

Machine Learning and Grammar Induction

CSC2540S Machine Learning and Universal Grammar
Department of Computer Science, University of Toronto

Shalom Lappin*
Department of Philosophy
King's College, London

* Joint work with Alex Clark
Department of Computer Science
Royal Holloway College, London

March 3, 2009

Outline

- 1 The Machine Learning Paradigm
- 2 Supervised Learning with a Probabilistic Grammar
- 3 Unsupervised Parsing
- 4 NL Engineering and Human Language Acquisition
- 5 Conclusions

Machine Learning Algorithms and Models of the Learning Domain

- A machine learning system implements a learning algorithm that defines a function from a domain of input samples to a range of output values.
- A corpus of examples is divided into a training and a test set.
- The learning algorithm is specified in conjunction with a model of the phenomenon to be learned.

Machine Learning Algorithms and Models of the Learning Domain

- A machine learning system implements a learning algorithm that defines a function from a domain of input samples to a range of output values.
- A corpus of examples is divided into a training and a test set.
- The learning algorithm is specified in conjunction with a model of the phenomenon to be learned.

Machine Learning Algorithms and Models of the Learning Domain

- A machine learning system implements a learning algorithm that defines a function from a domain of input samples to a range of output values.
- A corpus of examples is divided into a training and a test set.
- The learning algorithm is specified in conjunction with a model of the phenomenon to be learned.

Machine Learning Algorithms and Models of the Learning Domain

- This model defines the space of possible hypotheses that the algorithm can generate from the input data.
- The values of its parameters are set through training of the algorithm on the training set.
- When the parameter values of the model are determined, an element of the hypothesis space is selected.

Machine Learning Algorithms and Models of the Learning Domain

- This model defines the space of possible hypotheses that the algorithm can generate from the input data.
- The values of its parameters are set through training of the algorithm on the training set.
- When the parameter values of the model are determined, an element of the hypothesis space is selected.

Machine Learning Algorithms and Models of the Learning Domain

- This model defines the space of possible hypotheses that the algorithm can generate from the input data.
- The values of its parameters are set through training of the algorithm on the training set.
- When the parameter values of the model are determined, an element of the hypothesis space is selected.

Evaluating a Parsing Algorithm

Supervised Learning

- If one has a gold standard of correct parses in a corpus, then it is possible to compute the percentage of correct parses that the algorithm produces for a blind test subpart of this corpus.
- A more common procedure for scoring an ML algorithm on a test set is to determine its performance for *recall* and *precision*.

Evaluating a Parsing Algorithm

Supervised Learning

- If one has a gold standard of correct parses in a corpus, then it is possible to compute the percentage of correct parses that the algorithm produces for a blind test subpart of this corpus.
- A more common procedure for scoring an ML algorithm on a test set is to determine its performance for *recall* and *precision*.

Precision, Recall, and F-Score

- The *recall* of a parsing algorithm \mathcal{A} is the percentage of labeled brackets of the test set that it correctly identifies.
- \mathcal{A} 's *precision* is the percentage of the brackets that it returns which correspond to those in the gold standard.
- A unified score for \mathcal{A} , known as an *F score*, can be computed as a weighted average of its recall and its precision.

Precision, Recall, and F-Score

- The *recall* of a parsing algorithm \mathcal{A} is the percentage of labeled brackets of the test set that it correctly identifies.
- \mathcal{A} 's *precision* is the percentage of the brackets that it returns which correspond to those in the gold standard.
- A unified score for \mathcal{A} , known as an *F score*, can be computed as a weighted average of its recall and its precision.

Precision, Recall, and F-Score

- The *recall* of a parsing algorithm \mathcal{A} is the percentage of labeled brackets of the test set that it correctly identifies.
- \mathcal{A} 's *precision* is the percentage of the brackets that it returns which correspond to those in the gold standard.
- A unified score for \mathcal{A} , known as an *F score*, can be computed as a weighted average of its recall and its precision.

Learning Biases and Priors

- The choice of parameters and their range of values defines a bias for the language model by imposing prior constraints on the set of possible hypotheses.
- All learning requires some sort of bias to restrict the set of possible hypotheses for the phenomenon to be learned.
- This bias can express strong assumptions about the nature of the domain of learning.
- Alternatively, it can define comparatively weak domain-specific constraints, with learning driven primarily by domain-general procedures and conditions.

Learning Biases and Priors

- The choice of parameters and their range of values defines a bias for the language model by imposing prior constraints on the set of possible hypotheses.
- All learning requires some sort of bias to restrict the set of possible hypotheses for the phenomenon to be learned.
- This bias can express strong assumptions about the nature of the domain of learning.
- Alternatively, it can define comparatively weak domain-specific constraints, with learning driven primarily by domain-general procedures and conditions.

Learning Biases and Priors

- The choice of parameters and their range of values defines a bias for the language model by imposing prior constraints on the set of possible hypotheses.
- All learning requires some sort of bias to restrict the set of possible hypotheses for the phenomenon to be learned.
- This bias can express strong assumptions about the nature of the domain of learning.
- Alternatively, it can define comparatively weak domain-specific constraints, with learning driven primarily by domain-general procedures and conditions.

Learning Biases and Priors

- The choice of parameters and their range of values defines a bias for the language model by imposing prior constraints on the set of possible hypotheses.
- All learning requires some sort of bias to restrict the set of possible hypotheses for the phenomenon to be learned.
- This bias can express strong assumptions about the nature of the domain of learning.
- Alternatively, it can define comparatively weak domain-specific constraints, with learning driven primarily by domain-general procedures and conditions.

Prior Probability Distributions on a Hypothesis Space

- One way of formalising this learning bias is a prior probability distribution on the elements of the hypothesis space that favours some hypotheses as more likely than others.
- The paradigm of Bayesian learning in cognitive science implements this approach.
- The simplicity and compactness measure that Perfors et al. (2006) use is an example of a very general prior.

