NNLG

Neural Natural Language Generation

Yannis Konstas

Joint work with
Srinivasan Iyer, Mark Yatskar, Rik Koncel-Kedziorcki, Li Zilles,
Luke Zettlemoyer, Yejin Choi, Hannaneh Hajishirzi
NLG Pipeline

Input → Content Planning → Sentence Planning → Surface Realisation → Text

Communicative Goal

Content Selection
Document Planning
Content Planning
Lexicalization
Reordering/Linearization
Splitting/Aggregation
NLG Pipeline

Input

- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multiple Sources

Content Planning

- Content Selection
- Document Planning

Sentence Planning

- Lexicalization
- Reordering/Linearization
- Splitting/Aggregation

Surface Realisation

Text

Communicative Goal
NLG Pipeline

Input
- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multiple Sources

Content Planning
- Content Selection
- Document Planning

Sentence Planning
- Lexicalization
- Reordering/Linearization
- Splitting/Aggregation

Surface Realisation
- Single utterance
- Single (complex) sentence
- Multiple sentences
- Multiple paragraphs

Text
NLG Pipeline

- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multiple Sources

Communicative Goal

Content Planning

Content Selection
Document Planning

Sentence Planning

Lexicalization
Reordering/Linearization
Splitting/Aggregation

Surface Realisation

- Single utterance
- Single (complex) sentence
- Multiple sentences
- Multiple paragraphs

Text
NLG Pipeline

Input

- Records / Fields / Values
- Source Code
- Predicate-Argument Structure
- Algebra equation
- Text / Script
- Multiple Sources

Content Planning

- Communicative Goal

Content Selection

Document Planning

Sentence Planning

- Lexicalization
- Reordering/Linearization
- Splitting/Aggregation

Surface Realisation

Text

- Single utterance
- Single (complex) sentence
- Multiple sentences
- Multiple paragraphs
NLG is everywhere

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

Concept-to-Text Generation

Input: Machine-generated Representation

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

Concept-to-Text Generation

*Input*: Machine-generated Representation

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

Concept-to-Text Generation

*Input*: Machine-generated Representation

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

Concept-to-Text Generation

*Input*: Machine-generated Representation

<table>
<thead>
<tr>
<th>source</th>
<th>block:</th>
<th>hk</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>block:</td>
<td>ms</td>
</tr>
<tr>
<td>pos</td>
<td>RP:</td>
<td>W</td>
</tr>
<tr>
<td></td>
<td>scale:</td>
<td>small</td>
</tr>
</tbody>
</table>

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

Concept-to-Text Generation

*Input*: Machine-generated Representation

<table>
<thead>
<tr>
<th>source</th>
<th>target</th>
<th>pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>block: hk</td>
<td>block: ms</td>
<td>RP: W, scale: small</td>
</tr>
</tbody>
</table>

Place the heineken block west of the mercedes block.

(A Global Model for Concept-to-Text Generation. Konstas and Lapata, JAIR 2013.)
NLG is everywhere

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
NLG is everywhere

Code-to-Text Generation

Input: Source Code

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
NLG is everywhere

**Code-to-Text Generation**

*Input*: Source Code

CODE-NN

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
NLG is everywhere

Code-to-Text Generation

Input: Source Code

CODE-NN

```csharp
public int TextWidth (string text) {
    TextBlock t = new TextBlock();
    t.Text = text;
    return (int) Math.Ceiling(t.ActualWidth);
}
```

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
NLG is everywhere

Code-to-Text Generation

Input: Source Code

CODE-NN

```csharp
public int TextWidth (string text) {
  TextBlock t = new TextBlock();
  t.Text = text;
  return (int) Math.Ceiling(t.ActualWidth);
}
```

Get rendered width of string rounded up to the nearest integer.

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
NLG is everywhere

Meaning Representation Generation

**Input:** Predicate - Argument Structure

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)
NLG is everywhere

Meaning Representation Generation

**Input:** Predicate - Argument Structure

I knew a planet that was inhabited by a lazy man.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)
NLG is everywhere

Meaning Representation Generation

Input: Predicate - Argument Structure

I knew a planet that was inhabited by a lazy man.

I have known a planet that was inhabited by a lazy man.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)
NLG is everywhere

Meaning Representation Generation

Input: Predicate - Argument Structure

I knew a planet that was inhabited by a lazy man.

I have known a planet that was inhabited by a lazy man.

There is a lazy man who inhabited a planet I know about.

