information

Article

Learning the Morphological and Syntactic Grammars for Named Entity Recognition

Mengtao Sun [1,*], Qiang Yang [2], Hao Wang [3], Mark Pasquine [4] and Ibrahim A. Hameed [1]

[1] Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 6009 Ålesund, Norway; ibib@ntnu.no
[2] China Telecom (Middle East) FZ-LLC., Dubai 500482, United Arab Emirates; yangqianghk@gmail.com
[3] Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway; hawa@ntnu.no
[4] Department of International Business, Norwegian University of Science and Technology, 6009 Ålesund, Norway; mapa@ntnu.no
* Correspondence: mengtao.sun@ntnu.no

Abstract: In some languages, Named Entity Recognition (NER) is severely hindered by complex linguistic structures, such as inflection, that will confuse the data-driven models when perceiving the word's actual meaning. This work tries to alleviate these problems by introducing a novel neural network based on morphological and syntactic grammars. The experiments were performed in four Nordic languages, which have many grammar rules. The model was named the NorG network (Nor: Nordic Languages, G: Grammar). In addition to learning from the text content, the NorG network also learns from the word writing form, the POS tag, and dependency. The proposed neural network consists of a bidirectional Long Short-Term Memory (Bi-LSTM) layer to capture word-level grammars, while a bidirectional Graph Attention (Bi-GAT) layer is used to capture sentence-level grammars. Experimental results from four languages show that the grammar-assisted network significantly improves the results against baselines. We also investigate how the NorG network works on each grammar component by some exploratory experiments.

Keywords: named entity recognition; morphology; syntax; language processing; deep learning

Citation: Sun, M.; Yang, Q.; Wang, H.; Pasquine, M.; Hameed, I.A. Learning the Morphological and Syntactic Grammars for Named Entity Recognition. *Information* **2022**, *13*, 49. https://doi.org/10.3390/info13020049

Academic Editor: Ricardo Ribeiro

Received: 10 December 2021
Accepted: 17 January 2022
Published: 20 January 2022

1. Introduction

Machine Learning models have widely applied Natural Language Processing (NLP) techniques, which replace the previous rule-based models and show better performances. NLP techniques, such as recommendation and sentiment analysis, have become ubiquitous and necessary among businesses across industries [1]. Named Entity Recognition (NER) is a type of NLP technique based on machine learning models that extracts entities from sentences [2]. Entities are generally considered the keywords in marketing, and recognizing 'named entities', i.e., the name of a person, an organization, a place, and all other entities identified by a name, is vital for the potential business strategies in the big data era [3,4]. On one hand, NER has seen considerable development in English, and many data-driven models have been proposed. On the other hand, there is insufficient research to support languages other than English [5,6]. Compared with English, some languages have many linguistic structures. Aiming at these grammar rules, this work proposes a grammar-based network for named entity recognition and selected four Nordic languages in experiments.

In the Norwegian Bokmål language, one of the experimental languages, some words have feminine, masculine, and neutral varieties. Moreover, there are many compound words, that is, one word consists of several separated words, and the boundaries in compound words are hard to discern. As is shown in Table 1, English words cannot make a one-to-one alignment with Norwegian Bokmål language words, which will hinder the model's performance. Here, 'Høyesterettsjustitiarius' comprises three word-tokens, 'Supreme',

'Court', and 'Justice' in English. However, data-driven models cannot well learn these linguistic structures. This work proposes a novel method incorporating the missing grammar information and discusses how the grammars influence NER performance.

Table 1. Examples of one-to-one alignment in English and Norwegian Bokmål sentences.

English	Norwegian Bokmål
Supreme Court Justice Carsten Smith had ex-Queen Anne-Marie of Greece as her table lady.	Høyesterettsjustitiarius Carsten Smith hadde eks-dronning Anne-Marie av Hellas som sin borddame.
Pürische Nacht is obviously drawn by the pattern of the Crystal Night.	Pürische Nacht er åpenbart tegnet etter mønster av Krystallnatten.
We caught up this threat, and decided to evacuate the school, says police inspector Heidi L. Arneberg at Fredrikstad police station to Aftenposten.no.	Vi fanget opp denne trusselen, og besluttet å evakuere skolen, sier politiinspektør Heidi L. Arneberg ved Fredrikstad politistasjon til Aftenposten.no.

Traditionally, a bidirectional Long Short-Term Memory (Bi-LSTM) layer and a conditional random field (CRF) layer are applied, which encode the sentence in a sequential pattern [7] (see Figure 1). However, as Shen et al. argued, the Bi-LSTM- and CRF-based models also suffer some problems because a sentence does not follow a 'front-to-end' sequential pattern [8]. The NorG network can break through the sequential pattern according to the graphical dependency grammars (see Figure 2). Recently, stronger representations have been proposed. For example, BERT embedding achieved state-of-the-art performance in the NER model [9], but knowledge of the grammar is not well considered during training.

Figure 1. Sequential Pattern.

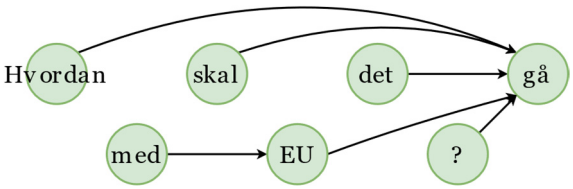

Figure 2. Dependency parsing.

To verify the effectiveness of the linguistic structure for NER, we built a new structure, called the NorG network, and conducted experiments on four Nordic languages. The NorG network incorporates morphological and syntactic grammars. Morphological and syntactic grammars can be obtained by existing tools, such as NLTK [10]. NorG consists of a bidirectional LSTM layer to produce embeddings with word-level grammar, while a bidirectional Graph Attention (Bi-GAT) layer is used to capture the sentence-level dependency grammar. The NorG network reduces the reliability of the word embedding, shows good robustness in the four Nordic languages, and can alleviate the abovementioned problems.

The involved named entities in this study are defined in Table 2. We used ten types of entities. The entity annotation IOB2 is explained in Table 3.

Table 2. Named entity labels.

Entities	Abbreviation	Explanation
Person	PER	Real or fictional characters and animals
Organization	ORG	Any collection of people, such as firms, institutions, and organizations.
Location	LOC	Places, buildings, facilities, etc.
Geo-political entity	GPE	Geographical regions defined by political and/or social groups
	GPE_LOC	GPE with a locative sense
	GPE_ORG	GPE with an organization sense
Product	PROD	Artificially produced entities are regarded as products
Event	EVT	Festivals, cultural events, sports events, weather phenomena, wars, etc.
Derived	DRV	Words that are derived from a name, but arenot a name in themselves.
Miscellaneous	MISC	Other named entities

Table 3. IOB2 entity annotation.

