Semantics in Machine Translation

Integrating AMR and UCCA into MT

Abelardo Carlos Martínez Lorenzo
martinez@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Semantics in Machine Translation

Machine Translation

- Task of automatically converting source text in one language to text in another language
- One of the most challenging NLP tasks given the fluidity of human language.

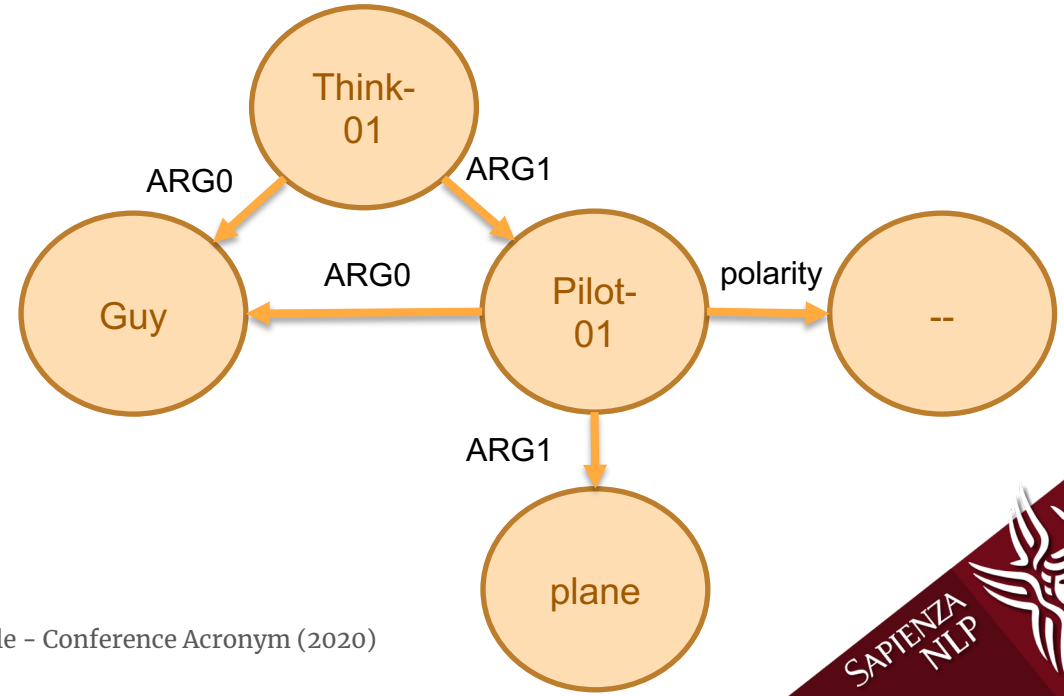
Rule-based Systems → Statistical Models → Deep Neural Networks

AMR

- It tries to embed the semantics of the sentence in a directed acyclic graph.
- The main idea is that two different sentences with the same meaning are going to be translated to the same graph representation.

Example:

- The guy doesn't think he will pilot a plane
- The guy thinks he won't pilot a plane



Benefits of Semantic in NMT

- Meaning Preservation
- Reduce Data Sparsity in low-resource languages

- Marcheggiani et al. (2018):
  - Introduce SRL into NMT
  - Predicate-argument structure of sentences gives valuable semantic information to MT

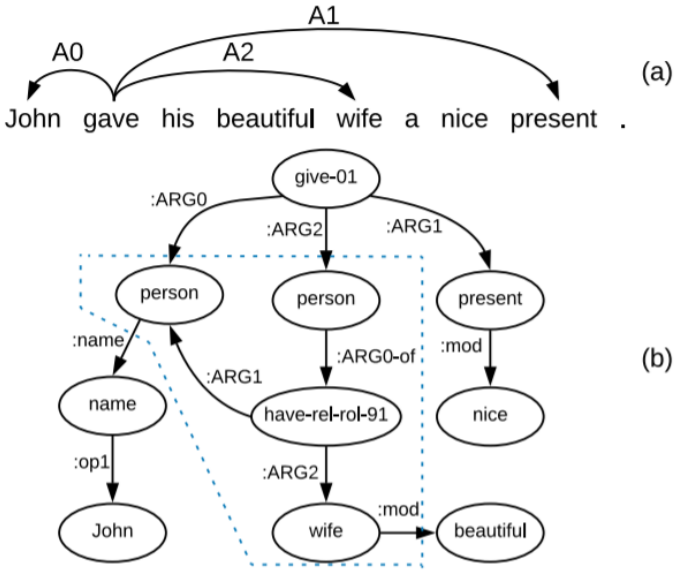


Figure 1: (a) A sentence with semantic roles annotations; (b) the corresponding AMR graph of that sentence.

Semantic Neural Machine Translation Using AMR

Linfeng Song et al.

AMR into NMT: GRN

Structural semantic information from AMRs can be complementary to the source textual input by introducing a higher level of information abstraction.

