Language Understanding

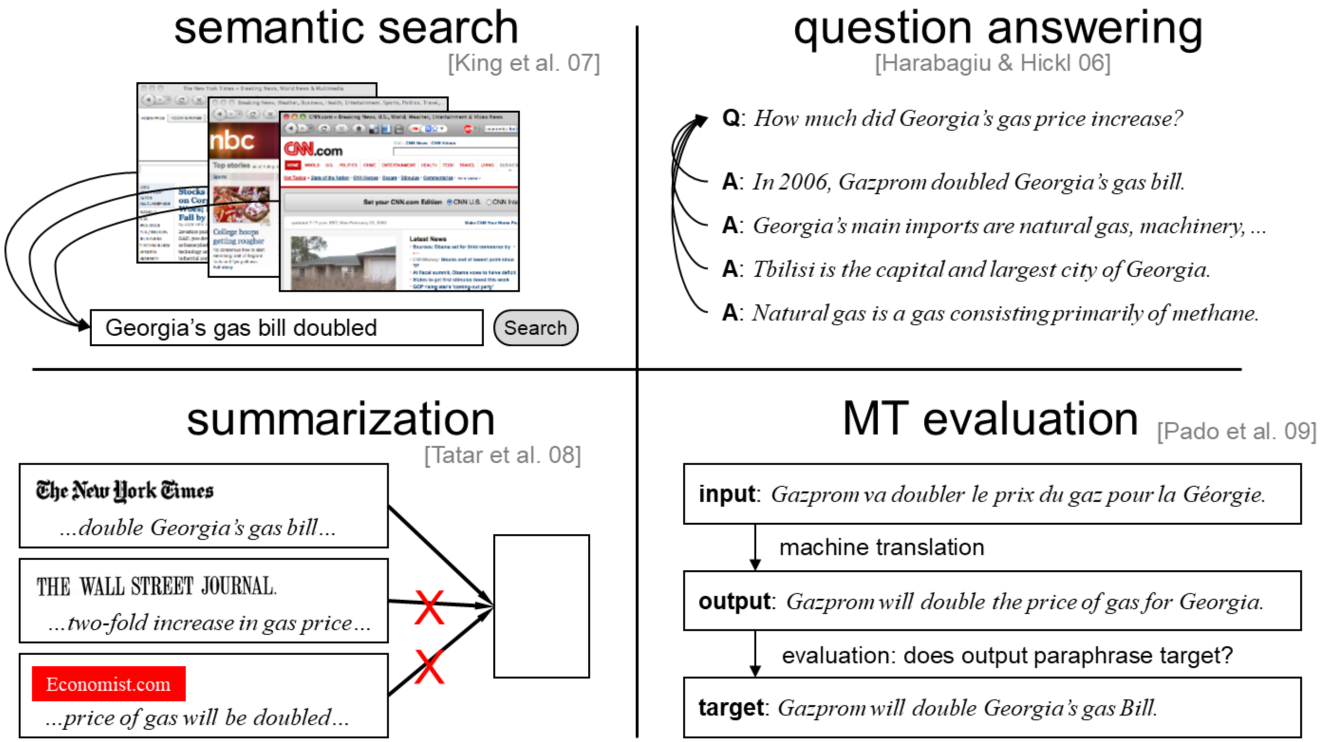
12 - Natural Language Inference

Hossein Zeinali

Natural Language Inference (NLI)

- Aka recognizing textual entailment (RTE)
- Does premise (Text or Sentence A) **P** justify an inference to hypothesis (Sentence B) **H**?
  - An informal, intuitive notion of inference: not strict logic
  - Emphasis on variability of linguistic expression
  - We say that P entails H if, typically, a human reading P would infer that H is most likely true.
- P Several airlines polled saw costs grow more than expected, even after adjusting for inflation.
- H Some of the companies in the poll reported cost increases.
  Yes
- Necessary to goal of natural language understanding (NLU)
- Many more immediate applications …

Applications of NLI



Judging Understanding with NLI

- To reliably perform well at NLI, your method for sentence understanding must be able to interpret and use the full range of phenomena about compositional semantics:
    - Lexical entailment (cat vs. animal, cat vs. dog)
    - Quantification (all, most, fewer than eight)
    - Lexical ambiguity and scope ambiguity (bank, ...)
    - Modality (might, should, ...)
    - Common sense background knowledge
    - ...
- Other tasks like sentiment analysis do not require models to deal with the full complexity of compositional semantics.

Entailments and Truth Conditions

- Most formal semantics research (and some semantic parsing research) deals with truth conditions.
- In this view understanding a sentence means (roughly) characterizing the set of situations in which that sentence is true.
- Truth-conditional semantics is strictly harder than NLI.