IOB2	Explanation
I-Entity	This token is inside the entity
O	This token is outside the entity
B-Entity	This token is the first token of the entity

The main contributions of this paper can be summarized as follows:

(1) In addition to using embeddings from content, we propose the use of embeddings from different grammars for NER.
(2) We propose the NorG network, which integrates the text content, morphology, and syntax. We found that bidirectional LSTM can capture the morphological knowledge well, and bidirectional GAT can capture the syntactic dependency knowledge well.
(3) Experimental results demonstrate the effectiveness of the proposed method in four languages and some exploratory experiments were conducted to discover the influences of different grammar components on the NER performance.

The rest of the paper is organized as follows. Section 2 introduces related works about Nordic NER resources, Nordic NER algorithms, and general NER algorithms. Section 3 presents the details of how we used the grammar features and explains the process from the input to the output. Section 4 introduces the dataset and experimental setting. Section 5 presents the results and a discussion. Finally, our conclusions on the NorG network can be found in Section 6.

2. Related Works

With the recent release of several Nordic NER databases, more experiments on Nordic languages have become possible. Most notably, ref. [11] published the first Norwegian Bokmål and Nynorsk NER datasets and gave some experimental benchmarks. Ref. [12] presents a Danish NER dataset, and a few tools for Danish NER [13–15] are available. For Finnish NER, ref. [16] collected a new dataset across ten field corpora. FiNER trigger [17,18] is a dictionary- and rule-based method for Finnish NER based on a combination of morphological analysis and an extensive dictionary. However, the existing Nordic NER methods are mostly based on grammar rules or basic deep neural networks.

In recent studies, some researchers have released deep models with a multilingual version that supports Nordic languages. Some pretrained multilingual models can be directly used. BERT [9], the well-known embedding, has been successfully applied to more than 100 languages in different NLP tasks. Moreover, some models can be transferred from English. The FLAIR framework [19] is a Bi-LSTM + CRF based on Huang et al.'s method [7]

and has shown state-of-the-art performance in English NER [20,21]. Hvingelby et al. [12] found that this model works well on the Danish NER dataset. These models are compatible with existing resources. However, they lack a specific deep learning structure that improves the performance by using grammar information.

The milestone of the NER method is CRF [22], which is a probabilistic sequence labeling model. Huang et al. [7] found that bidirectional LSTM can best obtain the sequence features. Their experiments showed a better performance when CRF received the hidden state of the Bi-LSTM layer. The recent success of graph neural networks on node classification has made it possible to handle graph-structured data [23]. Zhang et al. [24] proposed using dependency parse trees to construct a graph for relation extraction. Recently, multi-head attention mechanisms [25] have been widely used by graph neural networks during the fusion process [26,27], which can aggregate graph information by assigning different weights to neighboring nodes or associated edges. In this study, we utilized a bidirectional Graph Attention network, a type of graph neural network, to capture the dependency and improve the performance by word-level and sentence-level grammars.

Most leading NER models are based on BERT [9], a type of word embedding pre-trained by the Transformer architecture [25]. Since its release, it has attracted the attention of many researchers. The state-of-the-art NER models for the Norwegian, Danish, and Finnish languages are based on BERT. Kutuzov et al. [6] introduced the first large-scale monolingual BERT model for Norwegian. They tested their model in a Norwegian NER task and obtained the best results. Hvingelby et al. [12] trained the first monolingual BERT model for Danish, and their results were better than those of the best traditional NER models. Virtanen et al. [28] introduced the first Finnish BERT model. They evaluated their model in a Finnish NER task and obtained the best performance.

3. Materials and Methods

In this Section, we introduce the NorG network that incorporates morphological and syntactic information. A sentence example in the Norwegian Bokmål NER dataset is shown in Table 4.

Table 4. A sentence example in the Norwegian Bokmål NER annotation.

# Text = Hvordan Skal det gå Med EU? (How Will It Go with EU?)					
Word_id	**Word Segments**	**Lemma**	**POS Tag**	**Dependency**	**NER Label**
1	Hvordan (How)	hvordan	ADV	4	name = O
2	skal (will)	skulle	AUX	4	name = O
3	det (it)	det	PRON	4	name = O
4	gå (go)	gå	VERB	0 (root)	name = O
5	med (with)	med	ADP	6	name = O
6	EU (EU)	EU	PROPN	4	name = B-GPE_ORG
7	? (?)	$?	PUNCT	4	name = O

The overall flowchart of message parsing is shown in Figure 3. The explanation is based on the sentence example shown in Table 4. The NorG embedding comprises different morphological structures, and Bi-LSTM is used to mix and produce the word-level embedding. Second, a bi-GAT layer captures the sentence-level dependency, and the output of the bi-GAT layer comprises syntactic knowledge. The implementation of the Bi-GAT layer for nodes and edges is similar to that of the multi-head attention mechanism in Transformer [25].