Prior Probability Distributions on a Hypothesis Space

- One way of formalising this learning bias is a prior probability distribution on the elements of the hypothesis space that favours some hypotheses as more likely than others.
- The paradigm of Bayesian learning in cognitive science implements this approach.
- The simplicity and compactness measure that Perfors et al. (2006) use is an example of a very general prior.

Prior Probability Distributions on a Hypothesis Space

- One way of formalising this learning bias is a prior probability distribution on the elements of the hypothesis space that favours some hypotheses as more likely than others.
- The paradigm of Bayesian learning in cognitive science implements this approach.
- The simplicity and compactness measure that Perfors et al. (2006) use is an example of a very general prior.

Learning Bias and the Poverty of Stimulus

- The poverty of stimulus issue can be formulated as follows:

What are the minimal domain-specific linguistic biases that must be assumed for a reasonable learning algorithm to support language acquisition on the basis of the training set available to the child?

- If a model with relatively weak language-specific biases can sustain effective grammar induction, then this result undermines poverty of stimulus arguments for a rich theory of universal grammar.

Learning Bias and the Poverty of Stimulus

- The poverty of stimulus issue can be formulated as follows:

What are the minimal domain-specific linguistic biases that must be assumed for a reasonable learning algorithm to support language acquisition on the basis of the training set available to the child?

- If a model with relatively weak language-specific biases can sustain effective grammar induction, then this result undermines poverty of stimulus arguments for a rich theory of universal grammar.

Learning Bias and the Poverty of Stimulus

- The poverty of stimulus issue can be formulated as follows:

What are the minimal domain-specific linguistic biases that must be assumed for a reasonable learning algorithm to support language acquisition on the basis of the training set available to the child?

- If a model with relatively weak language-specific biases can sustain effective grammar induction, then this result undermines poverty of stimulus arguments for a rich theory of universal grammar.

Supervised Learning

- When the samples of the training set are annotated with the classifications and structures that the learning algorithm is intended to produce as output for the test set, then learning is supervised.
- Supervised grammar induction involves training an ML procedure on a corpus annotated with the parse structures of the gold standard.
- The learning algorithm infers a function for assigning input sentences to appropriate parse output on the basis of a training set of sentence argument-parse value pairs.

Supervised Learning

- When the samples of the training set are annotated with the classifications and structures that the learning algorithm is intended to produce as output for the test set, then learning is supervised.
- Supervised grammar induction involves training an ML procedure on a corpus annotated with the parse structures of the gold standard.
- The learning algorithm infers a function for assigning input sentences to appropriate parse output on the basis of a training set of sentence argument-parse value pairs.

Supervised Learning

- When the samples of the training set are annotated with the classifications and structures that the learning algorithm is intended to produce as output for the test set, then learning is supervised.
- Supervised grammar induction involves training an ML procedure on a corpus annotated with the parse structures of the gold standard.
- The learning algorithm infers a function for assigning input sentences to appropriate parse output on the basis of a training set of sentence argument-parse value pairs.

Unsupervised Learning

- If the test set is not marked with the properties to be returned as output for the test set, then learning is unsupervised.
- Unsupervised learning involves using clustering patterns and distributional regularities in a training set to identify structure in the data.

Unsupervised Learning

- If the test set is not marked with the properties to be returned as output for the test set, then learning is unsupervised.
- Unsupervised learning involves using clustering patterns and distributional regularities in a training set to identify structure in the data.

Supervised Learning and Language Acquisition

- It could be argued that supervised grammar induction is not directly relevant to poverty of stimulus arguments.
- It requires that target parse structures be represented in the training set, while children have no access to such representations in the data they are exposed to.
- If negative evidence of the sort identified by Saxton (1997, 2000) and Chouinard and Clark (2003) is available and plays a role in grammar induction, then it is possible to model the acquisition process as a type of supervised learning.
- If, however, children achieve language solely on the basis of positive evidence, then it is necessary to treat acquisition as unsupervised learning.

Supervised Learning and Language Acquisition

- It could be argued that supervised grammar induction is not directly relevant to poverty of stimulus arguments.
- It requires that target parse structures be represented in the training set, while children have no access to such representations in the data they are exposed to.
- If negative evidence of the sort identified by Saxton (1997, 2000) and Chouinard and Clark (2003) is available and plays a role in grammar induction, then it is possible to model the acquisition process as a type of supervised learning.
- If, however, children achieve language solely on the basis of positive evidence, then it is necessary to treat acquisition as unsupervised learning.

Supervised Learning and Language Acquisition

- It could be argued that supervised grammar induction is not directly relevant to poverty of stimulus arguments.
- It requires that target parse structures be represented in the training set, while children have no access to such representations in the data they are exposed to.
- If negative evidence of the sort identified by Saxton (1997, 2000) and Chouinard and Clark (2003) is available and plays a role in grammar induction, then it is possible to model the acquisition process as a type of supervised learning.
- If, however, children achieve language solely on the basis of positive evidence, then it is necessary to treat acquisition as unsupervised learning.

Supervised Learning and Language Acquisition

- It could be argued that supervised grammar induction is not directly relevant to poverty of stimulus arguments.
- It requires that target parse structures be represented in the training set, while children have no access to such representations in the data they are exposed to.
- If negative evidence of the sort identified by Saxton (1997, 2000) and Chouinard and Clark (2003) is available and plays a role in grammar induction, then it is possible to model the acquisition process as a type of supervised learning.
- If, however, children achieve language solely on the basis of positive evidence, then it is necessary to treat acquisition as unsupervised learning.

