• There is not much other than (lexical) semantics in a phrase based MT system.
• What people actually mean when they semantics is often generalisation.
The Machine Translation Pyramid:

- German Syntax
- German Semantics
- Interlingua
- English Syntax
- English Semantics
Introduction

The Machine Translation Pyramid:

- German Syntax
- German Semantics
- Interlingua
- English Semantics
- English Syntax
- English

Phrase based
The Machine Translation Pyramid:

Hierarchical (Hiero) MT
The Machine Translation Pyramid:

- German Syntax
- German Semantics
- Interlingua
- English Semantics
- English Syntax

String to tree
The Machine Translation Pyramid:

German Syntax → Interlingua → English Syntax

German Semantics → Interlingua → English Semantics

Semantic transfer
The Machine Translation Pyramid:

- German Syntax
- German Semantics
- Interlingua
- English Semantics
- English Syntax

Interlingua: the language of God/lambda calculus
请给我一杯白葡萄酒。
Generalisation in MT

Lambda Calculus

Generalisation

请给我一杯白酒。
Generalisation in MT

I'd like a glass of white wine, please.

Generation

Lambda Calculus

Generalisation

Please give me a glass of white wine.
i 'd like a glass of white wine , please .

Generation

Generalisation

Formal logical representations are very hard to learn from data. Let's just assume a vector space and see how we go.
1 Distributed Representations in Compositional Semantics

2 From Vector Space Compositional Semantics to MT
How to Represent Meaning in NLP

We can represent words using a number of approaches:

- Characters
- POS tags
- Grammatical roles
- Named Entity Recognition
- Collocation and distributional representations
- Task-specific features

All of these representations can be encoded in vectors. Some of these representations capture meaning.
Q: Do two words (roughly) mean the same? “Cat” ≡ “Dog”?

A: Use a distributional representation to find out.

Given a vector representation, we can calculate the similarity between two things using their cosine. We know that

\[ A \cdot B = \|A\| \|B\| \cos(\theta) \]

Where \( \cos(\theta) \) is the cosine of the angle between the two vectors \( A \) and \( B \). From this it follows that:

\[ \text{Sim}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \]

\(^1\text{http://en.wikipedia.org/wiki/Cosine_similarity} \)
Word-Word Similarity

$\cos(\theta)$ lies on a range between -1 and 1, with 1 indicating full similarity and 0 indicating no relation and -1 indicating exact opposites.

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td></td>
<td>0.7</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6</td>
<td>0.1</td>
</tr>
</tbody>
</table>

$\text{Sim}(\text{cat}, \text{dog}) = 0.437$

<table>
<thead>
<tr>
<th></th>
<th>Villa</th>
<th>House</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

$\text{Sim}(\text{villa}, \text{house}) = 0.998$
A different task: paraphrase detection

Q: Do two sentences (roughly) mean the same? 
   “He enjoys Jazz music” \equiv “He likes listening to Jazz” ?

A: Use a distributional representation to find out?
A different task: paraphrase detection

Q: Do two sentences (roughly) mean the same?
   “He enjoys Jazz music” ≡ “He likes listening to Jazz” ?

A: Use a distributional representation to find out?

<table>
<thead>
<tr>
<th>Most representations not sensible on the sentence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Characters</td>
</tr>
<tr>
<td>• POS tags</td>
</tr>
<tr>
<td>• Grammatical roles</td>
</tr>
<tr>
<td>• Named Entity Recognition</td>
</tr>
<tr>
<td>• Collocation and distributional representations</td>
</tr>
<tr>
<td>• Task-specific features</td>
</tr>
</tbody>
</table>
Why can’t we extract hierarchical features?

The curse of dimensionality

As the dimensionality of a representation increases, learning becomes less and less viable due to sparsity.

Dimensionality for collocation

- One word per entry: Size of dictionary (small)
- One sentence per entry: Number of possible sentences (infinite)

⇒ We need a different method for representing sentences
Deep Learning for Language

Learning a hierarchy of features, where higher levels of abstraction are derived from lower levels.
A door, a roof, a window: It’s a house
Composition

Lots of possible ways to compose vectors

- Addition
- Multiplication
- Kronecker Product
- Tensor Magic
- Matrix-Vector multiplication
- ...

Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not commutative</td>
<td>Mary likes John $\neq$ John likes Mary</td>
</tr>
<tr>
<td>Encode its parts?</td>
<td>Magic carpet $\equiv$ Magic + Carpet</td>
</tr>
<tr>
<td>More than parts?</td>
<td>Memory lane $\neq$ Memory + Lane</td>
</tr>
</tbody>
</table>
Autoencoders

We want to ensure that the joint representation captures the meaning of its parts. We can achieve this by autoencoding our data at each step:

For this to work, our autoencoder minimizes an objective function over inputs $x_i, i \in N$ and their reconstructions $x_i'$:

$$J = \frac{1}{2} \sum_{i}^{N} \| x_i' - x_i \|^2$$  \hspace{1cm} (1)
We still want to learn how to represent a full sentence (or house). To do this, we chain autoencoders to create a recursive structure.

We use a composition function $g(W \ast \text{input} + \text{bias})$

$g$ is a non-linearity (tanh, sigm)
$W$ is a weight matrix
$b$ is a bias
A different task: paraphrase detection

Q: Do two sentences (roughly) mean the same? “He enjoys Jazz music” ≡ “He likes listening to Jazz” ?

A: Use deep learning to find out!
Other Applications: Stick a label on top

1. Combine label and reconstruction error

\[ E(N, l, \theta) = \sum_{n \in N} E_{rec}(n, \theta) + E_{lbl}(v_n, l, \theta) \]
\[ E_{rec}(n, \theta) = \frac{1}{2} \| [x_n \| y_n] - r_n \|_2^2 \]
\[ E_{lbl}(v, l, \theta) = \frac{1}{2} \| l - v \|_2^2 \]

2. State of the art for a number of tasks:
   Sentiment Analysis
   Paraphrase Detection
   Image Search
   ...
A Convolution Sentence Model

Open the pod bay doors HAL
A Convolution Sentence Model

$m = 2$

Open the pod bay doors HAL
A Convolution Sentence Model

Open the pod bay doors HAL

\[ m = 3 \]
\[ m = 2 \]
A Convolution Sentence Model

$m = 3$

$m = 2$

Open the pod bay doors HAL

$m = 2$

$m = 3$
A: My favourite show is Masterpiece Theatre.

A: Do you like it by any chance?

B: Oh yes!

A: You do!

B: Yes, very much.

A: Well, wouldn't you know.

B: As a matter of fact, I prefer public television.

B: And, uh, I have, particularly enjoy English comedies.
Dave: Hello HAL, do you read me HAL? 
HAL: Affirmative, Dave, I read you. 
Dave: Open the pod bay doors, HAL. 
HAL: I'm sorry, Dave, I'm afraid I can't do that.
Dave: Hello HAL, do you read me HAL?

HAL: Affirmative, Dave, I read you.

Dave: Open the pod bay doors, HAL.

HAL: I'm sorry, Dave, I'm afraid I can't do that.
Dave: Hello HAL, do you read me HAL?

HAL: Affirmative, Dave, I read you.

Dave: Open the pod bay doors, HAL.

HAL: I'm sorry, Dave, I'm afraid I can't do that.
A CSM for Dialogue Act Tagging

Dave: Hello HAL, do you read me HAL?
HAL: Affirmative, Dave, I read you.
Dave: Open the pod bay doors, HAL.
HAL: I'm sorry, Dave, I'm afraid I can't do that.

\[ h_i = g(Ix_{i-1} + H^{i-1}h_{i-1} + Ss_i) \]
\[ p_i = \text{softmax}(O^i h_i) \]
State of the art results while allowing online processing of dialogue.

\[ h_i = g(Ix_{i-1} + H^{i-1}h_{i-1} + Ss_i) \]

\[ p_i = \text{softmax}(O^i h_i) \]
1. Distributed Representations in Compositional Semantics

2. From Vector Space Compositional Semantics to MT
Generalisation in MT

i'd like a glass of white wine, please.

