Scope in an incremental context

Lecture 4: computational linguistics and scope

Asad Sayeed

University of Gothenburg

Looking at English Resource Grammar

Part 1: crash course in NLP machine learning

Part 1.1: classification

What classifiers do...

- Given an object, assign a category.
- Such tasks are pervasive in NLP.

Example: classification of documents

- "Classic" sentiment analysis: develop a program that groups customer reviews into positive and negative classes (given the text only)

Just plain lame., August 14, 2007

By **Gary Smith "Editor, Handgun Hunter Magazine"** (Texas) - See all my reviews

This review is from: Garden & Gun (Magazine)

This magazine has a catchy title and very nice graphics and photography. What the premier issue lacks is anything of any substance about guns or hunting. I wonder if they actually read their own title. In my opinion these guys are nothing more than posers from the guns/hunting standpoint and many of the photographs appear to be staged. In particular, there are a couple pictures of a woman shooting a bow and arrow. Not only is she showing extremely poor form she's using the equipment shown in the photographs incorrectly. This is tantamount to using spinning gear with the reel positioned over the top of the fishing pole. If they want to cover hunting they should at least hire a photo editor that knows what (s)he's looking at. If you want a hunting magazine buy something else...

Example: classification of documents

- "Classic" sentiment analysis: develop a program that groups customer reviews into positive and negative classes (given the text only)

Just plain lame., August 14, 2007

By **Gary Smith "Editor, Handgun Hunter Magazine"** (Texas) - See all my reviews

This review is from: Garden & Gun (Magazine)

This magazine has a catchy title and very nice graphics and photography. What the premier issue lacks is anything of any substance about guns or hunting. I wonder if they actually read their own title. In my opinion these guys are nothing more than posers from the guns/hunting standpoint and many of the photographs appear to be staged. In particular, there are a couple pictures of a woman shooting a bow and arrow. Not only is she showing extremely poor form she's using the equipment shown in the photographs incorrectly. This is tantamount to using spinning gear with the reel positioned over the top of the fishing pole. If they want to cover hunting they should at least hire a photo editor that knows what (s)he's looking at. If you want a hunting magazine buy something else...

- other examples:

Example: classification of documents

- "Classic" sentiment analysis: develop a program that groups customer reviews into positive and negative classes (given the text only)

  ★☆☆☆☆ **Just plain lame.**, August 14, 2007

  By **Gary Smith "Editor, Handgun Hunter Magazine"** (Texas) - See all my reviews

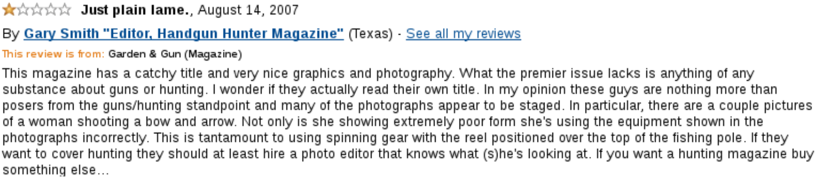
  This review is from: Garden & Gun (Magazine)

  This magazine has a catchy title and very nice graphics and photography. What the premier issue lacks is anything of any substance about guns or hunting. I wonder if they actually read their own title. In my opinion these guys are nothing more than posers from the guns/hunting standpoint and many of the photographs appear to be staged. In particular, there are a couple pictures of a woman shooting a bow and arrow. Not only is she showing extremely poor form she's using the equipment shown in the photographs incorrectly. This is tantamount to using spinning gear with the reel positioned over the top of the fishing pole. If they want to cover hunting they should at least hire a photo editor that knows what (s)he's looking at. If you want a hunting magazine buy something else...

- other examples:
  - Reuters, $\sim$ 100 hierarchical categories

Example: classification of documents

- "Classic" sentiment analysis: develop a program that groups customer reviews into positive and negative classes (given the text only)



Just plain lame., August 14, 2007

By Gary Smith "Editor, Handgun Hunter Magazine" (Texas) - See all my reviews

This review is from: Garden & Gun (Magazine)

This magazine has a catchy title and very nice graphics and photography. What the premier issue lacks is anything of any substance about guns or hunting. I wonder if they actually read their own title. In my opinion these guys are nothing more than posers from the guns/hunting standpoint and many of the photographs appear to be staged. In particular, there are a couple pictures of a woman shooting a bow and arrow. Not only is she showing extremely poor form she's using the equipment shown in the photographs incorrectly. This is tantamount to using spinning gear with the reel positioned over the top of the fishing pole. If they want to cover hunting they should at least hire a photo editor that knows what (s)he's looking at. If you want a hunting magazine buy something else...

- other examples:
  - Reuters, $\sim$ 100 hierarchical categories
  - Classification according to a library system (LCC, SAB)

Example: classification of documents

- "Classic" sentiment analysis: develop a program that groups customer reviews into positive and negative classes (given the text only)

★☆☆☆☆ **Just plain lame.**, August 14, 2007

By Gary Smith "Editor, Handgun Hunter Magazine" (Texas) - See all my reviews

This review is from: Garden & Gun (Magazine)

This magazine has a catchy title and very nice graphics and photography. What the premier issue lacks is anything of any substance about guns or hunting. I wonder if they actually read their own title. In my opinion these guys are nothing more than posers from the guns/hunting standpoint and many of the photographs appear to be staged. In particular, there are a couple pictures of a woman shooting a bow and arrow. Not only is she showing extremely poor form she's using the equipment shown in the photographs incorrectly. This is tantamount to using spinning gear with the reel positioned over the top of the fishing pole. If they want to cover hunting they should at least hire a photo editor that knows what (s)he's looking at. If you want a hunting magazine buy something else...

- other examples:
  - Reuters, $\sim$ 100 hierarchical categories
  - Classification according to a library system (LCC, SAB)
  - ...by target group (e.g. CEFR readability) or some property of the author (e.g. gender, native language)

Example: disambiguation of word meaning in context

A woman and child suffered minor injuries after the car they were riding in crashed into a **rock** *wall Tuesday morning.*

Example: disambiguation of word meaning in context

A woman and child suffered minor injuries after the car they were riding in crashed into a **rock** *wall Tuesday morning.*

- what is the meaning of *rock* in this context?

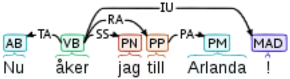
Example: disambiguation of word meaning in context

A woman and child suffered minor injuries after the car they were riding in crashed into a **rock** *wall Tuesday morning.*

- what is the meaning of *rock* in this context?

- S: (n) **rock**, stone (a lump or mass of hard consolidated mineral matter) *"he threw a rock at me"*
- S: (n) **rock**, stone (material consisting of the aggregate of minerals like those making up the Earth's crust) *"that mountain is solid rock"; "stone is abundant in New England and there are many quarries"*
- S: (n) **Rock**, John Rock (United States gynecologist and devout Catholic who conducted the first clinical trials of the oral contraceptive pill (1890-1984))
- S: (n) **rock** ((figurative) someone who is strong and stable and dependable) *"he was her rock during the crisis"; "Thou art Peter, and upon this rock I will build my church"--Gospel According to Matthew*
- S: (n) rock candy, **rock** (hard bright-colored stick candy (typically flavored with peppermint))
- S: (n) rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, **rock**, rock music (a genre of popular music originating in the 1950s; a blend of black rhythm-and-blues with white country-and-western) *"rock is a generic term for the range of styles that evolved out of rock'n'roll."*
- S: (n) **rock**, careen, sway, tilt (pitching dangerously to one side)

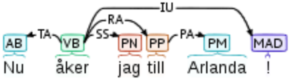
Example: classification of grammatical relations



Example: classification of grammatical relations

- What is the grammatical relation between *åker* and *till*?

Example: classification of grammatical relations



- What is the grammatical relation between *åker* and *till*?
    - e.g. subject, object, adverbial, . . .

Example: classification of discourse relations

Mary had to study hard. Her exam was only one week away.

Example: classification of discourse relations

Mary had to study hard. Her exam was only one week away.

- What is the discourse/rhetorical relation between the two sentences?

Example: classification of discourse relations

Mary had to study hard. Her exam was only one week away.

- What is the discourse/rhetorical relation between the two sentences?
  - e.g. IF, THEN, AND, BECAUSE, BUT, ...

Features for classification

- To be able to classify an object, we must describe its properties: **features**

Features for classification

- To be able to classify an object, we must describe its properties: **features**
- Useful information that we believe helps us tell the classes apart.