Probabilistic Context-Free Grammars

- A Probabilistic Context-Free Grammar (PCFG) conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- It provides conditional probabilities of the form $P(X_1 \cdots X_n \mid N)$ for each nonterminal N and sequence $X_1 \cdots X_n$ of items from the vocabulary of the grammar.
- It also specifies a probability distribution over the label of the root of the tree $P_S(N)$.
- The conditional probabilities $P(X_1 \cdots X_n \mid N)$ correspond to probabilistic parameters that govern the expansion of a node in a parse tree according to a context free rule $N \rightarrow X_1 \cdots X_n$.

Probabilistic Context-Free Grammars

- A Probabilistic Context-Free Grammar (PCFG) conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- It provides conditional probabilities of the form $P(X_1 \cdots X_n \mid N)$ for each nonterminal N and sequence $X_1 \cdots X_n$ of items from the vocabulary of the grammar.
- It also specifies a probability distribution over the label of the root of the tree $P_S(N)$.
- The conditional probabilities $P(X_1 \cdots X_n \mid N)$ correspond to probabilistic parameters that govern the expansion of a node in a parse tree according to a context free rule $N \rightarrow X_1 \cdots X_n$.

Probabilistic Context-Free Grammars

- A Probabilistic Context-Free Grammar (PCFG) conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- It provides conditional probabilities of the form $P(X_1 \cdots X_n \mid N)$ for each nonterminal N and sequence $X_1 \cdots X_n$ of items from the vocabulary of the grammar.
- It also specifies a probability distribution over the label of the root of the tree $P_S(N)$.
- The conditional probabilities $P(X_1 \cdots X_n \mid N)$ correspond to probabilistic parameters that govern the expansion of a node in a parse tree according to a context free rule $N \rightarrow X_1 \cdots X_n$.

Probabilistic Context-Free Grammars

- A Probabilistic Context-Free Grammar (PCFG) conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- It provides conditional probabilities of the form $P(X_1 \cdots X_n \mid N)$ for each nonterminal N and sequence $X_1 \cdots X_n$ of items from the vocabulary of the grammar.
- It also specifies a probability distribution over the label of the root of the tree $P_S(N)$.
- The conditional probabilities $P(X_1 \cdots X_n \mid N)$ correspond to probabilistic parameters that govern the expansion of a node in a parse tree according to a context free rule $N \rightarrow X_1 \cdots X_n$.

Probabilistic Context-Free Grammars

- The probabilistic parameter values of a PCFG can be learned from a parse annotated training corpus by computing the frequency of CFG rules in accordance with a Maximum Likelihood Expectation (MLE) condition.

$$\frac{c(A \rightarrow \beta_1 \dots \beta_k)}{c(A \rightarrow \gamma)}$$

- Statistical models of this kind have achieved F-measures in the low 70% range against the Penn Tree Bank (Marcus (1993)).

Probabilistic Context-Free Grammars

- The probabilistic parameter values of a PCFG can be learned from a parse annotated training corpus by computing the frequency of CFG rules in accordance with a Maximum Likelihood Expectation (MLE) condition.

$$\frac{c(A \rightarrow \beta_1 \dots \beta_k)}{c(A \rightarrow \gamma)}$$

- Statistical models of this kind have achieved F-measures in the low 70% range against the Penn Tree Bank (Marcus (1993)).

Probabilistic Context-Free Grammars

- The probabilistic parameter values of a PCFG can be learned from a parse annotated training corpus by computing the frequency of CFG rules in accordance with a Maximum Likelihood Expectation (MLE) condition.

$$\frac{c(A \rightarrow \beta_1 \dots \beta_k)}{c(A \rightarrow \gamma)}$$

- Statistical models of this kind have achieved F-measures in the low 70% range against the Penn Tree Bank (Marcus (1993)).

Lexicalized Probabilistic Context-Free Grammars

- It is possible to significantly improve the performance of a PCFG by adding additional bias to the language model that it defines.
- Collins (1999) constructs a Lexicalized Probabilistic Context-Free Grammar (LPCFG) in which the probabilities of the CFG rules are conditioning on lexical heads of the phrases that nonterminal symbols represent.
- In Collins' LPCFGs nonterminals are replaced by nonterminal/head pairs.

Lexicalized Probabilistic Context-Free Grammars

- It is possible to significantly improve the performance of a PCFG by adding additional bias to the language model that it defines.
- Collins (1999) constructs a Lexicalized Probabilistic Context-Free Grammar (LPCFG) in which the probabilities of the CFG rules are conditioning on lexical heads of the phrases that nonterminal symbols represent.
- In Collins' LPCFGs nonterminals are replaced by nonterminal/head pairs.

Lexicalized Probabilistic Context-Free Grammars

- It is possible to significantly improve the performance of a PCFG by adding additional bias to the language model that it defines.
- Collins (1999) constructs a Lexicalized Probabilistic Context-Free Grammar (LPCFG) in which the probabilities of the CFG rules are conditioning on lexical heads of the phrases that nonterminal symbols represent.
- In Collins' LPCFGs nonterminals are replaced by nonterminal/head pairs.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \dots H/h \dots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \dots H/h \dots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \dots H/h \dots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.

Bias in a LPCFG

- Rather than encoding a particular categorical bias into his language model by excluding certain context-free rules, Collins allows all such rules.
- He incorporates bias by adjusting the prior distribution of probabilities over the lexicalized CFG rules.
- The model imposes the requirements that
 - sentences have hierarchical constituent structure,
 - constituents have heads that select for their siblings, and
 - this selection is determined by the head words of the siblings.

Bias in a LPCFG

- Rather than encoding a particular categorical bias into his language model by excluding certain context-free rules, Collins allows all such rules.
- He incorporates bias by adjusting the prior distribution of probabilities over the lexicalized CFG rules.
- The model imposes the requirements that
 - sentences have hierarchical constituent structure,
 - constituents have heads that select for their siblings, and
 - this selection is determined by the head words of the siblings.

