NLP @ University of Washington

Gordon Moon and Jie Zhao

NLP @ UW Introduction

Research Projects:

- **Code with Natural Language**
- **Interactive Learning for Semantic Parsing**
- **Language and Vision**
- **Language generation**
- **Social Science Applications: Political Science, Sociology, Psychology & more**
- Language Grounding in Robotics
- Multilingual Representations and Parsing
- Relation and Entity Extraction
- Detecting and Extracting Events
- Tools and Resource Development

Publications:

- 11 publications in 2016 so far

People:

- 5 Professors
- 8 Adjunct Faculty
- 16 Ph.D. Students

Faculty Members and Research Areas

Yejin Choi

- Language Grounding with Vision
- Knowledge Extraction
- Instructions to Action Diagrams
- Situated Language Generation, Conversation, and Storytelling
- Connotation and Intention

Pedro Domingos

- Statistical Relational Learning
- Tractable Deep Learning
- Machine Reading
- Collective Knowledge Bases
- Large-Scale Machine Learning

Noah Smith

- Parsing sentences in different languages into syntactic representations
- Semantic representations
- Cross-cutting techniques for unsupervised language learning
- Automatic translation

Daniel S. Weld

- Decision-theoretic crowdsourcing
- Artificial intelligence
- Relation Extraction
- Human-Computer Interaction with an emphasis on building intelligent user interfaces

Luke Zettlemoyer

- Designing learning algorithms for recovering representations of the meaning of natural language text
- Intersections of natural language processing
- Machine Learning
- Decision making under uncertainty

Oren Etzioni

- Artificial Intelligence
- Web Search
- As of January 1, 2014 he becomes the CEO of the Allen Institute for AI (AI2)

Document-level Sentiment Inference with Social, Faction, and Discourse Context

E. Choi, H. Rashkin, L. Zettlemoyer, Y. Choi, Conference of the Association for Computational Linguistics (ACL), 2016.

Tasks

Document-level sentiment inference to predict directed opinions (who feels positively or negatively towards whom) for all entities mentioned in a text

Russia criticized *Belarus* for permitting *Georgian* President *Mikheil Saakhashvili* to appear on *Belorussian* television. “The appearance was an unfriendly step towards *Russia*,” the speaker of *Russian* parliament *Boris Gryzlov* said. . . . *Saakhashvili* announced Thursday that *he* did not understand *Russia*’s claims. *Moscow* refused to have any business with *Georgia*’s president after the armed conflict in 2008 . . .

A Document-level Sentiment Model

- Given a news document d, and named entities $e_1, \ldots, e_n$ in d, where each entity e_i has mentions $m_{i1}, \ldots, m_{ik}$, the task is to decide directed sentiment between all pairs of entities

- Predicting the directed sentiment from e_i to e_j at the document level, i.e., sent $(e_i \rightarrow e_j) \in$ {positive, unbiased, negative}, for all e_i, $e_j \in$ d where $i \neq j$, assuming that sentiment is consistent within the document

- **Integer Linear Programming (ILP)** model jointly combines three complementary types of evidence:
  - Entity-pair sentiment classification
  - Template-based faction extraction
  - Sentiment dynamics in social groups motivated by social science theories
    - **Homophily** (Lazarsfeld and Merton, 1954)
    - Triadic social dynamics with **social balance theory** (Heider, 1946)
    - Dyadic social constraints – The likely **reciprocity of opinions** (Gouldner, 1960)

A Document-level Sentiment Model

- Document-level ILP is used for easily incorporating different types of soft social constraints, $\boldsymbol{\varphi_{fact}}$ and $\boldsymbol{\varphi_{social}}$
  - $\boldsymbol{\varphi_{fact}}$ models the fact that entities in supportive social relations tend to share similar sentiment towards others, and are often positive towards each other (Lazarsfeld and Merton, 1954)
  - $\boldsymbol{\varphi_{social}}$ models social balance in an interpersonal network where entities on positive terms have similar opinions towards other entities and those on negative terms have opposing opinions

    (Heider, 1946)
  - $\boldsymbol{\varphi_{social}}$ models reciprocity of sentiment, social stability (Gouldner, 1960)