- If you know the truth conditions of two sentences, can you work out whether one entails the other?
- Can you work out whether one sentence entails another without knowing their truth conditions?

NLI Datasets

FraCaS Test Suite

- 346 NLI problems, constructed by semanticists in mid-90s
- 55% have single premise; remainder have 2 or more premises
- 3-way classification: entailment, contradiction, compatibility

- P: *Smith wrote a report in two hours.*
- H: *Smith spend more than two hours writing the report.*

  Label: No entailment

- P: No delegate finished the report.
- H: Some delegate finished the report on time.

  Label: No entailment

Recognizing Textual Entailment (RTE)

- Annual competitions, each with dev & test sets
- Some variation in format, but about 5000 NLI examples total
- Premises (texts) drawn from naturally occurring text, often long/complex
  - Expert-constructed hypotheses
- Balanced 2-way classification: entailment vs. non-entailment

- P: Cavern Club sessions paid the Beatles £15 evenings and £5 lunchtime.
- H: The Beatles perform at Cavern Club at lunchtime.

  Label: Entailment

The Stanford NLI Corpus (SNLI)

- Premises derived from image captions (Flickr 30k), hypotheses created by crowd-workers
- About 550,000 examples; first NLI corpus to see encouraging results with neural networks

- P: *A black race car starts up in front of a crowd of people..*
- H: *A man is driving down a lonely road.*

  Label: Contradiction

FarsTail

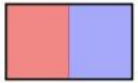
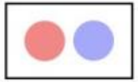
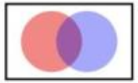
- A total of 10,367 samples are generated from a collection of 3,539 multiple-choice questions.
- The train, validation, and test portions include 7,266, 1,537, and 1,564 instances, respectively.
- E, C, and N stand for entailment, contradiction, and neutral classes, respectively.
- Example:

- فرضیه: امام خمینی (ره) عملیات طریق‌القدس را «فتح‌الفتوح» نامیدند.
  - عملیات طریق‌القدس را «فتح‌الفتوح » می‌نامند. E
  - نام دیگر عملیات طریق‌القدس، مرصاد است. C
  - عملیات طریق‌القدس در روز ۸ آذر ۱۳۶۰ آغاز شد. N

Entailment Relations

	$\frac{X \text{ is a couch}}{X \text{ is a sofa}}$	$\frac{X \text{ is a crow}}{X \text{ is a bird}}$	$\frac{X \text{ is a fish}}{X \text{ is a carp}}$	$\frac{X \text{ is a hippo}}{X \text{ is hungry}}$	$\frac{X \text{ is a man}}{X \text{ is a woman}}$
2-way RTE1,2,3	**Yes** entailment		**No** non-entailment		
3-way FraCaS, PARC, RTE4	**Yes** entailment		**Unknown** compatibility		**No** contradiction
containment Sánchez-Valencia	**P = Q** equivalence	**P < Q** forward entailment	**P > Q** reverse entailment	**P # Q** non-entailment	

MacCartney's Natural Logic Label Set

	X ≡ Y	equivalence	couch ≡ sofa
	X ⊏ Y	forward entailment	crow ⊏ bird
	X ⊐ Y	reverse entailment	European ⊐ French
	X ^ Y	negation	human ^ non-human
	X | Y	alternation	cat | dog
	X ‿ Y	cover	animal ‿ non-human
	X # Y	independence	hungry # hippo

Alignment for NLI

Alignment for NLI

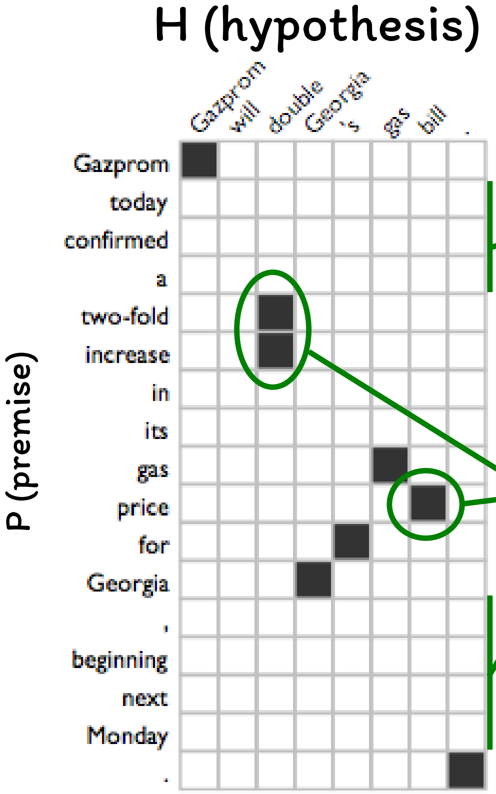
- Most approaches to NLI depends on a facility for *alignment*