Features for classification

- To be able to classify an object, we must describe its properties: **features**
- Useful information that we believe helps us tell the classes apart.
- This is an art more than a science.

Features for classification

- To be able to classify an object, we must describe its properties: **features**
- Useful information that we believe helps us tell the classes apart.
- This is an art more than a science.
- Examples:

Features for classification

- To be able to classify an object, we must describe its properties: **features**
- Useful information that we believe helps us tell the classes apart.
- This is an art more than a science.
- Examples:
  - In document classification, typically the **words**

Features for classification

- To be able to classify an object, we must describe its properties: **features**
- Useful information that we believe helps us tell the classes apart.
- This is an art more than a science.
- Examples:
  - In document classification, typically the **words**
  - ...But also stylistic features such as sentence length, word variation, syntactic complexity

Representation of features

- depending on the task we are trying to solve, features may be viewed in different ways

Representation of features

- depending on the task we are trying to solve, features may be viewed in different ways
- **bag of words**: `["I", "love", "this", "film"]`

Representation of features

- depending on the task we are trying to solve, features may be viewed in different ways
- **bag of words**: `["I", "love", "this", "film"]`
- **attribute–value pairs**: {`"age"=63, "gender"="F", "income"=25000`}

Representation of features

- depending on the task we are trying to solve, features may be viewed in different ways
- **bag of words**: `["I", "love", "this", "film"]`
- **attribute–value pairs**: {`"age"=63, "gender"="F", "income"=25000`}
- **geometric vector**: `[0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 1]`

A note on terminology

- We want to develop some NLP system (a classifier, a tagger, a parser, . . .) by getting some parameters from the data instead of hard-coding (**data-driven**).

A note on terminology

- We want to develop some NLP system (a classifier, a tagger, a parser, ...) by getting some parameters from the data instead of hard-coding (**data-driven**).
- A statistician would say that we **estimate** parameters of a model.

A note on terminology

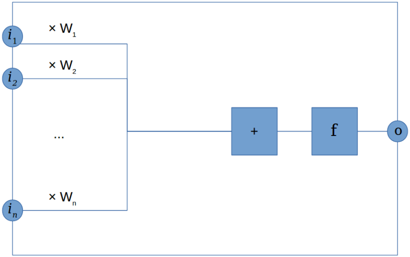
- We want to develop some NLP system (a classifier, a tagger, a parser, . . .) by getting some parameters from the data instead of hard-coding (**data-driven**).
- A statistician would say that we **estimate** parameters of a model.
- A computer scientist would say that we **train** the model.

A note on terminology

- We want to develop some NLP system (a classifier, a tagger, a parser, ...) by getting some parameters from the data instead of hard-coding (**data-driven**).
- A statistician would say that we **estimate** parameters of a model.
- A computer scientist would say that we **train** the model.
  - Or conversely, that we apply a **machine learning** algorithm.

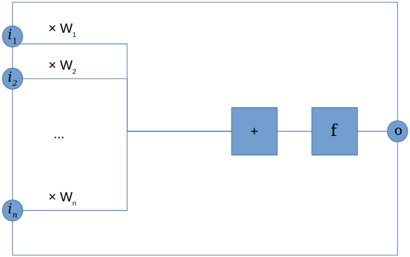
The perceptron: a very simple neural network

(from Wikipedia)



The perceptron: a very simple neural network

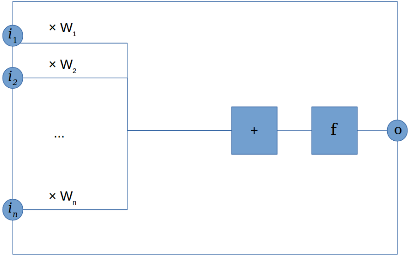
(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs *i* to the network.

The perceptron: a very simple neural network

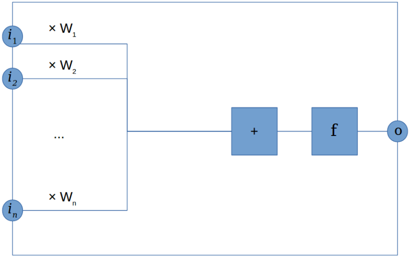
(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs i to the network.
- Feature function f is applied (remember: 1 or 0 output).

The perceptron: a very simple neural network

(from Wikipedia)



- Each instance vector **x**'s values are fed as inputs i to the network.
- Feature function f is applied (remember: 1 or 0 output).
- Weights adjusted based on output correctness.

Perceptron algorithm (roughly)

Initialize weights $\mathbf{w}$ and bias (usually to (close to) 0).
Given n feature vectors $\mathbf{x}$ and corresponding "ground truth" values d, for vector $\mathbf{x}_i$:

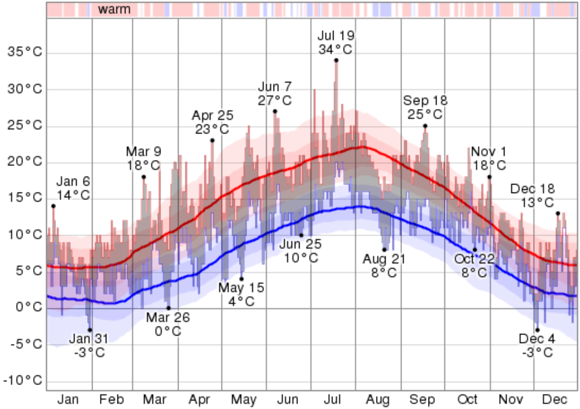
- Calculate $f(\mathbf{x})$ as 1 or 0 using $\mathbf{w} \cdot \mathbf{x}_i + b$.
- Update weights as $\mathbf{w} \leftarrow \mathbf{w} + (d_i - f(\mathbf{x_i}))\mathbf{x_i}$.
- Move to next $\mathbf{x}$ feature vector, cycling through vectors until convergence.

(There is a theoretical upper bound on how many iterations are required to converge.)

Part 1.2: language modeling

We have expectations about changes.

We know that yesterday is a good clue about today.
Temperatures in Amsterdam in 2014:



The daily temperature is a Markov process.

Let T_d = temperature T on day d.
We can represent the probability conditionally.

Probability of today's temperature given universe

$p(T_d | T_{d-1}, T_{d-2}, \ldots, T_{d-\infty})$

The daily temperature is a Markov process.

Let T_d = temperature T on day d.
We can represent the probability conditionally.

Probability of today's temperature given 2 previous days

$$p(T_d | T_{d-1}, T_{d-2}, \ldots, T_{d-\infty}) \approx p(T_d | T_{d-1}, T_{d-2})$$

But we only need a few days to give us a trend. So we make a Markov assumption.

The daily temperature is a Markov process.

Let T_d = temperature T on day d.
We can represent the probability conditionally.

Probability of today's temperature given 2 previous days

$$p(T_d | T_{d-1}, T_{d-2}, \ldots, T_{d-\infty}) \approx p(T_d | T_{d-1}, T_{d-2})$$

But we only need a few days to give us a trend. So we make a Markov assumption.
Then we can calculate the joint probability of a sequence of days:

Markov chain

$$p(T_d, T_{d-1}, T_{d-2}) =$$
$$p(T_d | T_{d-1}, T_{d-2}) p(T_{d-1} | T_{d-2}, T_{d-3}) p(T_{d-2} | T_{d-3}, T_{d-4})$$

Modeling Markovianishly

The Markov assumption is an assumption of ignorance.

Modeling Markovianishly

The Markov assumption is an assumption of ignorance.

- There is a process "generating" the data, but we don't know what it is.

Modeling Markovianishly

The Markov assumption is an assumption of ignorance.

- There is a process "generating" the data, but we don't know what it is.
- That process has states we observe, but we don't know what hidden states might actually give us those outcomes.

Modeling Markovianishly

The Markov assumption is an assumption of ignorance.

- There is a process "generating" the data, but we don't know what it is.
- That process has states we observe, but we don't know what hidden states might actually give us those outcomes.
- The probabilities allow us to "infer" the hidden states, after making some assumptions...

Modeling Markovianishly

The Markov assumption is an assumption of ignorance.

- There is a process "generating" the data, but we don't know what it is.
- That process has states we observe, but we don't know what hidden states might actually give us those outcomes.
- The probabilities allow us to "infer" the hidden states, after making some assumptions...