Bias in a LPCFG

- Rather than encoding a particular categorical bias into his language model by excluding certain context-free rules, Collins allows all such rules.
- He incorporates bias by adjusting the prior distribution of probabilities over the lexicalized CFG rules.
- The model imposes the requirements that
 - sentences have hierarchical constituent structure,
 - constituents have heads that select for their siblings, and
 - this selection is determined by the head words of the siblings.

Bias in a LPCFG

- Rather than encoding a particular categorical bias into his language model by excluding certain context-free rules, Collins allows all such rules.
- He incorporates bias by adjusting the prior distribution of probabilities over the lexicalized CFG rules.
- The model imposes the requirements that
 - sentences have hierarchical constituent structure,
 - constituents have heads that select for their siblings, and
 - this selection is determined by the head words of the siblings.

Bias in a LPCFG

- Rather than encoding a particular categorical bias into his language model by excluding certain context-free rules, Collins allows all such rules.
- He incorporates bias by adjusting the prior distribution of probabilities over the lexicalized CFG rules.
- The model imposes the requirements that
 - sentences have hierarchical constituent structure,
 - constituents have heads that select for their siblings, and
 - this selection is determined by the head words of the siblings.

LPCFG as a Weak Bias Model

- The bias that Collins, and Charniak and Johnson specify for their respective LPCFGs do not express the complex syntactic parameters that have been proposed as elements of a strong bias view of UG.
- So, for example, these models do not contain a head-complement directionality parameter.
- However, they still learn the correct generalizations concerning head-complement order.
- The bias of a statistical parsing model has implications for the design of UG.

LPCFG as a Weak Bias Model

- The bias that Collins, and Charniak and Johnson specify for their respective LPCFGs do not express the complex syntactic parameters that have been proposed as elements of a strong bias view of UG.
- So, for example, these models do not contain a head-complement directionality parameter.
- However, they still learn the correct generalizations concerning head-complement order.
- The bias of a statistical parsing model has implications for the design of UG.

LPCFG as a Weak Bias Model

- The bias that Collins, and Charniak and Johnson specify for their respective LPCFGs do not express the complex syntactic parameters that have been proposed as elements of a strong bias view of UG.
- So, for example, these models do not contain a head-complement directionality parameter.
- However, they still learn the correct generalizations concerning head-complement order.
- The bias of a statistical parsing model has implications for the design of UG.

LPCFG as a Weak Bias Model

- The bias that Collins, and Charniak and Johnson specify for their respective LPCFGs do not express the complex syntactic parameters that have been proposed as elements of a strong bias view of UG.
- So, for example, these models do not contain a head-complement directionality parameter.
- However, they still learn the correct generalizations concerning head-complement order.
- The bias of a statistical parsing model has implications for the design of UG.

An All Possible Constituents Approach

- Initial experiments with unsupervised grammar induction (like those described in Carroll and Charniak (1992)) were not particularly encouraging.
- Far more promising results have been achieved in recent work.
- Klein and Manning (K&M) (2002) propose a method that learns constituent structure from POS tagged input by unsupervised techniques.
- It assigns probability values to all subsequences of tagged elements in an input string, construed as possible constituents in a tree.

An All Possible Constituents Approach

- Initial experiments with unsupervised grammar induction (like those described in Carroll and Charniak (1992)) were not particularly encouraging.
- Far more promising results have been achieved in recent work.
- Klein and Manning (K&M) (2002) propose a method that learns constituent structure from POS tagged input by unsupervised techniques.
- It assigns probability values to all subsequences of tagged elements in an input string, construed as possible constituents in a tree.

An All Possible Constituents Approach

- Initial experiments with unsupervised grammar induction (like those described in Carroll and Charniak (1992)) were not particularly encouraging.
- Far more promising results have been achieved in recent work.
- Klein and Manning (K&M) (2002) propose a method that learns constituent structure from POS tagged input by unsupervised techniques.
- It assigns probability values to all subsequences of tagged elements in an input string, construed as possible constituents in a tree.

An All Possible Constituents Approach

- Initial experiments with unsupervised grammar induction (like those described in Carroll and Charniak (1992)) were not particularly encouraging.
- Far more promising results have been achieved in recent work.
- Klein and Manning (K&M) (2002) propose a method that learns constituent structure from POS tagged input by unsupervised techniques.
- It assigns probability values to all subsequences of tagged elements in an input string, construed as possible constituents in a tree.

Constraints on the K&M (2002) Model

- The model imposes the constraint of binary branching on all non-terminal elements of a parse tree.
- It also partially characterizes phrase structure by the condition that sister phrases do not have non-empty intersections.
- These conditions are learning biases of the model.

Constraints on the K&M (2002) Model

- The model imposes the constraint of binary branching on all non-terminal elements of a parse tree.
- It also partially characterizes phrase structure by the condition that sister phrases do not have non-empty intersections.
- These conditions are learning biases of the model.

Constraints on the K&M (2002) Model

- The model imposes the constraint of binary branching on all non-terminal elements of a parse tree.
- It also partially characterizes phrase structure by the condition that sister phrases do not have non-empty intersections.
- These conditions are learning biases of the model.

Co-Occurrence and Constituency Structure

- K&M (2002) invoke an Expectation Maximization (EM) algorithm to select the most likely parse for a sentence.
- They identify (unlabeled) constituents through the distributional co-occurrence of POS sequences in the same contexts.
- The set of POS sequences is partitioned into classes on the basis of frequency of appearance in the same environments.
- The most highly valued parse for a string is the one which maximizes the likelihood of its constituents in the contexts in which they appear in the parse.

