

Explaining Character-Aware Neural Networks for Word- Level Prediction

Do They Discover Linguistic Rules?

Mohammad Samavatian

CSE-5559

What is Paper About?

What?

- Character-level patterns in neural network based natural language processing algorithms

Why?

- Qualitative analysis is missing
- Investigate which character-level patterns neural networks learn

How?

- Extend the contextual decomposition technique to convolutional neural networks
- Visualizing the contributions of each character
- Allows us to compare CNN and BiLSTM

Motivation

- Character-level features are an essential for NLP
 - Part-of-speech (PoS) tagging
 - Morphological tagging
- Character-level features
 - Rule-based taggers
 - Easily trace back why the tagger made a certain decision
- Neural network-based generation of part-of-speech and morphological taggers
 - Not traceable

Example: Rule-based Tagger for PoS Tagging

- Brill (1994)'s transformation-based error-driven tagger

Template

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



Rule

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

Neural Network Based Taggers

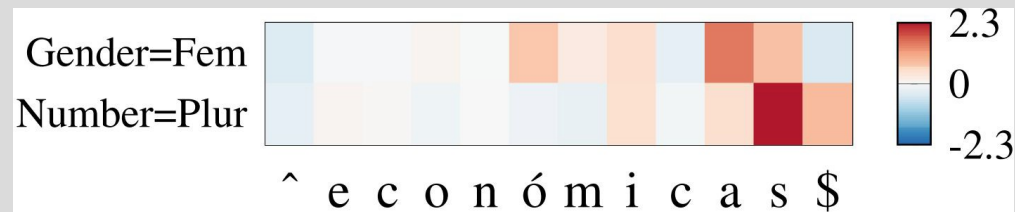
- Words are split into individual characters
- Aggregated using a BiLSTM or CNN

Shortcoming

- Unknown which character-level patterns learned?
- Coincide with our linguistic knowledge?
- Lack a qualitative analysis over different networks

Proposed Method in Nutshell

- Present contextual decomposition (CD) for CNNs
 - Extends CD for LSTMs (Murdoch et al. 2018)
 - White box approach to interpretability
- Trace back morphological tagging decisions to the character-level
 - Which characters are important?
 - Same patterns as linguistically known?
 - Difference CNN and BiLSTM?



NLP Neural Network Visualization


- Interpretation techniques have been proposed for images
- Not applicable because
 - NLP uses LSTMs mainly
 - Gradient-based techniques are not trustworthy when strongly saturating activation functions such as tanh and sigmoid
 - Only visualizing the LSTM hidden states
 - Provide limited local interpretations
 - Not model fine-grained interactions of groups of inputs

Solution:

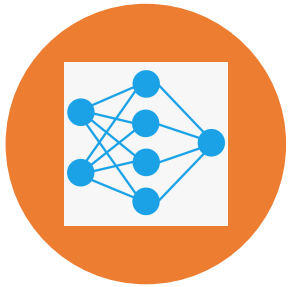
- Contextual Decomposition (CD) for LSTM

Contextual Decomposition

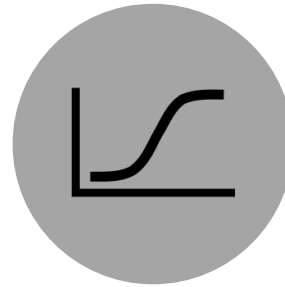
- Decompose the output value of the network for a certain class
 - Relevant: contributions originating from a specific character or set of characters within a word
 - Irrelevant: contributions originating from all the other characters within the same word
- Decompose every output $z = \beta + \gamma$

The diagram shows the equation $z = \beta + \gamma$ from the previous block. Below the term β is the word "relevant" in red text, with a blue arrow pointing from "relevant" up to β . Below the term γ is the word "irrelevant" in black text, with a blue arrow pointing from "irrelevant" up to γ .