P *Gazprom today confirmed a two-fold increase in its gas price for Georgia, beginning next Monday.*

H *Gazprom will double Georgia's gas bill.* yes

- Linking corresponding words & phrases in two sentences
- Alignment problem is familiar in machine translation (MT)

Alignment for NLI



unaligned content:
"deletions" from P

approximate match:
price ~ bill

phrase alignment:
two-fold increase ~ double

Approaches to NLI Alignment

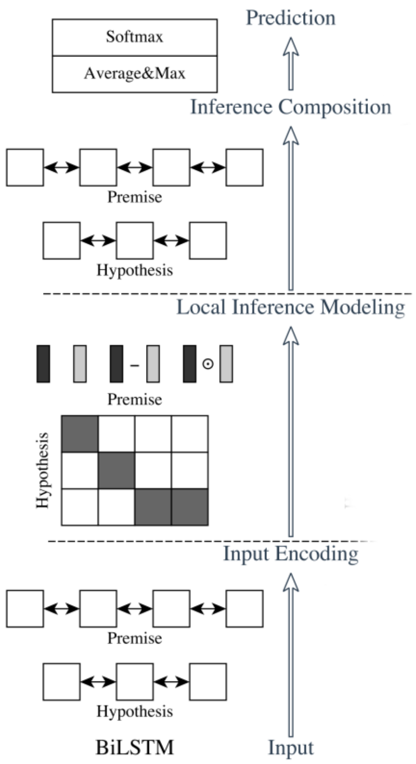
- Alignment addressed variously by current NLI systems
- In some approaches to NLI, alignments are *implicit*:
  - NLI via lexical overlap
  - NLI as proof search
- Other NLI systems make alignment step *explicit*:
  - Align first, then determine inferential validity
- What about using an MT aligner?
  - Alignment is familiar in MT, with extensive literature
  - Can tools & techniques of MT alignment transfer to NLI?
    - Not very well

Deep-Learning for NLI

Deep Learning Models for NLI

- Can roughly be organized in two categories:
- **Category I**: NLI models that explore both sentence representation and cross-sentence statistics (e.g., cross-sentence attention).
- **Category II**: NLI models that do not use cross-sentence information. (Sentence-vector-based models)
  - This category of models is of interest because NLI is a good test bed for learning representation for sentences.

Enhanced Sequential Inference Models (ESIM)



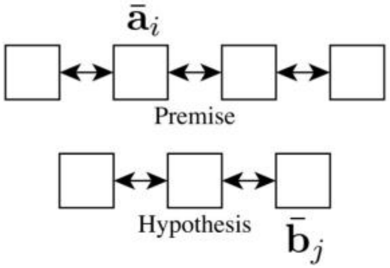
- **Layer 3**: Inference Composition/Aggregation
  - Perform composition/aggregation over local inference output to make the global judgement.
- **Layer 2**: Local Inference Modeling
  - Collect information to perform "local" inference between words or phrases.
- **Layer 1**: Input Encoding
  - ESIM uses BiLSTM, but different architectures can be used here, e.g., transformer-based, ELMo, densely connected CNN, tree-based models, etc.

Encoding Premise and Hypothesis

- For a premise sentence **a** and a hypothesis sentence **b**:

$$a = \left(a_1, \dots, a_{l_a}\right)$$
$$b = \left(b_1, \dots, b_{l_b}\right)$$

- We can apply different encoders (e.g., here BiLSTM):



$$\bar{\mathbf{a}}_i = \{\text{BiLSTM}(\mathbf{a})\}_i, i \in (1, \dots, \ell_a)$$
$$\bar{\mathbf{b}}_j = \{\text{BiLSTM}(\mathbf{b})\}_j, j \in (1, \dots, \ell_b)$$

where $\bar{a}_i$ denotes the output vector of BiLSTM at the position i of premise, which encodes word a_i and its context.