Co-Occurrence and Constituency Structure

- K&M (2002) invoke an Expectation Maximization (EM) algorithm to select the most likely parse for a sentence.
- They identify (unlabeled) constituents through the distributional co-occurrence of POS sequences in the same contexts.
- The set of POS sequences is partitioned into classes on the basis of frequency of appearance in the same environments.
- The most highly valued parse for a string is the one which maximizes the likelihood of its constituents in the contexts in which they appear in the parse.

Co-Occurrence and Constituency Structure

- K&M (2002) invoke an Expectation Maximization (EM) algorithm to select the most likely parse for a sentence.
- They identify (unlabeled) constituents through the distributional co-occurrence of POS sequences in the same contexts.
- The set of POS sequences is partitioned into classes on the basis of frequency of appearance in the same environments.
- The most highly valued parse for a string is the one which maximizes the likelihood of its constituents in the contexts in which they appear in the parse.

Co-Occurrence and Constituency Structure

- K&M (2002) invoke an Expectation Maximization (EM) algorithm to select the most likely parse for a sentence.
- They identify (unlabeled) constituents through the distributional co-occurrence of POS sequences in the same contexts.
- The set of POS sequences is partitioned into classes on the basis of frequency of appearance in the same environments.
- The most highly valued parse for a string is the one which maximizes the likelihood of its constituents in the contexts in which they appear in the parse.

Evaluating the K&M (2002) Parser

- Evaluated against Penn Treebank parses as the gold standard, the K&M (2002) parser achieves an F-measure of 71%.
- This score is achieved despite the fact that the Penn Treebank allows for non-binary branching for many constituents.
- Consequently, a binary branching parse algorithm can only achieve a maximum F-score of 87% against this standard.

Evaluating the K&M (2002) Parser

- Evaluated against Penn Treebank parses as the gold standard, the K&M (2002) parser achieves an F-measure of 71%.
- This score is achieved despite the fact that the Penn Treebank allows for non-binary branching for many constituents.
- Consequently, a binary branching parse algorithm can only achieve a maximum F-score of 87% against this standard.

Evaluating the K&M (2002) Parser

- Evaluated against Penn Treebank parses as the gold standard, the K&M (2002) parser achieves an F-measure of 71%.
- This score is achieved despite the fact that the Penn Treebank allows for non-binary branching for many constituents.
- Consequently, a binary branching parse algorithm can only achieve a maximum F-score of 87% against this standard.

Evaluating the K&M (2002) Parser

- In fact, many of the algorithm's binary constituent analyses that are excluded by the gold standard are linguistically defensible.
- For example the Treebank analyses noun phrases as having flat structure, but the iterated binary branching constituent structure that the parser assigns to NPs is well motivated on syntactic grounds.
- Therefore, when the K&M parser is evaluated against the Penn Tree Bank, its F-measure understates its success in producing linguistically viable parses of the data.

Evaluating the K&M (2002) Parser

- In fact, many of the algorithm's binary constituent analyses that are excluded by the gold standard are linguistically defensible.
- For example the Treebank analyses noun phrases as having flat structure, but the iterated binary branching constituent structure that the parser assigns to NPs is well motivated on syntactic grounds.
- Therefore, when the K&M parser is evaluated against the Penn Tree Bank, its F-measure understates its success in producing linguistically viable parses of the data.

Evaluating the K&M (2002) Parser

- In fact, many of the algorithm's binary constituent analyses that are excluded by the gold standard are linguistically defensible.
- For example the Treebank analyses noun phrases as having flat structure, but the iterated binary branching constituent structure that the parser assigns to NPs is well motivated on syntactic grounds.
- Therefore, when the K&M parser is evaluated against the Penn Tree Bank, its F-measure understates its success in producing linguistically viable parses of the data.

The Role of POS Tagging in Unsupervised Parsing

- The K&M parser is, in fact, a case of semi-supervised rather than fully unsupervised learning.
- Its input is a corpus annotated with the POS tagging of the Penn Treebank.
- If POS annotation is, in turn, provided by a tagger that uses unsupervised learning, then the entire parsing procedure can be construed as a sequenced process of unsupervised grammar induction.

The Role of POS Tagging in Unsupervised Parsing

- The K&M parser is, in fact, a case of semi-supervised rather than fully unsupervised learning.
- Its input is a corpus annotated with the POS tagging of the Penn Treebank.
- If POS annotation is, in turn, provided by a tagger that uses unsupervised learning, then the entire parsing procedure can be construed as a sequenced process of unsupervised grammar induction.

The Role of POS Tagging in Unsupervised Parsing

- The K&M parser is, in fact, a case of semi-supervised rather than fully unsupervised learning.
- Its input is a corpus annotated with the POS tagging of the Penn Treebank.
- If POS annotation is, in turn, provided by a tagger that uses unsupervised learning, then the entire parsing procedure can be construed as a sequenced process of unsupervised grammar induction.

Unsupervised POS Tagging

- An unsupervised POS tagger will also rely on morphological analysis of the words in a corpus.
- This can be provided by an unsupervised morphological analyzer.
- Goldsmith (2001), and Schone and Jurafsky (2001) each describe an unsupervised morphological analyzer, and show that they perform well on corpora from several languages.

Unsupervised POS Tagging

- An unsupervised POS tagger will also rely on morphological analysis of the words in a corpus.
- This can be provided by an unsupervised morphological analyzer.
- Goldsmith (2001), and Schone and Jurafsky (2001) each describe an unsupervised morphological analyzer, and show that they perform well on corpora from several languages.

Unsupervised POS Tagging

- An unsupervised POS tagger will also rely on morphological analysis of the words in a corpus.
- This can be provided by an unsupervised morphological analyzer.
- Goldsmith (2001), and Schone and Jurafsky (2001) each describe an unsupervised morphological analyzer, and show that they perform well on corpora from several languages.

