Natural Language Processing for Low-Resource and Marginalized Language Varieties

Pillar II: Learning

Sina Ahmadi (sina.ahmadi@uzh.ch)

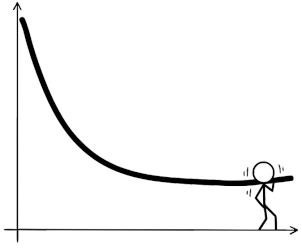
Department of Computational Linguistics

March 4, 2026

Recap

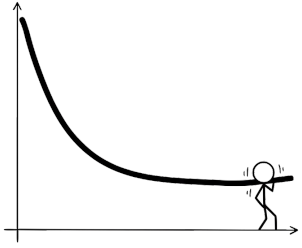
Pillar I: Data

- Low-resource languages suffer from the **long tail** of data availability



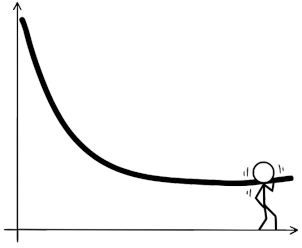
Pillar I: Data

- Low-resource languages suffer from the **long tail** of data availability
- Innovative approaches to bridge the gap:



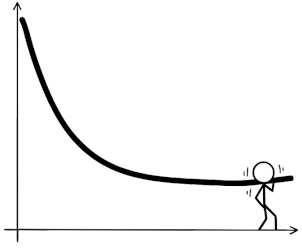
Pillar I: Data

- Low-resource languages suffer from the **long tail** of data availability
- Innovative approaches to bridge the gap:
  - **Collection**: crowdsourcing, community-driven efforts



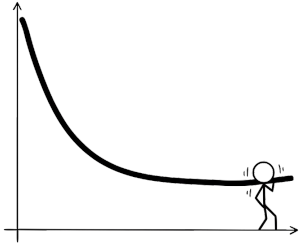
Pillar I: Data

- Low-resource languages suffer from the **long tail** of data availability
- Innovative approaches to bridge the gap:
  - **Collection**: crowdsourcing, community-driven efforts
  - **Annotation**: active learning, cross-lingual projection, LLM-as-annotator



Pillar I: Data

- Low-resource languages suffer from the **long tail** of data availability
- Innovative approaches to bridge the gap:
  - **Collection**: crowdsourcing, community-driven efforts
  - **Annotation**: active learning, cross-lingual projection, LLM-as-annotator
  - **Augmentation**: back-translation, bitext mining, BLI, gamification



Pillar II: Learning

Learn a model to map an input X into an output Y:

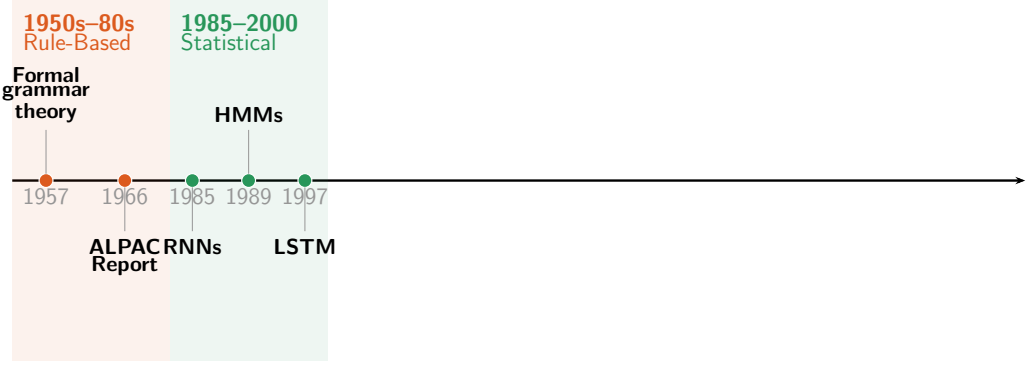
Input X	Output Y	Task
Text	Text in another language	Machine Translation
Text	Response	Dialog
Speech	Transcript	Speech Recognition
Text	Linguistic Structure	Language Analysis

But how?

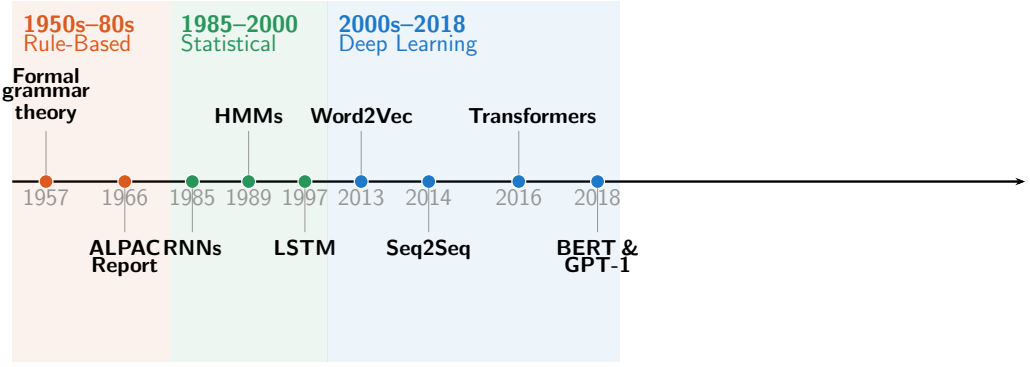
From X to Y

Paradigm Shifts in NLP

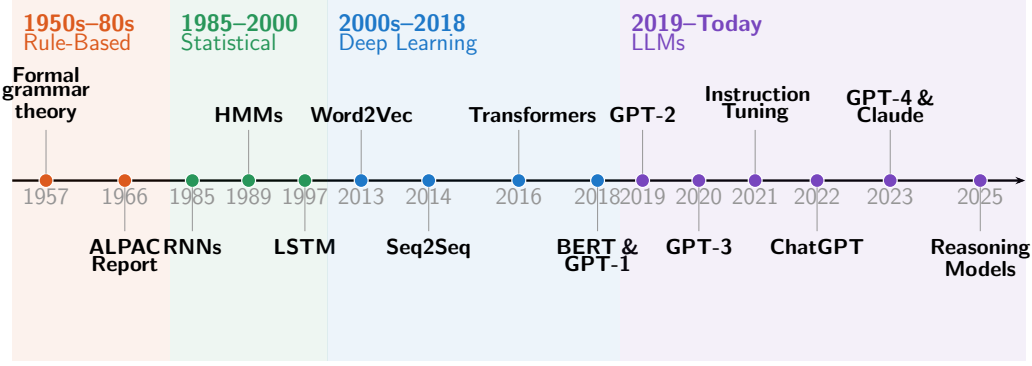
Paradigm Shifts in NLP



Paradigm Shifts in NLP



Paradigm Shifts in NLP



Rule-based Systems

- Develop rules from simple scripts to more complicated systems

Rule-based Systems

- Develop rules from simple scripts to more complicated systems
- Generally must be developed for each language by linguists

Rule-based Systems

- Develop rules from simple scripts to more complicated systems
- Generally must be developed for each language by linguists
- Appropriate for some tasks, e.g., spell checking with Hunspell

Rule-based Systems

- Develop rules from simple scripts to more complicated systems
- Generally must be developed for each language by linguists
- Appropriate for some tasks, e.g., spell checking with Hunspell
- Rule-based systems $\neq$ linguistically inspired approaches