Decomposing CNN Layers



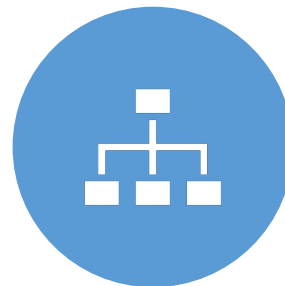
Decomposing the
convolution



Linearizing the activation
function



Max-pooling over time



Classification Layer

Decomposing Convolution

Decompose the output of the filter z_t

$$z_t = \beta_t + \gamma_t + b$$

$$z_t = \sum_{i=0}^{n-1} w_i \cdot x_{t+i} + b$$

$$\beta_t = \sum_{i=0}^{n-1} w_i \cdot x_{t+i} \quad (t+i) \in S$$

relevant

irrelevant

$$\gamma_t = \sum_{i=0}^{n-1} w_i \cdot x_{t+i} \quad (t+i) \notin S$$

^ e c o n o m i c **a** s \$

Indexes: 8 **9** 10 11

S: selected subset of characters

Linearizing Activation Function

- Linearize activation function to be able to split output
- CNNs uses ReLU
- $\beta_{c,t}$ and $\gamma_{c,t}$ is approximation due to the linearization

$$c_t = f_{ReLU}(z_t)$$

$$c_t = f_{ReLU}(\beta_{z,t} + \gamma_{z,t} + b)$$

$$c_t = L_{ReLU}(\beta_{z,t}) + [L_{ReLU}(\gamma_{z,t}) + L_{ReLU}(b)]$$

$$c_t = \beta_{c,t} + \gamma_{c,t}$$


relevant

irrelevant

Linearization formula:

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

Max-Pooling Over Time

- Max-pooling operation is executed over the time dimension
 - Resulting in a fixed-size representation that is independent of the sequence length
- Not applying a max operation over the $\beta_{c,t}$ and $\gamma_{c,t}$ contributions separately

Instead

- Determine the position t of the highest c_t value
- Propagate the corresponding $\beta_{c,t}$ and $\gamma_{c,t}$

$$\beta_c + \gamma_c = \max_t (\beta_{c,t} + \gamma_{c,t})$$

Contextual Decomposition classification Layer

- p_j of predicting class j :

$$p_j = \frac{e^{w_j \cdot x + b_j}}{\sum_{i=1}^C e^{w_j \cdot x + b_j}}$$

- The input x is either:
 - CNN output or
 - LSTM h_t
- Decompose x into β and γ contributions
- Only consider the pre-activation and decompose it:

$$w_j \cdot x + b_j = W_j \cdot \beta + W_j \cdot \gamma + b_j$$

Relevant contribution to class j

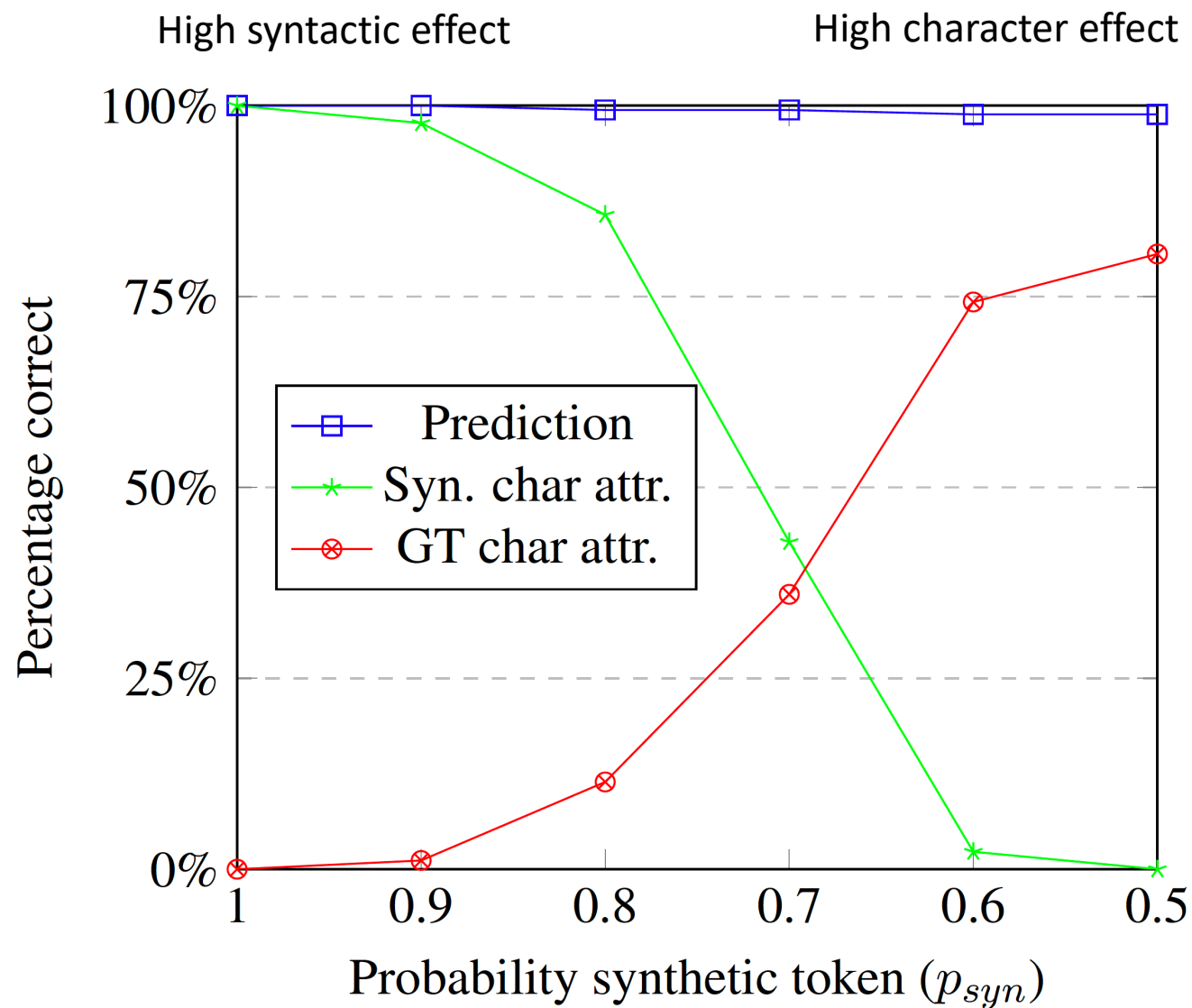
- Contribution of a set of characters with indexes S to the final score of class j .