The K&M (2002) Parser with an Unsupervised Tagger

- The K&M (2002) parser achieves an F-score of 63.2% on WSJ text annotated by an unsupervised POS tagger.
- They observe that this tagger is not particularly reliable.
- Other unsupervised taggers, like the one presented in Clark (2003), produce better results.
- These taggers might well allow the parser to perform at a level comparable to that which it achieves with Penn Treebank tags.

The K&M (2002) Parser with an Unsupervised Tagger

- The K&M (2002) parser achieves an F-score of 63.2% on WSJ text annotated by an unsupervised POS tagger.
- They observe that this tagger is not particularly reliable.
- Other unsupervised taggers, like the one presented in Clark (2003), produce better results.
- These taggers might well allow the parser to perform at a level comparable to that which it achieves with Penn Treebank tags.

The K&M (2002) Parser with an Unsupervised Tagger

- The K&M (2002) parser achieves an F-score of 63.2% on WSJ text annotated by an unsupervised POS tagger.
- They observe that this tagger is not particularly reliable.
- Other unsupervised taggers, like the one presented in Clark (2003), produce better results.
- These taggers might well allow the parser to perform at a level comparable to that which it achieves with Penn Treebank tags.

The K&M (2002) Parser with an Unsupervised Tagger

- The K&M (2002) parser achieves an F-score of 63.2% on WSJ text annotated by an unsupervised POS tagger.
- They observe that this tagger is not particularly reliable.
- Other unsupervised taggers, like the one presented in Clark (2003), produce better results.
- These taggers might well allow the parser to perform at a level comparable to that which it achieves with Penn Treebank tags.

Klein and Manning (2004): A Head Dependency Parser

- K&M (2004) present an unsupervised learning procedure for lexicalized head dependency grammars.
- It assigns probabilities to all possible dependency relations in a sentence S by estimating the likelihood that each word in S is a head for particular sequences of words to its left and to its right.

Klein and Manning (2004): A Head Dependency Parser

- K&M (2004) present an unsupervised learning procedure for lexicalized head dependency grammars.
- It assigns probabilities to all possible dependency relations in a sentence S by estimating the likelihood that each word in S is a head for particular sequences of words to its left and to its right.

K&M (2004): A Head Dependency Parser

- The probabilities for these alternative dependency relations are computed on the basis of the context, defined as adjacent words (word classes) on each side, in which each head occurs.
- Binary branching is a condition on dependency relations.
- The procedure achieves an F-measure of 52.1% on Penn Treebank test data.

K&M (2004): A Head Dependency Parser

- The probabilities for these alternative dependency relations are computed on the basis of the context, defined as adjacent words (word classes) on each side, in which each head occurs.
- Binary branching is a condition on dependency relations.
- The procedure achieves an F-measure of 52.1% on Penn Treebank test data.

K&M (2004): A Head Dependency Parser

- The probabilities for these alternative dependency relations are computed on the basis of the context, defined as adjacent words (word classes) on each side, in which each head occurs.
- Binary branching is a condition on dependency relations.
- The procedure achieves an F-measure of 52.1% on Penn Treebank test data.

A Combined Head Dependency-Constituency Parser

- K&M (2004) combine their dependency and constituent structure grammar induction systems into an integrated model that produces better results than either of its component parsers.
- The composite model computes the score for a tree as the product of the dependency and constituency structure grammars.

A Combined Head Dependency-Constituency Parser

- K&M (2004) combine their dependency and constituent structure grammar induction systems into an integrated model that produces better results than either of its component parsers.
- The composite model computes the score for a tree as the product of the dependency and constituency structure grammars.

A Combined Head Dependency-Constituency Parser

- The model uses both constituent clustering and head dependency relations to predict binary constituent parse structure.
- The combined parser achieves an F-measure of 77.6% for Penn Treebank POS tagging.
- It scores an F-measure of 72.9% for Schütze's (1995) unsupervised tagger.

A Combined Head Dependency-Constituency Parser

- The model uses both constituent clustering and head dependency relations to predict binary constituent parse structure.
- The combined parser achieves an F-measure of 77.6% for Penn Treebank POS tagging.
- It scores an F-measure of 72.9% for Schütze's (1995) unsupervised tagger.

A Combined Head Dependency-Constituency Parser

- The model uses both constituent clustering and head dependency relations to predict binary constituent parse structure.
- The combined parser achieves an F-measure of 77.6% for Penn Treebank POS tagging.
- It scores an F-measure of 72.9% for Schütze's (1995) unsupervised tagger.

Unsupervised Data Oriented Processing: An All Possible Binary Trees Approach

- Bod (2006, 2007a, 2007b) proposes Unsupervised Data Oriented Parsing (U-DOP).
- U-DOP generates all possible binary branching subtrees for a sentence S .
- The preferred parse for S is the one obtained through the smallest number of substitutions of subtrees into nodes in larger trees.

Unsupervised Data Oriented Processing: An All Possible Binary Trees Approach

- Bod (2006, 2007a, 2007b) proposes Unsupervised Data Oriented Parsing (U-DOP).
- U-DOP generates all possible binary branching subtrees for a sentence S .
- The preferred parse for S is the one obtained through the smallest number of substitutions of subtrees into nodes in larger trees.

Unsupervised Data Oriented Processing: An All Possible Binary Trees Approach

- Bod (2006, 2007a, 2007b) proposes Unsupervised Data Oriented Parsing (U-DOP).
- U-DOP generates all possible binary branching subtrees for a sentence S .
- The preferred parse for S is the one obtained through the smallest number of substitutions of subtrees into nodes in larger trees.