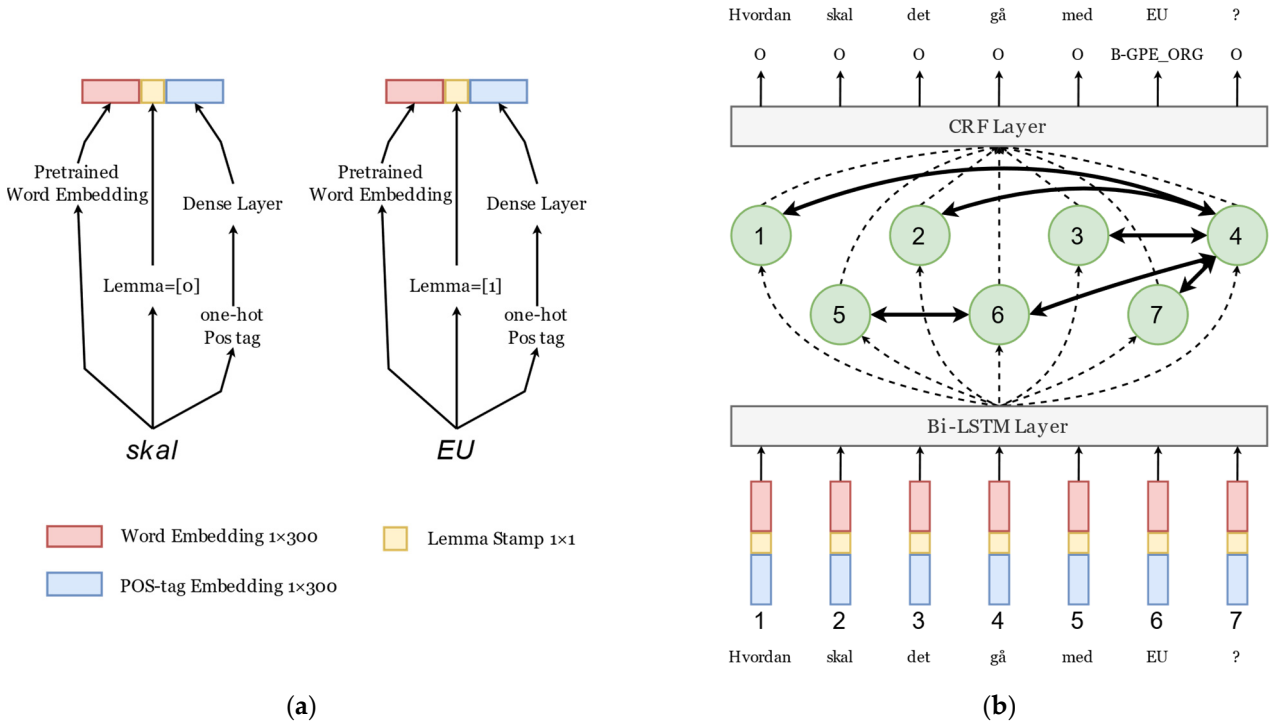

Figure 3. The structure of the NorG network. (**a**) The NorG embedding: word segment, lemma, and POS tag. (**b**) The NorG network: bi-LSTM for encoding the sentence, bi-GAT for receiving the dependency grammar, and the CRF layer for entity recognition.

3.1. NorG Embedding

The NorG embedding comprises different types of linguistic structure that help the model locate named entities.

1. Word Embedding: To obtain context knowledge, we use word-level segments and embeddings, which are pretrained on the corresponding unilingual datasets.
2. Lemma Stamp: A lemma is the base form of a word. Due to the rich morphological changes in languages, some transformations may help the model locate named entities. For example, if a word frequently changes in a corpus, the word can be a verb, an adjective, or a noun but will rarely be an entity. For this reason, a special mark is added to the pretrained embeddings; 1 is set if the lemma and word segments are the same, while 0 is set if the lemma and word segments are different.
3. POS Tag Embedding: Part-of-speech (POS) tags can disambiguate words and improve semantic expression. In universal standards, the POS tag contains 17 classes, such as NOUN (Noun) and ADV (Adverb). In the Penn tree standard, the POS tag contains 36 classes, such as NN (Noun, singular) and NNS (Noun, plural). We found that universal standard POS tags positively impacted named entities in experiments, but there are very few POS tag classes. We reclassified the POS tags through a dense layer. In this way, the NOUN tag was subdivided into more fine-grained dimensions. The meaning of the new POS tags is agnostic because they are produced by neural networks. In the end, a dense connection mapped the one-hot universal POS tag vector to 300 dimensions, so the POS tag provides 300 extra features in the pretrained word embedding (see Figure 3a).
4. Uppercasing or lowercasing leads to different NER results on named entities [29]. By investigating a large number of Nordic sentences, we found that many entities are displayed in capital letters (either the first letter, the entire word, or an abbreviation). For example, Person Name with the first letter capitalized is easier to recognize. Therefore, it is necessary to train the NER model alongside the capitalization, although lowercasing can reduce the vocabulary size and complexity of the neural network.

For a sentence, as shown in Figure 3a, features include a pretrained word embedding $x = \{ x_1, x_2, \ldots, x_N \}$, $x_i \in R_F$, where N is the sentence length and F is the embedding size in each word. Features include lemma stamps l, $l \in \{[0], [1]\}$, where 1 is set if word segments and the lemma are the same, and 0 is set if they are different. Features also include a POS tag $p = \{ p_1, p_2, \ldots, p_N \}$, $p_i \in P_{tag}$, where P_{tag} is the one-hot coding.

Ideally, the lemma feature will be trained as a critic symbol because an entity is usually unchangeable in different sentences.

POS tags are very useful. However, one-hot encoding is insufficient to represent information about them. To improve the expressive ability of POS features, we applied a single layer of dense connections with 300 units after one-hot encoding. That is,

$$p'_i = \sigma(W_p p_i + b_p) \tag{1}$$

To that end, a shared structure, parameterized by a weight matrix W_p and bias b_p, is applied to every step unit. σ is the element-wise ELU non-linearization. The ith word of the NorG embedding is e_i, which is concatenated by a pretrained word embedding, a lemma embedding, and a POS tag embedding. That is,

$$e_i = [\, x_i \parallel l_i \parallel p'_i \,] \tag{2}$$

where $\parallel$ is the concatenation operation, x_i is the pretrained word embedding, l_i is the lemma embedding, and p'_i is the POS tag embedding.

3.2. NorG Network

3.2.1. Bi-LSTM Layer

Figure 3b shows that a Bi-LSTM layer is applied to serialize sentences forward and backward. Bi-LSTM contains two-directional information and achieves a better performance compared with unidirectional LSTM. Following Equation (2), a sentence coding $e = (e_1, e_2, \ldots, e_N)$ goes into the LSTM layer. The layer produces a new set of features $h = (h_1, h_2, \ldots, h_N)$ as its output. Formally, the formulas to update a LSTM unit at time i are

$$I_i = \alpha(W_I e_i + U_I h_{i-1} + b_I) \tag{3}$$

$$F_i = \alpha(W_F e_i + U_F h_{i-1} + b_F) \tag{4}$$

$$O_i = \alpha(W_O e_i + U_O h_{i-1} + b_O) \tag{5}$$

$$\widetilde{c} = tanh(W_c e_i + U_c h_{i-1} + b_c) \tag{6}$$

$$c_i = F_i \cdot c_{i-1} + I_i \cdot \widetilde{c}_i \tag{7}$$

where α is the element-wise sigmoid function, $\cdot$ is the element-wise matrix multiplication, e_i is the input vector at time i, h_i represents the hidden state vector, and $W_I, W_F, W_O, W_c, U_I, U_F, U_O, U_c, b_I, b_F, b_O, b_c$ are trainable parameters. Given a sequence of input vectors $(e_1, e_2, \ldots, e_N)$, LSTM computes a context representation vector h_i for each input e_i. For bidirection, the final representation of a word is obtained by concatenating the left context $\overrightarrow{h_i}$ and the right context $\overleftarrow{h_i}$, that is, $h_i = [\overrightarrow{h_i} \parallel \overleftarrow{h_i}]$.