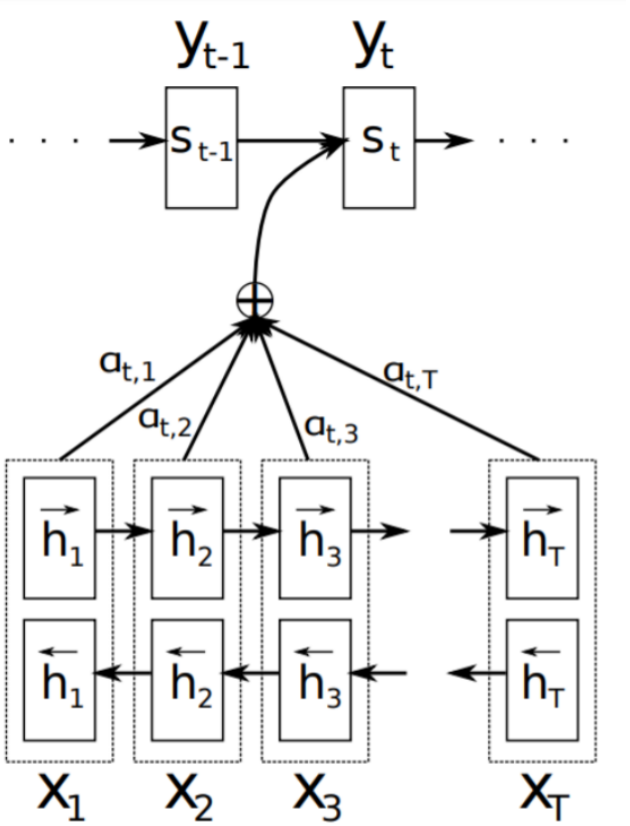
Architecture:

- A graph recurrent network (GRN) was leveraged to encode AMR graphs without breaking the original graph structure
- A sequential LSTM was used to encode the source input
- The decoder is a doubly attentive LSTM, taking the encoding results of both the graph encoder and the sequential encoder as attention memories.

Baseline

- Attention-based sequence-to-sequence model of Bahdanau et al. (2015)
  - Encoder:
    - Bidirectional LSTM
  - Decoder:
    - Attention-based LSTM decoder

- Performs attention over the encoder hidden states for each decoding step.



Incorporating AMR

- Encoder:
  - Bidirectional LSTM
  - Graph Recurrent Network (GRN): Model the state transition process
- Decoder:
  - Attention-based LSTM decoder

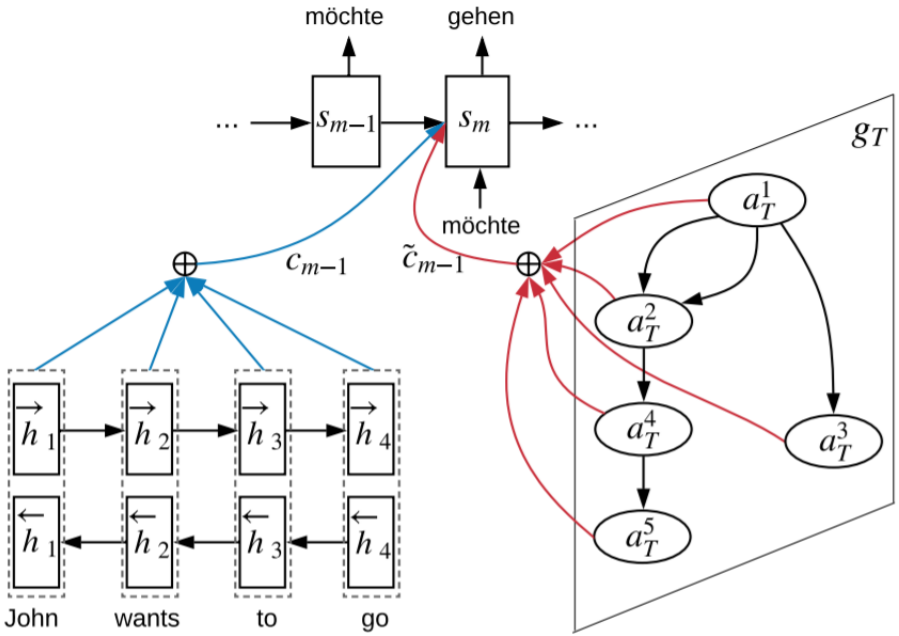


Figure 2: Overall architecture of our model.

 Martinez Lorenzo, Abelardo Carlos– AMR into NMT: Graph Recurrent Networks- TACL (2019)

Graph Recurrent Network

AMR Graph: $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$
Node State: $\boldsymbol{a}^j \quad v_j \in \boldsymbol{V}$
Graph State: $\boldsymbol{g} = \{\boldsymbol{a}^j\}|_{v_j \in \boldsymbol{V}}$

With this state transition mechanism, information of each node is propagated to all its neighbouring nodes after each step.

So after several transition steps, each node state contains the information of a large context, including its ancestors, descendants, and siblings.

