Graph Convolutional Encoders for Syntax-aware NMT

Joost Bastings

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
http://joost.ninja

Collaborators

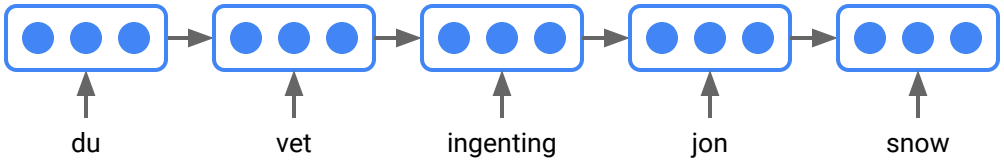
Ivan Titov

Wilker Aziz

Diego Marcheggiani

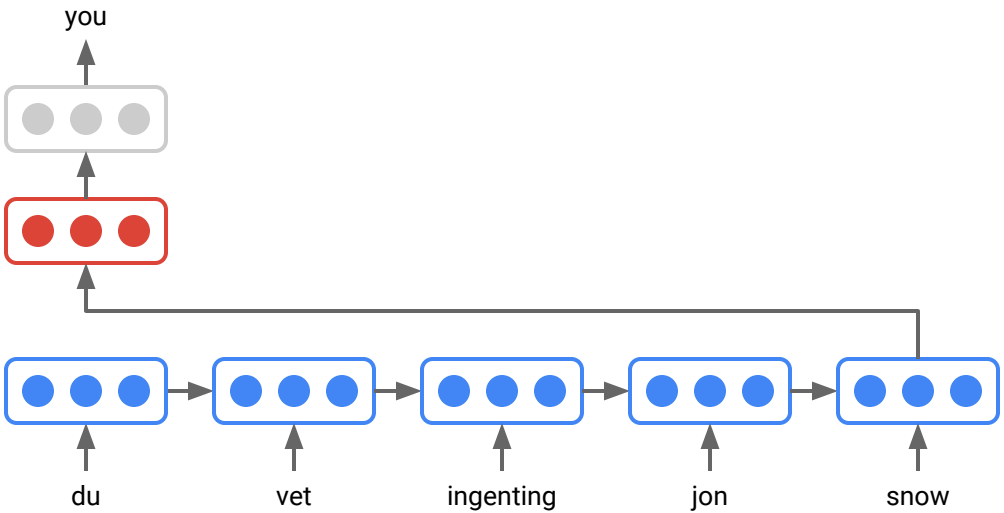
Khalil Sima’an

Encoder-Decoder models: no syntax, no hierarchical structure



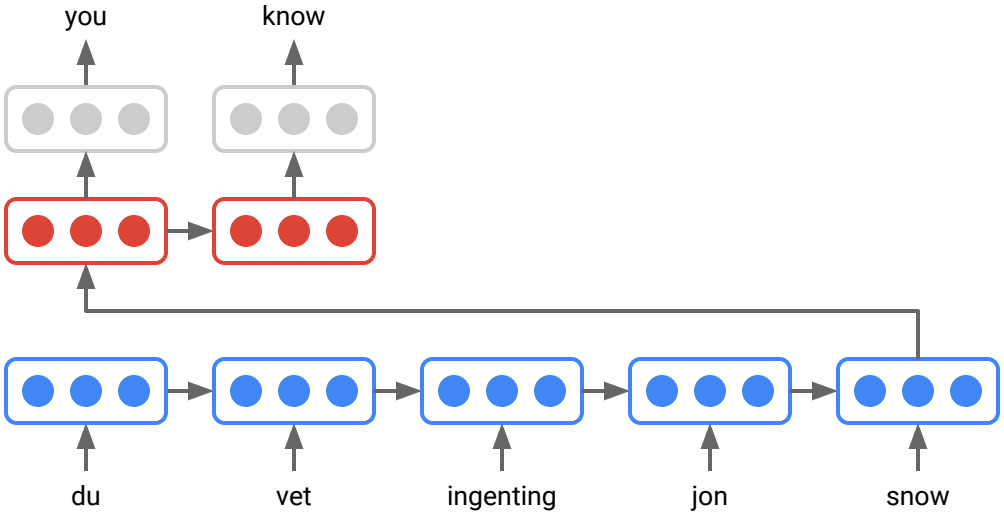
Sutskever et al. (2014), Cho et al. (2014)

Encoder-Decoder models: no syntax, no hierarchical structure



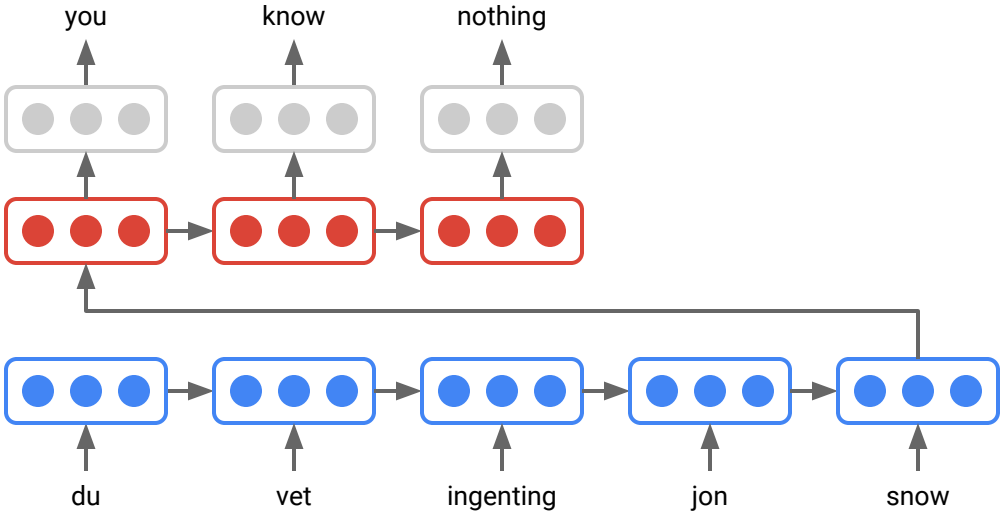
Sutskever et al. (2014), Cho et al. (2014)

Encoder-Decoder models: no syntax, no hierarchical structure



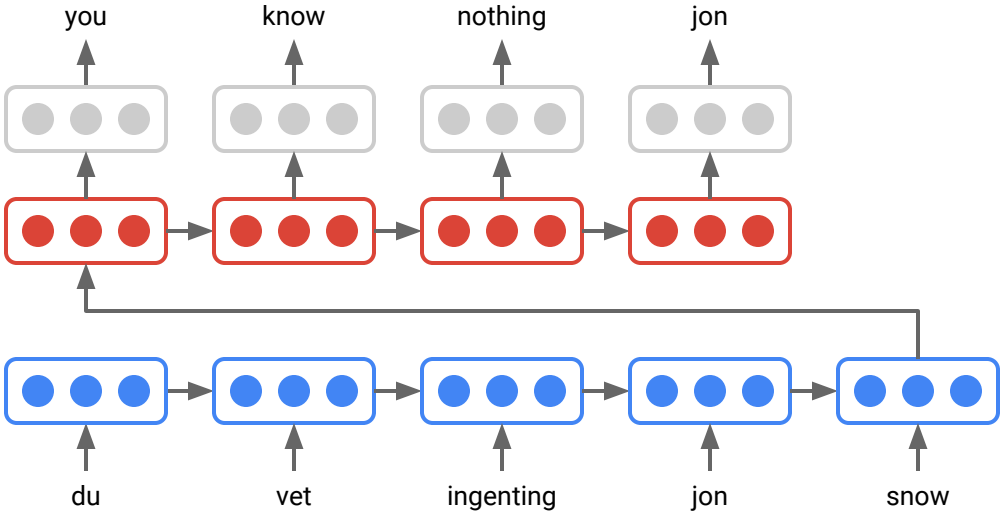
Sutskever et al. (2014), Cho et al. (2014)