A possible collection of states: POS tags

Tagging in general: the task

- We are given a list of words such as `['The', 'cat', 'sleeps']`

Tagging in general: the task

- We are given a list of words such as `['The', 'cat', 'sleeps']`
- Our task is to predict a list of tags such as `['DT', 'NN', 'VBZ']`

Tagging in general: the task

- We are given a list of words such as `['The', 'cat', 'sleeps']`
- Our task is to predict a list of tags such as `['DT', 'NN', 'VBZ']`
- This is a **sequence tagging** problem.

A probabilistic model of tagging

- The typical probabilistic formulation of a tagger starts from Bayes' rule:

$$\begin{aligned}\arg\max_T P(T|W) &= \arg\max_T \frac{P(W|T)P(T)}{P(W)} \\ &= \arg\max_T P(W|T)P(T)\end{aligned}$$

A probabilistic model of tagging

- The typical probabilistic formulation of a tagger starts from Bayes' rule:

$$\begin{aligned}\arg\max_T P(T|W) &= \arg\max_T \frac{P(W|T)P(T)}{P(W)} \\ &= \arg\max_T P(W|T)P(T)\end{aligned}$$

- $P(T)$ is like a language model, but for tag sequences instead of word sequences

Making the probabilities practical

- We need to make assumptions about $P(T)$ and $P(W|T)$.
- In a **bigram tagger**, the probability of the next tag depends **only** on the previous tag (Markov assumption):

$$P(t_n|t_1,\ldots,t_{n-1}) \approx P(t_n|t_{n-1})$$

Making the probabilities practical

- We need to make assumptions about $P(T)$ and $P(W|T)$.
- In a **bigram tagger**, the probability of the next tag depends **only** on the previous tag (Markov assumption):

$$P(t_n|t_1, \ldots, t_{n-1}) \approx P(t_n|t_{n-1})$$

  - This is called the **transition probability**.

Making the probabilities practical

- We need to make assumptions about $P(T)$ and $P(W|T)$.
- In a **bigram tagger**, the probability of the next tag depends **only** on the previous tag (Markov assumption):

$$P(t_n|t_1, \ldots, t_{n-1}) \approx P(t_n|t_{n-1})$$

  - This is called the **transition probability**.
- The probability of a word depends **only** on its tag:

$$P(w_n|\text{tags}, \text{other words}) \approx P(w_n|t_n)$$

Making the probabilities practical

- We need to make assumptions about $P(T)$ and $P(W|T)$.
- In a **bigram tagger**, the probability of the next tag depends **only** on the previous tag (Markov assumption):

$$P(t_n|t_1,\ldots,t_{n-1}) \approx P(t_n|t_{n-1})$$

  - This is called the **transition probability**.
- The probability of a word depends **only** on its tag:

$$P(w_n|\text{tags}, \text{other words}) \approx P(w_n|t_n)$$

  - This is called the **emission probability**.

Hidden Markov Models

$$P(t_n|t_{n-1}) \qquad P(w_n|t_n)$$

- A model where we have an unknown underlying sequence is called a **hidden Markov** model (HMM).

Hidden Markov Models

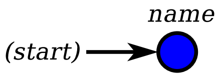
$$P(t_n|t_{n-1}) \qquad P(w_n|t_n)$$

- A model where we have an unknown underlying sequence is called a **hidden Markov** model (HMM).

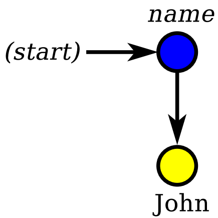
Generative story in hidden Markov models

(start)

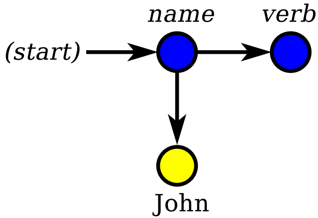
Generative story in hidden Markov models



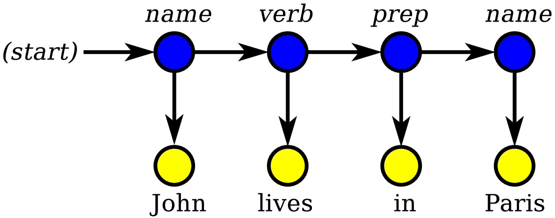
Generative story in hidden Markov models



Generative story in hidden Markov models



Generative story in hidden Markov models



How can we estimate the probabilities?

- to estimate $P(t_n|t_{n-1})$ and $P(w_n|t_n)$, we need a corpus where the part-of-speech tags have been annotated (by humans)

How can we estimate the probabilities?

- to estimate $P(t_n|t_{n-1})$ and $P(w_n|t_n)$, we need a corpus where the part-of-speech tags have been annotated (by humans)

The DT
rifles NNS
were VBD
n't RB
loaded VBN
. .
As IN
interest NN
rates NNS
rose VBD
, ,
...

Estimating the probabilities

- We estimate the probabilities by counting frequencies (maximum likelihood estimation; MLE):

$$P_{MLE}(\text{noun}|\text{verb}) = \frac{\text{count}(\text{verb, noun})}{\text{count}(\text{verb})} \qquad P_{MLE}(cat|\text{noun}) = \frac{\text{count}(\text{noun: } cat)}{\text{count}(\text{noun})}$$

Kelsey and other Grammers

- A **grammar** here is another word for a language model
- They consist of four sets $G = \langle \Sigma, N, S, P \rangle$

  terminals – word types; lowest nodes in syntax trees
  Examples: *dog, the, eats*

  non-terminals – phrasal types; middle nodes in syntax trees
  Examples: *VP, DET, NP*

  start symbol – "S"; the top node in syntax trees

Kelsey and other Grammers

- A **grammar** here is another word for a language model
- They consist of four sets $G = \langle \Sigma, N, S, P \rangle$

terminals – word types; lowest nodes in syntax trees
Examples: *dog, the, eats*

non-terminals – phrasal types; middle nodes in syntax trees
Examples: *VP, DET, NP*

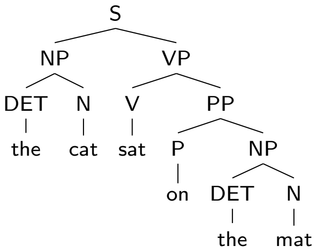
start symbol – "S"; the top node in syntax trees

production rules – recursive symbol substitutions
Examples:

$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow DET\ N \\
NP &\rightarrow ADJ\ N \\
VP &\rightarrow V\ NP \\
VP &\rightarrow V \\
N &\rightarrow dog \\
N &\rightarrow cat \\
V &\rightarrow barks
\end{aligned}
$$

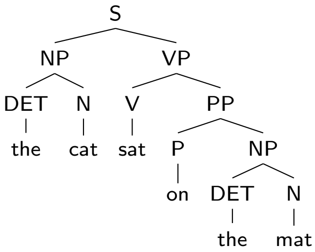
Visualization

- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:



Visualization

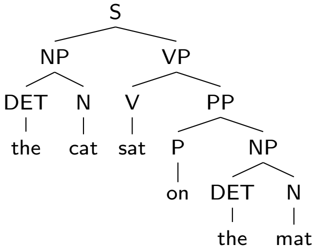
- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:



$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow DET\ N \\
DET &\rightarrow the \\
N &\rightarrow cat \\
VP &\rightarrow V\ PP \\
V &\rightarrow sat \\
PP &\rightarrow P\ NP \\
N &\rightarrow mat
\end{aligned}
$$

Visualization

- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:

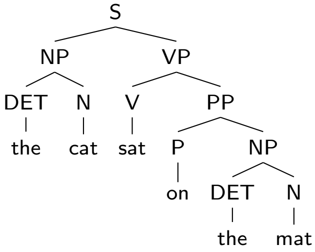


$$
\begin{array}{rcl}
S & \rightarrow & NP\ VP \\
NP & \rightarrow & DET\ N \\
DET & \rightarrow & the \\
N & \rightarrow & cat \\
VP & \rightarrow & V\ PP \\
V & \rightarrow & sat \\
PP & \rightarrow & P\ NP \\
N & \rightarrow & mat
\end{array}
$$

- Originally these trees were **mere visualizations** of how you could generate a grammatical sentence, given a grammar

Visualization

- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:

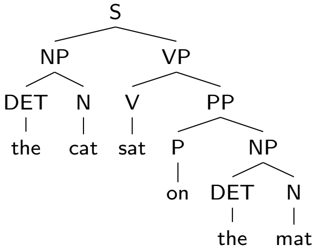


$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow DET\ N \\
DET &\rightarrow the \\
N &\rightarrow cat \\
VP &\rightarrow V\ PP \\
V &\rightarrow sat \\
PP &\rightarrow P\ NP \\
N &\rightarrow mat
\end{aligned}
$$