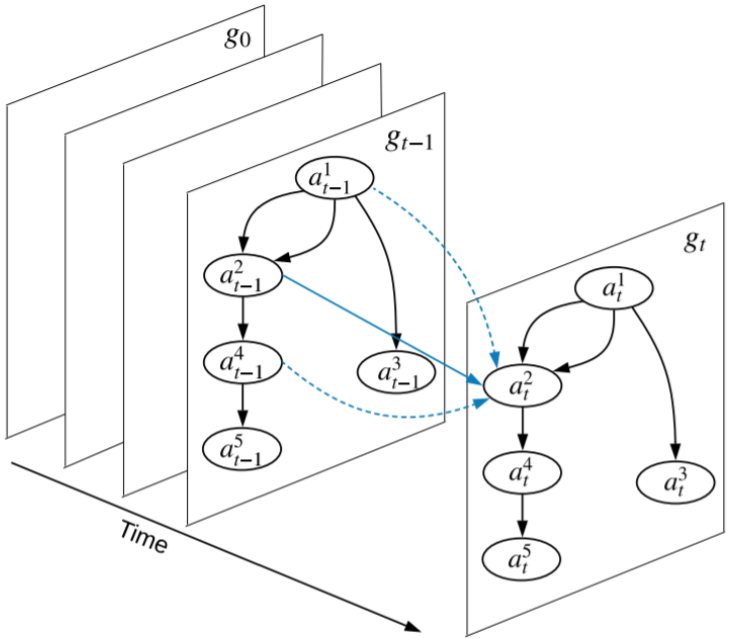


Figure 3: Architecture of the graph recurrent network.

Doubly Attentive Decoder

No one-to-one correspondence Between AMR nodes and words

Solution: external attention model

The context vector is incorporated Into the calculation of the output Probability distribution over the Target vocabulary

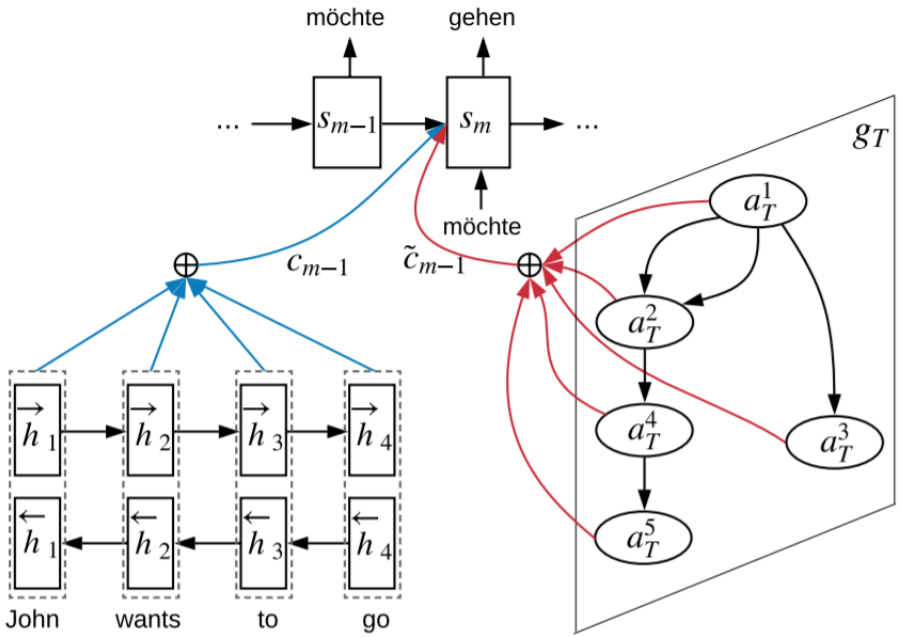


Figure 2: Overall architecture of our model.

Martinez Lorenzo, Abelardo Carlos– AMR into NMT: Graph Recurrent Networks- TACL (2019)

Experiments

- Effectiveness of AMR for English-to-German translation
- WMT16 English-to-German Dataset
    - 4,5 Millions sentence pairs

- Moses: for tokenizing English and German sentences
- JAMR: for parsing the Sentences into AMR
- BPE: for dealing with strange words

Experiments

System	NC-v11			FULL		
	BLEU	TER↓	Meteor	BLEU	TER↓	Meteor
OpenNMT-tf	15.1	0.6902	0.3040	24.3	0.5567	0.4225
Transformer-tf	17.1	0.6647	0.3578	25.1	0.5537	0.4344
Seq2seq	16.0	0.6695	0.3379	23.7	0.5590	0.4258
Dual2seq-LinAMR	17.3	0.6530	0.3612	24.0	0.5643	0.4246
Duel2seq-SRL	17.2	0.6591	0.3644	23.8	0.5626	0.4223
Dual2seq-Dep	17.8	0.6516	0.3673	25.0	0.5538	0.4328
Dual2seq	**19.2***	**0.6305**	**0.3840**	**25.5***	**0.5480**	**0.4376**

Table 3: TEST performance. *NC-v11* represents training only with the NC-v11 data, while *Full* means using the full training data. * represents significant (Koehn, 2004) result ($p < 0.01$) over *Seq2seq*. ↓ indicates the lower the better.