Source: Enhanced LSTM for Natural Language Inference (2017)

Local Inference Modeling

- The soft alignment layer computes the attention weights as the similarity of a hidden state of a premise and a hypothesis.
  - ESIM tried several more complicated functions of $e_{ij} = f\left(\bar{\boldsymbol{a}}_i, \bar{\boldsymbol{b}}_j\right)$, which did not further help.)

$$e_{ij} = \bar{\boldsymbol{a}}_i{}^T \bar{\boldsymbol{b}}_j$$

- The (cross-sentence) attention content is computed along both the premise-to-hypothesis and hypothesis-to-premise direction.

$$\tilde{\mathbf{a}}_i = \sum_{j=1}^{\ell_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_b} \exp(e_{ik})} \bar{\mathbf{b}}_j, \forall i \in [1, \ldots, \ell_a]$$

$$\tilde{\mathbf{b}}_j = \sum_{i=1}^{\ell_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{\ell_a} \exp(e_{kj})} \bar{\mathbf{a}}_i, \forall j \in [1, \ldots, \ell_b]$$

Source: Enhanced LSTM for Natural Language Inference (2017)

Local Inference Modeling

- With soft alignment ready, we can collect local inference information.
- Note that in various NLI models, the following heuristics have shown to work very well:

$$\mathbf{m_a} = [\bar{\mathbf{a}}; \tilde{\mathbf{a}}; \bar{\mathbf{a}} - \tilde{\mathbf{a}}; \bar{\mathbf{a}} \odot \tilde{\mathbf{a}}]$$

$$\mathbf{m}_b = [\bar{\mathbf{b}}; \tilde{\mathbf{b}}; \bar{\mathbf{b}} - \tilde{\mathbf{b}}; \bar{\mathbf{b}} \odot \tilde{\mathbf{b}}]$$

- For premise, at each time step i, concatenate $\bar{a}_i$ and $\tilde{a}_i$, together with their: (The same is performed for the hypothesis.)
  - Element-wise product
  - Element-wise difference

Source: Enhanced LSTM for Natural Language Inference (2017)

Inference Composition/Aggregation

- The next component is to perform composition/aggregation over local inference knowledge collected above.
- BiLSTM can be used here to perform "composition" over local inference:

$$\mathbf{v_a} = \text{BiLSTM}(\mathbf{m_a})$$

$$\mathbf{v_b} = \text{BiLSTM}(\mathbf{m_b})$$

- Then by concatenating the average and max-pooling of v_a and v_b, we obtain a vector v which is fed to a classifier.

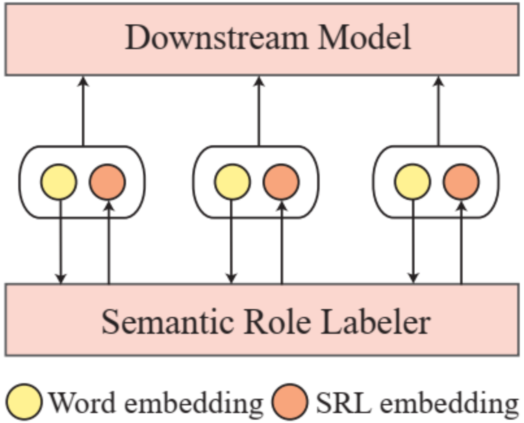
Source: Enhanced LSTM for Natural Language Inference (2017)

Performance of ESIM on SNLI

Model	#Para.	Train	Test
(1) Handcrafted features (Bowman et al., 2015)	-	99.7	78.2
(2) 300D LSTM encoders (Bowman et al., 2016)	3.0M	83.9	80.6
(3) 1024D pretrained GRU encoders (Vendrov et al., 2015)	15M	98.8	81.4
(4) 300D tree-based CNN encoders (Mou et al., 2016)	3.5M	83.3	82.1
(5) 300D SPINN-PI encoders (Bowman et al., 2016)	3.7M	89.2	83.2
(6) 600D BiLSTM intra-attention encoders (Liu et al., 2016)	2.8M	84.5	84.2
(7) 300D NSE encoders (Munkhdalai and Yu, 2016a)	3.0M	86.2	84.6
(8) 100D LSTM with attention (Rocktäschel et al., 2015)	250K	85.3	83.5
(9) 300D mLSTM (Wang and Jiang, 2016)	1.9M	92.0	86.1
(10) 450D LSTMN with deep attention fusion (Cheng et al., 2016)	3.4M	88.5	86.3
(11) 200D decomposable attention model (Parikh et al., 2016)	380K	89.5	86.3
(12) Intra-sentence attention + (11) (Parikh et al., 2016)	580K	90.5	86.8
(13) 300D NTI-SLSTM-LSTM (Munkhdalai and Yu, 2016b)	3.2M	88.5	87.3
(14) 300D re-read LSTM (Sha et al., 2016)	2.0M	90.7	87.5
(15) 300D btree-LSTM encoders (Paria et al., 2016)	2.0M	88.6	87.6
(16) 600D ESIM	4.3M	92.6	<u>88.0</u>