- Originally these trees were **mere visualizations** of how you could generate a grammatical sentence, given a grammar
- Then people started to think of these trees as the actual **structure** of a sentence

Visualization

- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:

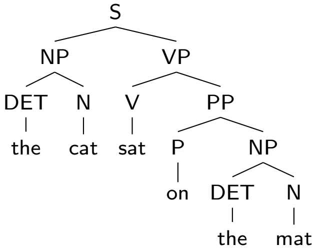


S	$\rightarrow$	$NP\ VP$
NP	$\rightarrow$	$DET\ N$
DET	$\rightarrow$	the
N	$\rightarrow$	cat
VP	$\rightarrow$	$V\ PP$
V	$\rightarrow$	sat
PP	$\rightarrow$	$P\ NP$
N	$\rightarrow$	mat

- Originally these trees were **mere visualizations** of how you could generate a grammatical sentence, given a grammar
- Then people started to think of these trees as the actual **structure** of a sentence
- Confusion ensued

Visualization

- Sentences are often visualized using **derivation trees**, also known as **parse trees** or **syntax trees**
- Example:



$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow DET\ N \\
DET &\rightarrow the \\
N &\rightarrow cat \\
VP &\rightarrow V\ PP \\
V &\rightarrow sat \\
PP &\rightarrow P\ NP \\
N &\rightarrow mat
\end{aligned}
$$

- Originally these trees were **mere visualizations** of how you could generate a grammatical sentence, given a grammar
- Then people started to think of these trees as the actual **structure** of a sentence
- Confusion ensued(Was it really confusion?)

Context-free Grammars

- A **context-free grammar** (CFG) is a generative model that can generate context-free languages, which are somewhere in the middle of the formal language hierarchy
- Many, but not all, phenomena in natural languages can be generated by CFGs

Context-free Grammars

- A **context-free grammar** (CFG) is a generative model that can generate context-free languages, which are somewhere in the middle of the formal language hierarchy
- Many, but not all, phenomena in natural languages can be generated by CFGs
- Context-free production rules have the general form of a non-termal rewriting to a sequence (string) of terminals and/or non-terminals ($A \rightarrow \alpha$)

Context-free Grammars

- A **context-free grammar** (CFG) is a generative model that can generate context-free languages, which are somewhere in the middle of the formal language hierarchy
- Many, but not all, phenomena in natural languages can be generated by CFGs
- Context-free production rules have the general form of a non-termal rewriting to a sequence (string) of terminals and/or non-terminals ($A \rightarrow \alpha$)
- CFGs can generate and recognize **center embedding**, but not more complex word order phenomena, so effectively CFG parse trees have **no crossing lines**

Context-free Grammars

- A **context-free grammar** (CFG) is a generative model that can generate context-free languages, which are somewhere in the middle of the formal language hierarchy
- Many, but not all, phenomena in natural languages can be generated by CFGs
- Context-free production rules have the general form of a non-termal rewriting to a sequence (string) of terminals and/or non-terminals ($A \rightarrow \alpha$)
- CFGs can generate and recognize **center embedding**, but not more complex word order phenomena, so effectively CFG parse trees have **no crossing lines**
- Non-projective dependency grammars are more or less equivalent to CFGs (they have the same weak generative capacity)

Treebanks

- It's a lot of work to define a language model by hand (including context-free grammars), so another way is to annotate treebanks
- Example: (S (NP (DET the) (N cat))(VP (V sat)(PP (P on)(NP (DET the) (N mat)))))

Treebanks

- It's a lot of work to define a language model by hand (including context-free grammars), so another way is to annotate treebanks
- Example: (S (NP (DET the) (N cat))(VP (V sat)(PP (P on)(NP (DET the) (N mat)))))
- There are treebanks for about 10–20 languages, the Penn Treebank being the most well-known for English

Treebanks

- It's a lot of work to define a language model by hand (including context-free grammars), so another way is to annotate treebanks
- Example: (S (NP (DET the) (N cat))(VP (V sat)(PP (P on)(NP (DET the) (N mat)))))
- There are treebanks for about 10–20 languages, the Penn Treebank being the most well-known for English
- Treebanks can be annotated with various grammatical annotations, like **constituency / phrase-structure** (as above), **dependency grammar**, categorial grammar, HPSG, etc.
- Most of these annotation styles can be approximately mapped to other styles

Treebanks

- It's a lot of work to define a language model by hand (including context-free grammars), so another way is to annotate treebanks
- Example: (S (NP (DET the) (N cat))(VP (V sat)(PP (P on)(NP (DET the) (N mat)))))
- There are treebanks for about 10–20 languages, the Penn Treebank being the most well-known for English
- Treebanks can be annotated with various grammatical annotations, like **constituency / phrase-structure** (as above), **dependency grammar**, categorial grammar, HPSG, etc.
- Most of these annotation styles can be approximately mapped to other styles
- Here is a link to a list of syntactic treebanks

PCFGs

- We can induce a **probabilistic context-free grammar** (PCFG) from the treebank
- With multiple annotated sentences, we can get probabilities for production rules. Example:

1.0	S	$\rightarrow NP\ VP$
0.6	NP	$\rightarrow DET\ N$
0.4	NP	$\rightarrow ADJ\ N$
0.7	VP	$\rightarrow V\ NP$
0.3	VP	$\rightarrow V$
0.8	N	$\rightarrow dog$
0.2	N	$\rightarrow cat$
1.0	V	$\rightarrow barks$
1.0	DET	$\rightarrow the$

PCFGs

- We can induce a **probabilistic context-free grammar** (PCFG) from the treebank
- With multiple annotated sentences, we can get probabilities for production rules. Example:

1.0	S	$\rightarrow NP\ VP$
0.6	NP	$\rightarrow DET\ N$
0.4	NP	$\rightarrow ADJ\ N$
0.7	VP	$\rightarrow V\ NP$
0.3	VP	$\rightarrow V$
0.8	N	$\rightarrow$ *dog*
0.2	N	$\rightarrow$ *cat*
1.0	V	$\rightarrow$ *barks*
1.0	DET	$\rightarrow$ *the*

- Notice that the probabilities for each left-hand side must sum to one

PCFGs vs. n-gram Language Models (Lexicalized Probabilistic Regular Grammars)

- PCFGs can better handle long-distance dependencies like subject-verb agreement and filler-gap dependencies
- PCFGs usually give worse perplexity than n-gram LMs. Why?

PCFGs vs. *n*-gram Language Models (Lexicalized Probabilistic Regular Grammars)

- PCFGs can better handle long-distance dependencies like subject-verb agreement and filler-gap dependencies
- PCFGs usually give worse perplexity than *n*-gram LMs. Why? Mostly because PCFGs are unlexicalized – they use pre-terminals (word classes / POS tags). Thus they fail to account for local co-occurrences like multiword expressions and proper names.

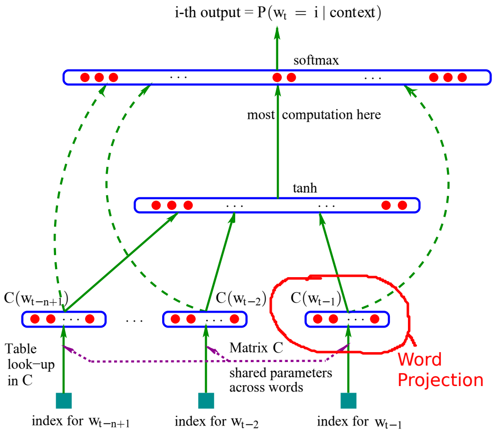
PCFGs vs. *n*-gram Language Models (Lexicalized Probabilistic Regular Grammars)

- PCFGs can better handle long-distance dependencies like subject-verb agreement and filler-gap dependencies
- PCFGs usually give worse perplexity than *n*-gram LMs. Why? Mostly because PCFGs are unlexicalized – they use pre-terminals (word classes / POS tags). Thus they fail to account for local co-occurrences like multiword expressions and proper names.
- PCFGs take longer to train
- PCFGs need manually-annotated treebanks to give decent results
- PCFG parsers (eg. CKY) are usually not incremental

Part 1.3: “deep” learning

Neural Language Modeling

- This was actually one of the earliest uses of word vectors. [Bengio et al., 2003]. Feed-forward neural network:



Neural Networks for Sequential Data

- Feedforward (FF) networks only indirectly deal with sequential data (like language)

Neural Networks for Sequential Data

- Feedforward (FF) networks only indirectly deal with sequential data (like language)
- FF Neural LMs are basically 'soft' *n*-gram LMs – their history is still fixed

Neural Networks for Sequential Data

- Feedforward (FF) networks only indirectly deal with sequential data (like language)
- FF Neural LMs are basically 'soft' *n*-gram LMs – their history is still fixed
- The model needs to 'remember' a longer history, with loops

Recurrent Neural Networks

A neural net with loops is called **recurrent**

Recurrent Neural Networks

- A neural net with loops is called **recurrent**
- The current hidden layer of the model is based on both the current word and the hidden layer of the previous timestep

Recurrent Neural Networks

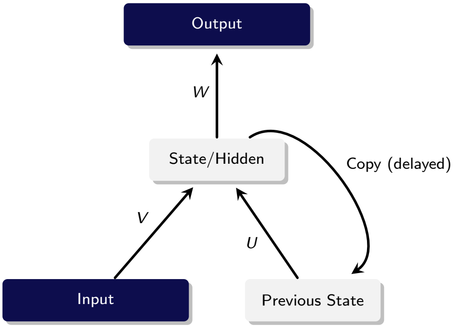
A neural net with loops is called **recurrent**

- The current hidden layer of the model is based on both the current word and the hidden layer of the previous timestep
- This is implemented by copying the hidden layer to another layer, overwriting the existing weights

Recurrent Neural Networks

A neural net with loops is called **recurrent**

- The current hidden layer of the model is based on both the current word and the hidden layer of the previous timestep
- This is implemented by copying the hidden layer to another layer, overwriting the existing weights
- This specific RNN is called an **Elman network** (or **simple RNN** / SRN)



To train an RNN, we first need to 'unroll' the loops

Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?

Sentence Vectors

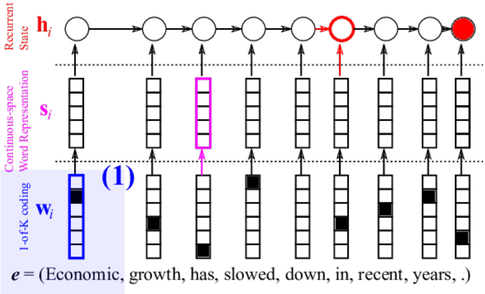
- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not?

Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ (ϕ_{enc} = LSTM or GRU)

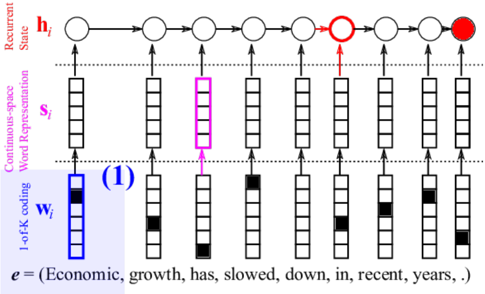
Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ (ϕ_{enc} = LSTM or GRU)



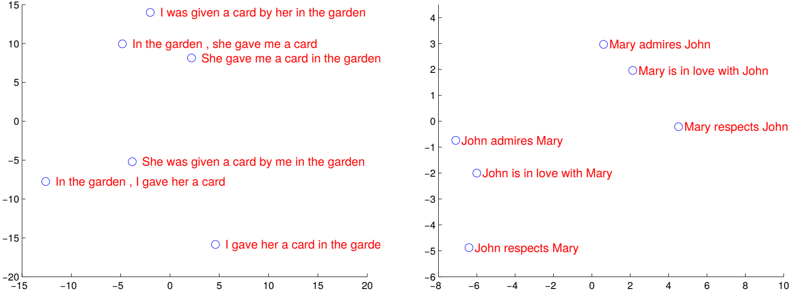
Sentence Vectors

- We've seen that words can be represented as vectors. Can sentences be represented as vectors?
- Sure, why not? How? From the hidden state at the end of a sentence: $\mathbf{h}_i = \phi_{\text{enc}}(\mathbf{h}_{i-1}, \mathbf{s}_i)$ ($\phi_{\text{enc}} = \text{LSTM or GRU}$)



- Are they any good? For Elman networks (SRNs), not so much. For LSTMs or GRUs, yes, they're pretty good

Sentence Vector Examples

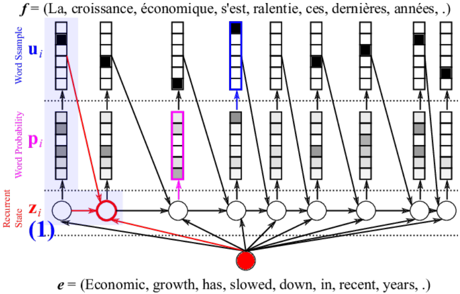


Sentence vectors were projected to two dimensions using PCA

Images courtesy of Sutskever, et al (2014)

Generating Sentences from Vectors

- We can also try to go the other direction, generating sentences from vectors
- How? Use an RNN to **decode**, rather than **encode** a sentence: $\mathbf{z}_i = \phi_{\text{dec}}(\mathbf{z}_{i-1}, \mathbf{u}_{i-1}, \mathbf{h}_T)$



- $\mathbf{h}_T$ ensures global sentence coherency (& adequacy in MT); $\mathbf{u}_{i-1}$ ensures local fluency

Encode and Decode to Translate

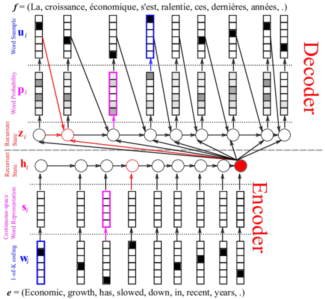
- We can combine the neural encoder and decoder of previous slides to form an **encoder–decoder model**

Encode and Decode to Translate

- We can combine the neural encoder and decoder of previous slides to form an **encoder–decoder model**
- This can be used for machine translation, summarization, chatbots/dialog systems, and **sequences-to-sequence** tasks

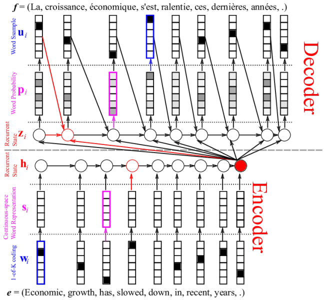
Encode and Decode to Translate

- We can combine the neural encoder and decoder of previous slides to form an **encoder–decoder model**
- This can be used for machine translation, summarization, chatbots/dialog systems, and **sequences-to-sequence** tasks



Encode and Decode to Translate

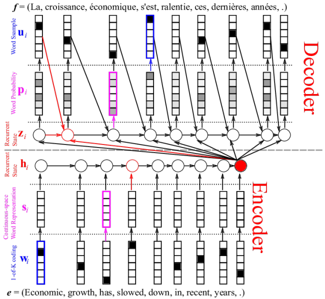
- We can combine the neural encoder and decoder of previous slides to form an **encoder–decoder model**
- This can be used for machine translation, summarization, chatbots/dialog systems, and **sequences-to-sequence** tasks



- Monolingual word projections (vectors/embeddings) are trained to maximize likelihood of next word

Encode and Decode to Translate

- We can combine the neural encoder and decoder of previous slides to form an **encoder–decoder model**
- This can be used for machine translation, summarization, chatbots/dialog systems, and **sequences-to-sequence** tasks



- Monolingual word projections (vectors/embeddings) are trained to maximize likelihood of next word
- Source-side word projections ($\mathbf{s}_i$) in an encoder–decoder setting are trained to maximize target-side likelihood

Bidirectional RNNs

- The basic encoder–decoder architecture doesn't handle long sentences very well

Bidirectional RNNs

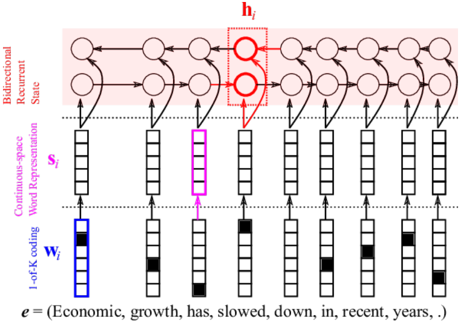
- The basic encoder–decoder architecture doesn't handle long sentences very well
- Everything must fit into a fixed-size vector,

Bidirectional RNNs

- The basic encoder–decoder architecture doesn't handle long sentences very well
- Everything must fit into a fixed-size vector,
- and RNNs remember recent items better

Bidirectional RNNs

- The basic encoder–decoder architecture doesn't handle long sentences very well
- Everything must fit into a fixed-size vector,
- and RNNs remember recent items better
- We can combine left-to-right and right-to-left RNNs to overcome these issues



What if . . .