(Flanigan et al, NAACL 2016, Pourdamaghani and Knight, INLG 2016, Song et al, EMNLP 2016.)
NLG is everywhere

Instructional Text Generation

Input: Goal Cue - Bag of concepts
NLG is everywhere

Instructional Text Generation

Input: Goal Cue - Bag of concepts

Spanakopita
(Greek Spinach Pie)

Ingredients

<table>
<thead>
<tr>
<th>3 tbsp olive oil</th>
<th>2 eggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 large onion, chopped</td>
<td>1/2 cup ricotta cheese</td>
</tr>
<tr>
<td>1 bunch green onions, chopped</td>
<td>1 cup feta cheese</td>
</tr>
<tr>
<td>2 cloves garlic, minced</td>
<td>8 sheets filo dough</td>
</tr>
<tr>
<td>2 pounds spinach</td>
<td>1/4 cup olive oil</td>
</tr>
<tr>
<td>1/2 cup chopped fresh parsley</td>
<td></td>
</tr>
</tbody>
</table>
NLG is everywhere

Instructional Text Generation

Input: Goal Cue - Bag of concepts

Spanakopita (Greek Spinach Pie)

Ingredients

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3 tbsp</td>
<td>olive oil</td>
</tr>
<tr>
<td>1 large</td>
<td>onion, chopped</td>
</tr>
<tr>
<td>1 bunch</td>
<td>green onions, chopped</td>
</tr>
<tr>
<td>2 cloves</td>
<td>garlic, minced</td>
</tr>
<tr>
<td>2 pounds</td>
<td>spinach</td>
</tr>
<tr>
<td>1/2 cup</td>
<td>chopped fresh parsley</td>
</tr>
<tr>
<td>2 eggs</td>
<td></td>
</tr>
<tr>
<td>1/2 cup</td>
<td>ricotta cheese</td>
</tr>
<tr>
<td>1 cup</td>
<td>feta cheese</td>
</tr>
<tr>
<td>8 sheets</td>
<td>phyllo dough</td>
</tr>
<tr>
<td>1/4 cup</td>
<td>olive oil</td>
</tr>
</tbody>
</table>

Preheat oven to 350 degrees F (175 degrees C). Lightly oil a 9x9 inch square baking pan.

Heat 3 tablespoons olive oil in a large skillet over medium heat. Saute onion, green onions and garlic, until soft and lightly browned. Stir in spinach and parsley, and continue to saute until spinach is limp, about 2 minutes. Remove from heat and set aside to cool.

In a medium bowl, mix together eggs, ricotta, and feta. Stir in spinach mixture. Lay 1 sheet of phyllo dough in prepared baking pan, and brush lightly with olive oil. Lay another sheet of phyllo dough on top, brush with olive oil, and repeat process with two more sheets of phyllo. The sheets will overlap the pan. Spread spinach and cheese mixture into pan and fold overhanging dough over filling. Brush with oil, then layer remaining 4 sheets of phyllo dough, brushing each with oil. Tuck overhanging dough into pan to seal filling.

Bake in preheated oven for 30 to 40 minutes, until golden brown. Cut into squares and serve while hot.

(Globally Coherent Text Generation with Neural Checklist Models. Kiddon et al, EMNLP 2016.)
NLG is everywhere

Storytelling Generation

*Input*: Script - Text - N/A
Jim was obsessed with super heroes. His sister told him if he tied a sheet on his back he could fly. She convinced Jim to climb the ladder to the roof and jump off. When he got up there he felt like he was superman.
Jim was obsessed with super heroes. His sister told him if he tied a sheet on his back he could fly. She convinced Jim to climb the ladder to the roof and jump off. When he got up there he felt like he was superman.

He ended up having a great time!
Jim was obsessed with super heroes. His sister told him if he tied a sheet on his back he could fly. She convinced Jim to climb the ladder to the roof and jump off. When he got up there he felt like he was superman.

He ended up having a great time!

Jim broke his arm and his sister was grounded for a year.
NLG is everywhere

Storytelling Generation

*Input*: Equation + Theme

(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)
NLG is everywhere

Storytelling Generation

*Input*: Equation + Theme

504 + x = 639 + STAR WARS

(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)
NLG is everywhere

Storytelling Generation

Input: Equation + Theme

Luke Skywalker has 639 blasters. Leia has 504 blasters. How many more blasters does Luke Skywalker have than Leia?

504 + x = 639

Luke Skywalker has 639 blasters. Leia has 504 blasters. How many more blasters does Luke Skywalker have than Leia?