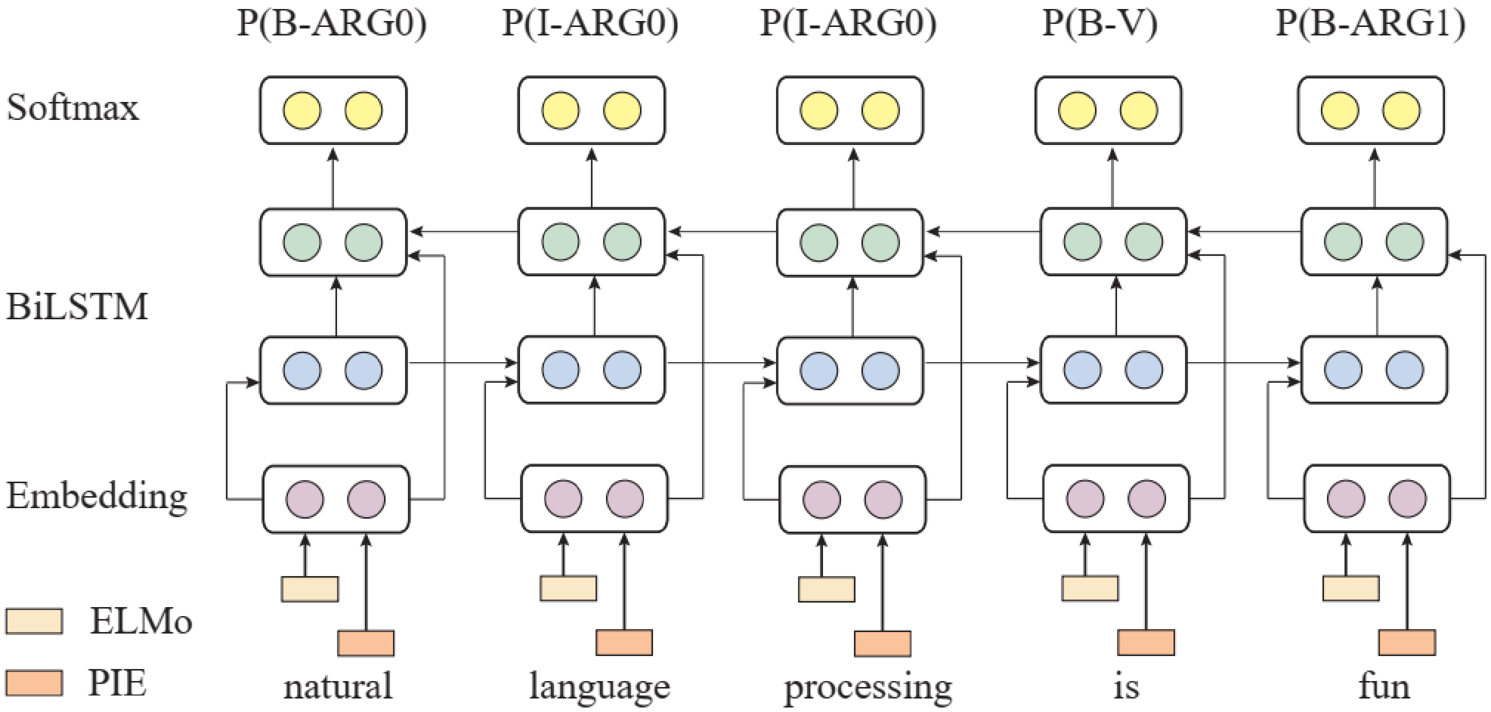
Models Enhanced with Semantic Roles

- Apply semantic role labeler to annotate the semantic tags for each token in the input sequence.
- Then the input sequence along with the corresponding semantic role labels is fed to downstream models.



Source: Explicit Contextual Semantics for Text Comprehension (2019)