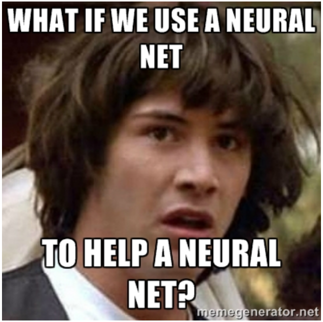
- Even bidirectional encoder–decoders have a hard time with long sentences

What if . . .

- Even bidirectional encoder–decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next

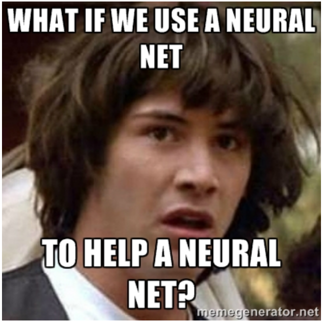
What if ...

- Even bidirectional encoder–decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next



What if . . .

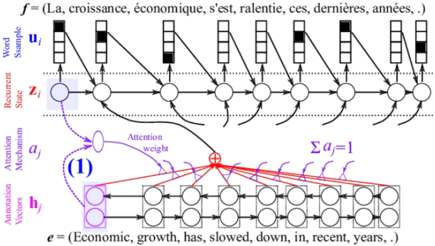
- Even bidirectional encoder–decoders have a hard time with long sentences
- We need a way to keep track of what's already been translated and what to translate next



- For neural nets, the solution is often more neural nets . . .

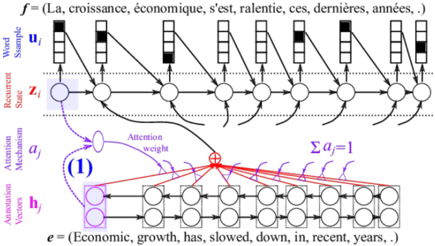
Achtung, Baby!

- Attention-based decoding adds another FF network (**a**) that takes as input the encoder's hidden state (**h**) and the decoder's hidden state (**z**), and outputs a probability for each source word at each time step (when and where to pay attention) :



Achtung, Baby!

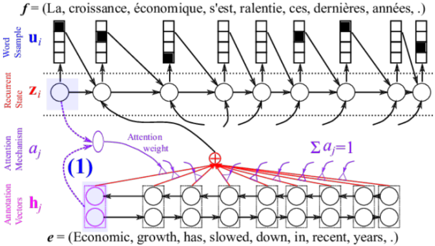
- Attention-based decoding adds another FF network (**a**) that takes as input the encoder's hidden state (**h**) and the decoder's hidden state (**z**), and outputs a probability for each source word at each time step (when and where to pay attention) :



- Because attention outputs probabilities, it requires expensive normalization (via softmax), at each decoding timestep

Achtung, Baby!

- Attention-based decoding adds another FF network (**a**) that takes as input the encoder's hidden state (**h**) and the decoder's hidden state (**z**), and outputs a probability for each source word at each time step (when and where to pay attention) :



- Because attention outputs probabilities, it requires expensive normalization (via softmax), at each decoding timestep
- The attention weights can also function as soft word alignments. They're trained on target-side MLE

Image Caption Generation

- You can use attention-based decoding to give textual descriptions of images

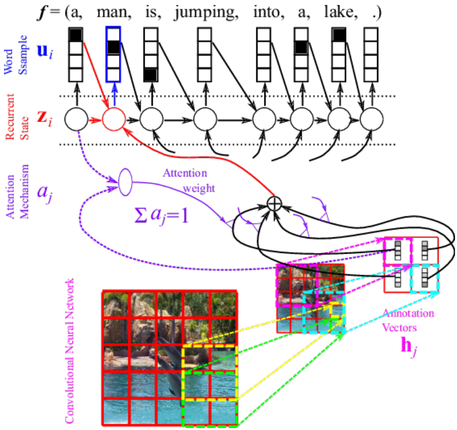
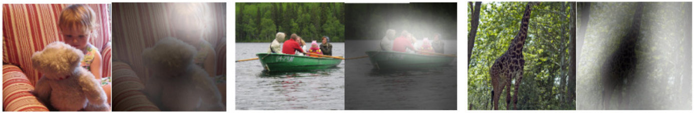


Image Caption Generation Examples

Figure 4. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

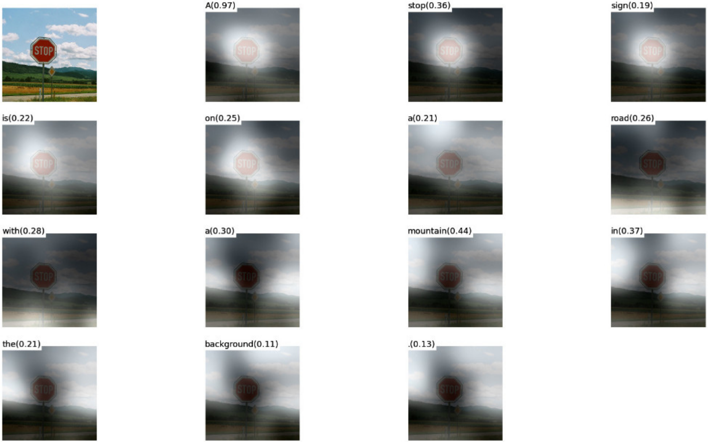
A woman is throwing a frisbee in a park. A dog is standing on a hardwood floor. A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear. A group of people sitting on a boat in the water. A giraffe standing in a forest with trees in the background.

Images courtesy of http://arxiv.org/abs/1502.03044

Image Caption Generation, Step by Step



(b) A stop sign is on a road with a mountain in the background.

And the bleeding edge: the transformer

"Multi-head self-attention" [Vaswani et al.]:

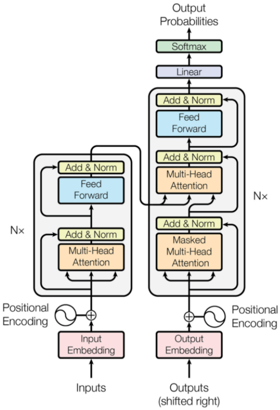
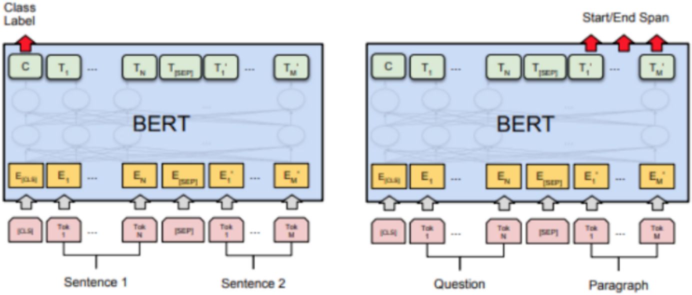


Figure 1: The Transformer - model architecture.

BERT

The bleedingest edge, based on transformer. Powerful, VERY expensive to train.



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(c) Question Answering Tasks: SQuAD v1.1

Why?

Why did we go through all that?

Why?

Why did we go through all that?

- For contextualization, just like for psycholinguistics. A bit more technical content.

Why?

Why did we go through all that?

- For contextualization, just like for psycholinguistics. A bit more technical content.
- To *model* incremental scope, we will likely need the "whole edifice" – it's higher-order semantics!

Why?

Why did we go through all that?

- For contextualization, just like for psycholinguistics. A bit more technical content.
- To *model* incremental scope, we will likely need the "whole edifice" – it's higher-order semantics!
- Current state of the art in *incremental* scope prediction is way below the "bleeding edge".

Part 2: corpora

The need for data

The sources of knowledge in incremental processing:

The need for data

The sources of knowledge in incremental processing:

- Formal syntax and semantics

The need for data

The sources of knowledge in incremental processing:

- Formal syntax and semantics
  - Opinions on how much this matters vary!

The need for data

The sources of knowledge in incremental processing:

- Formal syntax and semantics
    - Opinions on how much this matters vary!
- World knowledge ("Every jeweller appraised a diamond") and...