Experiments

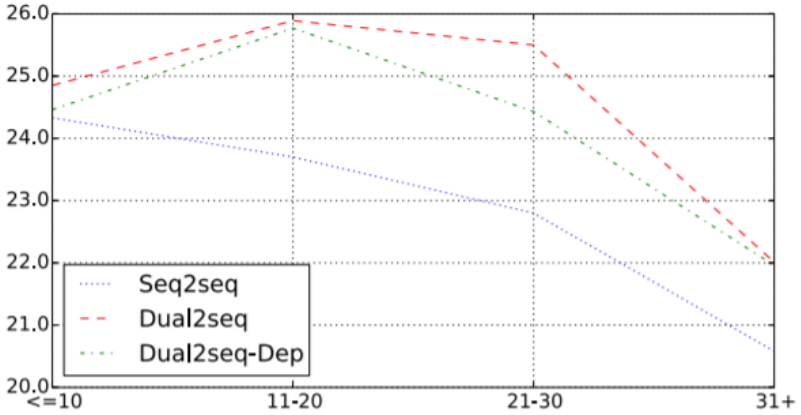


Figure 5: Test BLEU score of various sentence lengths.

Semantic Convolutional Neural Machine Translation Using AMR for English-Vietnamese

Viet Pham et al.

General Overview

An extension of the Convolutional NMT model to incorporate AMR

Convolutional Neural Networks in both encoder and decoder, is considered a better NMT model than Seq2Seq, which uses Recurrent Neural Networks.

In CNMT, elements can be fully parallelized during training to better exploit the GPU hardware and optimization is easier.

Baseline: ConvS2S

Graph Encoder

Node Embeddings:

1. Transform Nodes to Vectors
2. Categorize the neighbours $N_{\vdash}(v)$ $N_{\dashv}(v)$
3. Aggregate forward representations $\{h^{k-1}{}_{u\vdash}, \forall u \in N_{\vdash}(v)\}$ $h^{k}{}_{N_{\vdash}(v)}$
4. Concatenate current forward representation with new generated vector. Feed fully connected layer to update forwards representations $h^{k-1}{}_{v\vdash}$ $h^{k}{}_{N_{\vdash}(v)}$ $h^{k}{}_{v\vdash}$
5. Update the backward
6. Repeat steps (3)~(5) K times. Concatenation final forwards with backwards representations is used as the final bi-directional representation of the node

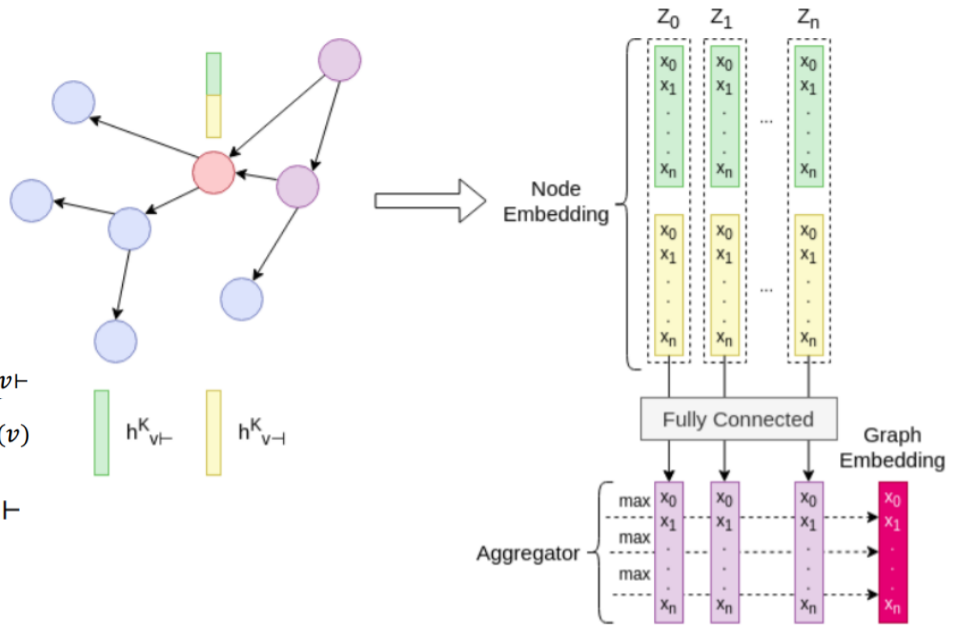


Figure 2. Graph encoder architecture.

Graph Encoder

The node embeddings are passed through a fully-connected layer and applied max-pooling aggregator to get the graph embedding.

