Explaining Character-Aware Neural Networks for Word-Level Prediction: Do They Discover Linguistic Rules?

Fréderic Godin, Kris Demuynck, Joni Dambre, Wesley De Neve and Thomas Demeester

IDLab, Ghent University - imec, Ghent, Belgium
firstname.lastname@ugent.be

Abstract

Character-level features are currently used in different neural network-based natural language processing algorithms. However, little is known about the character-level patterns those models learn. Moreover, models are often compared only quantitatively while a qualitative analysis is missing. In this paper, we investigate which character-level patterns neural networks learn and if those patterns coincide with manually-defined word segmentations and annotations. To that end, we extend the contextual decomposition (Murdoch et al., 2018) technique to convolutional neural networks which allows us to compare convolutional neural networks and bidirectional long short-term memory networks. We evaluate and compare these models for the task of morphological tagging on three morphologically different languages and show that these models implicitly discover understandable linguistic rules.

1 Introduction

Character-level features are an essential part of many Natural Language Processing (NLP) tasks. These features are for instance used for language modeling (Kim et al., 2016), part-of-speech tagging (Plank et al., 2016) and machine translation (Luong and Manning, 2016). They are especially useful in the context of part-of-speech and morphological tagging, where for example the suffix *-s* can easily differentiate plural words from singular words in English or Spanish.

The use of character-level features is not new. Rule-based taggers were amongst the earliest systems that used character-level features/rules for grammatical tagging (Klein and Simmons, 1963). Other approaches rely on fixed lists of affixes (Ratnaparkhi, 1996; Toutanova et al., 2003). Next, these features are used by a tagging model, such as a rule-based model or statistical model. Rule-based taggers are transparent models that allow us to easily trace back why the tagger made a certain decision (e.g., Brill (1994)). Similarly, statistical models are merely a weighted sum of features.

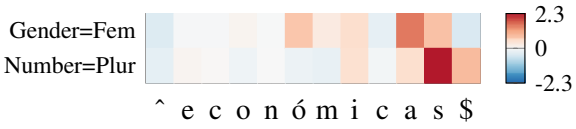


Figure 1: Individual character contributions of the Spanish adjective *económicas*. The character *a* has the highest positive (red) contribution for predicting the label *Gender=Fem*, and the character *s* for predicting the label *Number=Plur*. This coincides with our linguistic knowledge of Spanish.

For example, Brill (1994)'s transformation-based error-driven tagger uses a set of templates to derive rules by fixing errors. The following rule template:

*"Change the most-likely tag **X** to **Y** if the last (1,2,3,4) characters of the word are **x**"*,

resulted in the rule:

*"Change the tag **common noun** to **plural common noun** if the word has suffix **-s**"*.

Subsequently, whenever the tagger makes a tagging mistake, it is easy to trace back why this happened. Following the above rule, the word *mistress* will mistakingly be tagged as a plural common noun while it actually is a common noun[1].

This is in stark contrast with the most recent generation of part-of-speech and morphological taggers which mainly rely on neural networks.

[1] In Brill (1994), an additional rule encodes an exception to this rule to correctly tag the word *mistress*.

Words are split into individual characters and are in general either aggregated using a Bidirectional Long Short-Term Memory network (BiLSTM) (Plank et al., 2016) or Convolutional Neural Network (CNN) (dos Santos and Zadrozny, 2014). However, it is currently unknown which character-level patterns these neural network models learn and whether these patterns coincide with our linguistic knowledge. Moreover, different neural network architectures are currently only compared quantitatively and lack a qualitative analysis.

In this paper, we investigate which character patterns neural networks learn and to what extent those patterns comprise any known linguistic rules. We do this for three morphologically different languages: Finnish, Spanish and Swedish. A Spanish example is shown in Figure 1. By visualizing the contributions of each character, we observe that the model indeed uses the suffix *-s* to correctly predict that the word is plural.

Our main contributions are as follows:

- We show how word-level tagging decisions can be traced back to specific sets of characters and interactions between them.
- We extend the contextual decomposition method (Murdoch et al., 2018) to CNNs.
- We quantitatively compare CNN and BiLSTM models in the context of morphological tagging by performing an evaluation on three manually segmented and morphologically annotated corpora.
- We found out that the studied neural models are able to implicitly discover character patterns that coincide with the same rules linguists use to indicate the morphological function of subword segments.

Our implementation is available online[2].

2 Related Work

Neural network-based taggers currently outperform statistical taggers in morphological tagging (Heigold et al., 2017) and part-of-speech tagging (Plank et al., 2016) for a wide variety of languages. Character-level features form a crucial part of many of these systems. Generally, two neural network architectures are considered for aggregating the individual characters: a BiLSTM (Ling et al., 2015; Plank et al., 2016) or a CNN (dos Santos and Zadrozny, 2014; Bjerva et al., 2016; Heigold et al., 2017). These architectures outperform similar models that use manually defined features (Ling et al., 2015; dos Santos and Zadrozny, 2014). However, it is still unclear which useful character-level features they have learned. Architectures are compared quantitatively but lack insight into learned patterns. Moreover, Vania and Lopez (2017) showed in the context of language modeling that training a BiLSTM on ground truth morphological features still yields better results than eight other character-based neural network architectures. Hence, this raises the question which patterns neural networks learn and whether these patterns coincide with manually-defined linguistic rules.