(Koncel-Kedziorski, Konstas, Zettlemoyer, Hajishirzi. A Theme-Rewriting Approach for Generating Algebra Word Problems. EMNLP 2016.)
NNLG Framework

input
NNLG Framework

input ➔ Encoder
The A knew a planet...inhabit was...
The A… knew planet… a planet man… inhabit inhabited was…

think :arg0 you :arg1 quest
what
do
you
think
?
</s>
Encoding
Encoding

Bag of Words

CODE-NN

```
SELECT max(marks) FROM stud_records WHERE marks < 
(SELECT max(marks) FROM stud_records);
```
Encoding

Bag of Words

SELECT max(marks) FROM stud_records WHERE marks < (SELECT max(marks) FROM stud_records);

anonymization

SELECT max(col0) FROM tab0 WHERE col0 < (SELECT max(col1) FROM tab1);

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
Encoding

Bag of Words

SELECT max(marks) FROM stud_records WHERE marks < (SELECT max(marks) FROM stud_records);

anonymization

SELECT max(col0) FROM tab0 WHERE col0 < (SELECT max(col1) FROM tab1);

(Summarizing Source Code using a Neural Attention Model. Iyer, Konstas, Cheung, Zettlemoyer. ACL 2016.)
Encoding

Linearize —> RNN encoding

AMR Generation

ARG0

ARG1-of

ARG0

ARG1

know

planet

inhabit

man

mod

lazy

I
Encoding

Linearize $\rightarrow$ RNN encoding

ARG0
know
ARG1
planet
ARG1-of
inhabit
ARG0
man
mod
lazy

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy

AMR Generation
Encoding

Linearize —> RNN encoding

AMR Generation

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Linearize —> RNN encoding

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Linearize —> RNN encoding

AMR Generation

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Linearize —> RNN encoding

AMR Generation

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Linearize —> RNN encoding

AMR Generation

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Linearize —> RNN encoding

AMR Generation

know ARG0 I ARG1 planet ARG1-of inhabit ARG0 man mod lazy
Encoding

Hierarchical RNN encoding

Storytelling Generation
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.
Jim was obsessed with superheroes. His sister told him if he tied a sheet on his back he could fly. She convinced Jim to climb the ladder to the roof and jump off. When he got up there he felt like he was superman.
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was Superman.
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.
Jim was obsessed with superheroes.
His sister told him if he tied a sheet on his back he could fly.
She convinced Jim to climb the ladder to the roof and jump off.
When he got up there he felt like he was superman.
Decoding

Beam search (Left-to-Right)

$h_N^{(s)}$
Decoding

Beam search (Left-to-Right)
Decoding

Beam search (Left-to-Right)
Decoding

Beam search (Left-to-Right)

I
The
A
...

know
knew
planet
...

\[ \text{w}_{11}: I \]

\[ \text{w}_{12}: \text{The} \]

\[ \text{w}_{13}: \text{Man} \]

\[ \text{w}_{14}: \text{Tree} \]

\[ \ldots \]
Decoding

Beam search (Left-to-Right)

I

The

A

know

knew

planet

... ... ... ... ... ... ... ... ...

know

ARG0

I

ARG1

planet

w_{11}: I

w_{21}: I know

w_{12}: The

w_{22}: I knew

w_{13}: Man

w_{23}: The planet

w_{14}: Tree

w_{24}: Man planet

... ... ... ... ... ... ... ... ... ...
Decoding

Beam search (Left-to-Right)

The planet knew a man.

I know I knew the planet.

ARG0 ARG1

I know The

w11: I   w21: I know

w12: The   w22: I knew

w13: Man   w23: The planet

w14: Tree   w24: Man planet

...
Decoding

Beam search (Left-to-Right)

I know a planet

A man

w11: I
w21: I knew
w41: I know a planet

w12: The
w22: I knew
w42: I knew planets that

w13: Man
w23: The planet
w43: The planet I knew

w14: Tree
w24: Man planet
w44: Man know a planet that

...
Decoding

Beam search (Left-to-Right)

I know a planet

w_{41}: I know a planet

w_{42}: I knew planets that

w_{43}: The planet I knew

w_{44}: Man know a planet that

Attention

\[ h_2^{(t)} \rightarrow h_3^{(t)} \]

the planet man

\[ w_2: \text{ know} \]
Attention

know → h2
ARG0 → h2
I → h2
ARG1 → h2
planet → h2

h2 → h3

the planet man

w2: know
Attention

\[ a_i = \text{softmax}(f_i(h^{(s)}, h^{(t)}_{i-1})) \]