Rule-based Systems

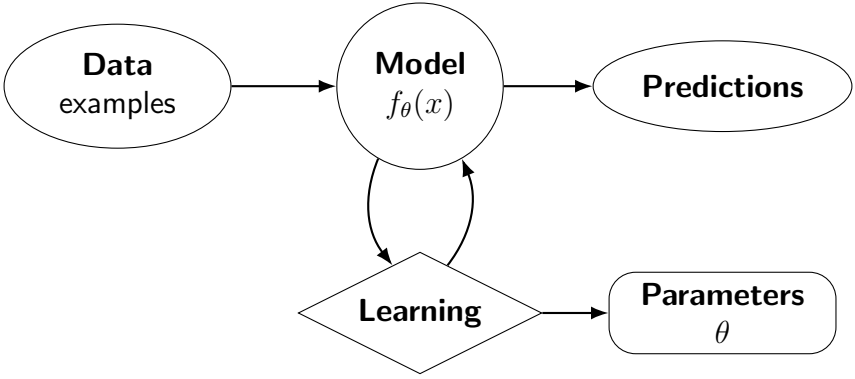
- Develop rules from simple scripts to more complicated systems
- Generally must be developed for each language by linguists
- Appropriate for some tasks, e.g., spell checking with Hunspell
- Rule-based systems $\neq$ linguistically inspired approaches

Example (Forms of feminine nouns in German in Hunspell)

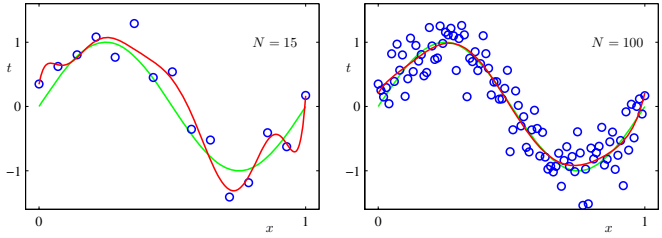
```
SFX F N 7
SFX F   0     nen          in
SFX F   e     in           e
SFX F   e     innen        e
SFX F   0     in           [^i]n
SFX F   0     innen        [^i]n
SFX F   0     in           [^en]
SFX F   0     innen        [^en]
```

Beyond rules: learn from the data

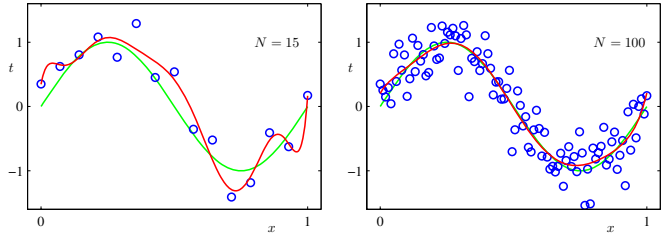
Learn a model to map an input X into an output Y: Adjust a model's parameters using data to capture patterns and improve prediction accuracy.



Learning: Example



Learning: Example



Efficiency of learning methods depends on:

- **Data**: quality, size, type
- **Task**: complexity, structure of the output space
- **Model**: capacity, inductive biases, number of parameters

Learning Paradigms

- **Supervised Learning**: Paired data $\langle X, Y \rangle$, source data X, target data Y e.g., machine translation (source $\rightarrow$ target)

Learning Paradigms

- **Supervised Learning**: Paired data $\langle X, Y \rangle$, source data X, target data Y
  e.g., machine translation (source $\rightarrow$ target)
- **Unsupervised Learning**: Find hidden patterns in unlabeled data X without Y
  e.g., clustering, topic modeling, language modeling

Learning Paradigms

- **Supervised Learning**: Paired data $\langle X, Y \rangle$, source data X, target data Y
  e.g., machine translation (source $\rightarrow$ target)
- **Unsupervised Learning**: Find hidden patterns in unlabeled data X without Y
  e.g., clustering, topic modeling, language modeling
- **Reinforcement Learning**: An agent learns by interacting with an environment and receiving rewards
  e.g., dialog systems

Statistical Learning

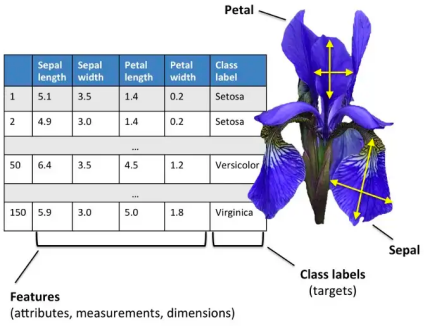
- Learn probability distributions over data

Statistical Learning

- Learn probability distributions over data
- Relies on **hand-crafted features**

Statistical Learning

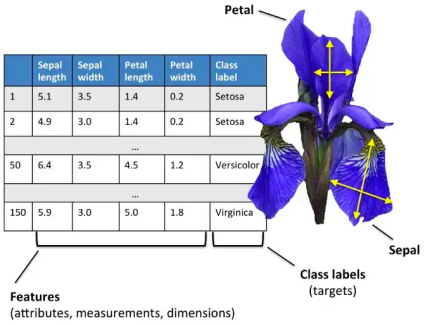
- Learn probability distributions over data
- Relies on **hand-crafted features**
- Naive Bayes, HMMs, CRFs, log-linear models, SVMs



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Statistical Learning

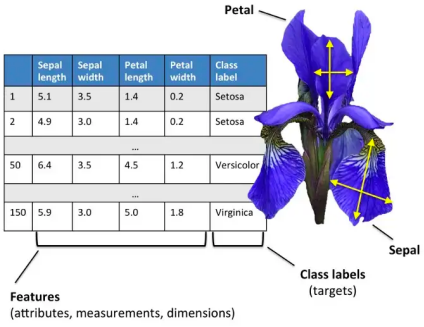
- Learn probability distributions over data
- Relies on **hand-crafted features**
- Naive Bayes, HMMs, CRFs, log-linear models, SVMs
- POS tagging, NER, statistical machine translation (e.g., Moses)



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Statistical Learning

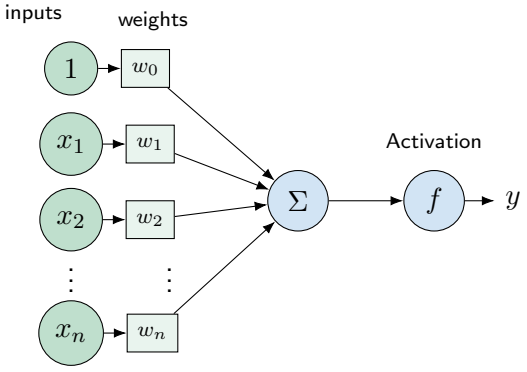
- Learn probability distributions over data
- Relies on **hand-crafted features**
- Naive Bayes, HMMs, CRFs, log-linear models, SVMs
- POS tagging, NER, statistical machine translation (e.g., Moses)
- Feature engineering is labor-intensive and does not scale across languages or domains