Encoder-Decoder models: no syntax, no hierarchical structure



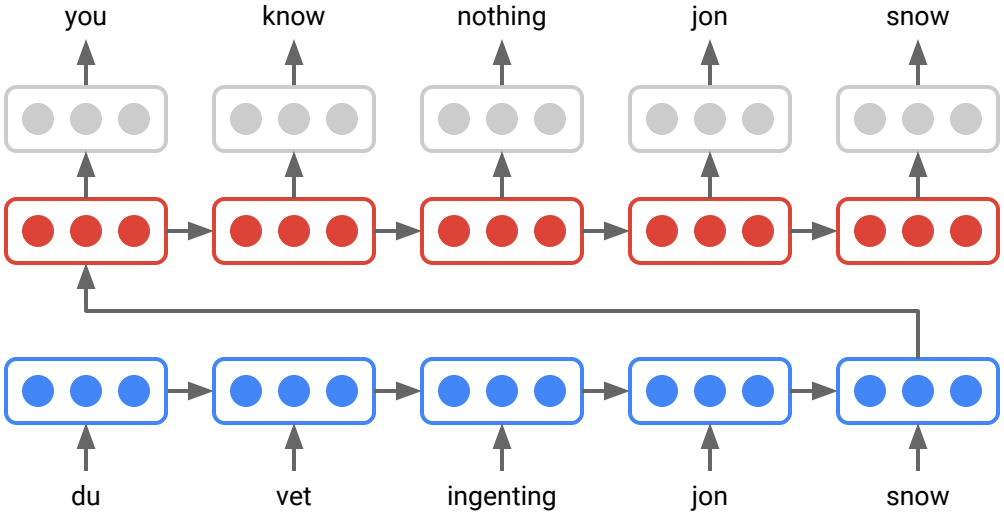
Sutskever et al. (2014), Cho et al. (2014)

Encoder-Decoder models: no syntax, no hierarchical structure



Sutskever et al. (2014), Cho et al. (2014)

Encoder-Decoder models: no syntax, no hierarchical structure



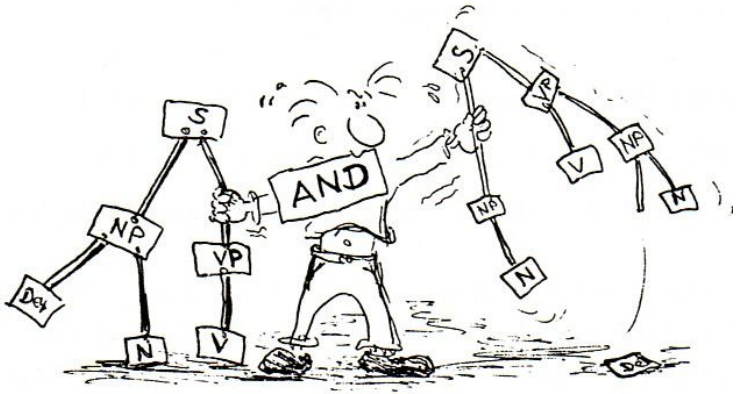
Sutskever et al. (2014), Cho et al. (2014)

Syntax can help!

Syntax:

- captures **non-local dependencies** between words
- captures **relations** between words

How to incorporate syntax into NMT?



Related Work

Multi-task learning / latent-graph:

- **Eriguchi et al.** (ACL, 2017). Learning to parse and translate improves NMT.
- **Hashimoto et al.** (EMNLP, 2017). NMT with Source-Side Latent Graph Parsing.
- **Nadejde et al.** (WMT, 2017). Predicting Target Language CCG Supertags Improves NMT.

Tree-RNN:

- **Eriguchi et al.** (ACL, 2016). Tree-to-Sequence Attentional NMT.
- **Chen et al.** (ACL, 2017). Improved NMT with a Syntax-Aware Encoder and Decoder.
- **Yang et al.** (EMNLP, 2017). Towards Bidirectional Hierarchical Representations for Attention-based NMT.

Serialized trees:

- **Aharoni & Goldberg** (ACL, 2017). Towards String-to-Tree NMT.
- **Li et al.** (ACL, 2017). Modeling Source Syntax for NMT.

Sequence-to-Dependency:

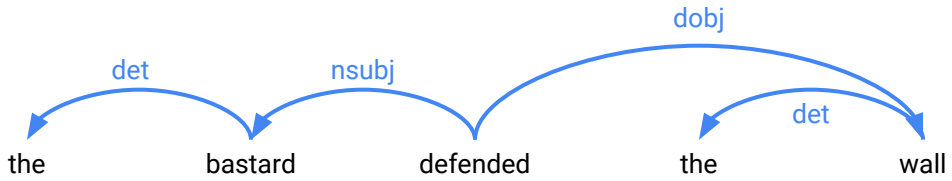
- **Wu et al.** (ACL, 2017). Sequence-to-Dependency NMT.

(((λ()(λ() 'yoav)))) @yoavgo 18 Apr

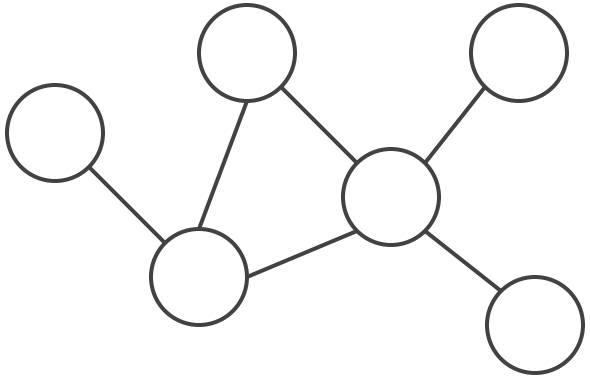
well, 2017 seems to be the year of syntax-in-NMT :)
We have it on target side, but seems effective also at source.
[syntax is good!]

Graph Convolutional Networks: Neural Message Passing

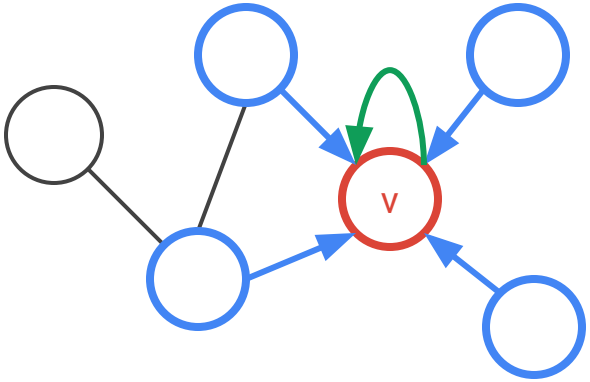
Many linguistic representations can be expressed as a graph



Graph Convolutional Networks: message passing



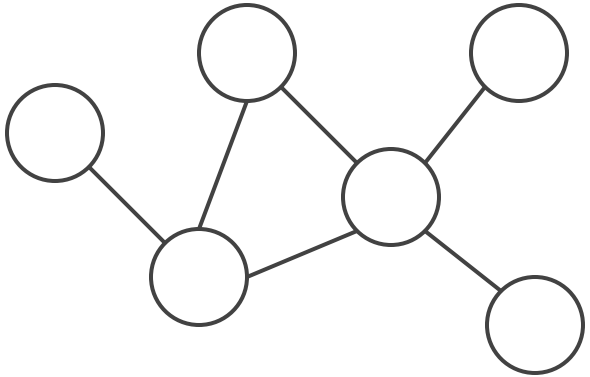
Undirected graph



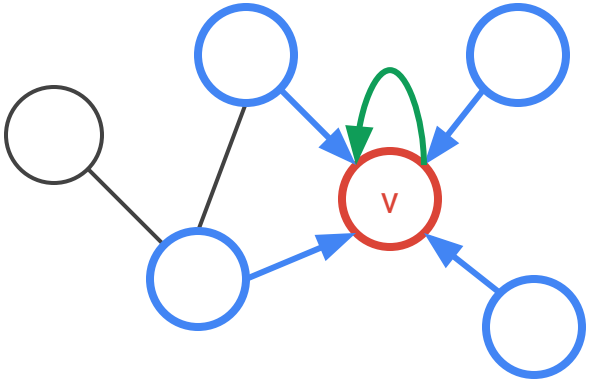
Update of node v

Kipf & Welling (2017). Related ideas earlier, e.g., Scarselli et al. (2009).