While a number of interpretation techniques have been proposed for images (Springenberg et al., 2014; Selvaraju et al., 2017; Shrikumar et al., 2017), these are generally not applicable in the context of NLP where LSTMs are mainly used. Moreover, gradient-based techniques are not trustworthy when strongly saturating activation functions such as *tanh* and *sigmoid* are used (e.g., Li et al. (2016a)). Hence, current interpretations in NLP are limited to visualizing the magnitude of the LSTM hidden states of each word (Linzen et al., 2016; Radford et al., 2017; Strobelt et al., 2018), removing words (Li et al., 2016b; Kádár et al., 2017) or changing words (Linzen et al., 2016) and measuring the impact, or training surrogate tasks (Adi et al., 2017; Chrupała et al., 2017; Belinkov et al., 2017). These techniques only provide limited local interpretations and do not model fine-grained interactions of groups of inputs or intermediate representations. In contrast, Murdoch et al. (2018) recently introduced an LSTM interpretation technique called Contextual Decomposition (CD), providing a solution to the aforementioned issues. We will build upon this interpretation technique and introduce an extension for CNNs, making it possible to compare different neural network architectures within a single interpretation framework.

3 Method

For visualizing the contributions of character sets, we use the recently introduced Contextual Decom-

[2] https://github.com/FredericGodin/ContextualDecomposition-NLP

position (CD) framework, as originally developed for LSTMs (Murdoch et al., 2018), and extend it to CNNs. First, we introduce the concept of CD, followed by the extension for CNNs. For details on CD for LSTMs, we refer the reader to the aforementioned paper. Finally, we explain how the CD of the final classification layer is done.

3.1 Contextual decomposition

The idea behind CD is that, in the context of character-level decomposition, we can decompose the output value of the network for a certain class into two distinct groups of contributions: (1) contributions originating from a specific character or set of characters within a word and (2) contributions originating from all the other characters within the same word.

More generally, we can decompose every output value z of every neural network component into a relevant contribution β and an irrelevant contribution γ:

$$z = \beta + \gamma \tag{1}$$

3.2 Decomposing CNN layers

A CNN typically consist of three components: the convolution itself, an activation function and an optional max-pooling operation. We will discuss each component in the next paragraphs.

Decomposing the convolution Given a sequence of character embeddings $\boldsymbol{x}_1, ..., \boldsymbol{x}_T \in \mathbb{R}^{d_1}$ of length T, we can calculate the convolution of size n of a single filter over the sequence $\boldsymbol{x}_{1:T}$ by applying the following equation to each n-length subsequence $\{\boldsymbol{x}_{t+i}, i = 0, .., n-1\}$, denoted as $\boldsymbol{x}_{t:t+n-1}$:

$$z_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} + b, \tag{2}$$

with $z_t \in \mathbb{R}$ and where $W \in \mathbb{R}^{d_1 \times n}$ and $b \in \mathbb{R}$ are the weight matrix and bias of the convolutional filter. W_i denotes the i-th column of the weight matrix W.

When we want to calculate the contribution of a subset of characters, where $\boldsymbol{S}$ is the set of corresponding character position indexes and $\boldsymbol{S} \subseteq \{1, ..., T\}$, we should decompose the output of the filter z_t into three parts:

$$z_t = \beta_t + \gamma_t + b. \tag{3}$$

That is, the relevant contribution β_t originating from the selected subset of characters with indexes $\boldsymbol{S}$, the irrelevant contribution γ_t originating from the remaining characters in the sequence, and a bias which is deemed neutral (Murdoch et al., 2018).

This can be achieved by decomposing the convolution itself as follows:

$$\beta_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} \qquad (t+i) \in \boldsymbol{S}, \tag{4}$$

$$\gamma_t = \sum_{i=0}^{n-1} W_i \cdot \boldsymbol{x}_{t+i} \qquad (t+i) \notin \boldsymbol{S}, \tag{5}$$

Linearizing the activation function After applying a linear transformation to the input, a non-linearity is typically applied. In CNNs, the ReLU activation function is often used.

In Murdoch et al. (2018), a linearization method for the non-linear activation function f is proposed, based on the differences of partial sums of all N components y_i involved in the pre-activation sum z_t. In other words, we want to split $f_{ReLU}(z_t) = f_{ReLU}(\sum_{i=1}^{N} y_i)$ into a sum of individual linearized contributions $L_{f_{ReLU}}(y_i)$, namely $f_{ReLU}(\sum_{i=1}^{N} y_i) = \sum_{i=1}^{N} L_{f_{ReLU}}(y_i)$. To that end, we compute $L_{f_{ReLU}}(y_k)$, the linearized contribution of y_k as the average difference of partial sums over all possible permutations $\pi_1, ..., \pi_{M_N}$ of all N components y_i involved:

$$L_f(y_k) = \frac{1}{M_N} \sum_{i=1}^{M_N} [f(\sum_{l=1}^{\pi_i^{-1}(k)} y_{\pi_i(l)}) - f(\sum_{l=1}^{\pi_i^{-1}(k)-1} y_{\pi_i(l)})] \tag{6}$$