- The ILP is solved by maximizing

$$\boldsymbol{F = \varphi_{social} + \varphi_{fact} + \sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_{ij}}$$

where pairwise potentials φ_{ij} defined as

$$\varphi_{ij} = \theta_{pos_{ij}} \cdot pos_{ij} + \theta_{neg_{ij}} \cdot neg_{ij} + \theta_{neu_{ij}} \cdot neu_{ij}$$

Results

	Development Set (KBP)						KBP						MPQA					
	Positive			Negative			Positive			Negative			Positive			Negative		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
KM_Gold	90.9	2.5	4.8	93.8	8.6	15.8	93.9	4.3	8.3	93.5	6.6	12.4	61.5	1.3	2.5	90.0	5.2	9.8
Random	16.6	13.1	14.7	4.9	4.0	4.4	13.3	12.7	13.0	10.1	6.9	8.2	10.9	15.4	12.8	8.9	6.7	7.7
Sentence	**60.0**	16.3	25.7	21.7	**43.1**	28.8	40.9	20.6	27.4	21.0	31.4	25.2	18.9	3.7	6.2	16.7	18.2	17.4
Pairwise	47.3	36.9	41.4	25.6	36.8	30.2	36.2	**35.5**	35.9	27.6	**41.2**	33.1	**28.7**	23.0	25.6	**23.2**	16.3	19.2
Global	58.2	**37.9**	**45.9**	**37.2**	35.1	**36.1**	**45.5**	32.7	**38.1**	**34.6**	36.8	**35.7**	25.2	**29.3**	**27.1**	17.6	**24.4**	**20.4**

Table 7: Performance on the evaluation datasets: including implicit and explicit sentiment.

	Positive			Negative		
	P	R	F1	P	R	F1
ILP base	56.7	25.2	34.9	36.9	27.6	31.6
+ Reci.	53.5	30.0	38.4	33.9	33.9	33.9
+ Balance	49.6	30.4	37.7	32.0	32.8	32.4
+ Faction	58.9	30.2	39.9	37.6	33.9	35.6

Table 8: ILP constraints ablation study.

Summarizing Source Code using a Neural Attention Model

S. Iyer, Y. Konstas, A. Cheung, L. Zettlemoyer, Annual Meeting of the Association for Computational Linguistics (ACL), 2016.

Tasks

Let U_C be the set of all code snippets.
Let U_N be the set of all summaries in NL.
For a training corpus with J code snippet and summary pairs (c_j, n_j), $1 \leq j \leq J$, $c_j \in U_c$, $n_j \in U_N$, define the following two tasks:

