation on a dataset from a different domain (section 7.4).

7.1. Evaluated approaches

We choose to evaluate the following 11 variants of the available NER approaches (section 5):

- BiLSTM-CRF with raw, BIO and IOBES labelling,
- BERT sequence labelling using raw, BIO and IOBES labelling,
- BERT sequence labelling with CRF using raw, BIO and IOBES labelling,
- LoNER using raw labelling,
- SpanNER.

7.2. Measures

For basic evaluation, we adopt the scheme proposed for Semeval 2020 Task 11 [40]. It compares the gold-standard and predicted sets of entities through precision, recall and F-score, accepting partially overlapping spans. Results from entity categories are aggregated using both micro- and macro-averaging, which helps to get a broader picture in the case of highly unbalanced dataset.

The F-score defined in such a way rewards a model for predicting even one word from a long sequence of words (or sentences) comprising the original entity, which often happens (see figure 2). We argue that such 'matches' may not be helpful for an end user and the evaluation needs to be extended with a measure that specifically targets the mismatch between the predicted and gold-standard distributions of span lengths.

In order to achieve that, we introduce *entity length discrepancy*, which is computed as Kullback-Leibler (KL) divergence [41] between the distributions of span lengths in gold-standard and predicted entities. To represent the two compared sets of entities as length distributions, their combined range is divided into 5-character-long bins, each of which contains the number of entities with spans of that length. These are then scaled to sum up to 1, creating a discrete probability distribution, which can be used to compute KL divergence.

Entity length discrepancy is computed in two variants. In *global*, the length distributions for comparison are built based on all the entities in predicted and gold-standard sets, as shown in Figure 2. In *local*, the comparisons are made within each category, e.g. predicted LL entities compared to gold-standard LL entities, and the obtained KL values are averaged over all categories.

7.3. Model training setup

All of the evaluated approaches are trained on the training set for 50 epochs. The micro-averaged F-score on the development set is used to choose the best epoch, for which the test set results are reported. Because of the modest size of the dataset, the results may vary between runs. In order to minimise the impact of this on the conclusions, we run each experiment ten times with different random seeds and provide the averaged results.

For BiLSTM-CRF, we use the implementation published by Genthial [42]². We keep the original parameters of the process, except for the dropout rate, which was set to 0.1. SpanNER is evaluated using the code published alongside the article³, using the sp2 variant, which performed best in the original evaluation [37].

²https://github.com/guillaumegenthial/tf_ner

³<https://github.com/neulab/SpanNER>

Method	Enc.	Micro-averaged			Macro-averaged			L. discrepancy	
		P	R	F1	P	R	F1	local	global
BiLSTM-CRF	raw	0.2719	0.1345	0.1798	0.2220	0.0972	0.1278	1.0226	0.1541
	BIO	0.2751	0.1306	0.1767	0.2215	0.0945	0.1231	1.0336	0.1628
	IOBES	0.2630	0.1242	0.1681	0.2317	0.0940	0.1215	1.0199	0.2032
SpanNER	–	0.2690	0.1336	0.1779	0.1317	0.0685	0.0824	1.0488	0.4289
BERT	raw	0.2666	0.2800	0.2724	0.2078	0.1507	0.1554	1.6296	0.4686
	BIO	0.2549	0.2777	0.2645	0.2015	0.1481	0.1501	1.5841	0.5014
	IOBES	0.2592	0.2772	0.2672	0.2054	0.1458	0.1506	1.6661	0.5287
BERT-CRF	raw	0.2530	0.2925	0.2702	0.2111	0.1564	0.1565	1.4791	0.4016
	BIO	0.2522	0.2841	0.2668	0.1983	0.1458	0.1478	1.5116	0.4269
	IOBES	0.2525	0.2731	0.2619	0.1995	0.1496	0.1541	1.5870	0.4985
LoNER	–	0.2815	0.2684	0.2731	0.2338	0.1488	0.1573	1.3216	0.3150

Table 2

The results of the evaluation of propaganda detection. Each method-encoding combination is measured with respect to precision, recall and F-measure (micro- or macro-averaged) and length discrepancy (local and global). The best results for each measure (highest P, R or F1; lowest discrepancy) is in boldface.

All of the BERT-based model are implemented using our own code in *TensorFlow*. The optimisation is performed using Adam [43] with learning rate equal 3×10^{-5} . The categorical cross-entropy loss is applied to the per-token predictions.

7.4. Additional evaluation

In order to broaden our evaluation, we include an additional experiment involving the EMB-NLP dataset [24]. It consists of abstracts describing clinical trial with NER categories covering elements that are of interest from the point of view of systematic reviews, including a description of participants, a tested intervention (e.g. a drug) and the observed results. In total, entities of 18 categories are included, covering 18.79 characters on average – not as long as propaganda techniques, but noticeably more than classic NER.