Consequently, we can decompose the output c_t after the activation function as follows:

$$\begin{aligned} c_t &= f_{ReLU}(z_t) && (7)\\ &= f_{ReLU}(\beta_{z,t} + \gamma_{z,t} + b) && (8)\\ &= L_{ReLU}(\beta_{z,t}) \\ &\quad + [L_{ReLU}(\gamma_{z,t}) + L_{ReLU}(b)] && (9)\\ &= \beta_{c,t} + \gamma_{c,t} && (10) \end{aligned}$$

Following Murdoch et al. (2018), $\beta_{c,t}$ contains the contributions that can be directly attributed to the specific set of input indexes $\boldsymbol{S}$. Hence, the bias b is

part of $\gamma_{c,t}$. Note that, while the decomposition in Eq. (10) is exact in terms of the total sum, the individual attribution to relevant ($\beta_{c,t}$) and irrelevant ($\gamma_{c,t}$) is an approximation, due to the linearization.

Max-pooling over time When applying a fixed-size convolution over a variable-length sequence, the output is again of variable size. Hence, a max-pooling operation is executed over the time dimension, resulting in a fixed-size representation that is independent of the sequence length:

$$c = \max_t(c_t). \tag{11}$$

Instead of applying a max operation over the $\beta_{c,t}$ and $\gamma_{c,t}$ contributions separately, we first determine the position t of the highest c_t value and propagate the corresponding $\beta_{c,t}$ and $\gamma_{c,t}$ values.

3.3 Calculating the final contribution scores

The final layer is a classification layer, which is the same for a CNN- or LSTM-based architecture. The probability p_j of predicting class j is defined as follows:

$$p_j = \frac{e^{W_j \cdot \boldsymbol{x} + b_j}}{\sum_{i=1}^{C} e^{W_i \cdot \boldsymbol{x} + b_i}}, \tag{12}$$

in which $W \in \mathbb{R}^{d_2 \times C}$ is a weight matrix and W_i the i-th column, $\boldsymbol{x} \in \mathbb{R}^{d_2}$ the input, $\boldsymbol{b} \in \mathbb{R}^{d_2}$ the bias vector and b_i the i-th element, d_2 the input vector size and C the total number of classes.

The input $\boldsymbol{x}$ is either the output $\boldsymbol{c}$ of a CNN or $\boldsymbol{h}$ of a LSTM. Consequently, we can decompose $\boldsymbol{x}$ into $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ contributions. In practice, we only consider the preactivation and decompose it as follows:

$$W_j \cdot \boldsymbol{x} + b_j = W_j \cdot \boldsymbol{\beta} + W_j \cdot \boldsymbol{\gamma} + b_j. \tag{13}$$

Finally, the contribution of a set of characters with indexes $\boldsymbol{S}$ to the final score of class j is equal to $W_j \cdot \boldsymbol{\beta}$. The latter score is used throughout the paper for visualizing contributions of sets of characters.

4 Experimental Setup

We execute experiments on morphological tagging in three different languages: Finnish, Spanish and Swedish. We describe the dataset in Section 4.1, whereas model and training details can be found in Section 4.2.

4.1 Dataset

For our experiments, we use the Universal Dependencies 1.4 (UD) dataset (Nivre et al., 2016), which contains morphological features for a large number of sentences. Additionally, we acquired manually-annotated character-level morphological segmentations and labels for a subset of the test set for three morphological different languages: Finnish, Spanish and Swedish. [3]

For each language, Silfverberg and Hulden (2017) selected the first non-unique 300 words from the UD test set and manually segmented each word according to the associated lemma and morphological features in the dataset. Whenever possible, they assigned each feature to a specific subset of characters. For example, the Spanish word "económicas" is segmented as follows:

- económic : lemma=económico
- a : gender=feminine
- s : number=plural

For our experiments, we are only interested in word/feature pairs for which a feature can be assigned to a specific subset of characters. Hence, we filter the test set on those specific word/feature pairs. In the above example, we have two word/feature pairs. This resulted in 278, 340 and 137 word/feature pairs for Finnish, Spanish and Swedish, respectively. Using the same procedure, we selected relevant feature classes, resulting in 12, 6 and 9 feature classes for Finnish, Spanish and Swedish, respectively. For each class, when a feature was not available, we introduced an additional *Not Applicable* (NA) label. A complete overview of the feature classes can be found in Appendix B.

We always train and validate on the full UD dataset for which we have filtered out all duplicate words. After that, we perform our analysis on either the UD test set or the annotated subset of manually segmented and annotated words. An overview can be found in Table 1.

4.2 Model

We experiment with both a CNN and BiLSTM architecture for character-level modeling of words.

At the input, we split every word into characters and add a start-of-word (^) and an end-of-word

[3] Available online: http://github.com/mpsilfve/ud-segmenter/commit/5959214d494cbc13e53e1b26650813ff950d2ee3

Table 1: Overview of the training, validation and test set used.

	Finnish	Spanish	Swedish
Train words	53547	62556	16295
Valid words	2317	4984	1731
Test words	2246	956	3538
Annotated Test pairs	278	340	137

($) character. With every character, we associate a character embedding of size 50.