3.2.2. Bi-GAT Layer

Graph neural networks have been successfully applied to node classification. We used the GAT method proposed by [23,30] to obtain the syntactic grammar. Figure 3b shows that each node represents a word segment, and the dependency relationship between words can be treated as an edge. The dependency graph changes the situation such that the words can only be serialized forward or backward.

Given the node features h, GAT will generate the output s, which contains dependency grammars and preceding features. The hidden states $h = (h_1, h_2, \ldots, h_N)$ are converted into a bidirected graph, as shown in Figure 3b, where each node represents a word and

the connection $edge_{k,i}$ from syntax dependencies can be treated as an edge. The state of the ith node represents the features of the ith token in a text sequence. The state of each edge represents the features of a corresponding dependency word, which help the node remove ambiguity. In GAT, we used multi-head attention to aggregate the corresponding predecessor nodes h_k for each node h_i. The incoming edges and predecessor nodes are able to indicate the meaning of a variety of words. The node aggregation can be formulated as followed:

$$s_i = \text{MultiAtt}\left(h_i, \left\{\forall \text{k}[h_i \parallel edge_{k,i}]\right\}\right) \tag{8}$$

where i refers to the aggregation at the ith step and $\parallel$ represents the concatenation operation. We found that bidirectional GAT performs better compared with unidirectional GAT. The final outputs become the concatenation of two directions, i.e., $s_i = [\ \overrightarrow{s_i} \parallel \overleftarrow{s_i}\ ]$.

3.2.3. CRF Layer

After the graph message processing, a conditional random field (CRF) layer is used. The CRF layer adds some constraints to ensure that the predicted label has a legal order between the output tags. For example, the I-PER (I-person) tag should follow the B-PER (B-person) tag or the I-PER tag, while it cannot be behind the B-LOC (B-location) tag. In this stage, we obtain the sequence of final node states $s = (s_1,\ s_2, \ldots,\ s_N)$. The probability of a NER label sequence $y = (y_1,\ y_2\ ,\ \ldots,\ y_N)$ can be defined as follows:

$$p(y|X) = \frac{exp\left(\Sigma_{i=1}^{n}\phi(y_{i-1}, y_i, s_i)\right)}{\Sigma_{\widetilde{y}\in Y(X)} exp\left(\Sigma_{i=1}^{n}\phi(\widetilde{y}_{i-1}, \widetilde{y}_i, s_i)\right)} \tag{9}$$

where $\widetilde{y}$ is an arbitrary label sequence, and $Y(X)$ is the set of all possible output label sequences for the model input X.

$$\phi(y_{i-1}, y_i, s_i) = W_{y_{i-1},y_i} s_i + b_{y_{i-1},y_i} \tag{10}$$

where W_{y_{i-1},y_i} and b_{y_{i-1},y_i} are the weight and bias parameters specific to the labels y_{i-1}, y_i, $\widetilde{y}_{i-1}$, and $\widetilde{y}_i$, respectively.

For training, we minimized the sentence-level negative log-likelihood loss as follows:

$$L = -\sum_{i=1}^{N} log(p(\ y_i|X_i)) \tag{11}$$

For testing and decoding, we maximized the likelihood to find the optimal sequence,

$$y' = \underset{y\in Y(X)}{argmax} p(\ y|s) \tag{12}$$

To calculate the above equations, we used the Viterbi algorithm, which can reduce the computational complexity efficiently.

4. Experiments

This section describes the four selected Nordic language corpora (Norwegian Bokmål, Norwegian Nynorsk, Danish, and Finnish). Then, we introduce the baselines applied for comparison and detail the hyperparameter configuration of the proposed model.

4.1. NER Datasets

Our model was evaluated in the NorNE (Norwegian Bokmål), NorNE (Norwegian Nynorsk), DaNE (Danish), and Turku NER (Finnish) datasets whose linguistic structures are annotated in CONLL-U format. All the NER labels were manually annotated based on Tables 2 and 3. The linguistic structures applied for the NorG Network are listed in Table 5.

Table 5. Linguistic structures of the NorG network.

Inputs	Outputs
Word Segment POS Tag Lemma Upper/Lower Case Dependency	NER label

4.1.1. Norwegian Bokmål and Nynorsk

NorNE [11] comprises two types of Norwegian: the Bokmål language and the Nynorsk language. Their morphological and syntactic grammars were both obtained from the Norwegian Dependency Treebank [31]. The dependency was converted to the Universal Dependencies (UD) standard by [32,33]. The text resources were mostly extracted from Norwegian News. The Bokmål language contains 16,309 sentences, and the Nynorsk language contains 14,878 sentences. Each language has eight entity labels: PER, ORG, LOC, GPE_LOC, GPE_ORG, PROD, EVT, DRV, and MISC.

4.1.2. Danish

DaNE [12] is a moderate-size dataset for Danish NER. Grammars come from the Copenhagen Dependency Treebank proposed by [34]. The Danish dependency was converted to the UD standard by [35]. The source is texts from the Danish PAROLE corpus [36], which comprises a range of textual domains, both written and spoken, from the years 1983–1992. This dataset consists of 474 texts with 5512 sentences and contains PER, LOC, ORG, and MISC as its named entities.

4.1.3. Finnish

Turku NER [16] is a Finnish NER dataset. The grammars are presented in [37]. We selected six NER labels: PER, ORG, LOC, GPE, PROD, and EVT. The texts consist of 754 documents representing ten different genres of text with 15,136 sentences in total.

The division of the sentences into training, validation, and testing sentences in the four languages is shown in Table 6.

Table 6. Training, validation and testing sentences.

Dataset	Train	Val	Test
NorNE (Bokmål)	15,696	2410	1939
NorNE (Nynorsk)	14,174	1890	1511
DaNE (Danish)	4383	564	565
Turku NER (Finish)	12,217	1364	1555

Each word in the four languages is naturally space-segmented, and the texts were carefully cleaned to ensure that each word is correct. The embeddings of word tokens were pretrained using the FastText model (abbreviated as cbow) [38] and fixed during training. In the experiments, the pretrained cbow embedding showed a greater capability than the initialized embedding (abbreviated as ie).

We did not employ other segments on the NorG network because word-level tokens can align with grammar information. In the experiments, we still compared our results with recent works that utilize fine-grained segments, such as Byte Pair Encoding (BPE) and character-level segments. The NorG network is good enough to recognize entities compared with the recent works.