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

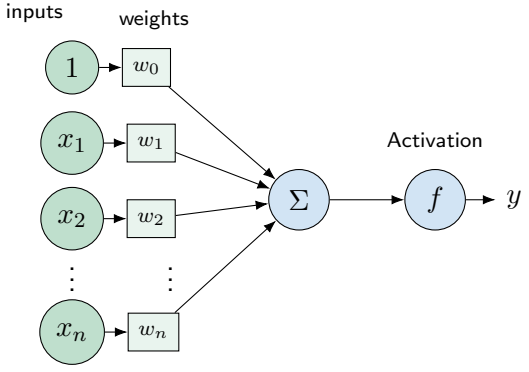
Neural Networks: The Basics

- A **perceptron**: weighted sum of inputs passed through an activation function



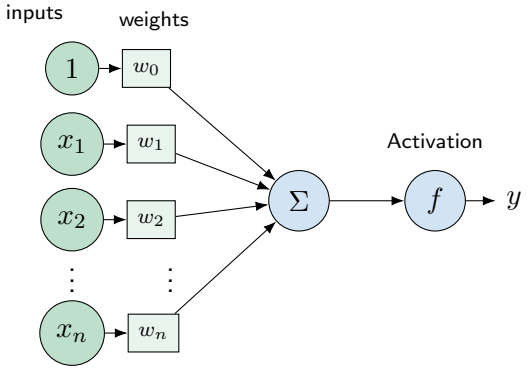
Neural Networks: The Basics

- A **perceptron**: weighted sum of inputs passed through an activation function
- A single perceptron can learn linearly separable functions (a single perceptron can't learn XOR)



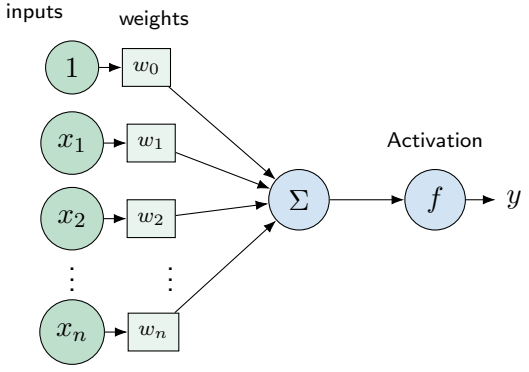
Neural Networks: The Basics

- A **perceptron**: weighted sum of inputs passed through an activation function
- A single perceptron can learn linearly separable functions (a single perceptron can't learn XOR)
- Stack layers $\rightarrow$ **Multi-Layer Perceptron** (MLP)



Neural Networks: The Basics

- A **perceptron**: weighted sum of inputs passed through an activation function
- A single perceptron can learn linearly separable functions (a single perceptron can't learn XOR)
- Stack layers $\rightarrow$ **Multi-Layer Perceptron** (MLP)
- Non-linear activation functions allow learning complex boundaries



Deep Learning

- Called "deep" because of **multiple stacked layers**

Deep Learning

- Called "deep" because of **multiple stacked layers**
- Each layer learns increasingly abstract representations

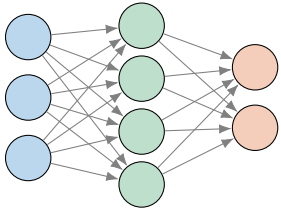
Deep Learning

- Called "deep" because of **multiple stacked layers**
- Each layer learns increasingly abstract representations
- No manual feature engineering: features are learned from data

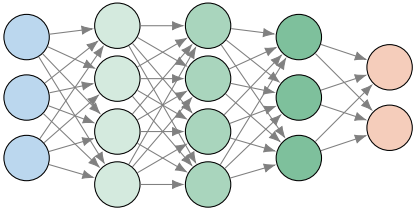
Deep Learning

- Called "deep" because of **multiple stacked layers**
- Each layer learns increasingly abstract representations
- No manual feature engineering: features are learned from data
- **Challenge**: requires large amounts of data and compute

Shallow



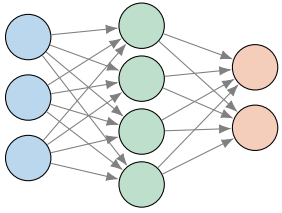
Deep



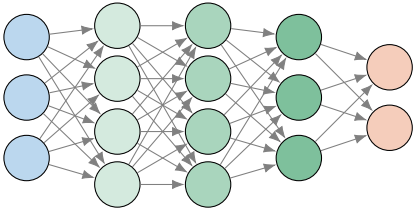
Deep Learning

- Called "deep" because of **multiple stacked layers**
- Each layer learns increasingly abstract representations
- No manual feature engineering: features are learned from data
- **Challenge**: requires large amounts of data and compute
- Breakthrough applications: image recognition, machine tranlsation, speech recognition

Shallow

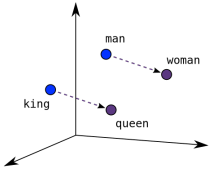


Deep

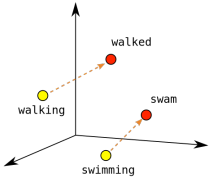


Learning Word Representations

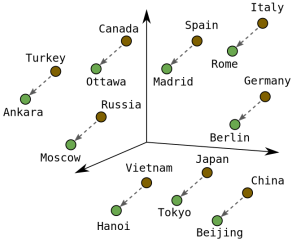
- Represent words as dense vectors in a continuous space



Male-Female



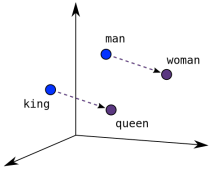
Verb Tense



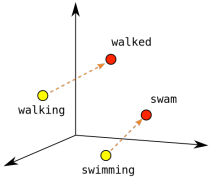
Country-Capital

Learning Word Representations

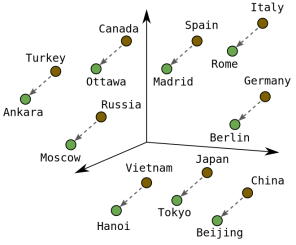
- Represent words as dense vectors in a continuous space
- Words appearing in similar contexts get **similar representations**



Male-Female



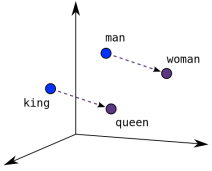
Verb Tense



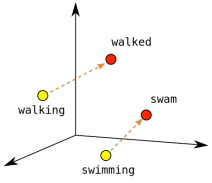
Country-Capital

Learning Word Representations

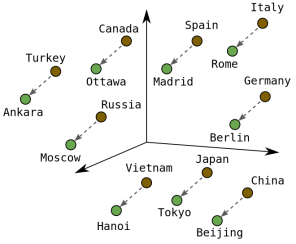
- Represent words as dense vectors in a continuous space
- Words appearing in similar contexts get **similar representations**
- Enabled transfer of lexical knowledge across tasks



Male-Female



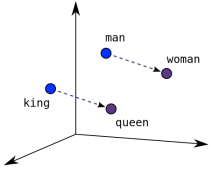
Verb Tense



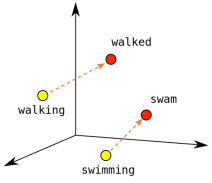
Country-Capital

Learning Word Representations

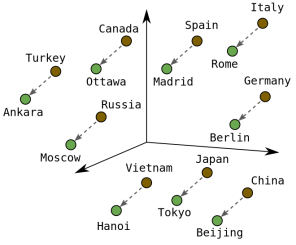
- Represent words as dense vectors in a continuous space
- Words appearing in similar contexts get **similar representations**
- Enabled transfer of lexical knowledge across tasks
- Foundation for all subsequent neural NLP work



Male-Female



Verb Tense



Country-Capital

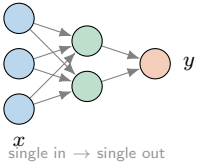
Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state

Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state
- Exploit the **sequential nature of language**: each word conditions on the previous context

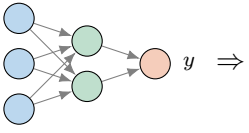
Feedforward



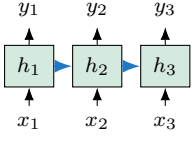
Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state
- Exploit the **sequential nature of language**: each word conditions on the previous context
- Language modeling, machine translation (seq2seq), text generation