Our CNN architecture is inspired by Kim et al. (2016) and consists of a set of filters of varying width, followed by a ReLU activation function and a max-over-time pooling operation. We adopt their small-CNN parameter choices and have 25, 50, 75, 100, 125 and 150 convolutional filters of size 1, 2, 3, 4, 5 and 6, respectively. We do not add an additional highway layer.

For the character-level BiLSTM architecture, we follow the variant used in Plank et al. (2016). That is, we simply run a BiLSTM over all the characters and concatenate the final forward and backward hidden state. To obtain a similar number of parameters as the CNN model, we set the hidden state size to 100 units for each LSTM.

Finally, the word-level representation generated by either the CNN or BiLSTM architecture is classified by a multinomial logistic regression layer. Each morphological class type has a different layer. We do not take into account context to rule out any influence originating from somewhere other than the characters of the word itself.

Training details For morphological tagging, we train a single model for all classes at once. We minimize the joint loss by summing the cross-entropy losses of each class. We orthogonally initialize all weight matrices, except for the embeddings, which are uniformly initialized ([-0.01;0.01]). All models are trained using Adam (Kingma and Ba, 2015) with minibatches of size 20 and learning rate 0.001. No specific regularization is used. We select our final model based on early stopping on the validation set.

5 Experiments

First, we verify that the CD algorithm works correctly by executing a controlled experiment with a synthetic token. Next, we quantitatively and qualitatively evaluate on the full test set.

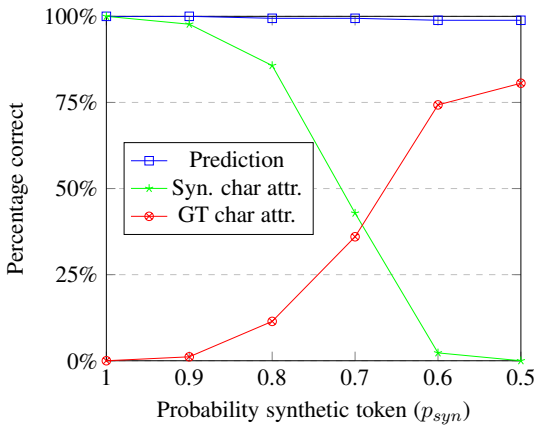


Figure 2: Comparison of the contribution of the synthetic character versus the Ground Truth (GT) character for the class $t = 1$. The prediction curve denotes the classification accuracy for class $t = 1$, and consequently, the prediction curve denotes the upper bound for the attributions.

5.1 Validation of contextual decomposition for convolutional neural networks

To verify that the contextual decomposition of CNNs works correctly, we devise an experiment in which we add a synthetic token to a word of a certain class, testing whether this token gets a high attribution score with respect to that specific class.

Given a word w and a corresponding binary label t, we add a synthetic character c to the beginning of word w with probability p_{syn} if that word belongs to the class $t = 1$ and with probability $1 - p_{syn}$ if that word belongs to the class $t = 0$. Consequently, if $p_{syn} = 1$, the model should predict the label with a 100% accuracy, thus attributing this to the synthetic character c. When $p_{syn} = 0.5$, the synthetic character does not provide any additional information about the label t, and c should thus have a small contribution.

Experimental setup We train a CNN model on the Spanish dataset and only use words having the morphological label *number*. This label has two classes *plur* and *sing*, and assign those classes to the binary labels zero and one, respectively. Furthermore, we add a synthetic character to each word with probability p_{syn}, varying p_{syn} from 1 to 0.5 with steps of 0.1. We selected 112 unique word/feature pairs from our test set with label *sing*

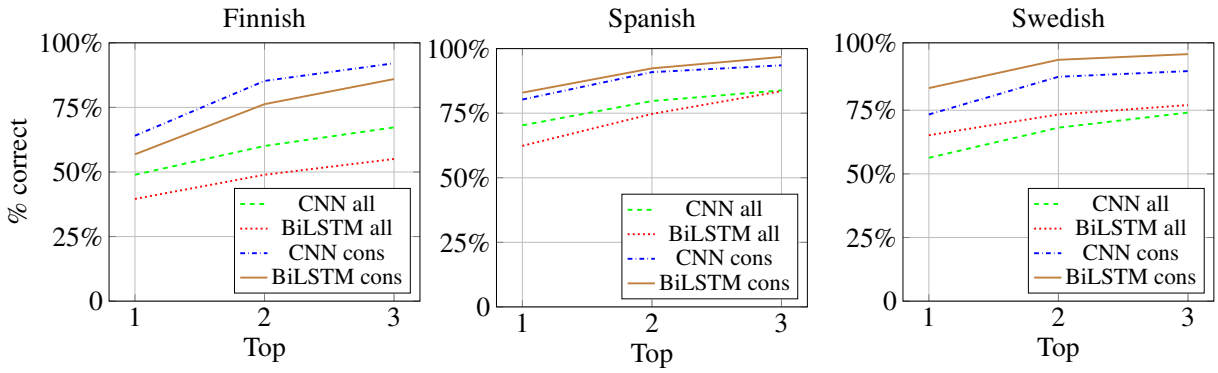


Figure 3: Evaluation of the attributions of CNN and BiLSTM models on the three different languages.

or *plur*. While plurality is marked by the suffix *s*, a variety of suffixes are used for the singular form. Therefore, we focus on the latter class ($t = 1$). The corresponding suffix is called the Ground Truth (GT) character.