Graph Convolutional Networks: message passing



Undirected graph

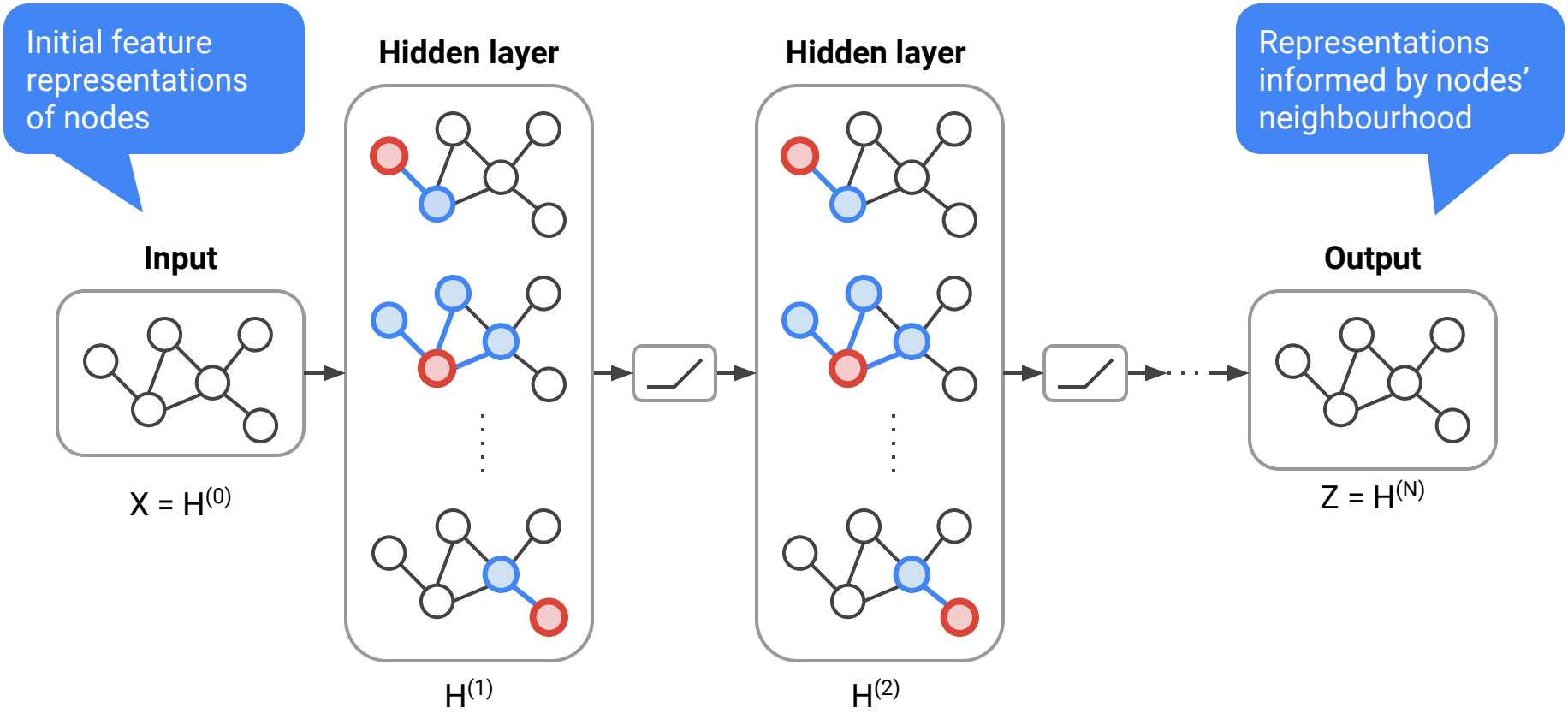


Update of node v

$$\mathbf{h}_v = \mathrm{ReLU}(W_{\mathrm{loop}}\mathbf{h}_v + \sum_{u \in \mathcal{N}(v)} W\mathbf{h}_u)$$

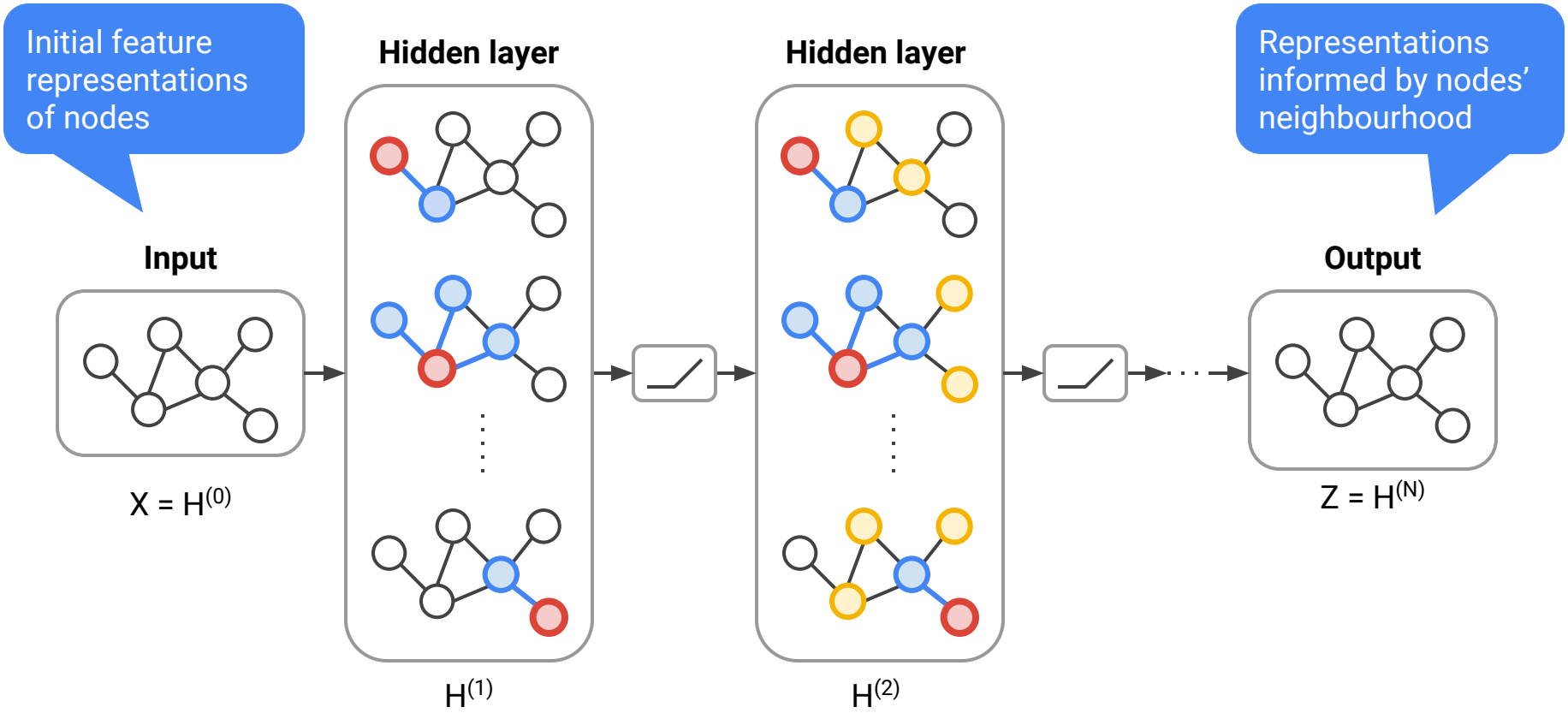
Kipf & Welling (2017). Related ideas earlier, e.g., Scarselli et al. (2009).

GCNs: multilayer convolution operation



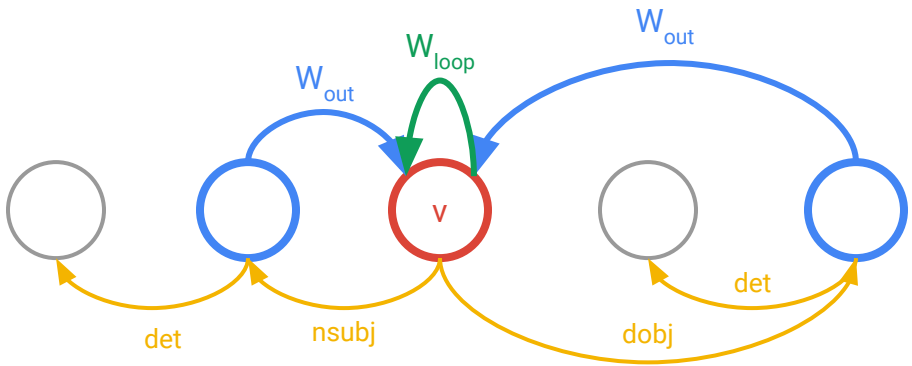
Adapted from Thomas Kipf. The computation is parallelizable can be made quite efficient (e.g., Hamilton, Ying and Leskovec (2017)).