Validation of Contextual Decomposition

- Given a word w and a corresponding binary label t
- Add a synthetic character c to the beginning of word w
 - Probability p_{syn} if class $t = 1$
 - Probability $1 - p_{syn}$ if class $t = 0$
- $p_{syn} = 1$
 - Model should predict the label with a 100% accuracy
 - C has high contribution
- $p_{syn} = 0.5$:
 - Synthetic character does not provide any additional information about the label t
 - c has a small contribution

Validation Result

- Singular/Plural class (0/1)
- varying p_{syn} from 1 to 0.5
- Measure the impact of p_{syn}
 - Add a synthetic character
 - Calculate the contribution of each character
 - The attribution is correct if the contribution of the synthetic/GT character is the highest contribution of all character contributions

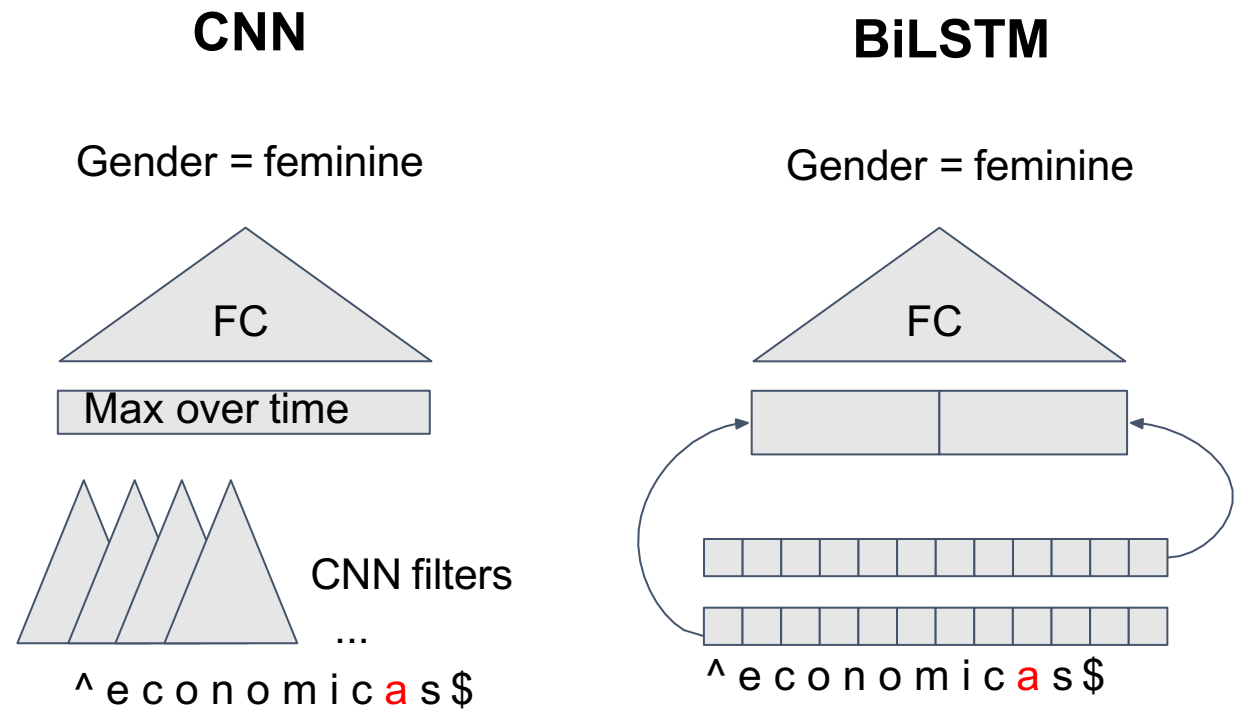


Experiments Setup

- Morphological tagging (gender, tense, singular/plural,...)