Feedforward

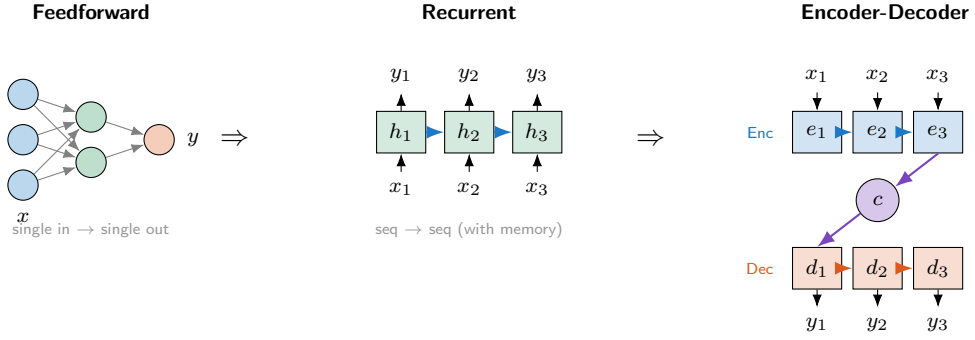


Recurrent



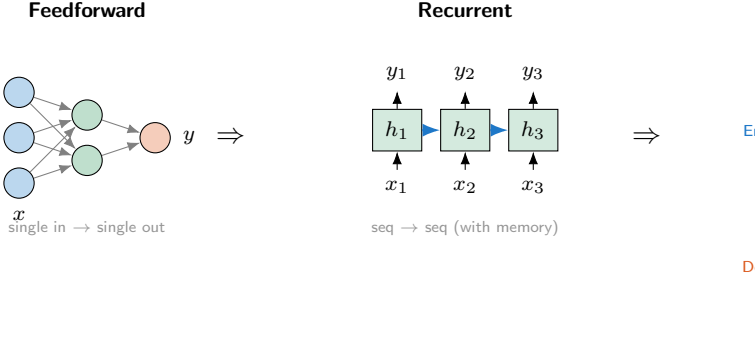
Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state
- Exploit the **sequential nature of language**: each word conditions on the previous context
- Language modeling, machine translation (seq2seq), text generation
- **Encoder-decoder**: one network reads input, another generates output



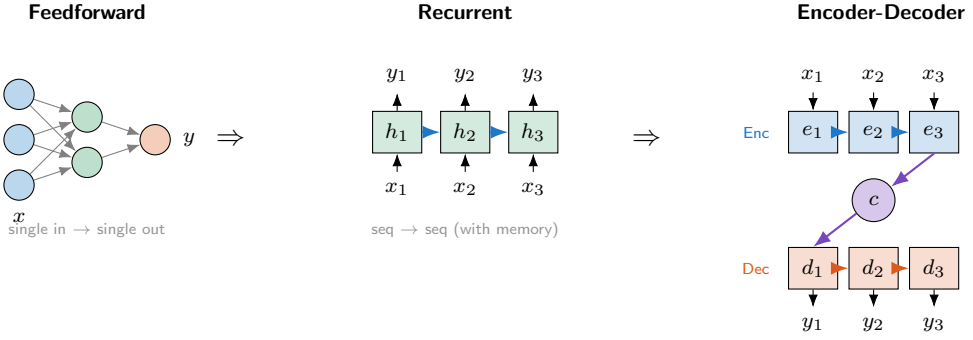
Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state
- Exploit the **sequential nature of language**: each word conditions on the previous context
- Language modeling, machine translation (seq2seq), text generation
- **Encoder-decoder**: one network reads input, another generates output
- This became the standard for machine translation (Sutskever et al., 2014)



Recurrent Neural Networks (RNNs)

- Process sequences step-by-step, maintaining a hidden state
- Exploit the **sequential nature of language**: each word conditions on the previous context
- Language modeling, machine translation (seq2seq), text generation
- **Encoder-decoder**: one network reads input, another generates output
- This became the standard for machine translation (Sutskever et al., 2014)
- Vanilla RNNs struggle with long-range dependencies ($\rightarrow$ LSTM)

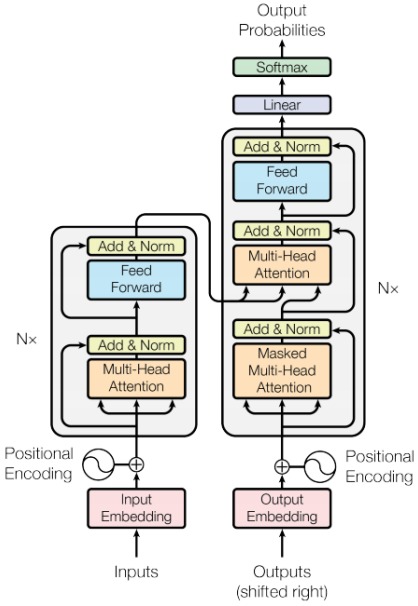


Attention Mechanism

- **Problem**: Encoder-decoder models compress entire input into a single fixed-size vector

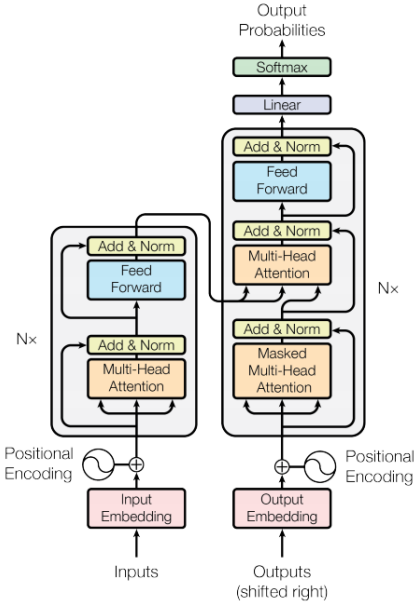
Attention Mechanism

- **Problem**: Encoder-decoder models compress entire input into a single fixed-size vector
- **Solution**: Let the decoder attend to **all** encoder states at each step (Bahdanau et al., 2014)



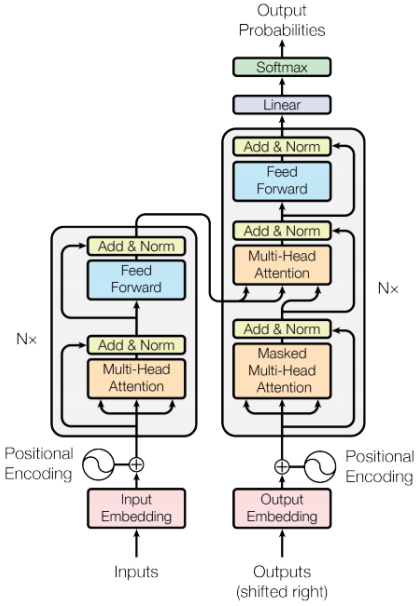
Attention Mechanism

- **Problem**: Encoder-decoder models compress entire input into a single fixed-size vector
- **Solution**: Let the decoder attend to **all** encoder states at each step (Bahdanau et al., 2014)
- The model learns *where to look* for each output token (Vaswani et al., 2017)