GCNs: multilayer convolution operation

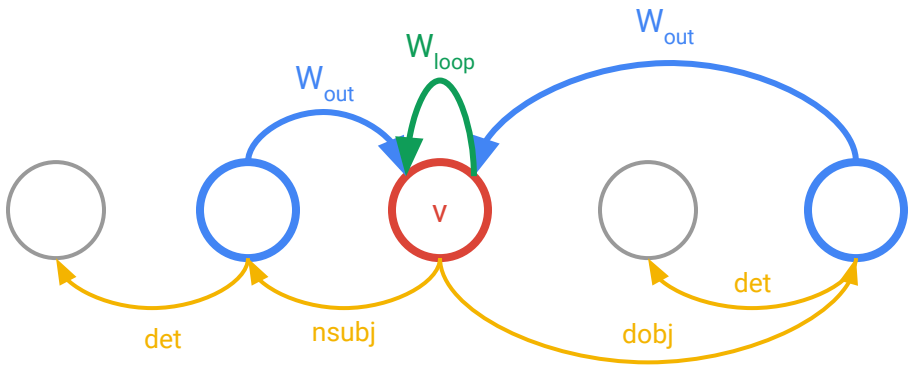


Adapted from Thomas Kipf. The computation is parallelizable can be made quite efficient (e.g., Hamilton, Ying and Leskovec (2017)).

Syntactic GCNs: directionality and labels



Syntactic GCNs: directionality and labels



Weight matrix for each direction: W_{out}, W_{in}, W_{loop}

Bias for each label, e.g. $\mathbf{b}_{\text{nsubj}}$

$$\mathbf{h}_v = \text{ReLU}\Big(\sum_{u \in \mathcal{N}(v)} W_{\text{dir}(u,v)}\, \mathbf{h}_u + \mathbf{b}_{\text{lab}(u,v)} \Big)$$

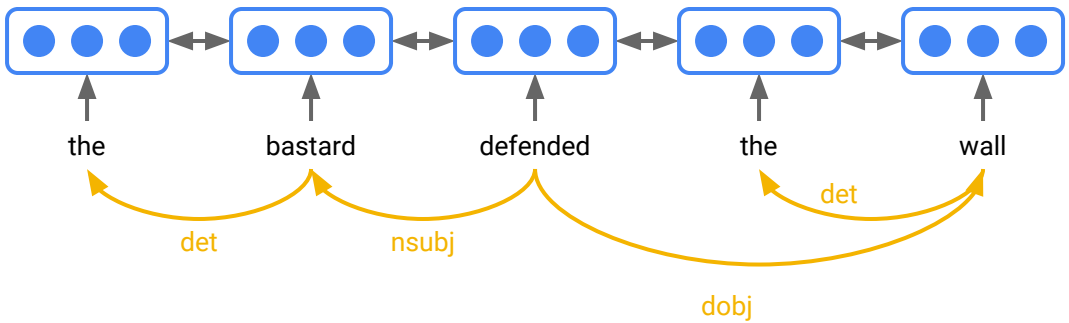
Syntactic GCNs: edge-wise gating

$$g_{u,v} = \sigma\left(\mathbf{h}_u \cdot \hat{\mathbf{w}}_{\mathrm{dir}(u,v)} + \hat{b}_{\mathrm{lab}(u,v)}\right)$$

$$\mathbf{h}_v = \mathrm{ReLU}\left(\sum_{u \in \mathcal{N}(v)} g_{u,v}\left(W_{\mathrm{dir}(u,v)}\,\mathbf{h}_u + \mathbf{b}_{\mathrm{lab}(u,v)}\right)\right)$$

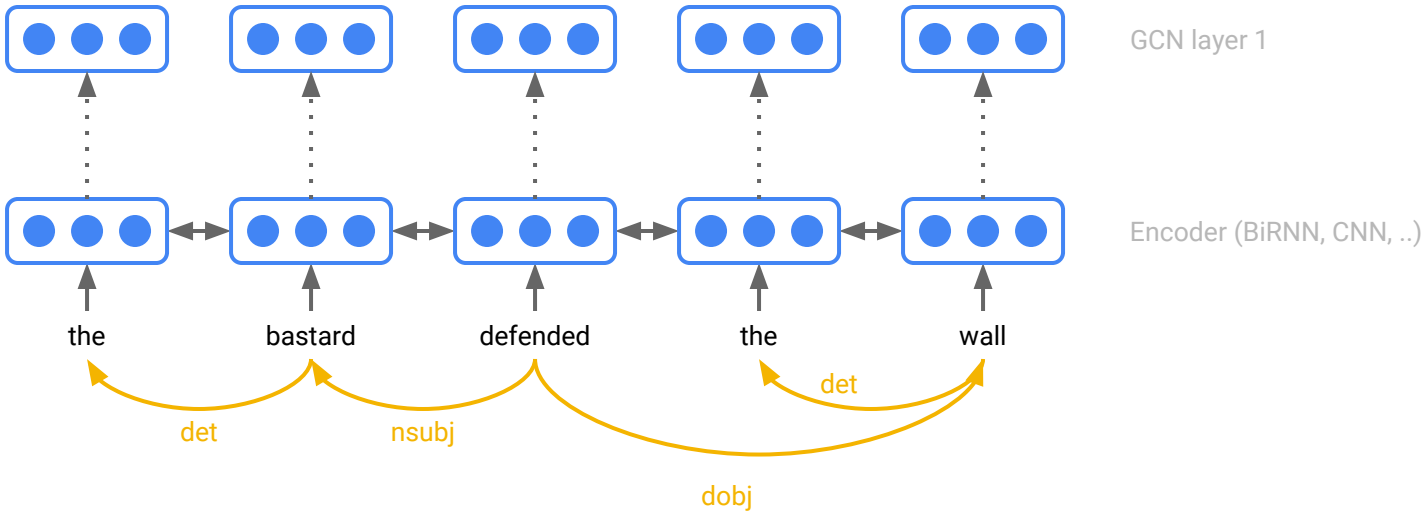
Graph Convolutional Encoders

Graph Convolutional Encoders

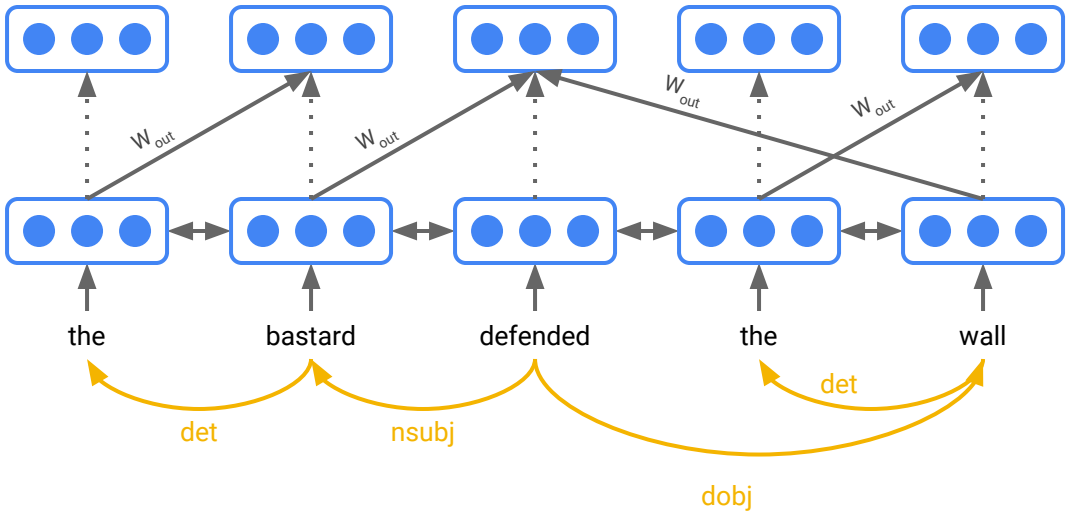


Encoder (BiRNN, CNN, ..)