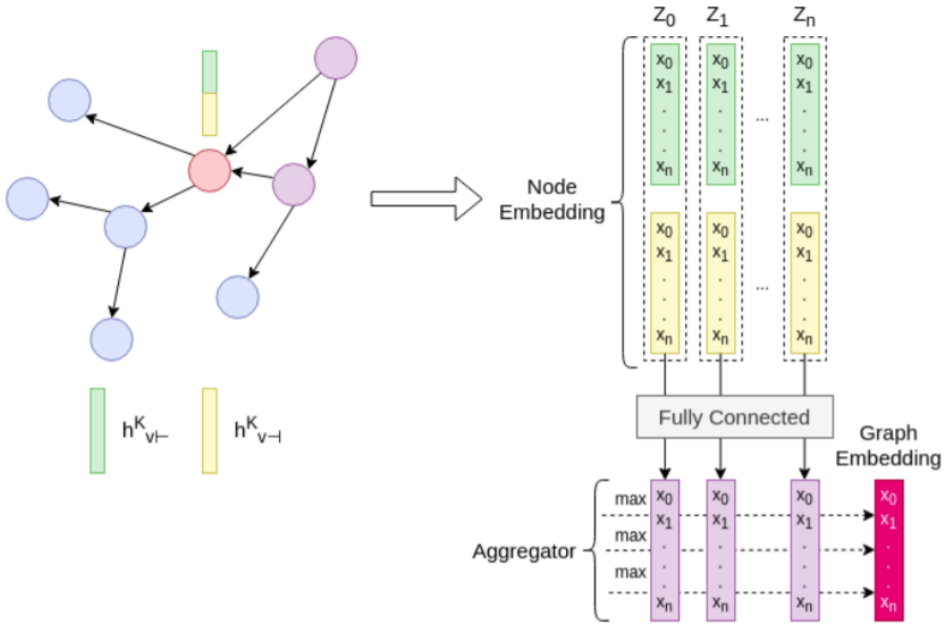


Figure 2. Graph encoder architecture.

CNN with AMR

The encoder remains the same that the baseline

Graph knowledge in order to enrich deep CNN representations

There is an attention mechanism because there is no one-to-one correspondence mapping between AMR nodes and words in a sentence

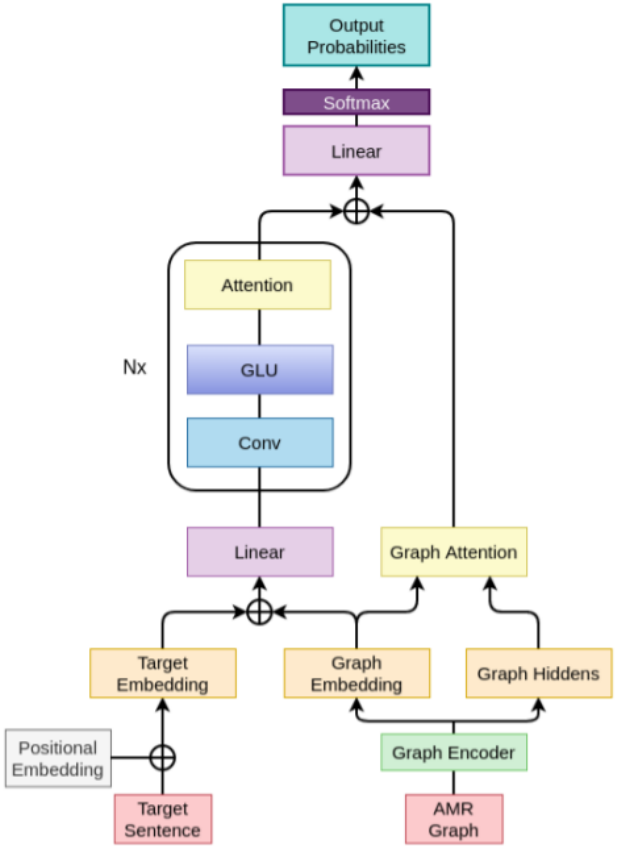


Figure 3. Decoder architecture.

Experiments: Dataset IWSLT 2015

Table 1. Statistics of the English-Vietnamese datasets

MT dataset	#tokens		#types		#sents
	en	vi	en	vi	
train	2,435,771	2,867,788	44,573	21,611	117,055
dev(tst2012)	27,988	34,298	3,518	2,170	1,553
test(tst2013)	26,729	33,683	3,676	2,332	1,268

Experiments: Results

Table 2. Experiments

Methods	bleu
ConvS2S	26.98 (61.6/35.7/21.9/13.7)
ConvS2S+AMR	**27.30** (62.2/36.4/22.6/14.3)

Results

- ConvS2S+AMR is just slightly better than the pure ConvS2S:
  - CNNs rely on convolutional mechanisms (parallelized computations)
    - decreased performance on syntax-sensitive tasks compared to RNN
  - Simplified version of AMR
- AMR can improve MT performance in low resource languages

Integrating AMR to Neural Machine Translation using Graph Attention Networks

Long H. B. Nguyen et al.

AMR into NMT: Graph Attention Networks

Doubly-attentive LSTM decoder, taking the results of both graph encoder and the sequential encoder as attention memories.

Graph encoder:

- Extend the node embedding algorithm GRAPH2SEQ (Kun xu et al., 2018). Use the edge information directly:
    - Not changing the topology of the graph
    - Not adding noise
- Graph Attention Networks

Graph Encoder Architecture

AMR graph G = (V, E)

1. Transform Nodes into features Vectors

$$h^v = W_E(v) \qquad \mathbf{h} = \{h^v\}, \forall v \in \mathcal{V}$$

2. Categorize the neighbours $N_{\vdash}(v)$ $N_{\dashv}(v)$

$$h_{bw} = \{W_E(u)\}, \forall u \in \mathcal{N}_{\dashv}(v) \qquad h_{fw} = \{W_E(u)\}, \forall u \in \mathcal{N}_{\vdash}(v)$$