To measure the impact of p_{syn}, we add a synthetic character to each word of the class $t = 1$ and calculate the contribution of each character by using the CD algorithm. We run the experiment five times with a different random seed and report the average correct attribution. The attribution is correct if the contribution of the synthetic/GT character is the highest contribution of all character contributions.

Results The results of our evaluation are depicted in Figure 2. When $p_{syn} = 1$, all words of the class $t = 1$ contain the synthetic character, and consequently, the accuracy for predicting $t = 1$ is indeed 100%. Moreover, the correct prediction is effectively attributed to the synthetic character ('syn. char attr.' in Figure 2 at 100%), with the GT character being deemed irrelevant. When the synthetic character probability p_{syn} is lowered, the synthetic character is less trustworthy and the GT character becomes more important (increasing 'GT char attr.' in Figure 2). Finally, when $p_{syn} = 0.5$, the synthetic character is equally plausible in both classes. Hence, the contribution of the synthetic character becomes irrelevant and the model attributes the prediction to other characters.

Consequently, we can conclude that whenever there is a clear character-level pattern, the model learns the pattern and the CD algorithm is able to accurately attribute it to the correct character.

Table 2: Average accuracy of all models trained on Finnish, Spanish and Swedish for the task of morphological feature prediction for all unique words in the full UD test set.

	Finnish	**Spanish**	**Swedish**
Maj. Vote	82.20%	72.39%	69.79%
CNN	94.81%	88.93%	**90.09%**
BiLSTM	**95.13%**	**89.33%**	89.45%

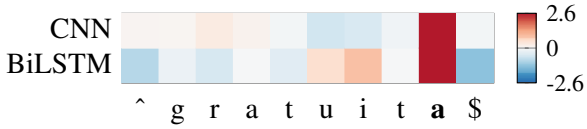
5.2 Evaluation of character-level attribution

In this section, we measure and analyze (1) which characters contribute most to the final prediction of a certain label and (2) whether those contributions coincide with our linguistic knowledge about a language. To that end, we train a model to predict morphological features, given a particular word. The model does not have prior word segmentation information and thus needs to discover useful character patterns by itself. After training, we calculate the attribution scores of each character pattern within a word with respect to the correct feature class using CD, and evaluate whether this coincides with the ground truth attribution.

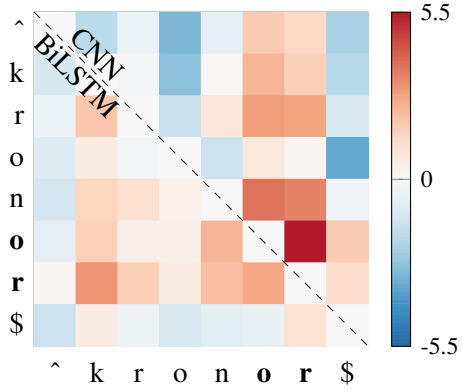
Model We train CNN and BiLSTM models on Finnish, Spanish and Swedish. The average accuracies on the full test set are reported in Table 2. The results for the individual classes can be found in Appendix C. As a reference for the trained models' ability to predict morphological feature classes, we provide a naive baseline, constructed from the majority vote for each feature type.

Overall, our neural models yield substantially higher average accuracies than the baseline and

(a) Example of Finnish. Word (verb): *olivat (were)*, target class: *Tense=Past*



(b) Example of Spanish. Word (adjective): *gratuita (free)*, target: *Gender=Fem*.



(c) Example of Swedish. Word (noun): *kronor (Swedish valuta as in dollars)*, target: *Number=Plur*.

Figure 4: Character-level contributions for predicting a particular class. Positive contributions are highlighted in red and negative contributions in blue. The ground truth character sequence is highlighted in bold.

perform very similar. Consequently, both the CNN and LSTM models learned useful character patterns for predicting the correct morphological feature classes. Hence, this raises the question whether these patterns coincide with our linguistic knowledge.

Evaluation For each annotated word/feature pair, we measure if the ground truth character sequence corresponds to the set or sequence of characters with the same length within the considered word that has the highest contribution for predicting the correct label for that word.

In the first setup, we only compare with character sequences having a consecutive set of characters (denoted *cons*). In the second setup, we compare with any set of characters (denoted *all*). We rank the contributions of each character set and report top one, two, and three scores. Because start-of-word and end-of-word characters are not annotated in the dataset, we do not consider them part of the candidate character sets.

Results The aggregated results for all classes and character sequence lengths are shown in Figure 3. In general, we observe that for almost all models and setups, the contextual decomposition attribution coincides with the manually-defined segmentations for at least half of the word/feature pairs. When we only consider the top two consecutive sequences (marked as *cons*), accuracies range from 76% up to 93% for all three languages. For Spanish and Swedish, the top two accuracies for character sets (marked as *all*) are still above 67%, despite the large space of possible character sets, whereas all ground truth patterns are consecutive sequences. While the accuracy for Finnish is lower, the top two accuracy is still above 50%.