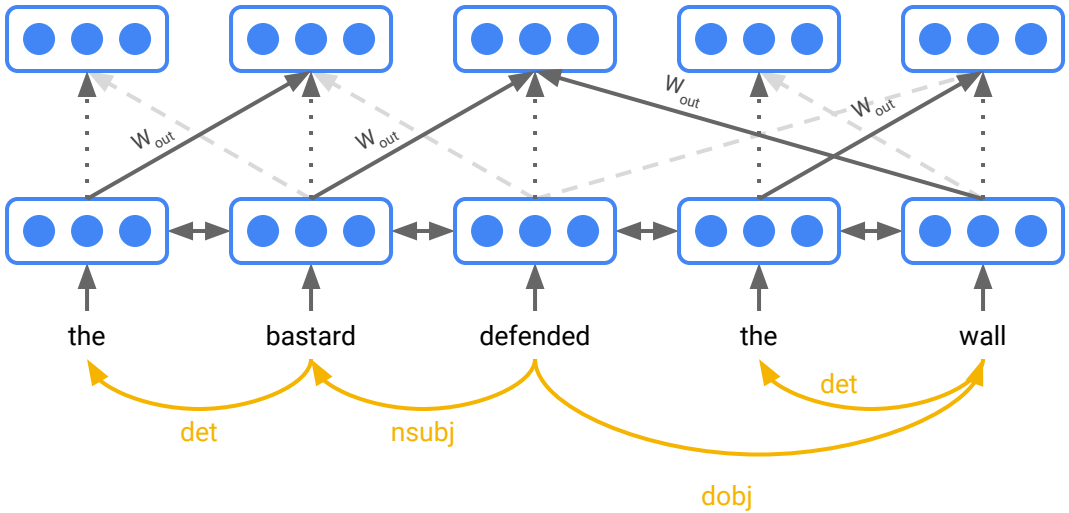
Graph Convolutional Encoders



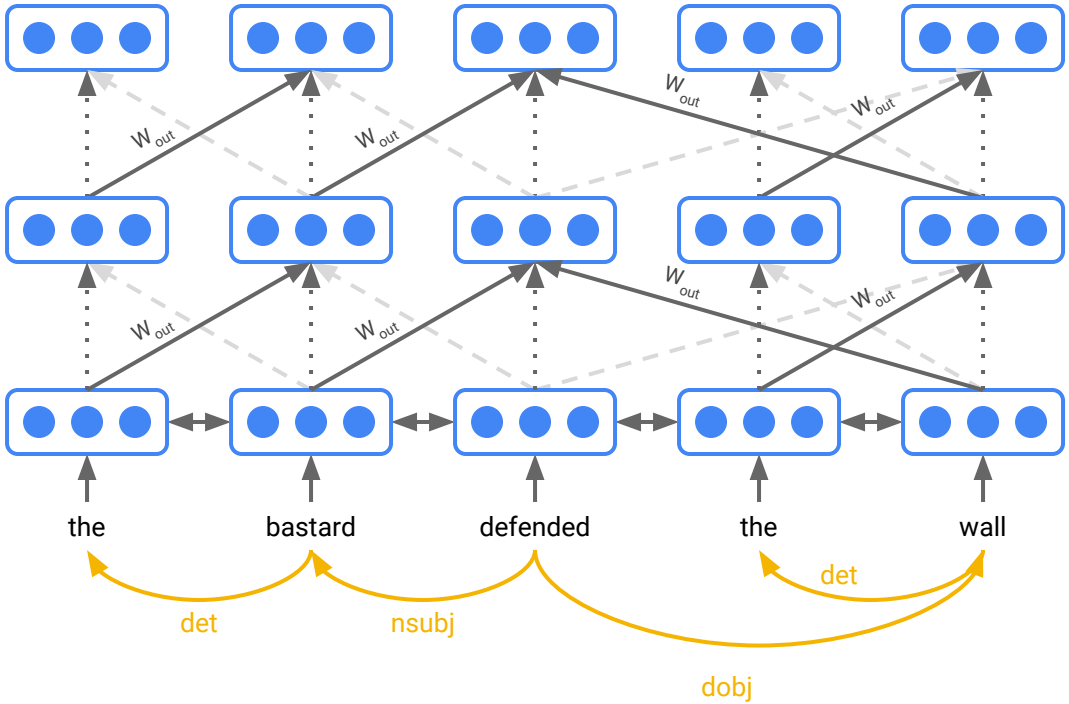
Graph Convolutional Encoders



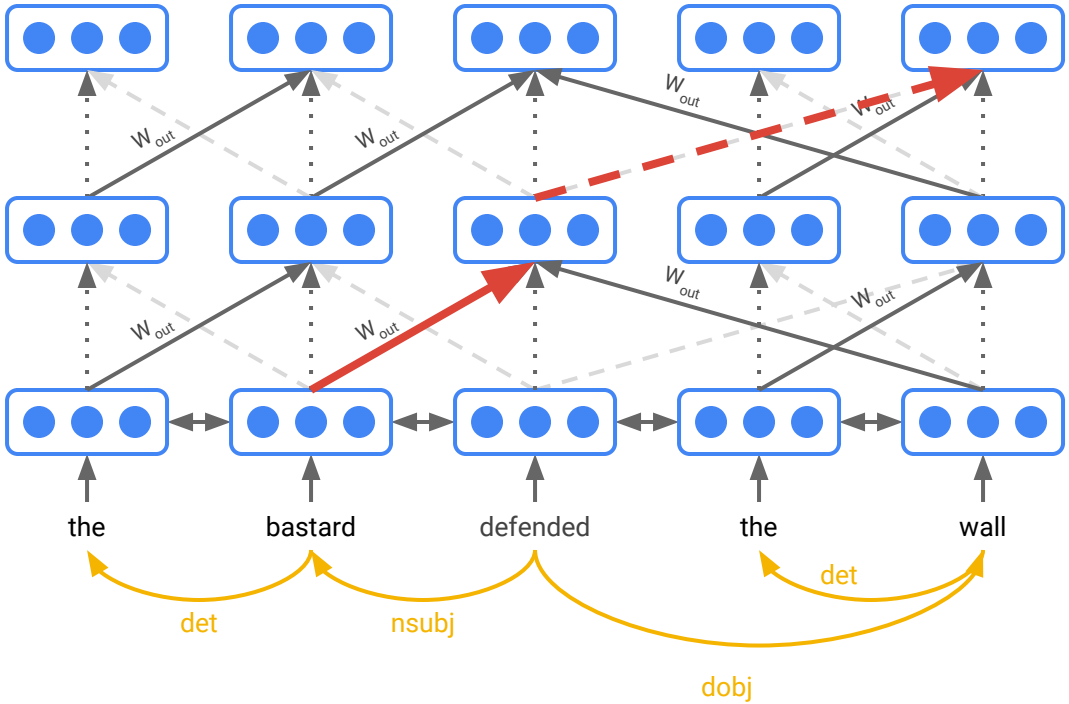
Graph Convolutional Encoders



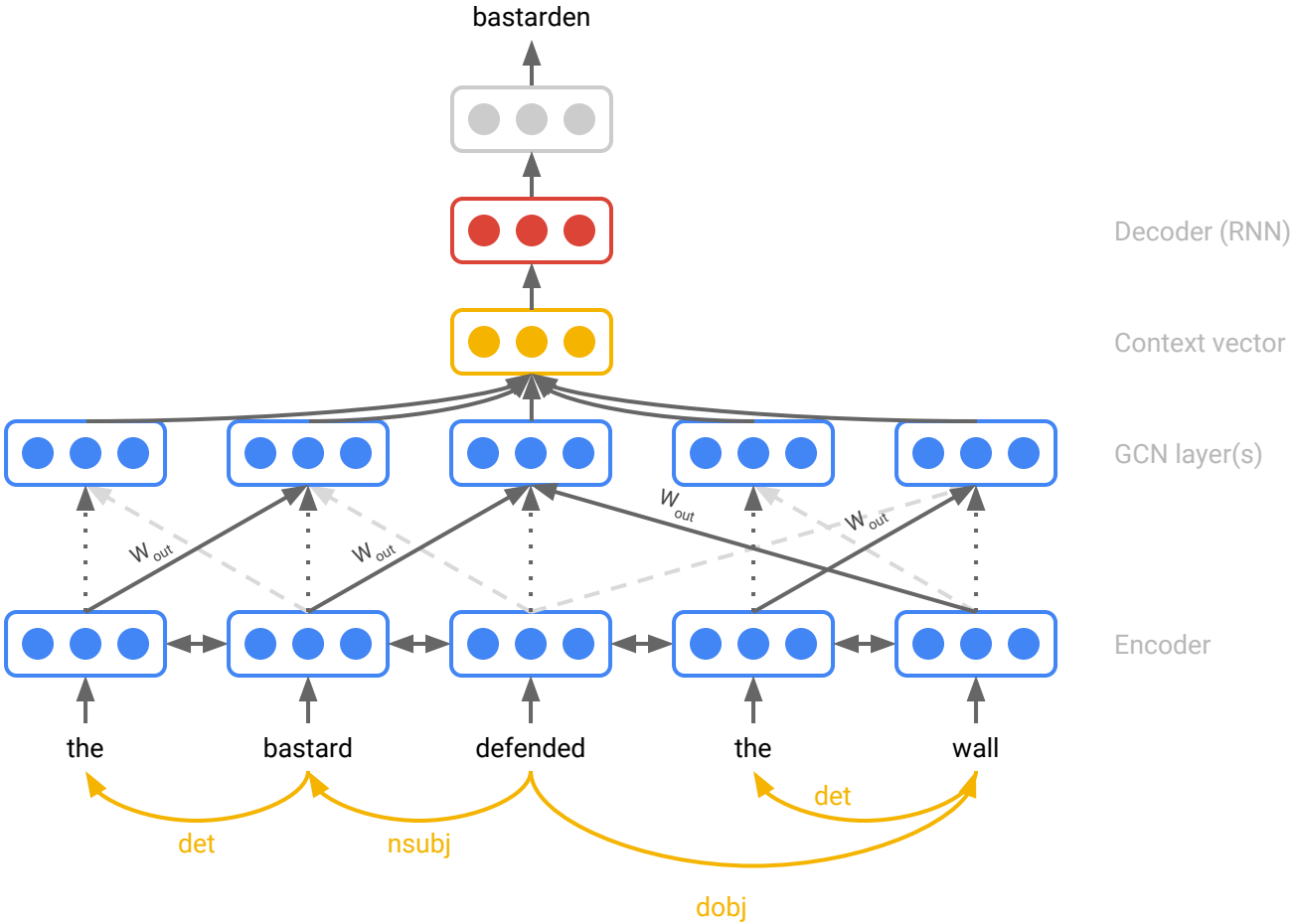
Graph Convolutional Encoders