 - Finnish, Spanish and Swedish
- Universal Dependencies 1.4 (UD) dataset
 - Morphological features for sentences
 - +manually-annotated character-level morphological segmentations and labels for test set (300 words)
 - **Economicas** → lemma=económico
 - Economicas → gender=feminine
 - Economicas → number=plural

Model Architecture: CNN vs BiLSTM

- Split every word into characters:
 - Start(^), End(\$)
- Use CNN and BiLSTM
- Multinomial Logistic Regression
 - Classify word-level representation generated by either the CNN or BiLSTM
- Train a single model for all classes at once

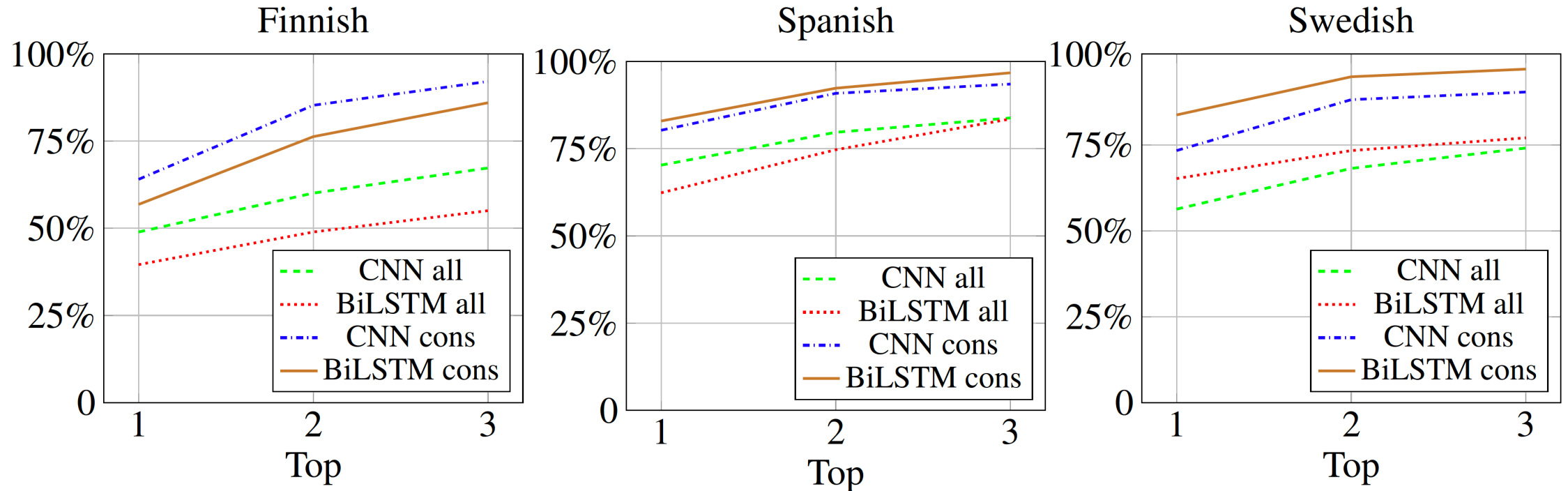


	Finnish	Spanish	Swedish
Maj. Vote	82.20%	72.39%	69.79%
CNN	94.81%	88.93%	90.09%
BiLSTM	95.13%	89.33%	89.45%

Average accuracy

All = every possible combination of characters

Cons = all consecutive character n-grams



Do the NN Patterns
Follow Manual
Segmentations?