Examples for Finnish, Spanish and Swedish are shown in Figure 4. For Finnish, the character with the highest contribution *i* coincides with the ground truth character for the CNN model. This is not the case for the BiLSTM model which focuses on the character *v*, even though the correct label is predicted. For Spanish, both models strongly focus on the ground truth character *a* for predicting the feminine gender. For Swedish, the ground truth character sequence is the suffix *or* which denotes plurality. Given that *or* consists of two characters, all contributions of character sets of two characters are visualized. As can be seen, the most important set of two characters is {o,r} for the CNN and {k,r} for the BiLSTM model. However, {o,r} is the second most important character set for the BiLSTM model. Consequently, the BiLSTM model deemed the interaction between a root and suffix character more important than between two suffix characters.

5.3 Analysis of learned patterns

In the previous section, we showed that there is a strong relationship between the manually-defined morphological segmentation and the patterns a neural network learns. However, there is still an accuracy gap between the results obtained using consecutive sequences only and results obtained using all possible character sets. Hence, this leads to the question which patterns the neural network

Table 3: The most frequent character sets used by a model for predicting a specific class. The frequency of occurrence is shown between brackets. An underscore denotes an unknown character.

		One character	Two characters	Three characters	Examples
Finnish Tense=Past	BiL.	i (69%), t (22%), v (4%), a (2%)	ti (13%), t_i (12%), v_t (9%), ui (6%)	tti (8%), iv_t (5%), t__ti (3%), sti (3%)	oli**vat**, näy**tti**kään
	CNN	i (71%), t (8%), s (6%), o (5%)	ui (12%), si (11%), ti (11%), oi (9%)	a__ui (3%), tii (3%), iv__$ (2%), ui__t (2%)	tie**si**, mei**dät**
Spanish Gend=Fem	BiL.	a (69%), i (16%), d (6%), e (4%)	as (23%), a$ (13%), ad (7%), ia (5%)	ia$ (4%), ad$ (3%), da$ (3%), ca$ (2%)	toleranc**ia**, ciud**ad**
	CNN	a (77%), ó (14%), n (4%), d (3%)	a$ (34%), as (20%), da (8%), ió (7%)	dad (5%), da$ (4%), a_ió (4%), sió (2%)	firm**as**, preci**sión**
Swedish Numb=Plur	BiL.	n (25%), r (19%), a (14%), g (7%)	na (13%), a__r (4%), or (3%), n__r (3%)	iga (5%), rna (3%), ner (1%), der (1%)	kron**or**, perio**der**
	CNN	n (21%), a (18%), r (15%), d (5%)	rn (8%), na (5%), or (4%), er (3%)	rna (7%), arn (3%), iga (2%), n_ar (2%)	krafte**rna**, sak**er**

focuses on, other than the manually defined patterns we evaluated before. To that end, for each of the three languages, we selected a morphological class of interest and evaluated for all words in the full UD test set that were assigned to that class what the most important character set of length one, two and three was. In other words, we evaluated for each word for which the class was correctly predicted, which character set had the highest positive contribution towards predicting that class. The results can be found in Table 3.

Finnish In Finnish, adding the suffix *i* to a verb, transforms it in the past tense. Sometimes the character *s* is added, resulting in the suffix *si*. The latter is a frequently used bigram pattern by the CNN but less by the BiLSTM. The BiLSTM combines the suffix *i* with another suffix *vat* which denotes third person plural in the character pattern *iv_t*.

Spanish While there is no single clear-cut rule for the Spanish gender, in general the suffix *a* denotes the feminine gender in adjectives. However, there exist many nouns that are feminine but do not have the suffix *a*. Teschner and Russell (1984) identify *d*, and *ión* as typical endings of feminine nouns, which our models identified too as for example *ad$* or *ió*/*sió*.

Swedish In Swedish, there exist four suffixes for creating a plural form: *or*, *ar*, *(e)r* and *n*. Both models identified the suffix *or*. However, similar to Finnish, multiple suffixes are merged. In Swedish, the suffix *na* only occurs together with one of the first three plural suffixes. Hence, both models correctly identified this pattern as an important pattern for predicting the class number=plural, rather than the linguistically-defined pattern.

5.4 Interactions of learned patterns

In the previous section, the pattern *a$* showed to be the most important pattern in 34% of the correctly-predicted feminine Spanish words in our dataset. However, there exist many words that end with the character *a* that are not feminine. For example the third person singular form of the verb *gustar* is *gusta*. Hence, this raises the question if the model will classify *gusta* wrongly as feminine or correctly as NA. As an illustration of the applicability of CD for morphological analysis, we will study this case in more detail.