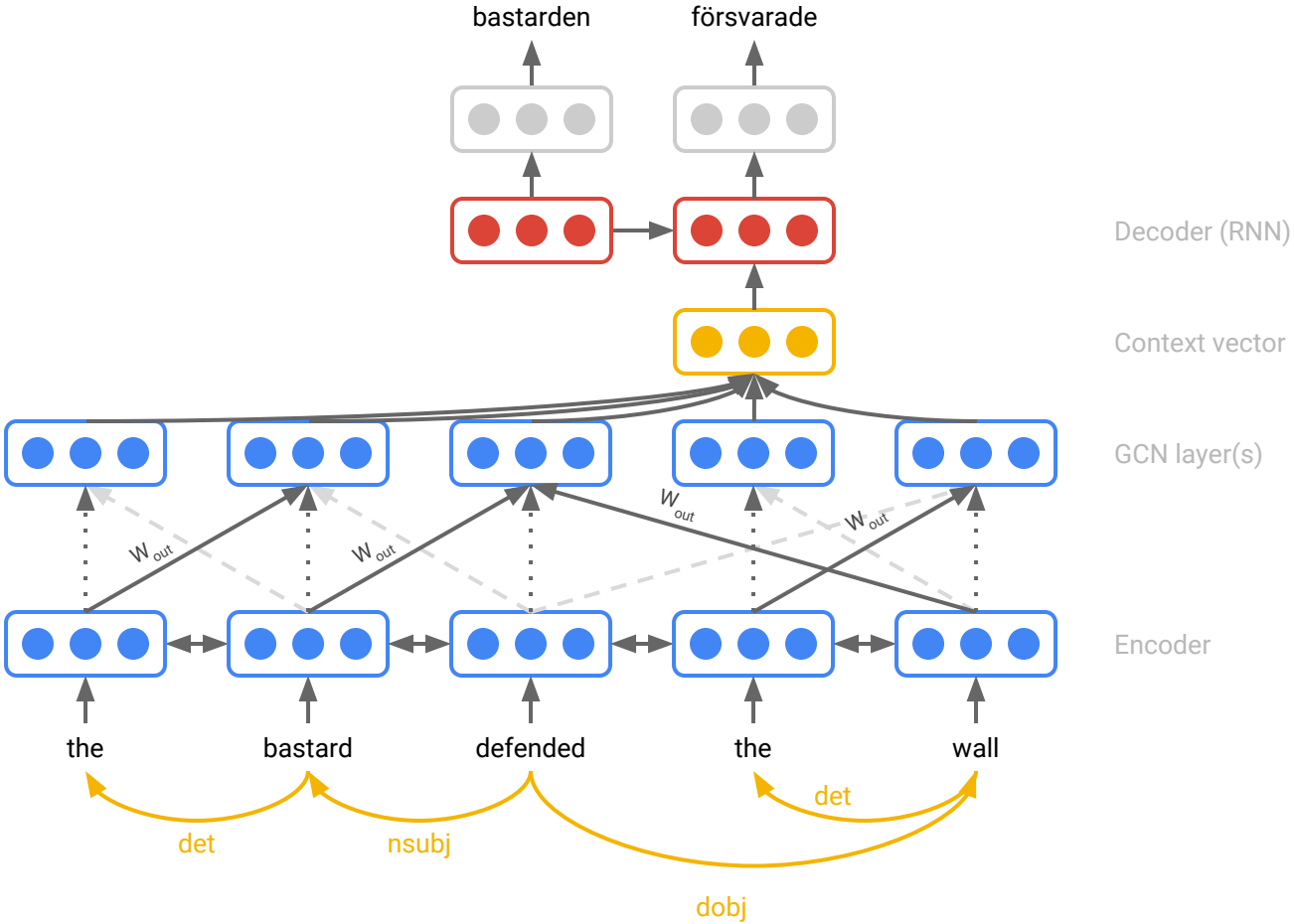
Graph Convolutional Encoders



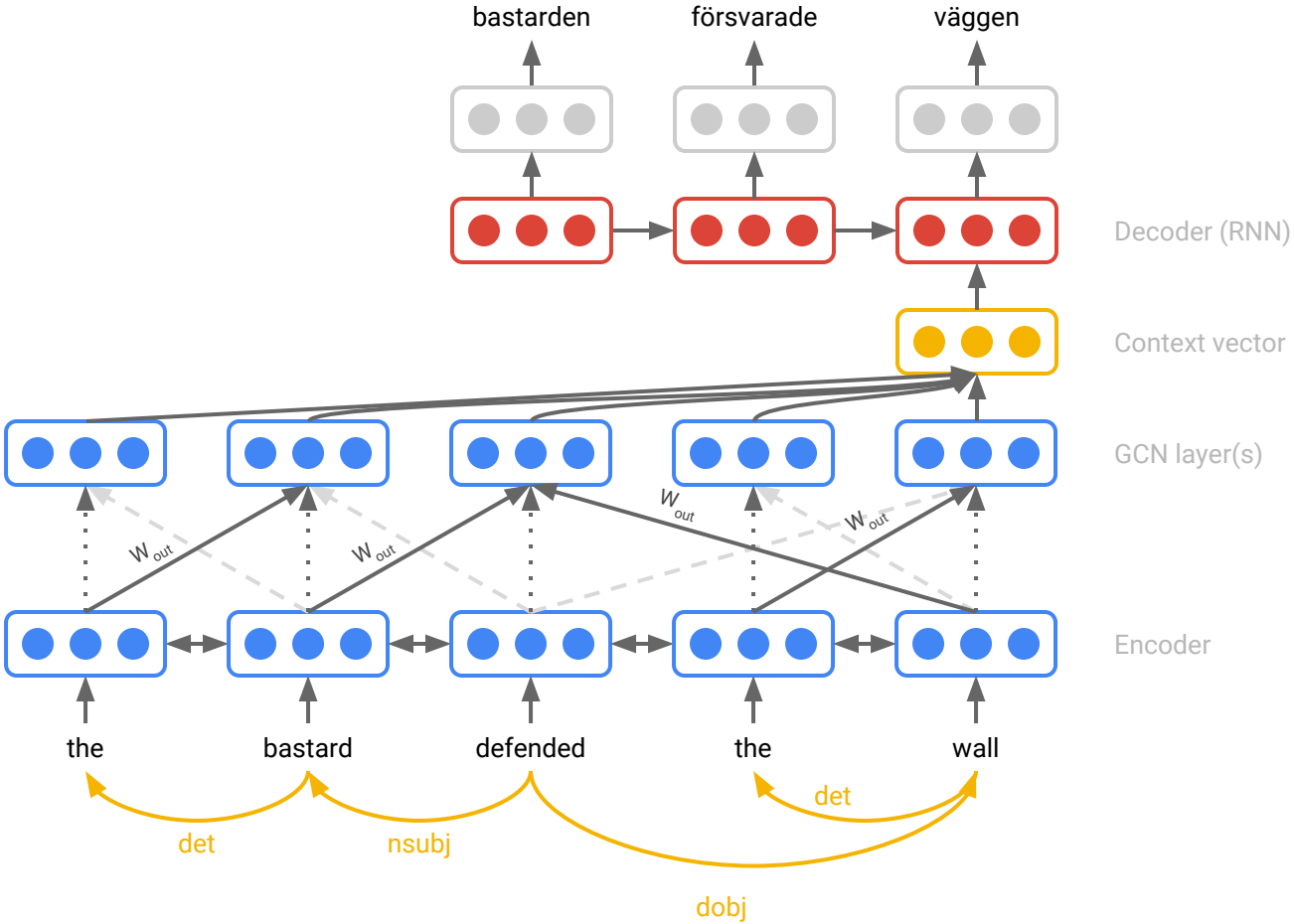
GCNs for NMT


bastarden
Decoder (RNN)
Context vector
GCN layer(s)
Encoder
W out
W out
W out
W out
the
bastard
defended
the
wall
det
nsubj
dobj
det