Models Enhanced with Semantic Roles



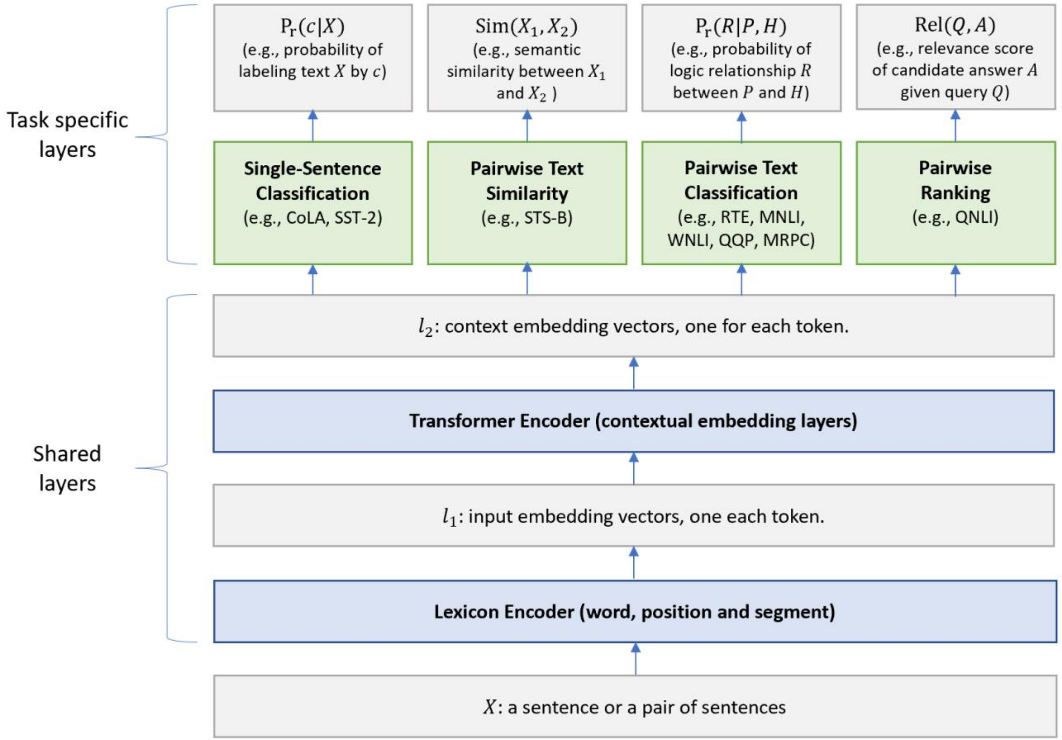
Predicate Indicator Embedding (PIE)

Models Enhanced with Semantic Roles

- Using Enhanced Sequential Inference Model (ESIM) as baseline
- Instead of Glove word embedding other embeddings were used

Model	Accuracy (%)
DIIN	88.0
DR-BiLSTM	88.5
CAFE	88.5
MAN	88.3
KIM	88.6
DMAN	88.8
ESIM + TreeLSTM	88.6
ESIM + ELMo	88.7
DCRCN	88.9
LM-Transformer	89.9
MT-DNN†	91.1
Baseline (ELMo)	88.4
+ SRL	89.1
Baseline ($\text{BERT}_{\text{BASE}}$)	89.2
+ SRL	89.6
Baseline ($\text{BERT}_{\text{LARGE}}$)	90.4
+ SRL	**91.3**

Multi-Task DNN for NLU



Source: Multi-Task Deep Neural Networks for Natural Language Understanding (2019)

Tasks Specific Layers

- **Single-Sentence Classification:**
  - Similar to BERT, use the contextual embedding of the token [CLS], as the input to a logistic regression with softmax

$$P_r(c|X) = \text{softmax}(\mathbf{W}_{SST}^\top \cdot \mathbf{x})$$

- **Text Similarity:**
  - Use the embedding of the token [CLS] to calculate dot product similarity with parameter vector w_{STS}

$$\text{Sim}(X_1, X_2) = \mathbf{w}_{STS}^\top \cdot \mathbf{x}$$

- **Relevance Ranking:**
  - Compute the relevance score the contextual embedding vector of [CLS] which is the semantic representation of a pair of question and its candidate answer (Q, A)
  - $$\text{Rel}(Q, A) = g(\mathbf{w}_{QNLI}^\top \cdot \mathbf{x})$$

Source: Multi-Task Deep Neural Networks for Natural Language Understanding (2019)