The number of word tokens in the corpora of Bokmål, Nynorsk, Danish, and Finnish languages is 301.9 k, 292.3 k, 100.7 k, and 202.1 k, respectively. The number of Bokmål entities, Nynorsk entities, Danish entities, and Finnish entities is 14.4 k, 13.9 k, 5.0 k, and 11.4 k, respectively. The cbow embeddings were pretrained on Common Crawl and

Wikipedia, with a dimension of 300, character n-grams of length 5, a window of size 5, and 10 negatives, using cbow300 released by [38]. The cased vocabulary lists of cbow300 embeddings in Bokmål, Nynorsk, Danish, and Finnish languages contain 35.2 k, 33.7 k, 19.1 k, and 55.4 k words, respectively.

4.2. Baselines

We applied recent NER models corresponding to each language and three general deep neural networks for comparison. The general baselines used were convolution neural network (CNN), Bi-LSTM, and bidirectional Gated Recurrent Unit (Bi-GRU). In the general baselines, we compared the pretrained cbow300 embeddings and the initialized embeddings.

The CNN method: This model has a word-level convolutional layer after the embedding layer. A CRF layer is used in the convolutional encoding.

The Bi-LSTM method: This model has a Bi-LSTM layer after the embedding layer. We concatenated the hidden vectors of the forward LSTM and the backward LSTM, and a standard CRF layer was used.

The Bi-GRU method: GRU is a simplified version of LSTM. Compared with LSTM, it reduces the model's complexity and maintains the efficacy of memories in long sequences. We employed Bi-GRU to replace the Bi-LSTM units.

4.3. Hyperparameters of the NorG Network

We set the four language models with the same hyperparameter configuration. We used Adam as the optimizer, with a default learning rate of 0.001 for the four languages. A dense layer was applied, which refined and specified the POS tag into 300 dimensions. We utilized bidirectional LSTM with 100 hidden states to composite the word embedding, POS tag, and lemma information. The number of heads for multi-head attention was 8. To further reduce overfitting, we employed Dropout with a rate of 0.5 after the embeddings and a rate of 0.6 after the GAT layer. ELU was applied as an activating function. The batch size was set to 32. The standard Precision (P), Recall (R), and F1-score (F1) were used as evaluation metrics.

5. Results

In this section, we present the main results of the NorG network on the four Nordic-language NER tasks. The model achieving the best results on the development set was chosen for the final evaluation on the test set. We also probed the effectiveness and interpretability of the NorG network by explanatory experiments.

5.1. Main Results

Table 7 shows the results of the NorG network and baselines on the Bokmål language. NCRF++ [39] is a popular NER toolkit that combines a character-level CNN and a word-level Bi-LSTM, which feed into a CRF inference layer. The authors of [11] found that, based on NCRF++, combining Bokmål and Nynorsk in the training set (i.e., BM + NN) can achieve a better F1 score. Ref. [6] is the Norwegian version of BERT and is known to be one of the best language models evaluated in most English benchmark tasks.

Our proposed NorG network incorporates grammars that enabled it to outperform the best baseline by 3.23% in terms of F1 score and provide an apparent increase in Precision and Recall. Finally, it had a precision of 98.10%, a recall of 94.76%, and an F1 score of 96.28%. We also found that the Bi-LSTM and bi-GRU models can better perceive the features compared with the CNN model in the Bokmål NER model.

Table 7. Main results on Norwegian–Bokmål NER.

Baseline	P	R	F1
NCRF++ (2018) [39]	-	-	89.47
BM + NN (2020) [11]	-	-	90.92
NorBERT (2021) [6]	-	-	85.50
cbow300 + CNN + crf	95.84	68.37	79.22
ie + CNN + crf	80.24	71.41	75.05
cbow300 + biLSTM + crf	96.27	90.20	92.97
ie + CNN + crf	83.74	74.20	78.19
cbow300 + biGRU + crf	96.28	89.42	92.55
ie + biGRU + crf	87.07	73.77	79.39
NorG network	98.10	94.76	96.28

Table 8 shows the results of the NorG network and baselines on the Nynorsk language. The NorG network obtained 98.72, 88.22, 92.92 in Precision, Recall, and F1 score, respectively, leading all the Nynorsk-language NER models.

Table 8. Main results on Norwegian–Nynorsk NER.

Baseline	P	R	F1
NCRF++ (2018) [39]	-	-	86.53
BM + NN (2020) [11]	-	-	88.03
NorBERT (2021) [6]	-	-	82.80
cbow300 + CNN + crf	95.24	44.94	59.74
ie + CNN + crf	83.49	67.90	73.96
cbow300 + biLSTM + crf	94.49	76.39	83.99
ie + CNN + crf	83.00	69.36	74.56
cbow300 + biGRU + crf	93.70	72.71	81.32
ie + biGRU + crf	90.21	63.58	73.72
NorG network	98.72	88.22	92.92

Table 9 shows the results of the NorG network and baselines on the Danish language. FLAIR is a toolkit that contains several pretrained NER models. The framework is Bi-LSTM + CRF with the option of passing concatenated embeddings of different types. FLAIR in Table 9 uses a concatenation of Word-Level FastText embeddings and FLAIR embeddings, which are 1024-dimensional hidden states extracted from a Bi-LSTM character-level language model. FLAIR + BPE uses FastText with BPE embeddings and pretrained FLAIR embeddings. BERT is a transformer-based architecture that was shown to obtain higher performance on NER. DanishBERT was pretrained on data from Common Crawl, Danish Wikipedia, OpenSubtitles, and various online forums. The experimental results on DaNE were obtained by [12].

Compared with the current methods and our baselines, the NorG network gave the best P, R, and F1 in Danish. The grammars are well integrated into the network and helped NorG network obtain a precision of 98.07, a recall of 80.80, and an F1 score of 88.33 F1, outperforming the other methods by a large margin.

Table 9. Main results on Danish NER.

Baseline	P	R	F1
FLAIR (2019) [19]	-	-	79.70
FLAIR + BPE (2020) [12]	-	-	78.05
DanishBERT (2020) [40]	-	-	83.76
cbow300 + CNN + crf	96.06	56.61	70.82
ie + CNN + crf	70.93	68.96	69.38
cbow300 + biLSTM + crf	95.22	78.60	85.97
ie + CNN + crf	84.58	65.48	73.56
cbow300 + biGRU + crf	94.87	77.79	85.19
ie + biGRU + crf	76.19	67.64	71.53
NorG network	98.07	80.80	88.33

Table 10 shows the results of the NorG network and baselines on the Finnish language. CRFsuite [41] is an implementation of simple CRF. NCRF++ [39] has the same structure as in Bokmål and Nynorsk. FiNER tagger [17] is a dictionary- and rule-based system that detects named entities based on known words. FinBERT [28], i.e., BERT in Finnish, is a state-of-the-art deep transfer learning model based on Transformer. Moreover, we applied the CNN/Bi-LSTM/Bi-GRU baselines on Finnish languages with two types of embedding.