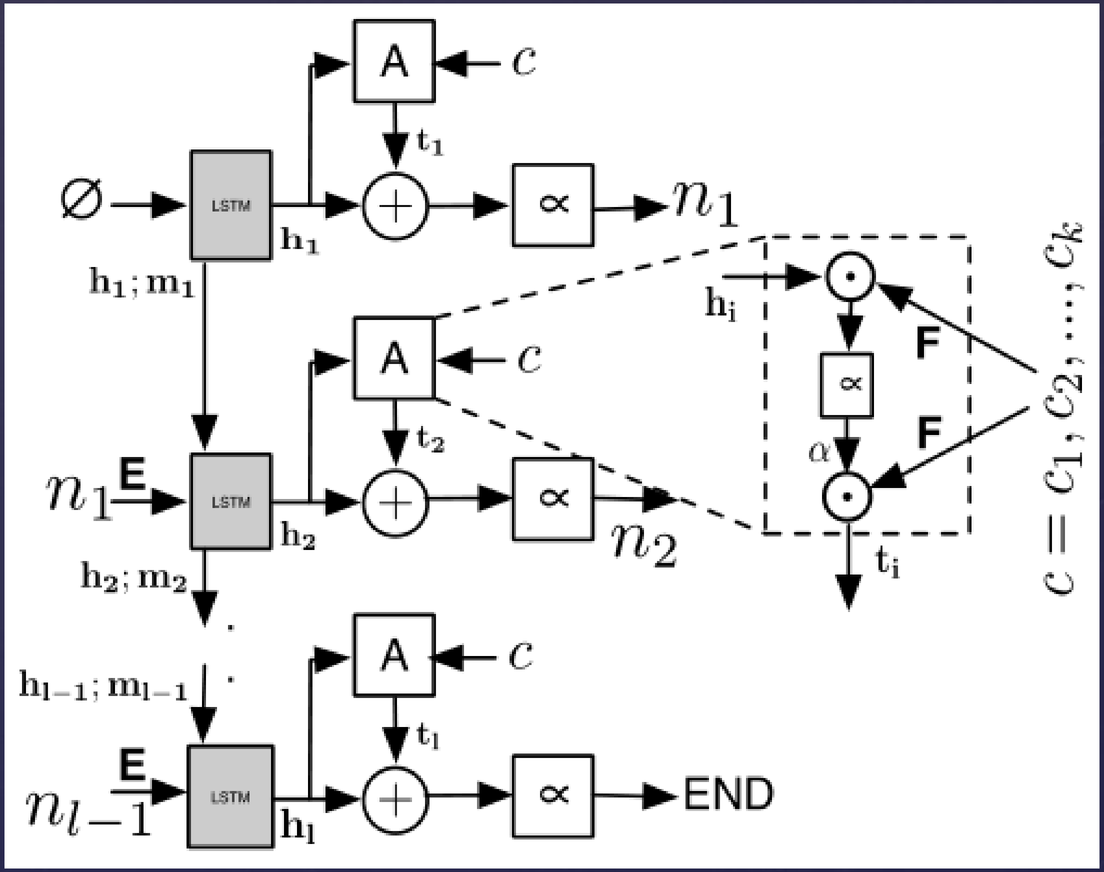
GEN For a given code snippet $c \in U_C$, the goal is to produce a NL sentence $n^* \in U_N$ that maximizes some scoring function $s \in (U_C \times U_N \to \mathbb{R})$:

$$n^* = \text{argmax}_n s(c, n).$$

RET Given a NL question $n \in U_N$, retrieve the highest scoring code snippet c_j^* from the training corpus, given a NL question $n \in U_N$ such that:

$$c_j^* = \text{argmax}_{c_j} s(c_j, n),\ 1 \leq j \leq J.$$

CODE-NN Model



CODE-NN Model

Scoring function s:

$$s(c, n) = \prod_{i=1}^{\ell} p(n_i | n_1, \ldots, n_{i-1})$$

with

- $p(n_i | n_1, \ldots, n_{i-1}) \propto W \tanh(W_1 h_i + W_2 t_i)$.
- H is the embedding dimensionality of the summaries.
- $W \in \mathbb{R}^{|N| \times H}$.
- $W_1, W_2 \in \mathbb{R}^{H \times H}$.
- h_i represents the hidden state of the LSTM cell at the current time step.
- t_i is the contribution from the attention model on the source code.

CODE-NN Model

h_i is computed as follows:

$$m_i; h_i = f(n_{i-1}E, m_{i-1}, h_{i-1}; \sigma).$$

where

- n_{i-1} is the previously generated word.
- m_{i-1} is the previous LSTM cell state.
- h_{i-1} is the previous LSTM hidden state.
- $E \in \mathbb{R}^{|N| \times H}$ is a word embedding matrix for the summaries.

Compute t_i, the contribution of the attention model:

$$t_i = \sum_{j=1}^{k} \alpha_{i,j} \cdot c_j \mathbf{F}$$

where

$$\alpha_{i,j} = \frac{\exp(h_i^T c_j \mathbf{F})}{\sum_{j=1}^{k} \exp(h_j^T c_j \mathbf{F})}$$

and $\mathbf{F} \in \mathbb{R}^{|C| \times H}$ is a token embedding matrix.