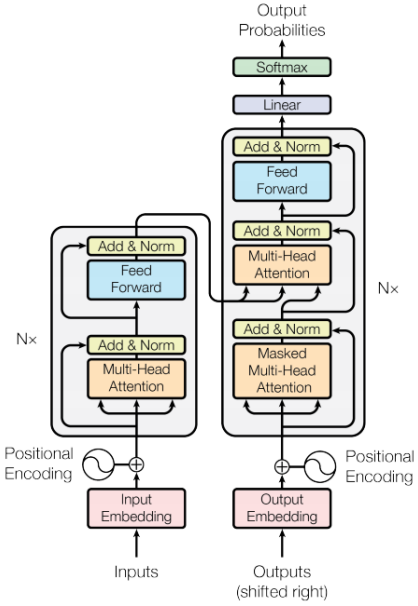
Attention Mechanism

- **Problem**: Encoder-decoder models compress entire input into a single fixed-size vector
- **Solution**: Let the decoder attend to **all** encoder states at each step (Bahdanau et al., 2014)
- The model learns *where to look* for each output token (Vaswani et al., 2017)
- Enabled scaling to billions of parameters

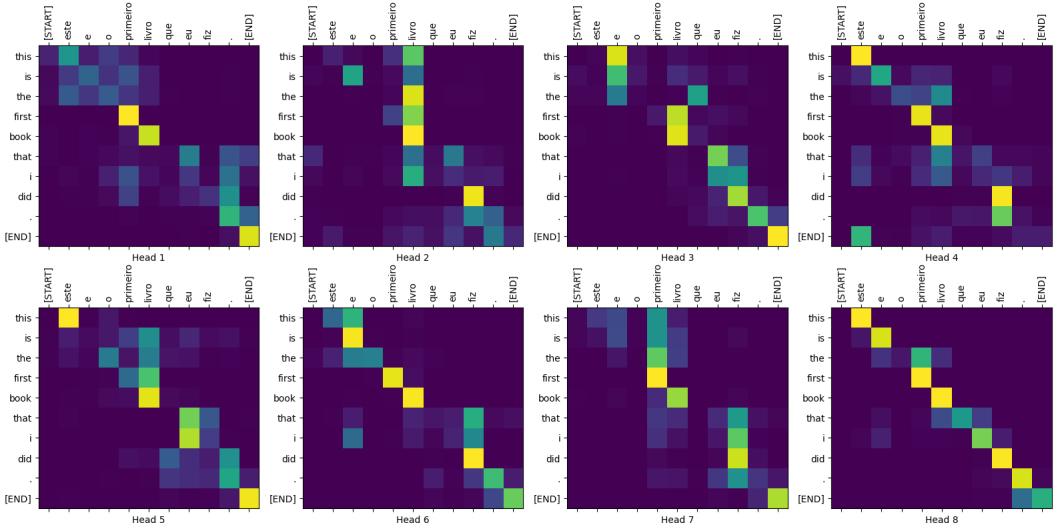


Attention Mechanism

- **Problem**: Encoder-decoder models compress entire input into a single fixed-size vector
- **Solution**: Let the decoder attend to **all** encoder states at each step (Bahdanau et al., 2014)
- The model learns *where to look* for each output token (Vaswani et al., 2017)
- Enabled scaling to billions of parameters
- See The Illustrated Transformer



Attention Mechanism: Example



Based on https://www.tensorflow.org/text/tutorials/transformer

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**
  - Predict the next token left-to-right

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**
  - Predict the next token left-to-right
  - Scaled up massively: GPT-2, GPT-3, GPT-4, ...

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**
  - Predict the next token left-to-right
  - Scaled up massively: GPT-2, GPT-3, GPT-4, ...
  - Excels at generation tasks: text completion, dialog, translation

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**
  - Predict the next token left-to-right
  - Scaled up massively: GPT-2, GPT-3, GPT-4, ...
  - Excels at generation tasks: text completion, dialog, translation
- Both achieved state-of-the-art on virtually all NLP benchmarks

BERT and GPT: Pre-trained Language Models

- **BERT** (Devlin et al., 2019): **Masked Language Modeling**
  - Mask random tokens, predict them from bidirectional context
  - Pre-train on large corpora, fine-tune on downstream tasks
  - Excels at understanding tasks: NER, classification, QA
- **GPT** (Radford et al., 2018): **Autoregressive Language Modeling**
  - Predict the next token left-to-right
  - Scaled up massively: GPT-2, GPT-3, GPT-4, ...
  - Excels at generation tasks: text completion, dialog, translation
- Both achieved state-of-the-art on virtually all NLP benchmarks
- **For low-resource**: multilingual variants (mBERT, XLM-R, BLOOM) are key

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):
  - **Zero-shot**: "Translate this sentence to Ladin: *The weather is nice.*"

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):
  - **Zero-shot**: "Translate this sentence to Ladin: *The weather is nice.*"
  - **Few-shot**: Provide a handful of input-output examples in the prompt

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):
  - **Zero-shot**: "Translate this sentence to Ladin: *The weather is nice.*"
  - **Few-shot**: Provide a handful of input-output examples in the prompt
  - **Chain-of-thought**: "Let's think step by step..."

Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):
  - **Zero-shot**: "Translate this sentence to Ladin: *The weather is nice.*"
  - **Few-shot**: Provide a handful of input-output examples in the prompt
  - **Chain-of-thought**: "Let's think step by step..."
- **Caveat**: LLM knowledge of low-resource languages is shallow

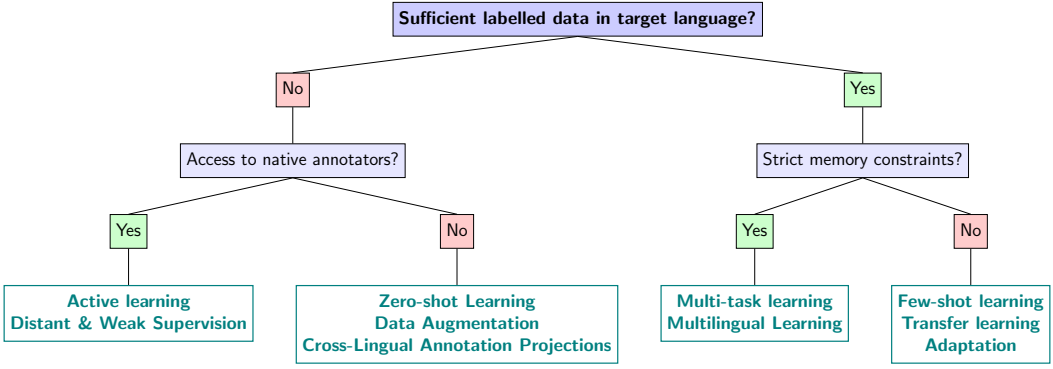
Prompt Engineering

- With large pre-trained models, **how you ask** matters as much as the model itself
- **Prompt**: a natural language instruction or template given to the model
- No fine-tuning needed $\rightarrow$ especially attractive for low-resource settings
- Examples (Liu et al., 2023):
  - **Zero-shot**: "Translate this sentence to Ladin: *The weather is nice.*"
  - **Few-shot**: Provide a handful of input-output examples in the prompt
  - **Chain-of-thought**: "Let's think step by step..."
- **Caveat**: LLM knowledge of low-resource languages is shallow
- Performance highly sensitive to prompt phrasing and example selection

Approaches to Low-Resource NLP

Approaches to Low-Resource NLP

How to effectively train models for low-resource languages?



Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest

Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.