3. No forward/backward neighbours

$$h_{fw} = \mathbf{0} \text{ or } h_{bw} = \mathbf{0}$$

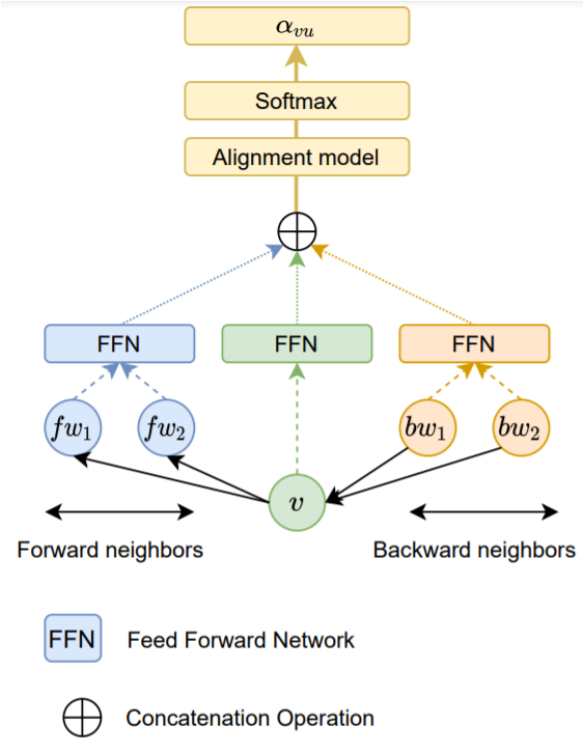


Figure 2: Graph encoder architecture.

Graph Encoder Architecture

- Linear Transformation -> High level features

$$h^{'v} = W_h h^v, \forall v \in \mathcal{V} \qquad h^{'u}_{bw} = W_{bw} h^u_{bw}, \forall u \in \mathcal{N}_{\dashv}(v)$$

$$h^{'u}_{fw} = W_{fw} h^u_{fw}, \forall u \in \mathcal{N}_{\vdash}(v)$$

- Perform self-attention -> indicate importance

$$e_{vu} = a([h^{'v} \oplus h^{'u}_{fw} \oplus h^{'u}_{bw}]) \qquad \alpha_{vu} = softmax(e_{vu})$$

- Compute the linear computation of the feature:

$$h^v = \sigma(\sum_{u \in \mathcal{N}_v} \alpha_{ij} W h^v)$$

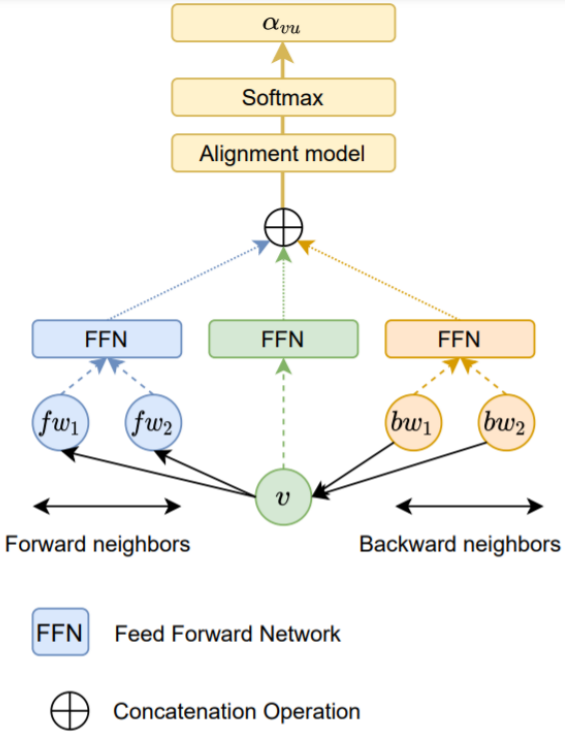


Figure 2: Graph encoder architecture.

Martinez Lorenzo, Abelardo Carlos - AMR into NMT: Graph Attention Networks IEEE (2020)

Decoder Architecture

- Attention Mechanism:
  - Help in the alignment
  - Address the bottleneck problem for largest sentences
- Graph Attention: takes the graph hidden states and the decoder states. Context vector:

$$\hat{e}_{ij} = a(s_{i-1}, \hat{h}_j) \qquad \hat{\alpha}_{ij} = \frac{\exp(\hat{e}_{ij})}{\sum_{k=1}^{\mathcal{V}} \exp(\hat{e}_{ik})} \qquad \hat{c}_i = \sum_{j=1}^{\mathcal{V}} \hat{\alpha}_{ij} h_j$$

Probability distribution:

$$P_{vocab} = softmax(W_o[s_i, c_i, \hat{c}_i] + b_o)$$

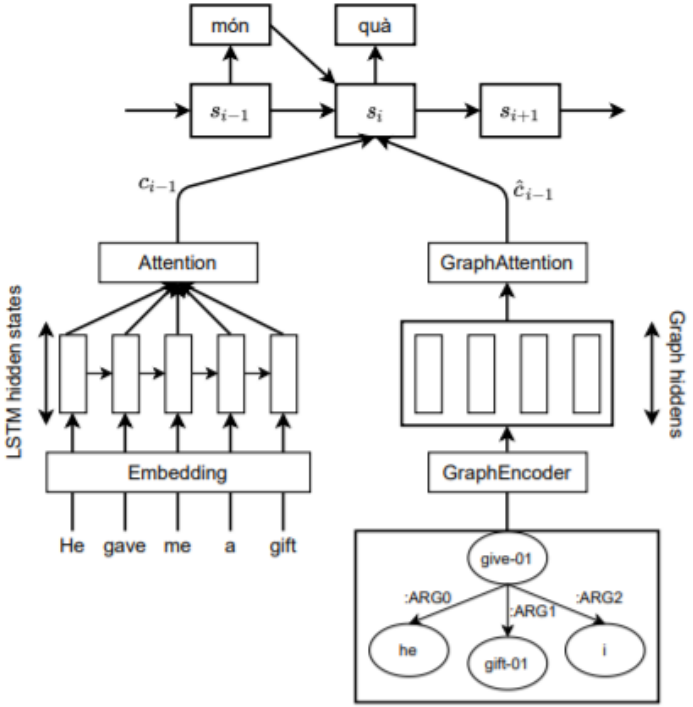


Figure 3: Our proposed decoder with dual attention mechianism.

Experiments

- Dataset IWSLT 2015
- BPE for pre-processing phase (dealing with rare and compound words)
- NeuralAmr toolkit for AMR parsing

Table I: Statistics of the English-Vietnamese datasets

Dataset	# tokens		#types		# sents
	en	vi	en	vi	
train	2,435,771	2,867,788	44,573	21,611	131,263
dev(tst2012)	27,988	34,298	3,518	2,170	1,553
test(tst2013)	26,729	33,683	3,676	2,332	1,268
test(tst2015)	20,850	26,235	3,127	2,059	1,080

Experiments: Results

- Metric: BLEU
- Compared with the AMR into NMT model using GRN

Model	tst2013	tst2015
Song et al.(2019)	26.12	23.58
Our method (w/ unidirectional LSTM)	26.72	24.34
Our method (w/ bidirectional LSTM)	**29.15**	**25.89**

Table II: Experimental results.

Incorporate Semantic Structures into Machine Translation Evaluation via UCCA

Jin Xu et al.

Machine Translation Evaluation

Evaluate the quality of sentence produced for MT System
Based on Lexical similarity
Neglect semantic structure

Most used:

- Bleu: Compare n-gram overlapping
- Meteor: align words and phrases to calculate a modified weighted F-score

Semantic Machine Translation Evaluation

Certain words or phrases appearing in all good translations of one source text, and these words tend to convey important semantic information.

SEMANTIC CORE WORDS

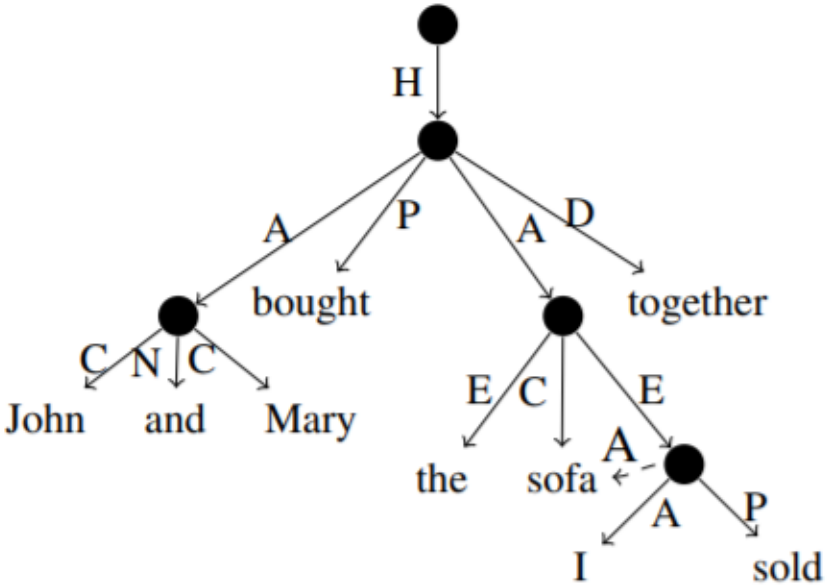
Semantically Weighted Sentence Similarity (SWSS): it is a new metric that leverages the power of UCCA to identify semantic core words, and then calculates sentence similarity scores on the overlap of semantic core words

UCCA

Represent the semantic of the sentence using directed acyclic graph (DAG) where end nodes correspond to specific words in the sentence.

Non end nodes represent the combination of meanings of its child nodes.

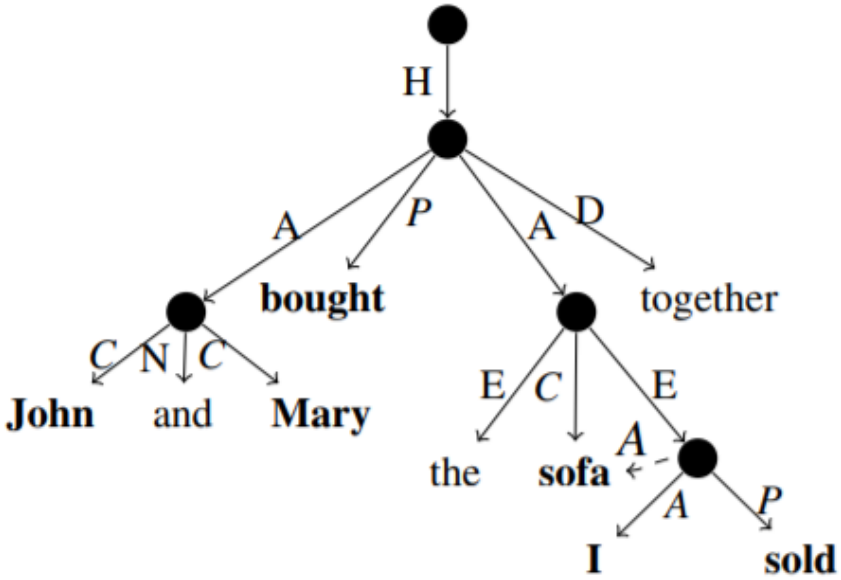
Scene.