Results on SNLI and SciTail

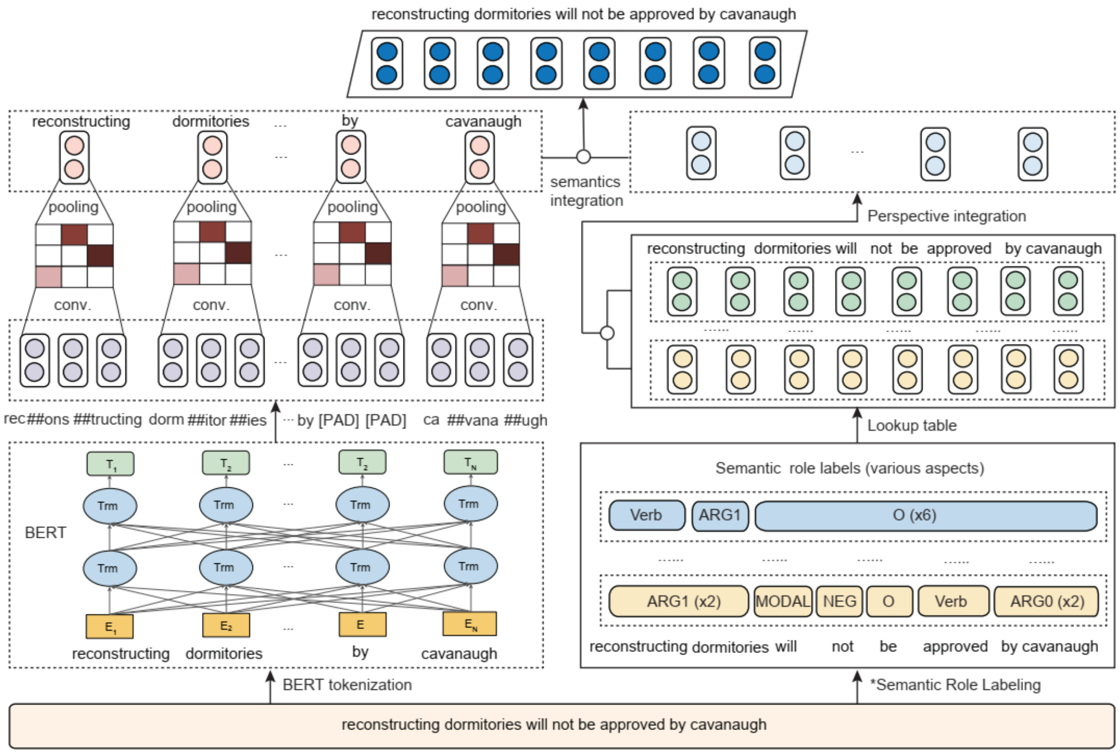
- We used the pre-trained BERT to initialize its shared layers, refined the model via MTL on all GLUE tasks, and fine-tuned the model for each GLUE task using task-specific data.

Model	0.1%	1%	10%	100%
SNLI Dataset (Dev Accuracy%)				
#Training Data	549	5,493	54,936	549,367
BERT	52.5	78.1	86.7	91.0
MT-DNN	82.1	85.2	88.4	91.5
SciTail Dataset (Dev Accuracy%)				
#Training Data	23	235	2,359	23,596
BERT	51.2	82.2	90.5	94.3
MT-DNN	81.9	88.3	91.1	95.7

Results on SNLI and SciTail

Model	Dev	Test
SNLI Dataset (Accuracy%)		
GPT (Radford et al., 2018)	-	89.9
Kim et al. (2018)*	-	90.1
$\text{BERT}_{\text{BASE}}$	91.0	90.8
$\text{MT-DNN}_{\text{BASE}}$	91.5	91.1
$\text{BERT}_{\text{LARGE}}$	91.7	91.0
$\text{MT-DNN}_{\text{LARGE}}$	**92.2**	**91.6**
SciTail Dataset (Accuracy%)		
GPT (Radford et al., 2018)*	-	88.3
$\text{BERT}_{\text{BASE}}$	94.3	92.0
$\text{MT-DNN}_{\text{BASE}}$	95.7	94.1
$\text{BERT}_{\text{LARGE}}$	95.7	94.4
$\text{MT-DNN}_{\text{LARGE}}$	**96.3**	**95.0**

Semantics-aware BERT



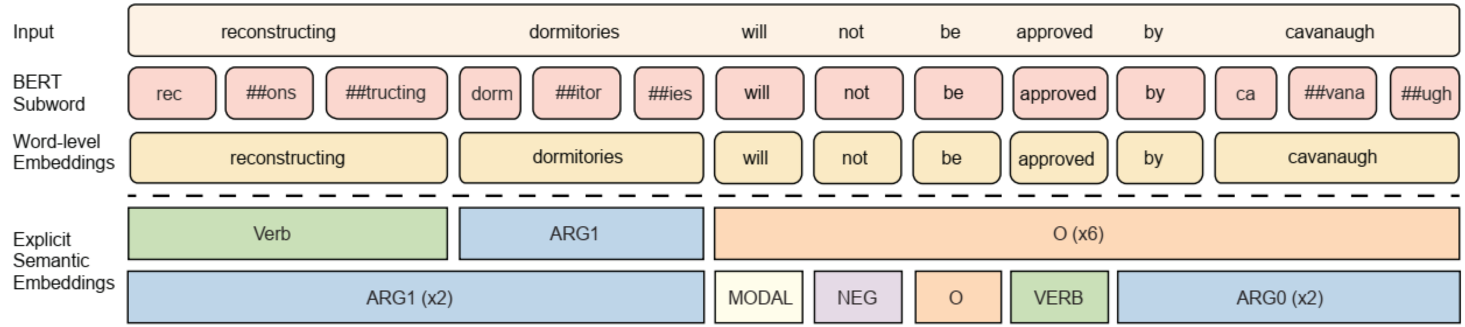
Source: Semantics-aware BERT for Language Understanding (2020)