GCNs for NMT


bastarden
försvarade
Decoder (RNN)
Context vector
GCN layer(s)
Encoder
W out
W out
W out
W out
the
bastard
defended
the
wall
det
nsubj
dobj
det

GCNs for NMT

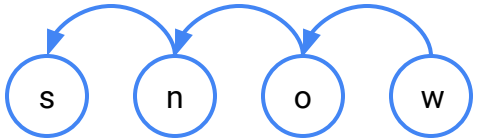

bastarden
försvarade
väggen
Decoder (RNN)
Context vector
GCN layer(s)
Encoder
W out
W out
W out
W out
the
bastard
defended
the
wall
det
nsubj
det
dobj

Experiments

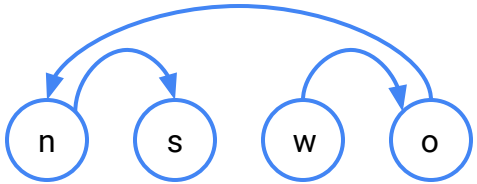
Artificial reordering experiment

Random sequences from **26 types**

Each token is linked to its **predecessor**



Sequences are randomly **permuted**

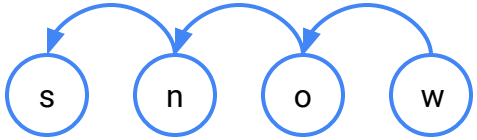


Each token is also linked to a **random** token with a “fake edge” using a different label set

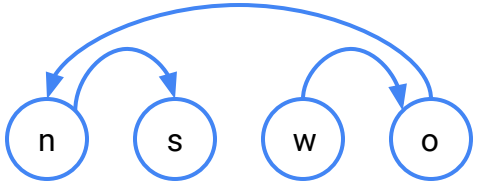
Artificial reordering experiment

Random sequences from **26 types**

Each token is linked to its **predecessor**



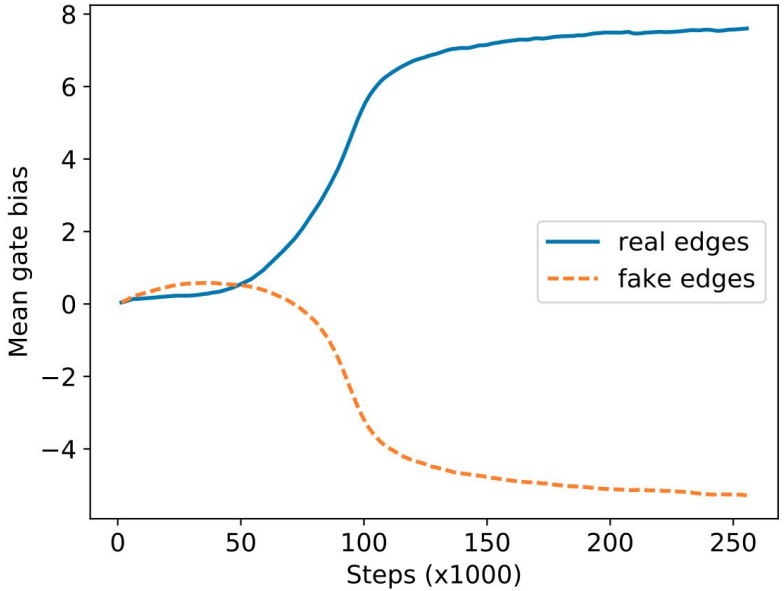
Sequences are randomly **permuted**



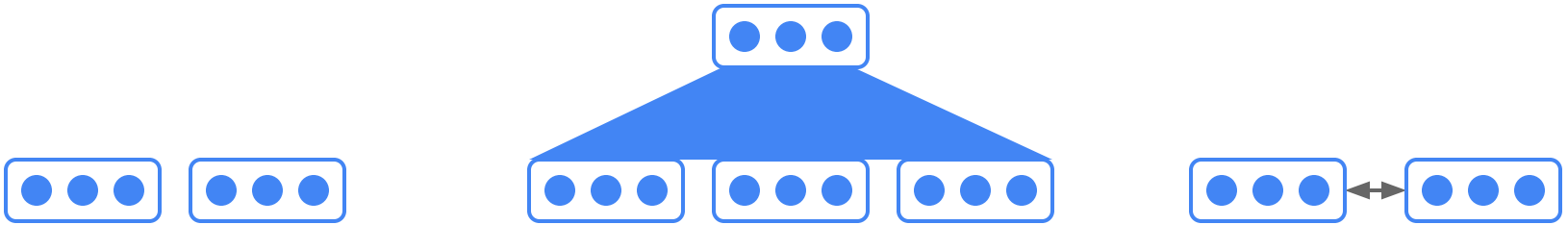
Each token is also linked to a **random** token with a “fake edge” using a different label set

A **BiRNN+GCN** model is able to learn how to put the permuted sequences **back into order**.

Real edges are distinguished from fake edges.



Three baselines



Bag of words

Convolutional

Recurrent

Machine Translation experiments

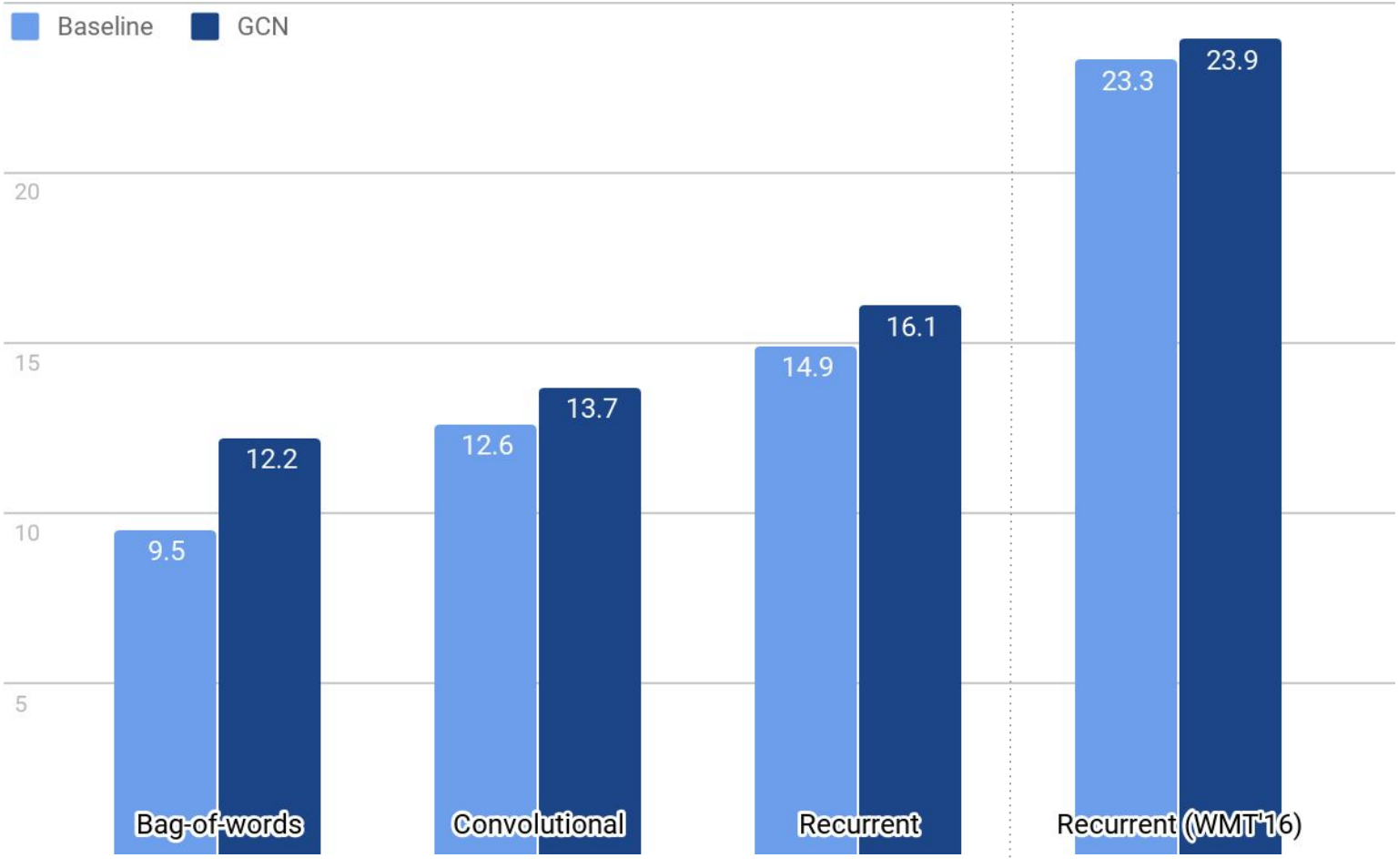
		Train	Validation *newstest2015*	Test *newstest2016*
English-German	NCv11	227 k	2169	2999
	WMT‘16	4.5 M		
English-Czech	NCv11	181 k	2656	2999