Attention Weights

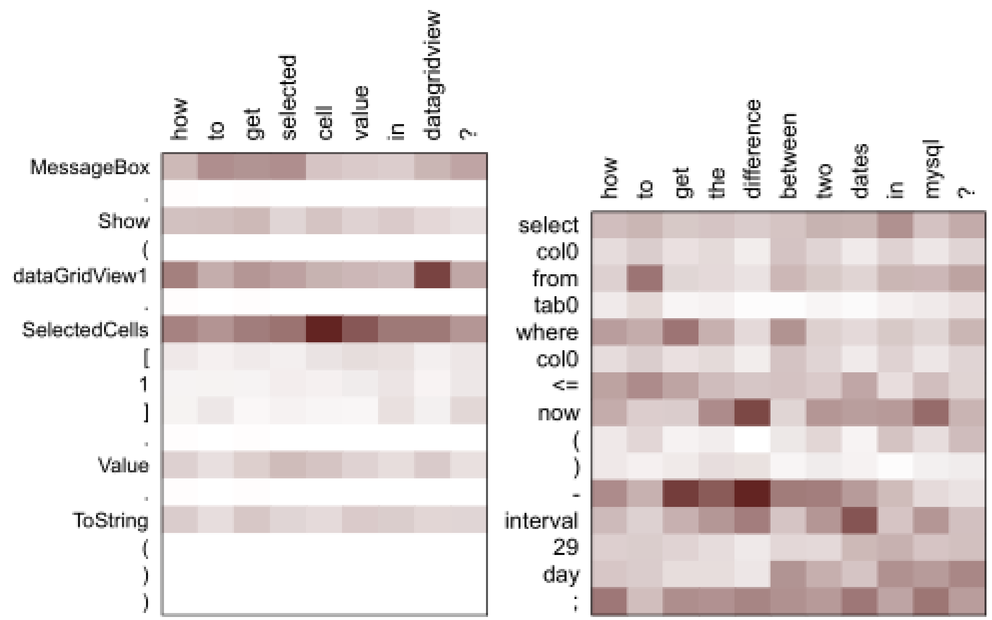


Figure 3: Heatmap of attention weights $\alpha_{i,j}$ for example C# (left) and SQL (right) code snippets. The model learns to align key summary words (like cell) with the corresponding tokens in the input (SelectedCells).

Experiments (GEN Task Results)

	Model	METEOR	BLEU-4
C#	IR	7.9 (6.1)	13.7 (12.6)
	MOSES	9.1 (9.7)	11.6 (11.5)
	SUM-NN	10.6 (10.3)	19.3 (18.2)
	CODE-NN	**12.3 (13.4)**	**20.5 (20.4)**
SQL	IR	6.3 (8.0)	13.5 (13.0)
	MOSES	8.3 (9.7)	15.4 (15.9)
	SUM-NN	6.4 (8.7)	13.3 (14.2)
	CODE-NN	**10.9 (14.0)**	**18.4 (17.0)**

Table 3: Performance on EVAL for the GEN task. Performance on DEV is indicated in parentheses.

	Model	Naturalness	Informativeness
C#	IR	3.42	2.25
	MOSES	1.41	2.42
	SUM-NN	**4.61***	1.99
	CODE-NN	**4.48**	**2.83**
SQL	IR	3.21	2.58
	MOSES	2.80	2.54
	SUM-NN	4.44	2.75
	CODE-NN	**4.54**	**3.12**

Table 4: Naturalness and Informativeness measures of model outputs. Stat. sig. between CODE-NN and others is computed with a 2-tailed Student's t-test; $p < 0.05$ except for *.

Experiments (RET Task Results)

	Model	MRR
C#	RET-IR	0.42 ± 0.02 (0.44 ± 0.01)
	CODE-NN	$\mathbf{0.58 \pm 0.01}$ ($\mathbf{0.66 \pm 0.02}$)
SQL	RET-IR	0.28 ± 0.01(0.4 ± 0.01)
	CODE-NN	$\mathbf{0.44 \pm 0.01}$ ($\mathbf{0.54 \pm 0.02}$)

Table 5: MRR for the RET task. Dev set results in parentheses.

	Model	MRR
L to C	Allamanis	0.182 ± 0.009
	CODE-NN	$\mathbf{0.590 \pm 0.044}$
C to L	Allamanis	0.434 ± 0.003
	CODE-NN	$\mathbf{0.461 \pm 0.046}$

Table 6: MRR values for the Language to Code (L to C) and the Code to Language (C to L) tasks using the C# dataset of Allamanis et al. (2015b)

Thank you!