From the full UD test set, we selected all words that end with the character *a* and that do not belong to the class gender=feminine. Using the Spanish CNN model, we predicted the gender class for each word and divided the words into two groups: predicted as feminine and predicted as not-feminine (_NA_ or masculine). The resulted in 44 and 199 words. Next, for each word in both groups we calculated the most positively and negatively contributing character set out of all possible character sets of any length within the considered word, using the CD algorithm. We compared the contribution scores in

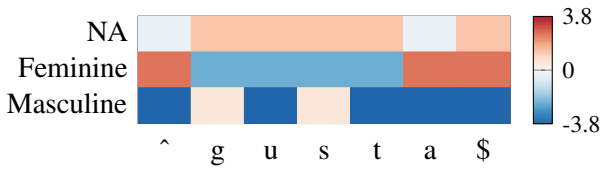


Figure 5: Visualization of the most positively and negatively contributing character set for each class of the morphological feature class gender for the Spanish verb *gusta* (likes).

both groups using a Kruskal-Wallis significance test.[4] While no significant ($p < 0.05$) difference could be found between the positive contributions of both groups (p=1.000), a borderline significant difference could be found between the negative contributions of words predicted as feminine and words predicted as not-feminine (p=0.070).

Consequently, the CNN model's classification decision is based on finding enough negative evidence to counteract the positive evidence found in the pattern *a$*, which CD was able to uncover.

A visualization of this interaction is shown in Figure 5 for the word *gusta*. While the positive evidence is the strongest for the class feminine, the model identifies the verb stem *gust* as negative evidence which ultimately leads to the correct final prediction NA.

6 Conclusion

While neural network-based models are part of many NLP systems, little is understood on how they handle the input data. We investigated how specific character sequences at the input of a neural network model contribute to word-level tagging decisions at the output, and if those contributions follow linguistically interpretable rules.

First, we presented an analysis and visualization technique to decompose the output of CNN models into separate input contributions, based on the principles outlined by Murdoch et al. (2018) for LSTMs. This allowed us then to quantitatively and qualitatively compare the character-level patterns the CNNs and BiLSTMs learned for the task of morphological tagging. We showed that these patterns generally coincide with the morphological segments as defined by linguists for three morphologically different languages, but that sometimes other linguistically plausible patterns are learned.

[4]The full statistical analysis is provided in Appendix D.

Finally, we showed that our CD algorithm for CNNs is able to explain why the model made a wrong or correct prediction.

By visualizing the contributions of each input unit or combinations thereof, we believe that much can be learned on how a neural network handles the input data, why it makes certain decisions, or even for debugging neural network models.

Acknowledgments

The authors would like to thank the anonymous reviewers and members of IDLab for their valuable feedback. FG would like to thank Kim Bettens for helping out with the statistical analysis.

The research activities as described in this paper were funded by Ghent University, imec, Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *International Conference on Representation Learning (ICLR)*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the Association for Computational Linguistics (ACL)*.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Eric Brill. 1994. A report of recent progress in transformation-based error-driven learning. In *Proceedings of the Workshop on Human Language Technology (HLT)*.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. 2017. Representations of language in a model of visually grounded speech signal. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Representation Learning (ICLR)*.

Sheldon Klein and Robert F. Simmons. 1963. A computational approach to grammatical coding of english words. *Journal of the Association for Computing Machinery*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in nlp. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of Association for Computational Linguistics (ACL)*.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. *International Conference on Learning Representations (ICLR)*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *International Confernce of Machine Learning (ICML)*.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Miikka Silfverberg and Mans Hulden. 2017. Automatic morpheme segmentation and labeling in universal dependencies resources. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2014. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806.

Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M. Rush. 2018. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*.

Richard Teschner and William Russell. 1984. The gender patterns of spanish nouns: an inverse dictionary-based analysis. *Hispanic Linguistics 1*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the Association for Computational Linguistics (ACL)*.

Appendix

A Notes on Dataset Selection

In the paper (Silfverberg and Hulden, 2017) that introduced the morphological segmentations for a subset of the Universal Dependencies dataset 1.4, sentences from the training dataset were selected for constructing the test set. Although the paper mentions that all test set sentences for Finnish, Spanish and Swedish were selected from the Universal Dependencies test sets, this is not the case for Finnish. The first 515 lines of *fi-ud-train.conllu* were used for selecting 300 test set words. Given that a new sentence starts at line 521, we removed the first 520 lines from the Finnish training set. This is only 0.3% of the full training set, and consequently, this will have a negligible impact on our conclusions. Note that, for Spanish and Swedish, the segmented words were indeed selected from their respective test sets. The above observations were also confirmed by the first author of the original paper.

B Overview Morphological Classes Used

In Table 4, Table 5 and Table 6, all class types and corresponding feature class values for Finnish, Spanish and Swedish are listed. During training, each class type has a specific multinomial regression layer which predicts a single value for that class type. However, all class types are jointly trained.

C Individual Results Morphological Tagging

In Table 7, Table 8 and Table 9, the individual results for each morphological feature class for Finnish, Spanish and Swedish can be found.

D Full statistical analysis for "Interactions of learned patterns"

From the full UD test set, we selected all words that end with the character *a* and evaluated the morphological feature type gender for all of them. We selected three groups:

- Words that have the label gender=feminine and are classified as gender=feminine, called wf_pf. This group contains 219 words.
- Words that do not have the label gender=feminine are classified as gen-

Table 4: Overview of classes used for Finnish.