In order for the EMB-NLP dataset to match the proportions of the PTC Corpus (60/20/20), we keep the original test portion of the EMB-NLP dataset (301 abstracts) and use the remaining data to randomly select subsets of 301 abstracts for development and 903 for training. To guarantee that each token is associated with only one label, we remove the overlapping spans (3% of the original data), prioritising the preservation of the longer span. We obtain 12007, 3853 and 4425 entities in the training, development and test portions, respectively.

8. Results

Table 2 shows the results of the main experiment. The values achieved by different methods vary greatly. Judging by the F1 measure, the superiority of the BERT-based approaches is quite clear. Among these, LoNER achieves the best micro-averaged F1 of 0.2731, but all BERT and BERT-CRF variants have a precision of around 0.26-0.27. BiLSTM-CRF and SpanNER clearly lag behind in this scenario, with F1 around 0.17-0.18. Interestingly, these methods deliver a solid

Method	Enc.	Micro-averaged			Macro-averaged			L. discrepancy	
		P	R	F1	P	R	F1	local	global
BERT	raw	0.3514	0.4118	0.3774	0.1686	0.1900	0.1695	0.4140	0.0663
LoNER	-	0.3569	0.3475	0.3519	0.1739	0.1611	0.1577	0.3321	0.0254

Table 3

The results of the additional evaluation using PICO entities detection.

precision, but less than half of the recall of BERT variants. The encoding schemes do not appear to improve the situation in this long-span task, with the simplest variant (raw) performing the best. It is also worth noting that despite the strong imbalance of the categories, all of the above observations hold both for micro- and macro-averaging.

The results of span length discrepancy validate the introduction of this measure. It turns out that the classic BERT-based methods, best-performing according to F1, are the worst in terms of preserving the length distribution. However, LoNER allows to mitigate this weakness, with global discrepancy of 0.3150 compared to 0.4686 of BERT.

Table 3 shows the results for the PICO entities detection. We can see that the situation resembles the case of propaganda detection. LoNER achieves similar values of F-score (higher precision, but lower recall) to BERT, but with a substantially better preservation of span length distribution. The difference is especially large for global length discrepancy, which is almost three times lower than for classic BERT.

9. Limitations

The overall goal of this work is to improve propaganda detection in online text. While we focused on proposing a new NER model, we need to emphasise that training data quality is just as crucial to achieving this aim. Persuasion techniques are one of the most subtle phenomena in language, which necessitates gathering many examples for training effective recognition. Unfortunately, as shown in table 1, several categories are represented by just a few dozen instances. This is the main factor limiting the performance of our solution. However, the consistency of outcomes between micro- and macro-averaged measures indicates that the benefits of LoNER are not affected by these small categories.

The main motivation behind LoNER was poor span length preservation by the mainstream BERT-based models. However, instead of designing an additional layer, one could investigate what this unsatisfactory performance stems from. In principle, Transformer-based language models, using attention to share information between faraway tokens, should be able to capture long-range contexts. The need to understand the workings of large language model has motivated a lot of research recently and we expect long entity recognition could be another use case for future work.

Another limitation of our work is not exploring the precision / recall tradeoff. As we can see in table 2, the models that are best in terms of length preservation (LSTM-based) have also relatively low recall. It might be beneficial to encourage other models to behave in a similar way, i.e. return less mentions, but of higher quality (in terms of length). It is however not clear how to compute model’s ‘confidence’ in a span in sequence labelling framework.

LoNER was designed specifically for propaganda detection, but could provide benefit in any NER task, especially when long entities are involved. Our additional evaluation is limited to PICO data, but it would be valuable to check other tasks, too. We hope some benefits of adaptive convolution would be noticeable even for classic entity types, but it is left for future work.

Finally, we emphasise the need to put the user in the loop when evaluating misinformation solutions. While data-based studies like this one can be valuable, they are limited to automatic performance measures and need to be followed by user studies in the expected application scenario. In case of propaganda detection, we are still a long way from understanding how to deploy such solutions to deliver real-world impact.

10. Conclusions

Representation using BERT vastly outperforms LSTMs. The advantages of pretrained language models for word representation are obviously well known across NLP. Here it is demonstrated through the significant gap between BiLSTM-CRM and BERT-based approaches.

CRF layers or labelling schemas provide no benefit. Both types of techniques are designed to manage spans through detecting tokens that occur in significant positions. But they do not seem to work for propaganda-length entities.

Models with high F-score have poor length preservation. Interestingly, the entity length is preserved the best by the models with comparatively low F-score, i.e. based on BiLSTM. This is achieved by not returning less certain entities, as demonstrated by precision-recall imbalance of LSTM-based solutions.

LoNER improves length preservation while maintaining F-score. Adaptive convolution works as expected, extending label information over longer spans. This allows us to keep the high F-score of BERT-based solutions (or even increase it) while improving the length preservation.