Unsupervised Data Oriented Processing

- In cases where more than one derivation satisfies the minimality condition, the derivation using subtrees with the highest frequency in previously parsed text is selected.
- Bod (2006) reports an F-score of 82.9% for U-DOP, combined with a maximum likelihood estimator and applied to the K&M WSJ test corpus.

Unsupervised Data Oriented Processing

- In cases where more than one derivation satisfies the minimality condition, the derivation using subtrees with the highest frequency in previously parsed text is selected.
- Bod (2006) reports an F-score of 82.9% for U-DOP, combined with a maximum likelihood estimator and applied to the K&M WSJ test corpus.

U-DOP and Discontinuous Structures

- U-DOP has an important advantage over simple PCFGs in its capacity to represent discontinuous syntactic structures.
- For example, it handles subject-auxiliary inversion in questions and complex determiners such as *more...than...*, as complete constructions.
- U-DOP incorporates binary branching tree recursion as the main bias of its model.
- It can parse structures not previously encountered, either
 - through the equivalent of PCFG rules, or
 - by identifying structural analogies between possible tree constructions for a current input and those assigned to previously parsed strings in a test set.

U-DOP and Discontinuous Structures

- U-DOP has an important advantage over simple PCFGs in its capacity to represent discontinuous syntactic structures.
- For example, it handles subject-auxiliary inversion in questions and complex determiners such as *more...than...*, as complete constructions.
- U-DOP incorporates binary branching tree recursion as the main bias of its model.
- It can parse structures not previously encountered, either
 - through the equivalent of PCFG rules, or
 - by identifying structural analogies between possible tree constructions for a current input and those assigned to previously parsed strings in a test set.

U-DOP and Discontinuous Structures

- U-DOP has an important advantage over simple PCFGs in its capacity to represent discontinuous syntactic structures.
- For example, it handles subject-auxiliary inversion in questions and complex determiners such as *more...than...*, as complete constructions.
- U-DOP incorporates binary branching tree recursion as the main bias of its model.
- It can parse structures not previously encountered, either
 - through the equivalent of PCFG rules, or
 - by identifying structural analogies between possible tree constructions for a current input and those assigned to previously parsed strings in a test set.

U-DOP and Discontinuous Structures

- U-DOP has an important advantage over simple PCFGs in its capacity to represent discontinuous syntactic structures.
- For example, it handles subject-auxiliary inversion in questions and complex determiners such as *more...than...*, as complete constructions.
- U-DOP incorporates binary branching tree recursion as the main bias of its model.
- It can parse structures not previously encountered, either
 - through the equivalent of PCFG rules, or
 - by identifying structural analogies between possible tree constructions for a current input and those assigned to previously parsed strings in a test set.

U-DOP and Discontinuous Structures

- U-DOP has an important advantage over simple PCFGs in its capacity to represent discontinuous syntactic structures.
- For example, it handles subject-auxiliary inversion in questions and complex determiners such as *more...than...*, as complete constructions.
- U-DOP incorporates binary branching tree recursion as the main bias of its model.
- It can parse structures not previously encountered, either
 - through the equivalent of PCFG rules, or
 - by identifying structural analogies between possible tree constructions for a current input and those assigned to previously parsed strings in a test set.

A Modified Version of U-DOP

- While U-DOP improves on the accuracy and coverage of K&M's (2004) combined unsupervised dependency-constituency model, it generates a very large number of possible subtrees for each parse that it produces, which renders it inefficient.
- Bod (2007b) describes a procedure for greatly reducing this number by converting a U-DOP model into a type of PCFG.
- The resulting parser produces far fewer possible subtrees for each sentence, but at the cost of performance.
- It yields a reported F-score of 77.9% on the WSJ test corpus.

A Modified Version of U-DOP

- While U-DOP improves on the accuracy and coverage of K&M's (2004) combined unsupervised dependency-constituency model, it generates a very large number of possible subtrees for each parse that it produces, which renders it inefficient.
- Bod (2007b) describes a procedure for greatly reducing this number by converting a U-DOP model into a type of PCFG.
- The resulting parser produces far fewer possible subtrees for each sentence, but at the cost of performance.
- It yields a reported F-score of 77.9% on the WSJ test corpus.

A Modified Version of U-DOP

- While U-DOP improves on the accuracy and coverage of K&M's (2004) combined unsupervised dependency-constituency model, it generates a very large number of possible subtrees for each parse that it produces, which renders it inefficient.
- Bod (2007b) describes a procedure for greatly reducing this number by converting a U-DOP model into a type of PCFG.
- The resulting parser produces far fewer possible subtrees for each sentence, but at the cost of performance.
- It yields a reported F-score of 77.9% on the WSJ test corpus.

A Modified Version of U-DOP

- While U-DOP improves on the accuracy and coverage of K&M's (2004) combined unsupervised dependency-constituency model, it generates a very large number of possible subtrees for each parse that it produces, which renders it inefficient.
- Bod (2007b) describes a procedure for greatly reducing this number by converting a U-DOP model into a type of PCFG.
- The resulting parser produces far fewer possible subtrees for each sentence, but at the cost of performance.
- It yields a reported F-score of 77.9% on the WSJ test corpus.

Accuracy vs. Cost in Parsing

- In general supervised learning algorithms achieve greater accuracy than unsupervised procedures.
- But hand annotating corpora for training supervised algorithms adds a significant cost that must be weighed against the accuracy that these procedures provide.
- By avoiding these costs unsupervised algorithms offer an important advantage, if they can sustain an acceptable level of performance in the applications for which they are designed.

Accuracy vs. Cost in Parsing

- In general supervised learning algorithms achieve greater accuracy than unsupervised procedures.
- But hand annotating corpora for training supervised algorithms adds a significant cost that must be weighed against the accuracy that these procedures provide.
- By avoiding these costs unsupervised algorithms offer an important advantage, if they can sustain an acceptable level of performance in the applications for which they are designed.