Table 10. Main results on Finnish NER.

Baseline	P	R	F1
CRFsuite (2007) [41]	74.53	63.18	68.39
NCRF++ (2018) [38]	82.92	80.20	81.54
FiNER tagger (2017) [17]	77.16	71.24	74.08
FinnishBERT (2019) [28]	90.87	92.44	91.65
cbow300 + CNN + crf	96.93	51.20	66.23
ie + CNN + crf	77.00	54.43	63.09
cbow300 + biLSTM + crf	90.61	78.51	83.78
ie + CNN + crf	82.98	52.45	63.27
cbow300 + biGRU + crf	91.18	77.45	83.49
ie + biGRU + crf	86.27	52.90	64.57
NorG network	96.06	87.25	91.24

The NorG network shows the best precision of all the methods, but the F1 score is slightly (0.41%) lower than that of the FinBERT model. One intuition is that the word-level segmentation is insufficient in the Finnish language because Finnish words contain more characters and the Finnish vocabulary is much larger (55.4 k). However, FinBERT is a very large architecture and has numerous hyperparameters. The NorG network still showed a good capability to utilize the morphological and syntactic grammars. It significantly outperformed the other word-level models. It had a precision of 96.06, a recall of 87.25, and an F1 score of 91.24 on the testing dataset.

5.2. Ablation Experiments

To study the contribution of each component, we conducted ablation experiments on the four language datasets. The results are displayed in Table 11. In the NorG network, we used the Bi-GAT layer to present syntactic dependency information. If the GAT layer is replaced with another graph layer to perceive the syntactic grammar (Graph Convolutional Network (GCN) [42], GCN + Skip Connection (GCS) [43], or GCN + Localized Spectral Filter (Chebnet) [44]), the model performances generally degrade. Among the models, GCN totally collapses when representing the dependency grammar. GCS slightly improves the Finnish language results but hurts the other languages' results. Overall, Chebnet showed

lower performance than GAT in the four languages. Therefore, Table 11 indicates that GAT is the best graph neural network for integrating the dependency grammar.

Table 11. Ablation study on the validation dataset.

Model	Bokmål	Nynorsk	Danish	Finnish
NorG	96.48	93.35	92.01	89.96
Use Initialized Embedding	94.00	92.03	89.40	87.72
Remove POS Tag	75.73	61.29	74.43	71.37
Remove Capitalization	94.04	88.42	85.27	89.11
Remove Lemma	94.38	90.28	85.65	88.51
Use Unidirectional Dependency	95.06	93.29	85.18	89.11
Use a Bi-GCN layer for Dependency	49.90	47.63	38.68	41.32
Use a Bi-GCS layer for Dependency	95.84	92.87	85.12	91.05
Use a Bi-Cheb layer for Dependency	94.23	91.94	85.58	89.25

Suppose we substitute cbow300 embedding with initialized embedding. The F1 scores decrease by 2.0% on average in the four languages. The decrease shown in Tables 7–10 is very slight (decreasing by 10% on average when using initialized embedding), which indicates that embedding will influence but not determine the performance when considering morphological and syntactic grammars. We found that the POS tag strongly positively impacted the NER performance. It caused more than a 20% decline in Bokmål, Danish, and Finnish and a 32.06% decline in Nynorsk. Using lowercase words or deleting the lemma token marks will also cause some errors. The NorG network showed better F1 performance with bidirectional dependency. The F1 score of the Danish language was 92.01% with bidirectional dependency and 85.18% with unidirectional dependency.

5.3. Performance against Sentence Length

Figure 4 shows the performance of the NorG network and baseline models on the Bokmål dataset. We split the dataset into five parts according to the sentence length. The results show that our proposed network outperforms the other word-level baselines over both short and long sentences. Moreover, the performance of each baseline is influenced by the sentence length and exhibits a general decrease when the sentences are short or long. In contrast, the NorG network yields better results than the baselines and demonstrates effectiveness and robustness when the sentence length changes. The F1 scores are stable at about 94%.

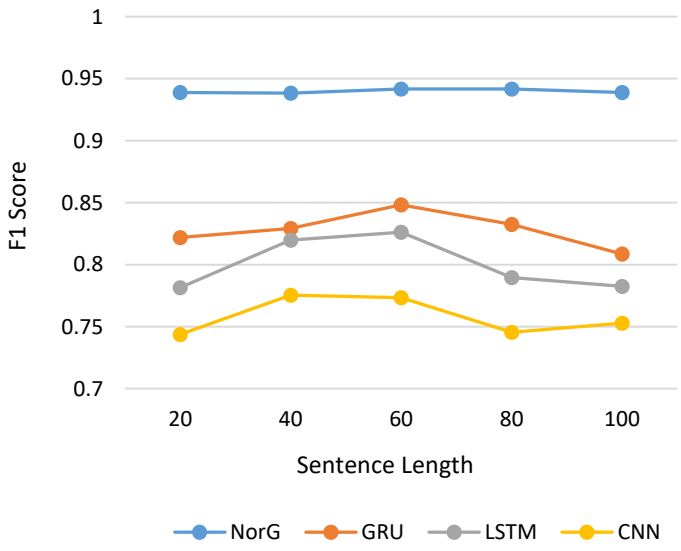

Figure 4. Performance against sentence length.

To sum up, the morphological and syntactic grammars are only slightly affected by different sentence lengths. The results indicate that the NorG network utilizes grammars well, yields a higher F1 score than the baselines, and is not disturbed by different sentence lengths.

5.4. Training Step

To investigate the influence of step numbers during the update process, we analyzed the performance of baselines and NorG networks on the Bokmål language under different training steps. Figure 5 illustrates the variation of the F1 score on the development sets when the step number increases. As used in [45], we applied D-F1 to represent the F1 scores at different steps minus the best results.