The planet man...

\[ \mathbf{w}_2: \text{know} \]
Attention

\[ a_i = \text{softmax} \left( f_i(h^{(s)}, h^{(t)}_{i-1}) \right) \]
Attention

\[ a_i = \text{softmax}(f_i(h^{(s)}_i, h^{(t)}_i)) \]
Attention

\[ a_i = \text{softmax}(f_i(h^{(s)}, h^{(t)}_{i-1})) \]
Attention

\[ a_i = \text{softmax}(f_i(h^{(s)}, h^{(t)}_{i-1})) \]
Max-probability search

\[ w^* = \arg \max_w sc(t^{(s)}, w) \]

where \[ sc(t^{(s)}, w) = p(w|t^{(s)}) \]
Issues to Address

Max-probability search

$$w^* = \arg \max_w sc(t^{(s)}, w)$$

where

$$sc(t^{(s)}, w) = p(w|t^{(s)})$$

Issues

- short / similar outputs
- no guarantee that input is covered
Issues to Address
Issues to Address

Max-probability search
- Length Penalty

\[ sc(t^{(s)}, w) = \frac{\log(p(w|t^{(s)}))}{|t^{(s)}|^{\alpha}} \]
Max-probability search

- Length Penalty

\[ sc(t^{(s)}, w) = \frac{\log(p(w|t^{(s)}))}{|t^{(s)}|^{\alpha}} \]

- Coverage Penalty

\[ cp(t^{(s)}, w) = \beta * \sum_{i=1}^{|t^{(s)}|} \log(\min(\sum_{j=1}^{|w|} a_{i,j}, 1.0)) \]
Issues to Address

**Max-probability search**
- Length Penalty
  \[
  sc(t^{(s)}, w) = \frac{\log(p(w|t^{(s)}))}{|t^{(s)}|^\alpha}
  \]
- Coverage Penalty
  \[
  cp(t^{(s)}, w) = \beta \cdot \sum_{i=1}^{|t^{(s)}|} \log(\min(\sum_{j=1}^{|w|} a_{i,j}, 1.0))
  \]

- Integrating in model
  - Neural Checklist Model  (Kiddon et al, EMNLP 2016)
  - Coverage Model   (Tu et al, ACL 2016)
  - Structural Biases  (Cohn et al, NAACL 2016)
  - Fertility, HMM bias
Issues to Address
Issues to Address

Sparsity
- Anonymize NE tokens
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that UK should keep ...

person_name_0 stated that country_name_1 should keep ...
Issues to Address

Sparsity
- Anonymize NE tokens

state ARG0 person_name_0 ARG1
keep ARG0 country_name_1 ...

President Obama stated that UK should keep ...

person_name_0 stated that country_name_1 should keep …

- Copy from input
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that UK should keep …

person_name_0 stated that country_name_1 should keep …

- Copy from input
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that UK should keep …

person_name_0 stated that country_name_1 should keep …

- Copy from input
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that UK should keep ...

person_name_0 stated that country_name_1 should keep ...

- Copy from input

![Diagram showing an example of copy from input with anonymized tokens]
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that \texttt{UK} should keep …

\texttt{person\_name\_0} stated that \texttt{country\_name\_1} should keep …

- Copy from input

```
input | output | prob  
inhabit | \textit{inhabits} | 0.6  
inhabit | 0.2  
inhabiting | 0.1  
...   | ...   
```
Issues to Address

Sparsity
- Anonymize NE tokens

President Obama stated that UK should keep …

person_name_0 stated that country_name_1 should keep …

- Copy from input

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>inhabit</td>
<td>inhabits</td>
<td>0.6</td>
</tr>
<tr>
<td>inhabit</td>
<td></td>
<td>0.2</td>
</tr>
<tr>
<td>inhabiting</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Issues to Address

Sparsity
- Anonymize NE tokens

- Copy from input

- Data Augmentation (Sennrich et al, ACL 2016)
Open Questions
Open Questions

Representations
- Probably shouldn’t treat all inputs as strings…
Open Questions

Representations
- Probably shouldn’t treat all inputs as strings...

Loss on some intermediate / latent goal
- Don’t want just good-looking string of [X_language]...
Open Questions

Representations
- Probably shouldn’t treat all inputs as strings…

Loss on some intermediate / latent goal
- Don’t want just good-looking string of [X_language]…

Document Plans
- Maybe shouldn’t treat output as stream of strings…
Thank You