This result is not limited to propaganda detection. We can see the reduced length discrepancy also in case of PICO entities. We hope that ideas behind LoNER will be used for challenging NER in more domains.

Acknowledgments

The work was supported by the *Polish National Agency for Academic Exchange* through a *Polish Returns* grant number PPN/PPO/2018/1/00006 and as part of ERINIA project that has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement No 101060930. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

- [1] F. Rangel, P. Rosso, Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling, in: L. Cappellato, N. Ferro, D. E. Losada, H. Müller (Eds.), CLEF 2019 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings, CEUR-WS.org, 2019.

- [2] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, L. Wang, Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), 2020, pp. 334–339. URL: <https://aclanthology.org/2020.acl-main.31>. doi:10.18653/V1/2020.ACL-MAIN.31. arXiv:2004.13826.
- [3] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, B. Stein, A Stylometric Inquiry into Hyperpartisan and Fake News, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2018, pp. 231–240. URL: <https://www.aclweb.org/anthology/P18-1022>.
- [4] P. Przybyła, A. J. Soto, When classification accuracy is not enough: Explaining news credibility assessment, *Information Processing & Management* 58 (2021). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0306457321001412>. doi:10.1016/j.ipm.2021.102653.
- [5] C. Bird, *Social Psychology*, D. Appleton and Company, New York, NY, USA, 1940.
- [6] T. Parsons, Propaganda and Social Control, *Psychiatry* 5 (1942) 551–572. URL: <https://doi.org/10.1080/00332747.1942.11022421>. doi:10.1080/00332747.1942.11022421.
- [7] G. Bennet, Propaganda and disinformation: how a historical perspective aids critical response development, in: *The SAGE Handbook of propaganda*, AGE Publications Ltd, 2020, pp. 244–260. doi:10.4135/9781526477170.n16.
- [8] G. S. Jowett, Propaganda and Communication: The Re-emergence of a Research Tradition, *Journal of Communication* 37 (1987) 97–114. doi:10.1111/j.1460-2466.1987.tb00971.x.
- [9] J. Wilke, Propaganda, in: W. Donsbach (Ed.), *The International Encyclopedia of Communication*, John Wiley & Sons, Ltd, 2008. doi:10.1002/9781405186407.wbiecp109.
- [10] E. B. Lee, A. M. Lee, *The Fine Art of Propaganda: A Study of Father Coughlin’s Speeches*, Harcourt Brace and Co, 1939. URL: <https://archive.org/details/LeeFineArt>.
- [11] P. F. Lazarsfeld, R. K. Merton, SECTION OF ANTHROPOLOGY: Studies in Radio and Film Propaganda*, *Transactions of the New York Academy of Sciences* 6 (1943) 58–74. doi:10.1111/j.2164-0947.1943.tb00897.x.
- [12] J. A. C. Brown, *Techniques of persuasion: from propaganda to brainwashing*, 6th ed., Penguin Books, Baltimore, USA, 1963.
- [13] T. J. Smith, *Propaganda: A Pluralistic Perspective*, Praeger, 1989.
- [14] A. Weston, *A Rule Book for Arguments*, Hackett Publishing Company, Inc, 2017.
- [15] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, Y. Choi, Truth of varying shades: Analyzing language in fake news and political fact-checking, *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings (2017)* 2931–2937. doi:10.18653/v1/d17-1317.
- [16] A. Barrón-Cedeño, G. da San Martino, I. Jaradat, P. Nakov, Proppy: A system to unmask propaganda in online news, *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019) (2019)* 9847–9848. doi:10.1609/aaai.v33i01.33019847. arXiv:1912.06810.
- [17] G. da San Martino, A. Barrón-Cedeño, P. Nakov, Findings of the NLP4IF-2019 Shared Task on Fine-Grained Propaganda Detection, in: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and*