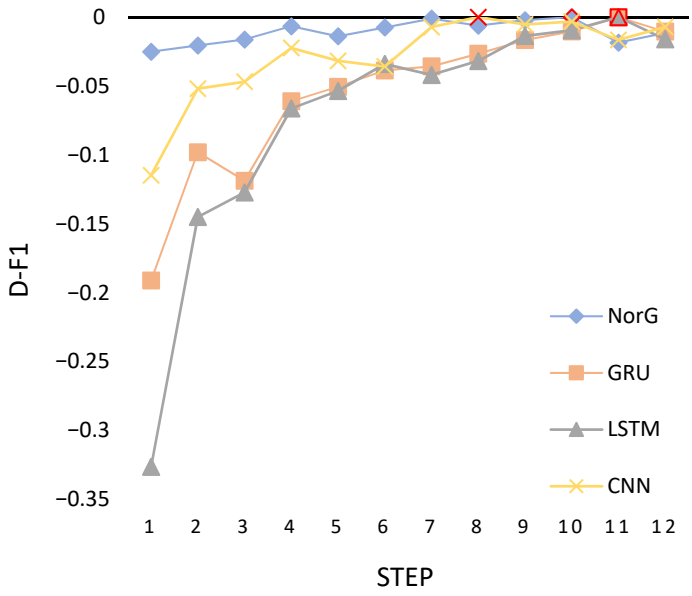

Figure 5. Training step on the validation dataset.

Figure 5 indicates that the NorG network can extract some entities in the early stage, while the number of update steps has a greater influence on the baseline models. The best results occur at the 8th step on the CNN, the 11th step on Bi-LSTM and Bi-GRU, and the 10th step on the NorG network. However, at the baselines, the F1 score decreases by more than 5.00% compared with the best results when the step number is less than 5, while the NorG network has less of an influence. The NorG curve is stable after the model training. The results indicate that morphological and syntactic grammars can help the NER model recognize entities at an early training step and do not influence the model learning process.

5.5. Performance on Automatically Obtained Grammars

Currently, linguistic features can be automatically obtained. Some grammars can be directly observed by a lexical search, while some grammars, such as Pos Tagging and Dependency Parsing, can be acquired using existing intelligent algorithms. In this section, we evaluate the performance of the NorG network on the Norwegian-Bokmål language to compare the performance under the condition of gold-standard and automatically obtained linguistic structures. Gold-standard linguistic structures are based on the abovementioned treebank, which was applied in previous experiments of this work. Automatically obtained linguistic structures are based on spaCy [46], a capable toolkit that contains various pretrained language models. The comparison is shown in Figure 6.

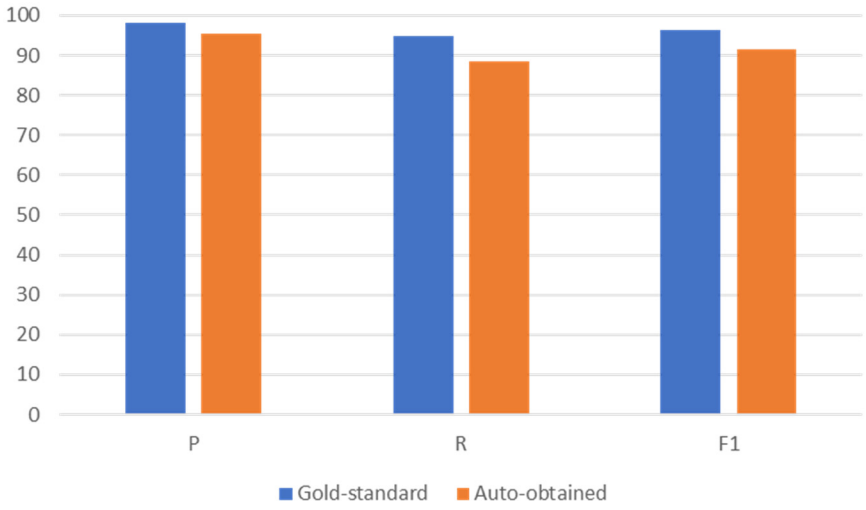

Figure 6. Comparison of the NorG network by gold-standard and automatically obtained grammars.

Automatically obtained grammars are preferable for industry-oriented research. As illustrated in the spaCy toolkit, POS Tagging achieved 97% accuracy and Dependency Parsing reached 89% accuracy in the Norwegian-Bokmål language. Figure 6 indicates that using automatically obtained grammars will cause a slight decrease in the evaluation, but the results are still satisfactory. The precision is about 95%, the recall is about 90%, and the F1 score is over 90%. In conclusion, the results of the automatically obtained grammar show that the NorG network can be used when the model receives a sentence so that it could be developed in an industrial environment.

6. Conclusions

The complex linguistic structure will hinder data-driven models from understanding the text's meaning accurately. This work investigated morphological and syntactic grammars and proposed a grammar-based model for named entity recognition. The proposed model is named the NorG network. Experimental results indicate that the NorG network can take advantage of morphological and syntactic grammars and help to identify named entities. In summary, the benefits of the NorG network are as follows.

(1) The results of the NorG network are the best results to be obtained in recent research.
(2) The NorG network is able to perceive the grammar features from each component.
(3) The NorG network shows good robustness and was only slightly influenced by sentence length.
(4) The NorG network can extract some entities during early training and shows good stability during training.

In the future, we will supplement the model with more linguistic structures to increase the model's performance. We will also improve the fusion structure so that knowledge of grammar can be better integrated into the NER models. Moreover, we will explore industrial applications by using the proposed method.