The need for data

The sources of knowledge in incremental processing:

- Formal syntax and semantics
  - Opinions on how much this matters vary!
- World knowledge ("Every jeweller appraised a diamond") and...
- A model of the "linear" sentence sequence itself!

The need for data

The sources of knowledge in incremental processing:

- Formal syntax and semantics
  - Opinions on how much this matters vary!
- World knowledge ("Every jeweller appraised a diamond") and...
- A model of the "linear" sentence sequence itself!

But that requires a lot of data.

BioScope

An early attempt: the BioScope corpus, Vincze et al. [2008].

BioScope

An early attempt: the BioScope corpus, Vincze et al. [2008].

- The problem: factual knowledge.

BioScope

An early attempt: the BioScope corpus, Vincze et al. [2008].

- The problem: factual knowledge.
  - Want to e.g. search medical databases for treatments.

BioScope

An early attempt: the BioScope corpus, Vincze et al. [2008].

- The problem: factual knowledge.
  - Want to e.g. search medical databases for treatments.
  - Need to detect uncertain and negative assertions as a filter.

BioScope

An early attempt: the BioScope corpus, Vincze et al. [2008].

- The problem: factual knowledge.
    - Want to e.g. search medical databases for treatments.
    - Need to detect uncertain and negative assertions as a filter.

BioScope is a negation and speculation scope corpus.

BioScope

Negation annotation in BioScope:

Stable appearance of the right kidney <without hydronephrosis>.
Surprisingly, however, <neither of these proteins bound in vitro to EBS1 or EBS2

BioScope

Negation annotation in BioScope:

Stable appearance of the right kidney <without hydronephrosis>.
Surprisingly, however, <neither of these proteins bound in vitro to EBS1 or EBS2

Speculation annotation in BioScope:

This is a 3 month old patient who had <possible pyelonephritis> with elevated fever.
<Atelectasis in the right mid zone is, however, possible>.

BioScope

Negation annotation in BioScope:

Stable appearance of the right kidney <without hydronephrosis>.
Surprisingly, however, <neither of these proteins bound in vitro to EBS1 or EBS2

Speculation annotation in BioScope:

This is a 3 month old patient who had <possible pyelonephritis> with elevated fever.
<Atelectasis in the right mid zone is, however, possible>.

Scope evidence may turn up at the end.

BioScope

More "interesting" cases:

The decrease was seen in patients who responded to the therapy as well as those who did <not>. ⇒ *ellipsis*

BioScope

More "interesting" cases:

The decrease was seen in patients who responded to the therapy as well as those who did <not>. ⇒ *ellipsis*

Overlapping scopes:

<Repression did <not seem to involve another factor whose activity is affected by the NSAIDs> >.
⇒ < <Repression did not seem to involve another factor whose activity is affected by the NSAIDs> >.

BioScope

More “interesting” cases:

The decrease was seen in patients who responded to the therapy as well as those who did <not>. ⇒ *ellipsis*

Overlapping scopes:

<Repression did <not seem to involve another factor whose activity is affected by the NSAIDs> >.
⇒ < <Repression did not seem to involve another factor whose activity is affected by the NSAIDs> >.

Avoid intersecting scopes by extending negation to the outermost scope.

BioScope

It's still not really that much data!

	Clinical	Full Paper	Abstract
#Documents	1954	9	1273
#Sentences	6383	2670	11871
Negation sentences	13.55%	12.70%	13.45%
#Negation cues	877	389	1848
Hedge sentences	13.39%	19.44%	17.70%
#Hedge cues	1189	714	2769

BioScope

It's still not really that much data!

	Clinical	Full Paper	Abstract
#Documents	1954	9	1273
#Sentences	6383	2670	11871
Negation sentences	13.55%	12.70%	13.45%
#Negation cues	877	389	1848
Hedge sentences	13.39%	19.44%	17.70%
#Hedge cues	1189	714	2769

And it's non-incremental...

- Was used in CoNNL-2010 shared task on detecting hedges.
- Simple classification to sequence models (HMMs etc) were applied often to high accuracy.

Quantifier annotations

Manshadi et al. (2011): corpus of text editor instructions

1. *Print [1/ every line] of [2/ the file] that starts with [3/ a digit] followed by [4/ punctuation].*

   QSD: {2>1, 2>3, 1>3, 2>4, 1>4}

2. *Delete [1/ the first character] of [2/ every word] and [3/ the first word] of [4/ every line] in [5/ the file].*

   QSD: {5>4, 5>3, 4>3, 5>2, 5>1, 2>1}

Figure 1. Two NP-chunked sentences with QSDs

Quantifier annotations

Manshadi et al. (2011): convert scope precedences to DAGs

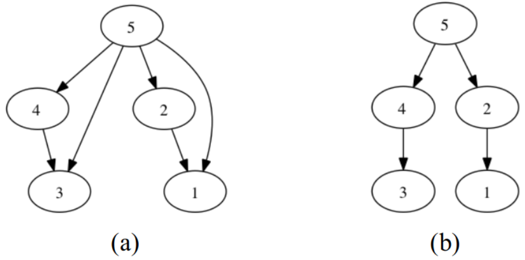


Figure 2. Scopings represented as DAGs

Quantifier annotations

Manshadi et al. (2011): classifier approach.

- For a given sentence, for every quantifier pair, they classify (SVM) whether or not there is a scope connection between them in the graph.
- Features: determiner type ("every", "the", etc.), head noun, syntactic dependency.
- Pairwise scope relation F-score: 73.2% (vs majority-class wide scope baseline of 30.3%).

Quantifier annotations

AnderBois et al. [2012]: investigate pragmatics of quantifier scope via corpus.

- Law School Admissions Test (LSAT) logic puzzles corpus.
- Not representative of English: contain a disproportionately high number of quantifiers relative to other English text.
- 497 quantifier scopes and resolutions.

Quantifier annotations

AnderBois et al. [2012]:

In the course of one month Garibaldi has exactly seven different meetings. Each of her meetings is with exactly one of five foreign dignitaries: Fuentes, Matsuba, Rhee, Soleimani, or Tbahi. The following constraints govern Garibaldi's meetings:

Introduction

She has exactly three meetings with Fuentes, and exactly one with each of the other dignitaries.
She does not have any meetings in a row with Fuentes.
Her meeting with Soleimani is the very next one after her meeting with Tbahi.
Neither the first nor last of her meetings is with Matsuba.

Laws

2. If Garibaldi's last meeting is with Rhee, then which one of the following could be true?

Question

(A) Garibaldi's second meeting is with Soleimani.
(B) Garibaldi's third meeting is with Matsuba.
(C) Garibaldi's fourth meeting is with Soleimani.
(D) Garibaldi's fifth meeting is with Matsuba.
(E) Garibaldi's sixth meeting is with Soleimani.

Answers

Quantifier annotations

AnderBois et al. [2012]

- They did not do a prediction task – instead, regression analyses.
- Confirmed previous literature that linear order and grammatical function have an effect on scope-taking.
- Confirmed that lexical effects are as important as lin. order and gramm. function.
- Confirmed that relations between quantifiers (remember the grammatical hierarchy) also affects interpretation.

(Just like psycholinguistic results from yesterday.)

Nothing so far has been strictly incremental. . .

Part 3: Filling the gap

Larger societal context: adaptiveness and usability

Dialog systems are an increasing part of daily life.

Larger societal context: adaptiveness and usability

Dialog systems are an increasing part of daily life. Consider Siri, Amazon Alexa, etc. Explicitly intended to handle general language.

Larger societal context: adaptiveness and usability

Dialog systems are an increasing part of daily life. Consider Siri, Amazon Alexa, etc. Explicitly intended to handle general language.

Potential scope ambiguities actually widespread in language [Koller and Thater, 2010], even if discourse and world context drastically reduces it.

Larger societal context: usability

Scope ambiguity with personal assistant.

Send every restaurant a reservation request.

Is there one reservation sent to all the restaurants (i.e. for single mass event)

or

does each restaurant receive a separate reservation request (as alternates)?

Larger societal context: usability

Scope ambiguity with personal assistant.

Send every restaurant a reservation request.

Is there one reservation sent to all the restaurants (i.e. for single mass event)

or

does each restaurant receive a separate reservation request (as alternates)? Normally the latter – our "common sense" tells us.

Larger societal context: usability

Scope ambiguity with personal assistant.

Send every restaurant a reservation request.

Is there one reservation sent to all the restaurants (i.e. for single mass event)

or

does each restaurant receive a separate reservation request (as alternates)? Normally the latter – our "common sense" tells us.