- Propaganda, Association for Computational Linguistics (ACL), 2019, pp. 162–170. URL: <http://arxiv.org/abs/1910.09982>. arXiv:1910.09982.
- [18] G. da San Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, P. Nakov, SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020), Online, 2020, pp. 1377–1414. URL: <http://propaganda.qcri.org/annotations/definitions.html><http://arxiv.org/abs/2009.02696>. arXiv:2009.02696.
- [19] D. Dimitrov, B. Bin Ali, S. Shaar, F. Alam, F. Silvestri, H. Firooz, P. Nakov, G. Da San Martino, SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), 2021, pp. 70–98. doi:10.18653/v1/2021.semeval-1.7. arXiv:2105.09284.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008. arXiv:1706.03762.
- [21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, Association for Computational Linguistics, 2018, pp. 4171–4186. URL: <http://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv: 1907.11692 (2019). URL: <https://arxiv.org/abs/1907.11692v1>. arXiv:1907.11692.
- [23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, volume 1, Association for Computational Linguistics (ACL), 2018, pp. 2227–2237. URL: <https://aclanthology.org/N18-1202>. doi:10.18653/v1/n18-1202. arXiv:1802.05365.
- [24] B. Nye, J. J. Li, R. Patel, Y. Yang, I. J. Marshall, A. Nenkova, B. C. Wallace, A Corpus with Multi-Level Annotations of Patients, Interventions and Outcomes to Support Language Processing for Medical Literature, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018. doi:10.18653/v1/P18-1019.
- [25] O. Shatalov, N. Ryabova, Named Entity Recognition Problem for Long Entities in English Texts, in: The 16th International Conference on Computer Sciences and Information Technologies (CSIT), Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 76–79. doi:10.1109/CSIT52700.2021.9648768.
- [26] F. Li, Z. Wang, S. C. Hui, L. Liao, D. Song, J. Xu, G. He, M. Jia, Modularized Interaction Network for Named Entity Recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 200–209. URL: <https://aclanthology.org/2021.acl-long.17>. doi:10.18653/v1/2021.acl-long.17.
- [27] E. F. Tjong Kim Sang, F. de Meulder, Introduction to the CoNLL-2003 Shared Task:

- Language-Independent Named Entity Recognition, in: Proceedings of the 7th Conference on Natural Language Learning, CoNLL 2003 at HLT-NAACL 2003, 2003, pp. 142–147. URL: <http://lcg-www.uia.ac.be/conll2003/ner/>. arXiv:0306050.
- [28] J. D. Kim, T. Ohta, Y. Tateisi, J. Tsujii, GENIA corpus—a semantically annotated corpus for bio-textmining, *Bioinformatics* 19 (2003). URL: https://academic.oup.com/bioinformatics/article/19/suppl_1/i180/227927. doi:10.1093/BIOINFORMATICS/BTG1023.
- [29] L. A. Ramshaw, M. P. Marcus, Text Chunking Using Transformation-Based Learning, in: ACL Third Workshop on Very Large Corpora, 1995, pp. 82–94. URL: <https://arxiv.org/abs/cmp-lg/9505040v1>. doi:10.1007/978-94-017-2390-9_10. arXiv:9505040.
- [30] A. Vlachos, Tackling the BioCreative2 gene mention task with conditional random fields and syntactic parsing, Proceedings of the Second BioCreative Challenge Evaluation Workshop (2007). URL: http://www.cl.cam.ac.uk/~av308/biocreative2_GM_vlachos.pdf.
- [31] L. Ratinov, D. Roth, Design Challenges and Misconceptions in Named Entity Recognition, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 147–155. URL: <http://l2r.cs.uiuc.edu/>.
- [32] H. C. Cho, N. Okazaki, M. Miwa, J. Tsujii, Named entity recognition with multiple segment representations, *Information Processing and Management* 49 (2013) 954–965. URL: <http://dx.doi.org/10.1016/j.ipm.2013.03.002>. doi:10.1016/j.ipm.2013.03.002.
- [33] M. Konkol, M. Konopík, Segment Representations in Named Entity Recognition, in: Proceedings of the International Conference on Text, Speech and Dialogue (TSD 2015), Springer, Cham, 2015, pp. 61–70. URL: https://link.springer.com/chapter/10.1007/978-3-319-24033-6_7. doi:10.1007/978-3-319-24033-6_7.
- [34] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, 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 (NAACL HLT 2016), 2016, pp. 260–270. URL: <https://github.com/>. doi:10.18653/v1/n16-1030. arXiv:1603.01360.
- [35] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation* 9 (1997) 1735–1780. URL: <http://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. doi:10.1162/neco.1997.9.8.1735.
- [36] J. Pennington, R. Socher, C. Manning, GloVe: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. URL: <https://aclanthology.org/D14-1162/>.
- [37] J. Fu, X. Huang, P. Liu, SpanNER: Named Entity Re-/Recognition as Span Prediction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 7183–7195. URL: <https://aclanthology.org/2021.acl-long.558>.
- [38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90. arXiv:1512.03385.
- [39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple

- Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* 15 (2014) 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [40] G. da San Martino, A. Barrón-Cedeño, P. Nakov, Evaluation of Propaganda Detection Tasks, Technical Report, 2020. URL: https://propaganda.qcri.org/semEval2020-task11/data/propaganda_tasks_evaluation.pdf.
- [41] S. Kullback, R. A. Leibler, On Information and Sufficiency, *The Annals of Mathematical Statistics* 22 (1951) 79–86. URL: <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full><https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694>. doi:10.1214/aoms/1177729694.
- [42] G. Genthial, Sequence Tagging with Tensorflow, 2017. URL: <https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html>.
- [43] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015. arXiv:1412.6980.