Semantics-aware BERT

- The m sequences of semantic label representation are concatenated and fed to a fully connected layer to obtain the refined joint representation:

$$e'(L_i) = W_2\,[e(t_1), e(t_2), \ldots, e(t_m)] + b_2,$$
$$e^t = \{e'(L_1), \ldots, e'(L_n)\},$$

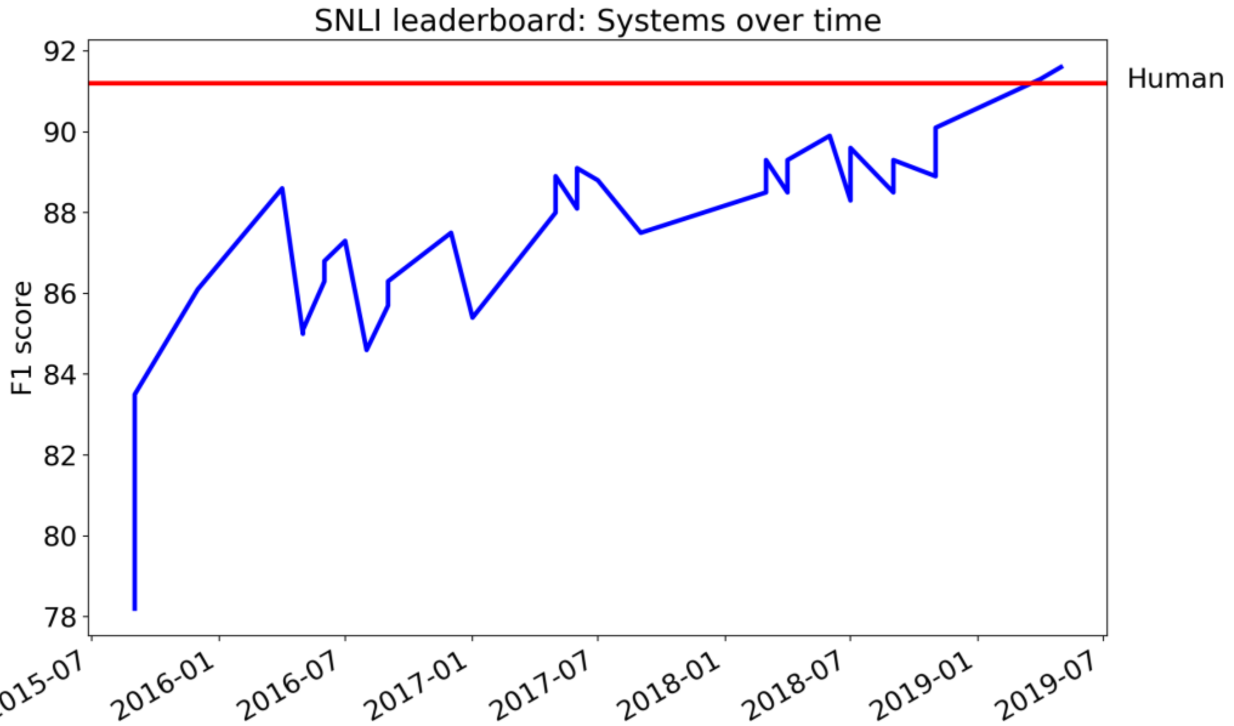
- The representation of sub-words for each word are grouped and a convolutional neural network (CNN) with a max pooling is used to obtain the representation in word-level.



Results on SNLI

Model	Dev	Test
In literature		
DRCN (Kim et al. 2018)	-	90.1
SJRC (Zhang et al. 2019)	-	91.3
MT-DNN (Liu et al. 2019)†	92.2	91.6
Our implementation		
$\text{BERT}_{\text{BASE}}$	90.8	90.7
$\text{SemBERT}_{\text{BASE}}$	91.2	91.0
$\text{BERT}_{\text{LARGE}}$	91.3	91.1
$\text{SemBERT}_{\text{LARGE}}$	92.0	91.6
BERT_{WWM}	92.1	91.6
$\text{SemBERT}_{\text{WWM}}$	92.2	91.9

Progress on SNLI



Thanks for your attention

References and IP Notice

- Some slides from Bill MacCartney's slides on NLI.
- Some slides from Sam Bowman's slides on DL for NLI.
- Some graphics from Slidesgo template