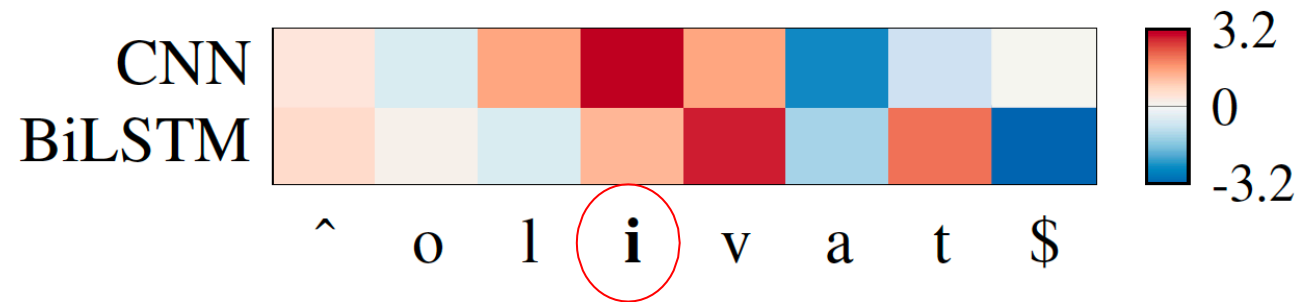
- Which characters contribute most to the final prediction of a certain label
- Whether those contributions coincide with our linguistic knowledge about a language

Contribution Visualization: One Character

Positive contributions: red
Negative contributions: blue

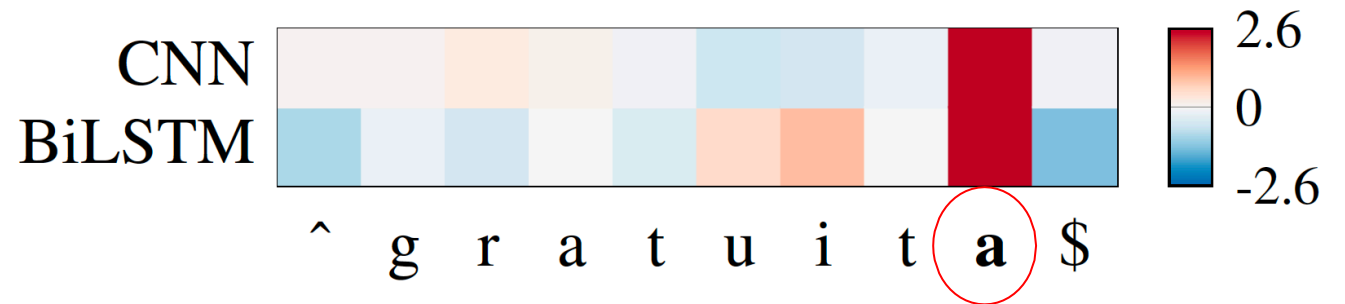
Finnish:

- verb
- olivat (→ were)
- Label: Tense=Past



Spanish:

- adjective
- gratuita (→ free)
- Label: Gender=Feminine

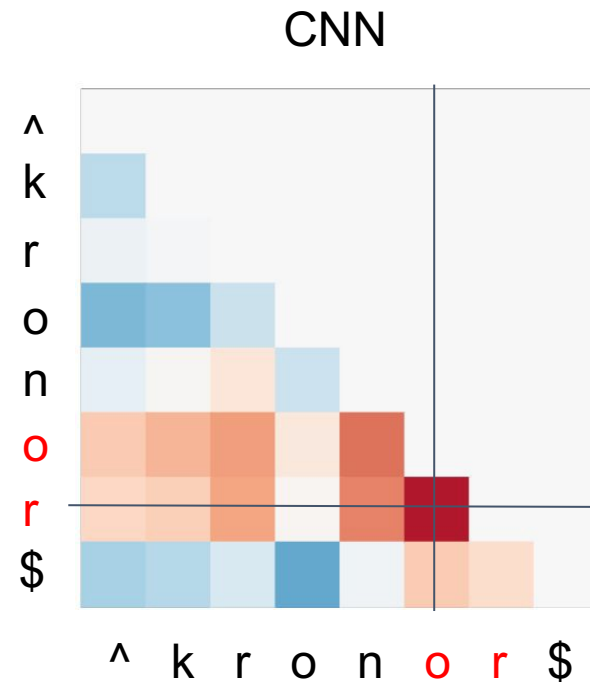


Contribution Visualization: Two Characters

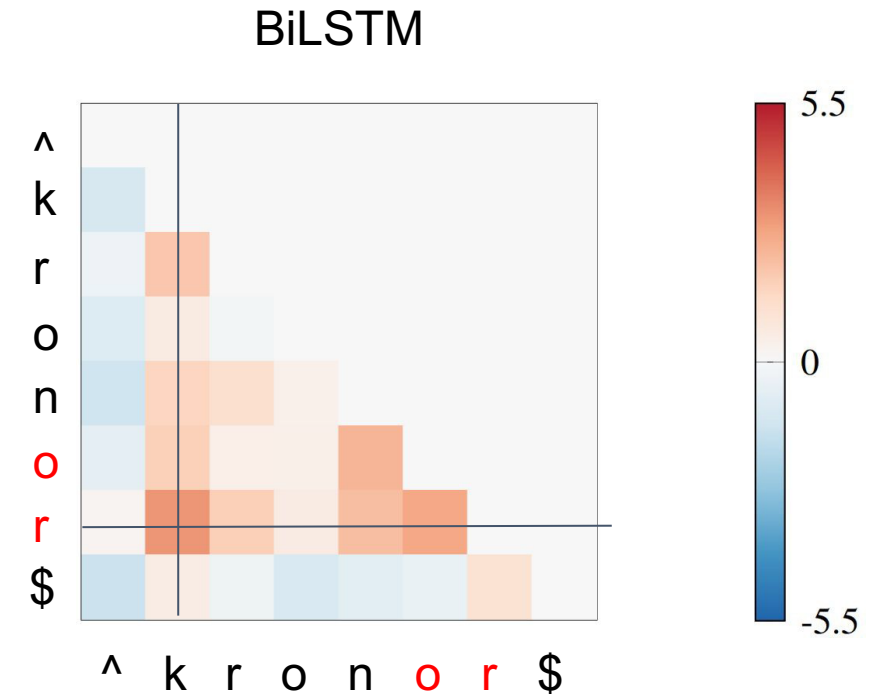
Swedish:

- Noun
- kronor (Swedish valuta as in dollars)
- target: Number=Plural

BiLSTM deemed the interaction between a root and suffix character more important than between two suffix characters



{o,r} for the CNN



{k,r} for the BiLSTM
{o,r} is the second

		One character	Two characters	Three characters	Examples
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%)	tolerancia, ciudad
	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%)	firmas, precisión

Most Important Patterns: Spanish

- Linguistic rules for feminine gender:
 - Feminine adjectives often end with “a”
 - Nouns ending with “dad” or “ión” are often feminine
- Found pattern:
 - “a” is a very important pattern
 - “dad” and “sió” are import trigrams

Most Important Patterns: Swedish

		One character	Two characters	Three characters	Examples
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%)	kronor, perioder
	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%)	krafterna, saker

- Linguistic rules for plural form:
 - 5 suffixes: or, ar, (e)r, n, and no ending
- Found pattern:
 - “or” and “ar”
 - Also “na” and “rn”
 - “na” is definite article in plural forms

Most Important Patterns: Finnish

		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%)	tii (8%), iv_t (5%), t__ti (3%), sti (3%)	olivat, näyttikään
	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%)	tiesi, meidät

- Linguistic rules for verb tense:
 - Past tense often end with “i”
 - Sometimes “s” added → “si”
- Found pattern:
 - “i”, “si”, “ti”, “ui”
 - “iv_t” → third person plural

Interactions/Compositions of Patterns

- Consider the Spanish verb “gusta”
 - Gender=Not Applicable (NA)
 - Suffix “a” is indicator for gender=feminine
- Whether the model will classify “gusta”
 - wrongly as feminine or
 - correctly as NA.
- Consider most positive/negative set of characters per class