Author Contributions: Conceptualization, M.S.; methodology, M.S.; software, M.S. and Q.Y.; validation, M.S.; formal analysis, M.S.; investigation, M.S. and Q.Y.; resources, M.S.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, H.W., M.P. and I.A.H.; visualization, M.S. and Q.Y.; supervision, H.W., M.P. and I.A.H.; project administration, H.W., M.P. and I.A.H.; funding acquisition, H.W., M.P. and I.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Norwegian University of Sciences and Technology [Prosjektnummer 70441595].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in reference [11,12,16].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pollák, F.; Dorčák, P.; Markovič, P. Corporate Reputation of Family-Owned Businesses: Parent Companies vs. Their Brands. *Information* **2021**, *12*, 89. [CrossRef]
2. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
3. Gu, K.; Vosoughi, S.; Prioleau, T. SymptomID: A framework for rapid symptom identification in pandemics using news reports. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2021**, *12*, 1–17. [CrossRef]
4. Lee, D.; Oh, B.; Seo, S.; Lee, K.H. News recommendation with topic-enriched knowledge graphs. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Conference, 19–23 October 2020; pp. 695–704.
5. Tanvir, H.M.; Kittask, C.; Sirts, K. EstBERT: A Pretrained Language-Specific BERT for Estonian. In Proceedings of the 23rd Nordic Conference on Computational Linguistics, Reykjavik, Iceland, 31 May–2 June 2021.
6. Kutuzov, A.; Barnes, J.; Velldal, E.; Ovrelid, L.; Oepen, S. Large-Scale Contextualised Language Modelling for Norwegian. In Proceedings of the 23rd Nordic Conference on Computational Linguistics, Reykjavik, Iceland, 31 May–2 June 2021.
7. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
8. Shen, Y.; Tan, S.; Sordoni, A.; Courville, A. Ordered neurons: Integrating tree structures into recurrent neural networks. In Proceedings of the ICLR, New Orleans, LA, USA, 6–9 May 2019.
9. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 3–5 June 2019; Volume 1, pp. 4171–4186.
10. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009.
11. Jørgensen, F.; Aasmoe, T.; Husevåg, A.S.R.; Øvrelid, L.; Velldal, E. NorNE: Annotating named entities for Norwegian. In Proceedings of the 12th Language Resources and Evaluation Conference, Le Palais du Pharo, France, 11–16 May 2020; pp. 4547–4556.
12. Hvingelby, R.; Pauli, A.B.; Barrett, M.; Rosted, C.; Lidegaard, L.M.; Søgaard, A. DaNE: A named entity resource for danish. In Proceedings of the 12th Language Resources and Evaluation Conference, Le Palais du Pharo, France, 11–16 May 2020; pp. 4597–4604.
13. Derczynski, L. Simple natural language processing tools for Danish. *arXiv* **2019**, arXiv:1906.11608.
14. Bick, E. A named entity recognizer for Danish. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, Lisbon, Portugal, 26–28 May 2004.
15. Johannessen, J.B.; Hagen, K.; Haaland, Å.; Jónsdottir, A.B.; Nøklestad, A.; Kokkinakis, D.; Meurer, P.; Bick, E.; Haltrup, D. Named entity recognition for the mainland Scandinavian languages. *Lit. Linguist. Comput.* **2005**, *20*, 91–102. [CrossRef]
16. Luoma, J.; Oinonen, M.; Pyykönen, M.; Laippala, V.; Pyysalo, S. A broad-coverage corpus for finnish named entity recognition. In Proceedings of the 12th Language Resources and Evaluation Conference, Le Palais du Pharo, France, 11–16 May 2020; pp. 4615–4624.
17. Kettunen, K.; Löfberg, L. Tagging named entities in 19th century and modern Finnish newspaper material with a Finnish semantic tagger. In Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden, 22–24 May 2017; pp. 29–36.
18. Ruokolainen, T.; Kauppinen, P.; Silfverberg, M.; Lindén, K. A Finnish news corpus for named entity recognition. *Lang. Resour. Eval.* **2020**, *54*, 247–272. [CrossRef]
19. Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; Vollgraf, R. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA, 3–5 June 2019; pp. 54–59.
20. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
21. Akbik, A.; Bergmann, T.; Vollgraf, R. Pooled contextualized embeddings for named entity recognition. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 3–5 June 2019; Volume 1, pp. 724–728.
22. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
23. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
24. Zhang, Y.; Qi, P.; Manning, C.D. Graph convolution over pruned dependency trees improves relation extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.

26. Zhang, J.; Shi, X.; Xie, J.; Ma, H.; King, I.; Yeung, D.Y. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In Proceedings of the Association for Uncertainty in Artificial Intelligence, Monterey, CA, USA, 7–9 August 2018; pp. 339–349.
27. Lee, J.B.; Rossi, R.; Kong, X. Graph classification using structural attention. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 19–23 August 2018; pp. 1666–1674.
28. Virtanen, A.; Kanerva, J.; Ilo, R.; Luoma, J.; Luotolahti, J.; Salakoski, T.; Ginter, F.; Pyysalo, S. Multilingual is not enough: BERT for Finnish. *arXiv* **2019**, arXiv:1912.07076.
29. Akhtyamova, L.; Martínez, P.; Verspoor, K.; Cardiff, J. testing contextualized word embeddings to improve NER in Spanish clinical case narratives. *IEEE Access* **2020**, *8*, 164717–164726. [CrossRef]
30. Yan, H.; Jin, X.; Meng, X.; Guo, J.; Cheng, X. Event detection with multi-order graph convolution and aggregated attention. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 5766–5770.
31. Solberg, P.E.; Skjærholt, A.; Øvrelid, L.; Hagen, K.; Johannessen, J.B. The Norwegian Dependency Treebank. In Proceedings of the Ninth International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 26–31 May 2014.
32. Øvrelid, L.; Hohle, P. Universal Dependencies for Norwegian. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, Portorož, Slovenia, 23–28 May 2016.
33. Velldal, E.; Øvrelid, L.; Hohle, P. Joint UD parsing of Norwegian Bokmål and Nynorsk. In Proceedings of the 21st Nordic Conference of Computational Linguistics, Gothenburg, Sweden, 22–24 May 2017; pp. 1–10.
34. Buch-Kromann, M. The danish dependency treebank and the DTAG treebank tool. In Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT), Växjö, Sweden, 14–15 November 2003; pp. 217–220.
35. Johannsen, A.; Alonso, H.M.; Plank, B. Universal dependencies for danish. In Proceedings of the International Workshop on Treebanks and Linguistic Theories (TLT14), Warsaw, Poland, 11–12 December 2015; p. 157.
36. Keson, B. *Vejledning til det Danske Morfosyntaktisk Taggede PAROLE-Korpus*; Technical Report; Det Danske Sprog og Litteraturselskab (DSL): Copenhagen, Denmark, 2000.
37. Nivre, J.; De Marneffe, M.C.; Ginter, F.; Goldberg, Y.; Hajic, J.; Manning, C.D.; McDonald, R.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al. Universal dependencies v1: A multilingual treebank collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 1659–1666.
38. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
39. Yang, J.; Zhang, Y. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 74–79.
40. Wolf, T.; Chaumond, J.; Debut, L.; Sanh, V.; Delangue, C.; Moi, A.; Cistac, P.; Funtowicz, M.; Davison, J.; Shleifer, S.; et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Virtual, 16–20 November 2020; pp. 38–45.
41. Okazaki, N. CRFsuite: A Fast Implementation of Conditional Random Fields. Software Package. 2007. Available online: http://www.chokkan.org/software/crfsuite (accessed on 20 May 2021).
42. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 2017 International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
43. Huang, W.; Zhang, T.; Rong, Y.; Huang, J. Adaptive sampling towards fast graph representation learning. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montréal, QC, Canada, 2–8 December 2018.
44. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Processing Syst.* **2016**, *29*, 3844–3852.
45. Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; Huang, X.J. A lexicon-based graph neural network for chinese ner. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 1040–1050.
46. Honnibal, M.; Montani, I.; Van Landeghem, S.; Boyd, A. spaCy: Industrial-Strength Natural Language Processing in Python. Software Package. 2020. Available online: https://spacy.io (accessed on 20 May 2021).