We parse English source sentences using **SyntaxNet**

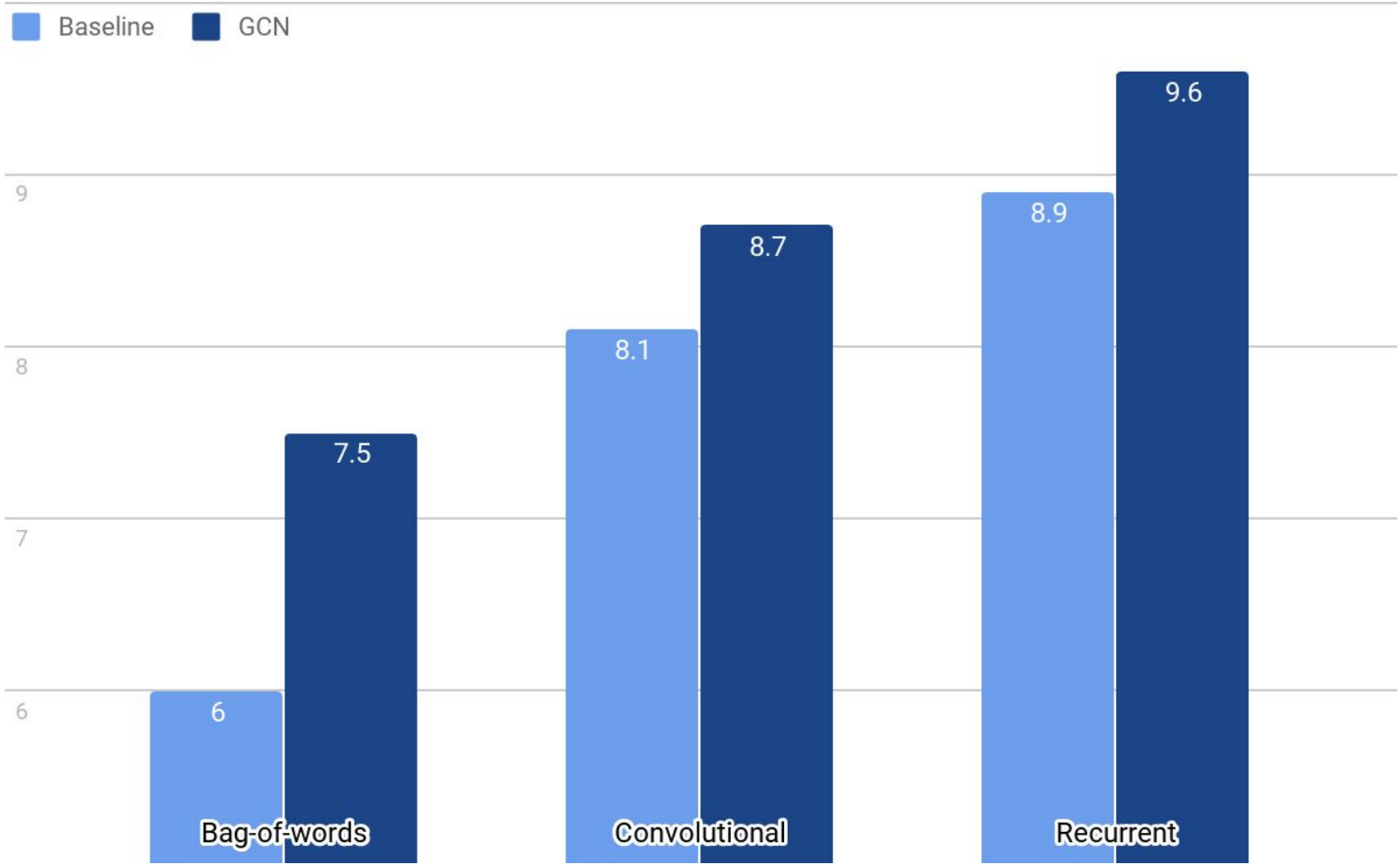
BPE on **target side** (8k merges, 16k for WMT’16)

Embeddings: 256 units, GRUs/CNNs: 512 units (800 for WMT’16)

Results English-German (BLEU)



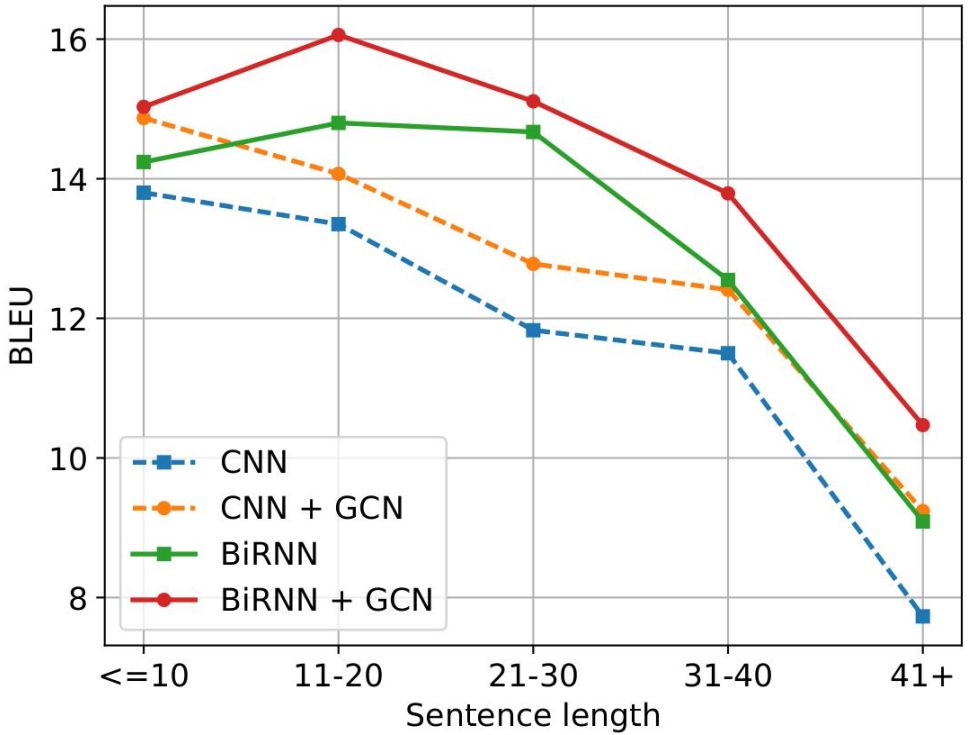
Results English-Czech (BLEU)



Effect of GCN layers

	English-German		English-Czech	
	$BLEU_1$	$BLEU_4$	$BLEU_1$	$BLEU_4$
BiRNN	44.2	14.1	37.8	8.9
+ GCN 1L	45.0	14.1	38.3	9.6
+ GCN 2L	**46.3**	**14.8**	**39.6**	**9.9**

Effect of sentence length



Flexibility of GCNs

In principle we can condition on **any kind** of graph-based linguistic structure:

Semantic Role Labels

Co-reference chains

AMR semantic graphs

Conclusion

Simple and effective approach to integrating syntax into NMT models

Consistent BLEU improvements for two challenging language pairs

GCNs are capable of encoding any kind of graph-based structure

Thank you!

Code: Neural Monkey + GCN (TensorFlow)

joost.ninja

Code: Semantic Role Labeling (Theano)

diegma.github.io

Ivan is hiring PhDs/Postdocs. Deadline tomorrow!

This work was supported by the European Research Council (ERC StG BroadSem 678254) and the Dutch National Science Foundation (NWO VIDI 639.022.518, NWO VICI 277-89-002).