Zero-shot Transfer

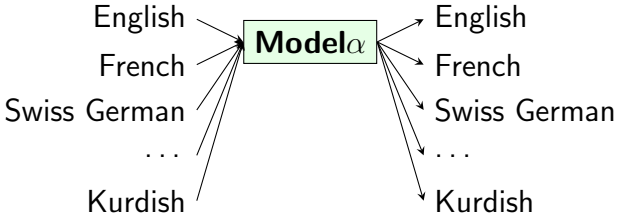
- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data

Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data
- Training

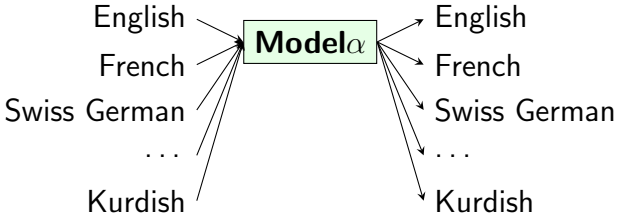
Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data
- Training



Zero-shot Transfer

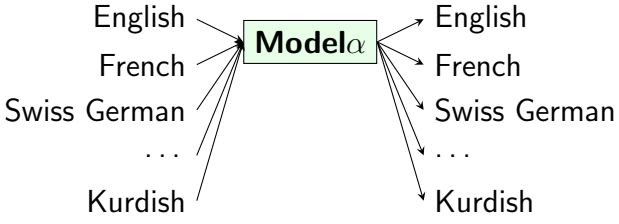
- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data
- Training



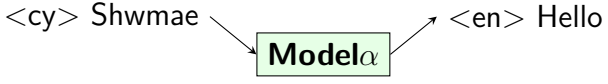
- Test

Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data
- Training

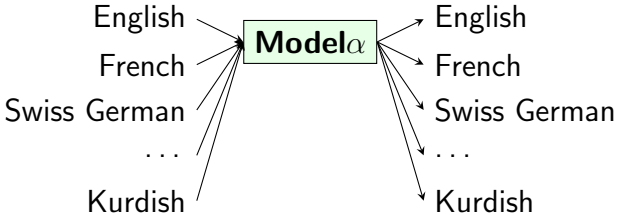


- Test

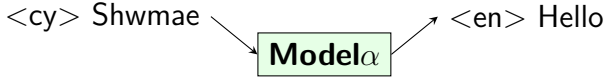


Zero-shot Transfer

- Suppose that we have no data for a language (pair) of interest
- **Idea:** leverage information from high-resource languages to help improve performance on low-resource languages.
- **Zero-shot transfer**: translate without any parallel data
- Training



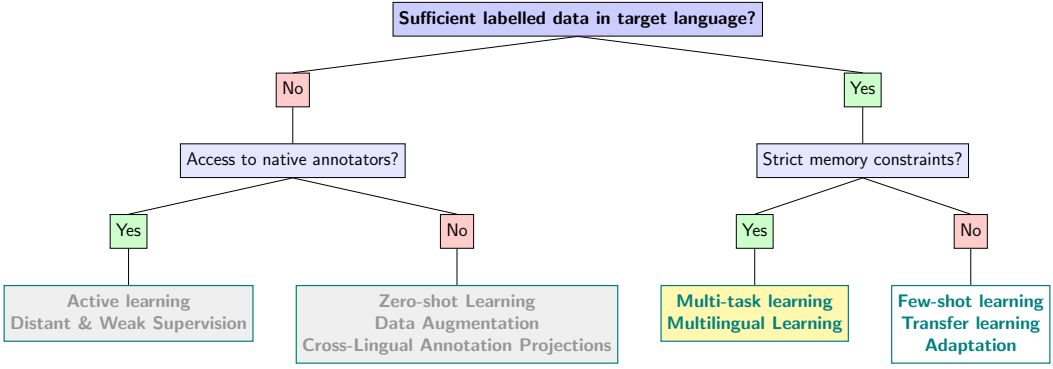
- Test



- Use monolingual data too: corrupt a sentence, train the model to reconstruct it (denoising)

Approaches to Low-Resource NLP

How to effectively train models for low-resource languages?



Multi-task Learning

- Train a single model on **multiple related tasks simultaneously**

Multi-task Learning

- Train a single model on **multiple related tasks simultaneously**
- Shared representations across tasks act as an inductive bias

Multi-task Learning

- Train a single model on **multiple related tasks simultaneously**
- Shared representations across tasks act as an inductive bias
- **Idea**: knowledge from auxiliary tasks can improve performance on the main task

Multi-task Learning

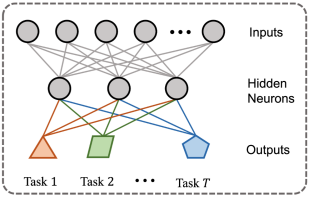
- Train a single model on **multiple related tasks simultaneously**
- Shared representations across tasks act as an inductive bias
- **Idea**: knowledge from auxiliary tasks can improve performance on the main task
- Especially helpful when the target task has limited data

Multi-task Learning

- Train a single model on **multiple related tasks simultaneously**
- Shared representations across tasks act as an inductive bias
- **Idea**: knowledge from auxiliary tasks can improve performance on the main task
- Especially helpful when the target task has limited data
- Example: jointly train NER, POS tagging, and dependency parsing

Multi-task Learning

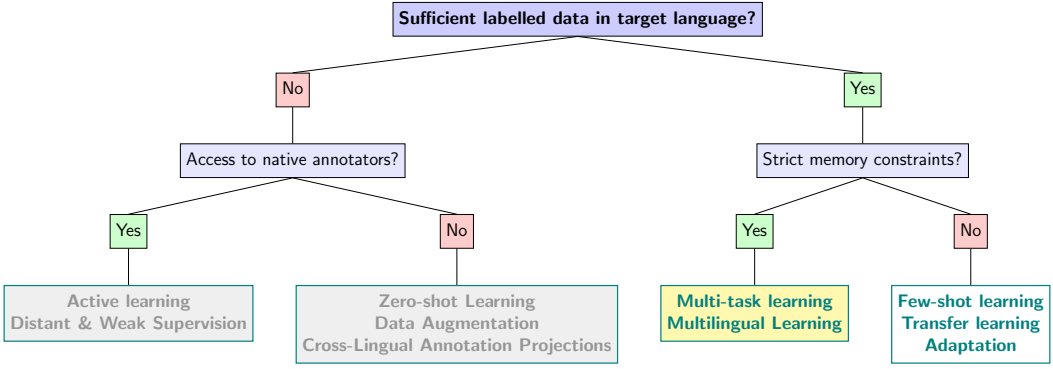
- Train a single model on **multiple related tasks simultaneously**
- Shared representations across tasks act as an inductive bias
- **Idea**: knowledge from auxiliary tasks can improve performance on the main task
- Especially helpful when the target task has limited data
- Example: jointly train NER, POS tagging, and dependency parsing



Multi-task learning framework (Yu et al., 2024)

Approaches to Low-Resource NLP

How to effectively train models for low-resource languages?