UCCA: Core Words

The lowest semantic role label in the representation for each word indicates the most basic semantic role of a word.

Process, State, Participant or Center is identified as semantic core words



Word Matching

- Stemming algorithm: Match words between sentence
- Precision: count how many semantic core words in a candidate sentence can be matched to any semantic core words in the reference sentence
- Recall: calculate the matched proportion of semantic core words in reference sentence
- Calculate F1 score

$$P = \frac{\sum_i w(h_i) \cdot m(h_i)}{\sum_i w(h_i)}$$

$$R = \frac{\sum_i w(r_i) \cdot m(r_i)}{\sum_i w(r_i)}$$

$$F_1 = \frac{2P \cdot R}{P + R}$$

Penalty and Combinations

- Penalties:
  - The ratio between counts of scenes. $P_s = \frac{\min(S1, S2)}{\max(S1, S2)}$
  - The ratio between counts of nodes. $P_s = \frac{\min(N1, N2)}{\max(N1, N2)}$
  - The ratio between counts of edges towards core roles. $P_s = \frac{\min(E1, E2)}{\max(E1, E2)}$
  - Average word count of a sentence pair $Score = F_1 \cdot \exp(-\alpha_1 \cdot P_S - \alpha_2 \cdot P_N - \alpha_3 \cdot P_E - \alpha_4 \cdot Len)$
- The SWSS score is calculated independently, so it can be combined

$$SWSS \star Meteor = Meteor + \beta \cdot Score$$

Experiments

- Dataset:
  - WMT15: 4 language pairs and each has 500 sentence pairs.
  - WMT16: 6 language pairs an each has 560 sentence pairs
  - WMT17: 7 language pairs an each has 560 sentence pairs
- Parameter:

α_1	0.2	α_4	0.01
α_2	1	β	0.2
α_3	0.5	ω	0.5

- Metric Evaluation: Pearson Correlation

Experiment: Results

Base Model	BLEU		Meteor		Meteor++	
Method	None	+UCCA	None	+UCCA	None	+UCCA
WMT15						
cs-en	0.377	**0.418**	0.605	**0.609**	0.610	**0.613**
de-en	0.420	**0.464**	0.620	**0.638**	0.637	**0.651**
fi-en	0.378	**0.444**	0.645	**0.668**	0.661	**0.679**
ru-en	0.445	**0.477**	0.628	**0.634**	0.620	**0.629**
Average	0.405	**0.451**	0.624	**0.637**	0.632	**0.643**
WMT16						
cs-en	0.484	**0.508**	**0.649**	0.646	**0.656**	0.651
de-en	0.367	**0.394**	0.503	**0.520**	0.507	**0.523**
fi-en	0.325	**0.368**	0.537	**0.548**	0.557	**0.564**
ro-en	0.418	**0.451**	0.626	**0.633**	0.625	**0.632**
ru-en	0.377	**0.413**	0.574	**0.578**	0.583	**0.585**
tr-en	0.333	**0.401**	0.609	**0.638**	0.600	**0.628**
Average	0.384	**0.423**	0.583	**0.594**	0.588	**0.597**

Experiment: Results

Method	+UCCA	-repr	-len	None
WMT15				
cs-en	**0.609**	0.599	0.606	0.605
de-en	0.638	**0.641**	0.631	0.620
fi-en	**0.668**	0.662	0.666	0.645
ru-en	**0.634**	0.622	**0.634**	0.628
Average	**0.637**	0.631	0.634	0.624
WMT16				
cs-en	0.646	0.648	0.645	**0.649**
de-en	**0.520**	0.512	0.512	0.503
fi-en	**0.548**	0.541	0.543	0.537
ro-en	**0.633**	0.631	0.627	0.626
ru-en	0.578	**0.581**	0.564	0.574
tr-en	**0.638**	0.632	0.627	0.609
Average	**0.594**	0.591	0.586	0.583

Table 3: Results of ablation experiments. "+UCCA" is the complete SWSS model combined with Meteor, "-repr" means the penalties based on UCCA representation (P_S, P_N, P_E) are removed, "-len" means the length penalty is removed, and "None" contains only Meteor without SWSS.

Conclusion

Conclusion

- Semantic structures have valuable information for improving MT.
- Each time, there are more research for applying semantics in NLP field.
- Improve Semantic Parsers is fundamental for adding better information.

Thank you for your attention!

Visit Project at project.com

Come visit us at http://nlp.uniroma1.it/

SAPIENZA
UNIVERSITÀ DI ROMA

SAPIENZA NLP