Class type	Values
Number	_NA_ Sing Plur
PartForm	_NA_ Past Pres Agt Neg
Case	_NA_ Ela Ine Ins Par Ill Com Nom All Acc Ade Gen Ess Abl Tra Abe
Person	_NA_ 1 2 3
Derivation	_NA_ Ja Minen Sti Vs Tar Llinen Inen U Ttaa Ttain Lainen Ton
Person[psor]	_NA_ 1 2 3
VerbForm	_NA_ Inf Part Fin
Mood	_NA_ Imp Cnd Pot Ind
Tense	_NA_ Past Pres
Clitic	_NA_ Pa,S Han Ko Pa Han,Pa Han,Ko Ko,S S Kin Kaan Ka
Degree	_NA_ Pos Cmp Sup
Voice	_NA_ Pass Act

Table 5: Overview of classes used for Spanish.

Class type	Values
Person	_NA_ 1 2 3
Mood	_NA_ Imp Ind Sub Cnd
Tense	_NA_ Fut Imp Pres Past
Gender	_NA_ Fem Masc
VerbForm	_NA_ Inf Ger Part Fin
Number	_NA_ Sing Plur

Table 6: Overview of classes used for Swedish.

Class type	Values
Gender	_NA_ Neut Masc Fem Com
Degree	_NA_ Sup Cmp Pos
Number	_NA_ Sing Plur
Case	_NA_ Gen Nom Acc
Poss	_NA_ Yes
Voice	_NA_ Act Pass
Tense	_NA_ Pres Past
Definite	_NA_ Ind Def
VerbForm	_NA_ Sup Part Inf Fin Stem

der=feminine, wnf_pf. This group contains 44 words.

- Words that do not have the label gender=feminine are classified as either gender=NA or gender=masc, i.e. not-feminine, called wnf_pnf. This group contains 199 words.

For each group, we calculated the contributions of all possible character sets of different length within each word and selected the highest contribution score and the lowest contribution score for each word. In other words, we look for the sets of characters that generate the strongest positive and negative contributions for predicting the class gender=feminine. These two contribution scores are the determining factors for certain classification decisions.

D.1 Maximum contribution scores

Based on a Kruskal-Wallis test, a statistically significant difference was found between the three groups, $H(2) = 50,600$, $p < 0.001$. Pairwise comparisons with adjusted p-values showed no significant difference in positive contributions scores between the groups wnf_pf and wnf_pnf ($p = 1.000$). Hence, non-feminine words have similar positive contribution scores, independent of the classification result. Furthermore, significant differences were found between the positive contribution scores of the groups wf_pf and wnf_pf ($p < 0.001$) and the groups wf_pf and wnf_pnf ($p < 0.001$), indicating a difference between the positive contributions of feminine words and non-feminine words.

D.2 Minimum contribution scores

Based on a Kruskal-Wallis test, an overall statistically significant difference was found between the three groups, $H(2) = 36.710$, $p < 0.001$. Pairwise comparisons with adjusted p-values showed that there was no significant difference between the groups wf_pf and wnf_pf ($p = 0.585$), showing that the negative contribution scores of words classified as feminine are similar despite that the fact that the words from wnf_pf are not feminine. A strong significant difference was found between the groups wf_pf and wnf_pnf ($p < 0.001$) and a borderline significant difference between the groups wnf_pnf and wnf_pf ($p < 0.070$). Consequently, there is a clear difference between the negative contributions of non-feminine words that are classified as not-feminine and words that are classified as feminine. Moreover, words that are wrongly classified as feminine have similar negative contribution scores as words classified correctly as feminine.

Table 7: Per class accuracy on the Finnish test set.

	Number	Partform	Case	Person	Derivation	Person[psor]
Maj. Vote	64.42%	94.33%	28.49%	89.17%	98.43%	96.05%
CNN	89.40%	96.97%	87.00%	95.81%	99.07%	98.49%
BiLSTM	89.67%	97.86%	87.89%	95.77%	99.11%	99.29%

	Verbform	Mood	Tense	Clitic	Degree	Voice
Maj. Vote	77.54%	87.77%	89.09%	98.49%	84.16%	78.49%
CNN	93.05%	95.90%	96.17%	99.51%	92.70%	93.59%
BiLSTM	93.19%	96.13%	95.99%	99.51%	92.97%	94.12%

Table 8: Per class accuracy on the Spanish test set.

	Person	Mood	Tense	Gender	Verbform	Number
Maj. Vote	85.26%	87.62%	85.99%	54.40%	75.49%	45.56%
CNN	91.84%	93.51%	91.11%	84.62%	88.08%	84.41%
BiLSTM	91.95%	93.41%	90.90%	84.31%	89.02%	86.40%

Table 9: Per class accuracy on the Swedish test set.

	Gender	Degree	Number	Case	Poss	Voice	Tense	Definite	Verbform
Maj. Vote	46.64%	84.57%	42.21%	62.73%	99.67%	83.99%	87.57%	41.54%	79.19%
CNN	86.18%	93.78%	79.45%	87.79%	99.94%	94.60%	94.29%	83.83%	90.98%
BiLSTM	83.97%	94.26%	78.72%	86.04%	99.97%	93.75%	93.84%	83.86%	90.64%