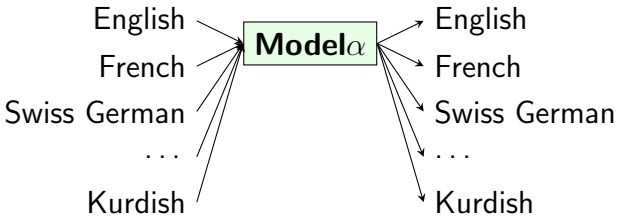


Multilingual Learning

- Training a single model on a mixed dataset from multiple languages

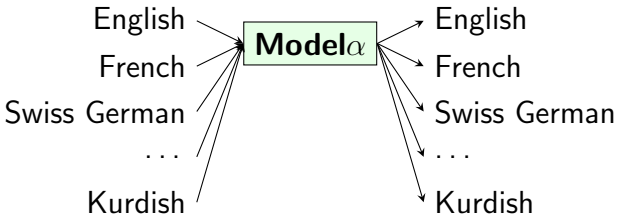
Multilingual Learning

- Training a single model on a mixed dataset from multiple languages



Multilingual Learning

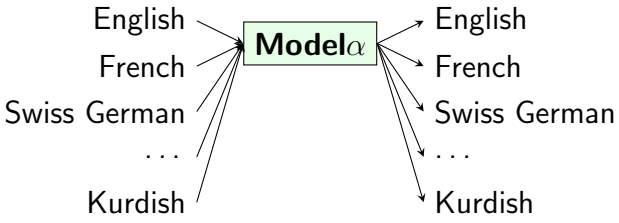
- Training a single model on a mixed dataset from multiple languages



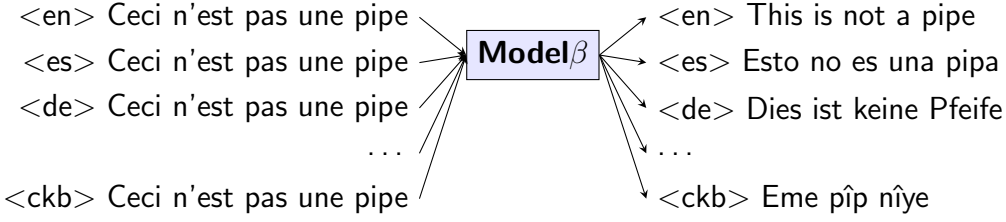
- To specify target language, simply add a language tag!

Multilingual Learning

- Training a single model on a mixed dataset from multiple languages



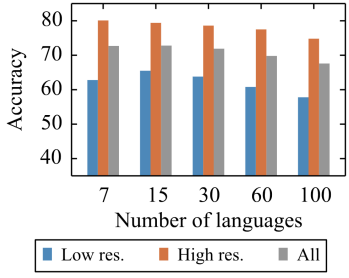
- To specify target language, simply add a language tag!



Multilingual Learning: A Zero-Sum Game?

- How to deal with the **curse of multilinguality** (Conneau et al., 2020) or **capacity bottleneck** (Arivazhagan et al., 2019)?
  Multilingual pretraining is characterized with severe data imbalance: In what proportions should we balance the pretraining languages?

Multilingual Learning: A Zero-Sum Game?



- How to deal with the **curse of multilinguality** (Conneau et al., 2020) or **capacity bottleneck** (Arivazhagan et al., 2019)?
  Multilingual pretraining is characterized with severe data imbalance: In what proportions should we balance the pretraining languages?
- Increasing the number of low-resource languages $\rightarrow$ decrease in the quality of high-resource language translations

Multilingual Learning: Weighting Strategies

Multilingual Learning: Weighting Strategies

- Find a trade-off between **language coverage** and **performance**

Multilingual Learning: Weighting Strategies

- Find a trade-off between **language coverage** and **performance**
- Upsample low-resource languages $\rightarrow$ model overfit

Multilingual Learning: Weighting Strategies

- Find a trade-off between **language coverage** and **performance**
- Upsample low-resource languages $\rightarrow$ model overfit
- Downsample high-resource languages $\rightarrow$ model underfit

Multilingual Learning: Weighting Strategies

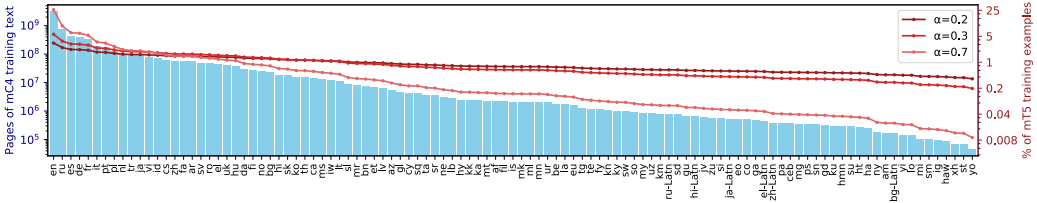
- Find a trade-off between **language coverage** and **performance**
- Upsample low-resource languages $\rightarrow$ model overfit
- Downsample high-resource languages $\rightarrow$ model underfit
- Uniform: Sample/weight all tasks with equal probability

Multilingual Learning: Weighting Strategies

- Find a trade-off between **language coverage** and **performance**
- Upsample low-resource languages $\rightarrow$ model overfit
- Downsample high-resource languages $\rightarrow$ model underfit
- Uniform: Sample/weight all tasks with equal probability
- Proportional: Sample/weight tasks according to data size

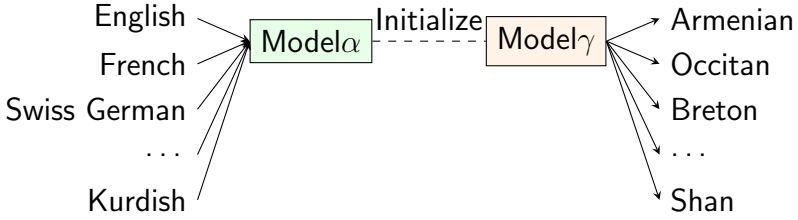
Multilingual Learning: Weighting Strategies

- Find a trade-off between **language coverage** and **performance**
- Upsample low-resource languages $\rightarrow$ model overfit
- Downsample high-resource languages $\rightarrow$ model underfit
- Uniform: Sample/weight all tasks with equal probability
- Proportional: Sample/weight tasks according to data size
- Temperature-based: Sample tasks according to data size exponentiated by $1/\tau$

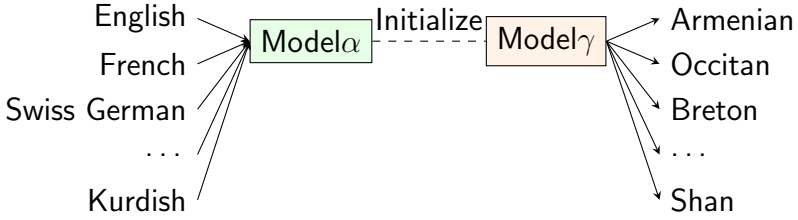


Vocabulary sampling in mT5 (Xue et al., 2020)

Multilingual Learning: Parameter Sharing

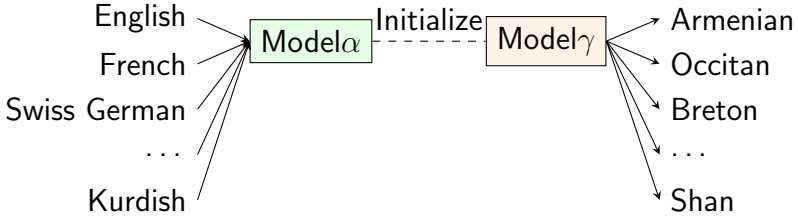


Multilingual Learning: Parameter Sharing



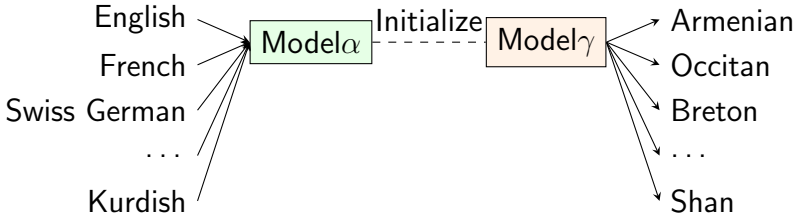
- Share all parameters
  e.g. single model for all domains