Generation

请给我一杯白葡萄酒。
A simple distributed representation language model:

\[ p_n = C_{n-2} R(w_{n-2}) + C_{n-1} R(w_{n-1}) \]

\[ p(w_n|w_{n-1}, w_{n-2}) \propto \exp \left( R(w_n)^T p_n \right) \]

This is referred to as a \textit{log-bilinear model}. 

\[ R(w_{n-2}) \times C_2 + R(w_{n-1}) \times C_1 = p_n \]
A simple distributed representation language model:

\[ p_n = C_{n-2} R(w_{n-2}) + C_{n-1} R(w_{n-1}) \]

\[ p(w_n | w_{n-1}, w_{n-2}) \propto \exp (R(w_n)^T \sigma (p_n)) \]

Adding a non-linearity gives a slightly more general version of what is often called a neural, or continuous space, LM.
$p_n = C_{n-2} R(t_{n-2}) + C_{n-1} R(t_{n-1}) + \text{CSM}(n, s)$

$p(t_n | t_{n-1}, t_{n-2}, s) \propto \exp \left( R(t_n)^T \sigma (p_n) \right)$
Conditional Generation: A Naive First Model

\[ p_n = C_2 R(t_{n-2}) + C_1 R(t_{n-1}) + \sum_{j=1}^{|s|} S(s_j) \]

\[ p(t_n \mid t_{n-1}, t_{n-2}, s) \propto \exp (R(t_n)^T \sigma(p_n)) \]
明天早上七点叫醒我好吗？
Conditional Generation: A Naive First Model
Conditional Generation: A Naive First Model

+ + + + + + +

明天 早上 七点 叫醒 我 好 吗 ？
may i have a wake-up call at seven tomorrow morning?
where's the currency exchange office?

CLM

II

货币 兑换处 在 哪里 ？
I'd like a glass of white wine, please.
i'm going to los angeles this afternoon.
I'd like to have a room under thirty dollars a night.
I'd like to have a room under thirty dollars a night.
I want a late thirties under $'s room.
you have to do something about it.
i can't urinate.
Conditional Generation: A Convolution N-Gram Model

\[
P(f|e)
\]

\[
P(f|m, e)
\]

RCTM I

RCTM II
## Perplexity results on the WMT News-Commentary test sets.

<table>
<thead>
<tr>
<th>Model</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knesser-Ney 5gram</td>
<td>218</td>
<td>213</td>
<td>222</td>
<td>225</td>
</tr>
<tr>
<td>RNNLM</td>
<td>178</td>
<td>169</td>
<td>178</td>
<td>181</td>
</tr>
<tr>
<td>IBM Model 1</td>
<td>207</td>
<td>200</td>
<td>188</td>
<td>197</td>
</tr>
<tr>
<td>fast_align (cdec/IBM Model 2)</td>
<td>153</td>
<td>146</td>
<td>135</td>
<td>144</td>
</tr>
<tr>
<td>RCTM I</td>
<td>143</td>
<td>134</td>
<td>140</td>
<td>142</td>
</tr>
<tr>
<td>RCTM II</td>
<td><strong>86</strong></td>
<td><strong>77</strong></td>
<td><strong>76</strong></td>
<td><strong>77</strong></td>
</tr>
</tbody>
</table>
A Convolution N-Gram Model

<table>
<thead>
<tr>
<th></th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>En → Fr</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knesser-Ney 5gram</td>
<td>218</td>
<td>213</td>
<td>222</td>
<td>225</td>
</tr>
<tr>
<td>RNNLM</td>
<td>178</td>
<td>169</td>
<td>178</td>
<td>181</td>
</tr>
<tr>
<td>IBM Model 1</td>
<td>207</td>
<td>200</td>
<td>188</td>
<td>197</td>
</tr>
<tr>
<td>fast_align (cdec/IBM Model 2)</td>
<td>153</td>
<td>146</td>
<td>135</td>
<td>144</td>
</tr>
<tr>
<td>RCTM I</td>
<td>143</td>
<td>134</td>
<td>140</td>
<td>142</td>
</tr>
<tr>
<td>RCTM II</td>
<td><strong>86</strong></td>
<td><strong>77</strong></td>
<td><strong>76</strong></td>
<td><strong>77</strong></td>
</tr>
</tbody>
</table>

Perplexity results on the WMT News-Commentary test sets.

In k-best rescoring experiments the RCTM II model achieves similar Bleu scores to a MERT trained cdec baseline.
### Advantages
- fast to train and decode, very compact models.
- a valid and tractable probability distribution over translations, making extensions easy to implement.
- distributed representations for words naturally include morphological properties.
- the conditional generation framework easily permits additional context such as dialogue and domain level vectors.

### Challenges
- better conditioning on sentence position and length.
- handling rare and unknown words.
Funded studentships are available for strong students interested in pursuing graduate study in Machine Learning and Computational Linguistics

http://www.cs.ox.ac.uk/admissions/dphil/