General-purpose models needed

Scope interaction with world/"common-sense" knowledge:

General-purpose models needed

Scope interaction with world/"common-sense" knowledge:

- Understanding $\Rightarrow$ being able to decide when linear scopes warranted.

General-purpose models needed

Scope interaction with world/"common-sense" knowledge:

- Understanding $\Rightarrow$ being able to decide when linear scopes warranted.
- Generation $\Leftarrow$ being able to present information without burdening the user with over- (or under-!) complex structures.

General-purpose models needed

Scope interaction with world/"common-sense" knowledge:

- Understanding $\Rightarrow$ being able to decide when linear scopes warranted.
- Generation $\Leftarrow$ being able to present information without burdening the user with over- (or under-!) complex structures.

(1) Every restaurant received a different reservation request.

General-purpose models needed

Scope interaction with world/"common-sense" knowledge:

- Understanding $\Rightarrow$ being able to decide when linear scopes warranted.
- Generation $\Leftarrow$ being able to present information without burdening the user with over- (or under-!) complex structures.

(1) Every restaurant received a different reservation request.

Needs to be made specific given user's common-sense knowledge?

Existing scope annotation schemes

Shows overall feasibility – but most existing approaches are non-incremental.

Existing scope annotation schemes

Shows overall feasibility – but most existing approaches are non-incremental.
Some existing efforts for quantifier scopes:

- AnderBois et al. [2012] – LSAT logic puzzles, 497 quantifier scopes and resolutions.
- Higgins and Sadock [2003] – 893 sentences from Penn Treebank.

Existing scope annotation schemes

Shows overall feasibility – but most existing approaches are non-incremental.
Some existing efforts for quantifier scopes:

- AnderBois et al. [2012] – LSAT logic puzzles, 497 quantifier scopes and resolutions.
- Higgins and Sadock [2003] – 893 sentences from Penn Treebank.

Other levels of scope representation:

- BioScope corpus [Vincze et al., 2008] – negation and uncertaintly in biomedical texts.
    - 20K sentences, 10% with potential meaning effects from scope ambiguity.
- Concill et al. – product reviews, 679 with negation annotations.

Adding another dimension: time

Elements of incremental scope annotation

- Unit of annotation: word-by-word, possibly focus on NP-boundaries.

Elements of incremental scope annotation

- Unit of annotation: word-by-word, possibly focus on NP-boundaries.
- Target of annotation: scope decisions

Elements of incremental scope annotation

- Unit of annotation: word-by-word, possibly focus on NP-boundaries.
- Target of annotation: scope decisions
  - How the processor updates expectation on an annotation-unit-by-unit basis.

Elements of incremental scope annotation

- Unit of annotation: word-by-word, possibly focus on NP-boundaries.
- Target of annotation: scope decisions
  - How the processor updates expectation on an annotation-unit-by-unit basis.
- General forms of scope decision annotation: Δ, $\Delta(\Gamma)$, or $\Delta(\Gamma, \Psi)$
  - Δ – decision "operation"
  - Γ – the specification made by the operation
  - Ψ – "justification" for the decision.

Decision operations

Possible values for Γ:

Decision operations

Possible values for Γ:

- **Quantifier introduction** ($\top$) – a quantified phrase enters the system, requiring a label for the quantifier and variable.

Decision operations

Possible values for Γ:

- **Quantifier introduction** (T) – a quantified phrase enters the system, requiring a label for the quantifier and variable.
- **Relation creation** (R) – two scope operators enter into a (possibly underspecified) scope precedence relationship.

Decision operations

Possible values for Γ:

- **Quantifier introduction** (T) – a quantified phrase enters the system, requiring a label for the quantifier and variable.
- **Relation creation** (R) – two scope operators enter into a (possibly underspecified) scope precedence relationship.
- **Specification** (S) – an underspecified relation is given a specified precedence is selected between two scope operators.

Decision operations

Possible values for Γ:

- **Quantifier introduction** (T) – a quantified phrase enters the system, requiring a label for the quantifier and variable.
- **Relation creation** (R) – two scope operators enter into a (possibly underspecified) scope precedence relationship.
- **Specification** (S) – an underspecified relation is given a specified precedence is selected between two scope operators.
- **Null** (N) – the word or phrase is not involved in a scope relation.

Specification

Part of annotation dependent on scope theory used.

Specification

Part of annotation dependent on scope theory used. Initially, use simple binary relations, where Q_n is a quantifier in the semantic representation:

Specification

Part of annotation dependent on scope theory used. Initially, use simple binary relations, where Q_n is a quantifier in the semantic representation:

- Q_1 – introduced quantifier
- $Q_1 = Q_2$ – potential scope relation exists, but is not specified.
- $Q_1 > Q_2$ – Q_1 scopes over Q_2

Specification

Part of annotation dependent on scope theory used. Initially, use simple binary relations, where Q_n is a quantifier in the semantic representation:

- Q_1 – introduced quantifier
- $Q_1 = Q_2$ – potential scope relation exists, but is not specified.
- $Q_1 > Q_2$ – Q_1 scopes over Q_2

Possible replace with e.g. quantifier raising operations?

Justification

Part of annotation dependent on pragmatic theory used. Simple initial scheme:

- Syntactic/structural (X) – the scope operation was applied because algorithmic or formal constraints require an interpretation to hold at that point.
- Pragmatic/knowledge-based (P) – the scope operation was applied because of information about the world or discourse context applied by the processor.

Annotation process

Word-by-word presentation to annotator.

(2) Every child climbed a tree ‖

$\mathrm{T}(\forall_1)$ N N $\mathrm{R}(\forall_1 = \exists_2,\mathrm{X})$ $\mathrm{S}(\forall_1 > \exists_2,\mathrm{P})$ ‖

First word introduces quantifier but no anticipatory information.

Annotation process

Word-by-word presentation to annotator.

(3)	Every	child	climbed	a	tree	$\|$
	$T(\forall_1)$	N	N	$R(\forall_1 = \exists_2, X)$	$S(\forall_1 > \exists_2, P)$	$\|$

Next two words do not introduce (possibly!) scope-relevant information.

Annotation process

Word-by-word presentation to annotator.

(4)	Every	child	climbed	a	tree	$\|$
	$T(\forall_1)$	N	N	$R(\forall_1 = \exists_2, X)$	$S(\forall_1 > \exists_2, P)$	$\|$

Article indicates there must be a relationship, but not which.

Annotation process

Word-by-word presentation to annotator.

(5)	Every	child	climbed	a	tree	$\|$
	$\mathrm{T}(\forall_1)$	N	N	$\mathrm{R}(\forall_1 = \exists_2, \mathrm{X})$	$\mathrm{S}(\forall_1 > \exists_2, \mathrm{P})$	$\|$

Finally, noun makes most implausible relationship obvious, for pragmatic reasons. Annotation of sentence completed.

Annotation process

Consider difference with:

(6) a. Every child a teacher picked climbed a tree

Every	child	a	teacher	picked	climbed	a	tree
$\mathrm{T}(\forall_1)$	N	$\mathrm{S}(\forall_1 > \exists_2,\mathrm{X})$	N	N	N	$\mathrm{R}(\forall_1 = \exists_3,\mathrm{X})$	$\mathrm{S}(\exists_3 > \forall_1,\mathrm{P})$

Annotation process

Consider difference with:

(6) a. Every child a teacher picked climbed a tree

Every	child	a	teacher	picked	climbed	a
$\mathrm{T}(\forall_1)$	N	$\mathrm{S}(\forall_1 > \exists_2,\mathrm{X})$	N	N	N	$\mathrm{R}(\forall_1 = \exists_3,\mathrm{X})$

tree
$\mathrm{S}(\exists_3 > \forall_1,\mathrm{P})$

Introduction of relative clause changes series of annotations by forcing some immediate specificaions early.

Open questions and future work

Which machine learning techniques to apply?

- For high-level Γ operations, is HMM-style sequence knowledge enough?

Open questions and future work

Which machine learning techniques to apply?

- For high-level Γ operations, is HMM-style sequence knowledge enough?
- More powerful deep learning for the actual scope specification operations (particularly if richer representation included)?

Open questions and future work

Which machine learning techniques to apply?

- For high-level Γ operations, is HMM-style sequence knowledge enough?
- More powerful deep learning for the actual scope specification operations (particularly if richer representation included)?
- Connection to knowledge-bases for justification annotations?

Tomorrow: more speculation, computational aspects