Multilingual Learning: Parameter Sharing



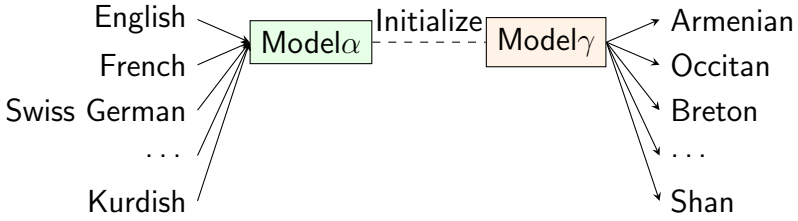
- Share all parameters
  e.g. single model for all domains
- Share some model components, not others
  e.g. share encoder or attention mechanism, separate decoder

Multilingual Learning: Parameter Sharing



- Share all parameters
  e.g. single model for all domains
- Share some model components, not others
  e.g. share encoder or attention mechanism, separate decoder
- Very small number of unshared parameters
  e.g. a single embedding specifying the language or domain

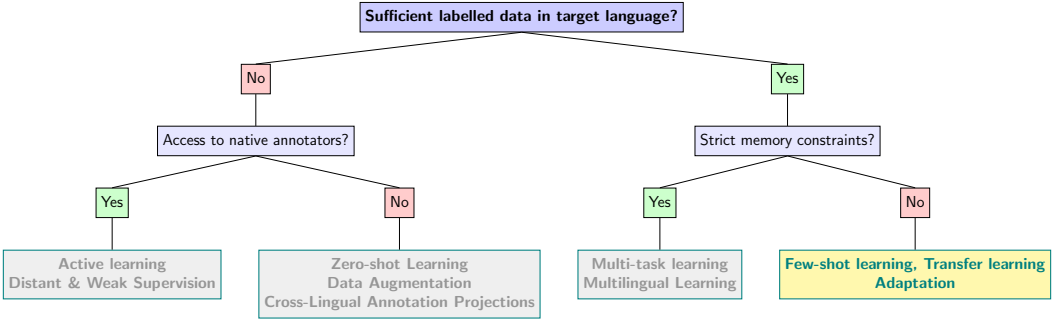
Multilingual Learning: Parameter Sharing



- Share all parameters
  e.g. single model for all domains
- Share some model components, not others
  e.g. share encoder or attention mechanism, separate decoder
- Very small number of unshared parameters
  e.g. a single embedding specifying the language or domain
- **Parameter efficient tuning**: prompt tuning, prefix tuning, adapters, compacters and etc

Approaches to Low-Resource NLP

How to effectively train models for low-resource languages?



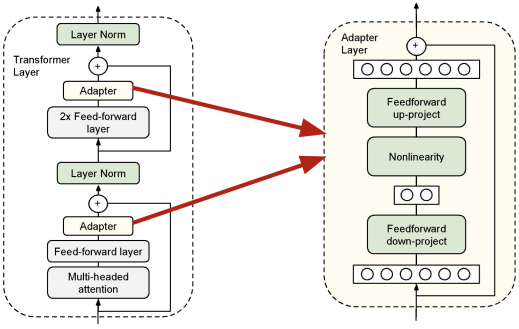
Adapters

Adapters

- Adaptation: Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages

Adapters

- Adaptation: Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages
- Add small trainable sub-networks that can be added post-hoc



"Parameter-Efficient Transfer Learning for NLP." Houslby et al. 2019.

Adapters

- Adaptation: Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages
- Add small trainable sub-networks that can be added post-hoc
- The original model weights are fixed; just the adapter modules are tuned

Adapters

- Adaptation: Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages
- Add small trainable sub-networks that can be added post-hoc
- The original model weights are fixed; just the adapter modules are tuned
- Can be used for multitask or multi-lingual learning

Multilingual and Zero-Shot Learning: Considerations

- When transferring from another language, let it be **similar** to the target language and **high-resourced**

Multilingual and Zero-Shot Learning: Considerations

- When transferring from another language, let it be **similar** to the target language and **high-resourced**
- Go beyond scripts: use phonological representations to make the similarity between languages apparent

Multilingual and Zero-Shot Learning: Considerations

- When transferring from another language, let it be **similar** to the target language and **high-resourced**
- Go beyond scripts: use phonological representations to make the similarity between languages apparent
- Zero shot transfers well to language with different scripts, esp. for languages with little vocabulary overlap

Multilingual and Zero-Shot Learning: Considerations

- When transferring from another language, let it be **similar** to the target language and **high-resourced**
- Go beyond scripts: use phonological representations to make the similarity between languages apparent
- Zero shot transfers well to language with different scripts, esp. for languages with little vocabulary overlap
- Zero shot does not work well for typologically different languages, e.g. fine-tune in English, test on Japanese

Multilingual and Zero-Shot Learning: Considerations

- When transferring from another language, let it be **similar** to the target language and **high-resourced**
- Go beyond scripts: use phonological representations to make the similarity between languages apparent
- Zero shot transfers well to language with different scripts, esp. for languages with little vocabulary overlap
- Zero shot does not work well for typologically different languages, e.g. fine-tune in English, test on Japanese
- When fine-tuning a pre-trained model on a new task or language, it may **forget** what it previously learned $\rightarrow$ catastrophic forgetting

Takeaways

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:
  - Zero-shot and cross-lingual transfer from high-resource languages

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:
  - Zero-shot and cross-lingual transfer from high-resource languages
  - Multi-task and multilingual learning to share knowledge across languages

Takeaways

- NLP has evolved from rules $\rightarrow$ statistical models $\rightarrow$ neural networks $\rightarrow$ pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:
  - Zero-shot and cross-lingual transfer from high-resource languages
  - Multi-task and multilingual learning to share knowledge across languages
  - Parameter-efficient adaptation (adapters) to avoid catastrophic forgetting

Next week: How do we know if our models actually work?
$\rightarrow$ *Pillar III: Evaluation*

Takeaways

- NLP has evolved from rules → statistical models → neural networks → pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:
  - Zero-shot and cross-lingual transfer from high-resource languages
  - Multi-task and multilingual learning to share knowledge across languages
  - Parameter-efficient adaptation (adapters) to avoid catastrophic forgetting
- No single approach works for all languages: choose based on available resources and typological similarity

Next week: How do we know if our models actually work?
→ *Pillar III: Evaluation*

Takeaways

- NLP has evolved from rules → statistical models → neural networks → pre-trained LLMs
- Each paradigm shift reduced the need for manual engineering but increased the need for data
- For low-resource languages, the key strategies are:
  - Zero-shot and cross-lingual transfer from high-resource languages
  - Multi-task and multilingual learning to share knowledge across languages
  - Parameter-efficient adaptation (adapters) to avoid catastrophic forgetting
- No single approach works for all languages: choose based on available resources and typological similarity
- The bottleneck is shifting from *models* to *data and evaluation*

Next week: How do we know if our models actually work?
→ *Pillar III: Evaluation*

References

N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9): 195:1–195:35, 2023. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815.

A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

J. Yu, Y. Dai, X. Liu, J. Huang, Y. Shen, K. Zhang, R. Zhou, E. Adhikarla, W. Ye, Y. Liu, et al. Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras. *arXiv preprint arXiv:2404.18961*, 2024.