Accuracy vs. Cost in Parsing

- In general supervised learning algorithms achieve greater accuracy than unsupervised procedures.
- But hand annotating corpora for training supervised algorithms adds a significant cost that must be weighed against the accuracy that these procedures provide.
- By avoiding these costs unsupervised algorithms offer an important advantage, if they can sustain an acceptable level of performance in the applications for which they are designed.

Semi-Supervised Learning

- Banko and Brill (2001) (B&B) use a method of semi-supervised learning that combines some of the benefits of both systems.
- They train 10 distinct classifiers for a word disambiguation problem on an annotated test set.
- They then run all the classifiers on an unannotated corpus and select the instances for which there is full agreement among them.
- This automatically annotated data is added to the original hand annotated corpus for a new cycle of training.
- The process is iterated with additional unannotated corpora.

Semi-Supervised Learning

- Banko and Brill (2001) (B&B) use a method of semi-supervised learning that combines some of the benefits of both systems.
- They train 10 distinct classifiers for a word disambiguation problem on an annotated test set.
- They then run all the classifiers on an unannotated corpus and select the instances for which there is full agreement among them.
- This automatically annotated data is added to the original hand annotated corpus for a new cycle of training.
- The process is iterated with additional unannotated corpora.

Semi-Supervised Learning

- Banko and Brill (2001) (B&B) use a method of semi-supervised learning that combines some of the benefits of both systems.
- They train 10 distinct classifiers for a word disambiguation problem on an annotated test set.
- They then run all the classifiers on an unannotated corpus and select the instances for which there is full agreement among them.
- This automatically annotated data is added to the original hand annotated corpus for a new cycle of training.
- The process is iterated with additional unannotated corpora.

Semi-Supervised Learning

- Banko and Brill (2001) (B&B) use a method of semi-supervised learning that combines some of the benefits of both systems.
- They train 10 distinct classifiers for a word disambiguation problem on an annotated test set.
- They then run all the classifiers on an unannotated corpus and select the instances for which there is full agreement among them.
- This automatically annotated data is added to the original hand annotated corpus for a new cycle of training.
- The process is iterated with additional unannotated corpora.

Semi-Supervised Learning

- Banko and Brill (2001) (B&B) use a method of semi-supervised learning that combines some of the benefits of both systems.
- They train 10 distinct classifiers for a word disambiguation problem on an annotated test set.
- They then run all the classifiers on an unannotated corpus and select the instances for which there is full agreement among them.
- This automatically annotated data is added to the original hand annotated corpus for a new cycle of training.
- The process is iterated with additional unannotated corpora.

Using Semi-Supervised Learning to Improve Performance

- B&B's system for word disambiguation improves accuracy through unsupervised extensions of a supervised base corpus up to a certain phase in the learning cycles.
- After this phase it begins to decline.
- B&B suggest that this effect may be due to the learning process reaching a point at which the benefits of additional data are outweighed by the distortion of bias imported with new samples.

Using Semi-Supervised Learning to Improve Performance

- B&B's system for word disambiguation improves accuracy through unsupervised extensions of a supervised base corpus up to a certain phase in the learning cycles.
- After this phase it begins to decline.
- B&B suggest that this effect may be due to the learning process reaching a point at which the benefits of additional data are outweighed by the distortion of bias imported with new samples.

Using Semi-Supervised Learning to Improve Performance

- B&B's system for word disambiguation improves accuracy through unsupervised extensions of a supervised base corpus up to a certain phase in the learning cycles.
- After this phase it begins to decline.
- B&B suggest that this effect may be due to the learning process reaching a point at which the benefits of additional data are outweighed by the distortion of bias imported with new samples.

Semi-Supervised Parsing

- B&B's approach can be generalized to grammar induction and parsing.
- It would involve training several supervised parsing systems on an initial parsed corpus.
- Then these procedures would be optimized through iterated parsing of text containing only POS tagging.
- The tagging can be done automatically using a reliable tagger.

Semi-Supervised Parsing

- B&B's approach can be generalized to grammar induction and parsing.
- It would involve training several supervised parsing systems on an initial parsed corpus.
- Then these procedures would be optimized through iterated parsing of text containing only POS tagging.
- The tagging can be done automatically using a reliable tagger.

Semi-Supervised Parsing

- B&B's approach can be generalized to grammar induction and parsing.
- It would involve training several supervised parsing systems on an initial parsed corpus.
- Then these procedures would be optimized through iterated parsing of text containing only POS tagging.
- The tagging can be done automatically using a reliable tagger.

Semi-Supervised Parsing

- B&B's approach can be generalized to grammar induction and parsing.
- It would involve training several supervised parsing systems on an initial parsed corpus.
- Then these procedures would be optimized through iterated parsing of text containing only POS tagging.
- The tagging can be done automatically using a reliable tagger.

Unsupervised Parsing and NLP

- There are good engineering reasons for investing more research effort in the development of robust unsupervised and semi-supervised learning procedures.
- Very large quantities of raw natural language text are now online and easily accessible at little or no cost.
- Generating the training corpora for supervised learning is expensive and time consuming.
- As the accuracy and coverage of unsupervised systems improve, they become increasingly attractive alternatives to supervised methods for a variety of NLP tasks.

Unsupervised Parsing and NLP

- There are good engineering reasons for investing more research effort in the development of robust unsupervised and semi-supervised learning procedures.
- Very large quantities of raw natural language text are now online and easily accessible at little or no cost.
- Generating the training corpora for supervised learning is expensive and time consuming.
- As the accuracy and coverage of unsupervised systems improve, they become increasingly attractive alternatives to supervised methods for a variety of NLP tasks.

Unsupervised Parsing and NLP

- There are good engineering reasons for investing more research effort in the development of robust unsupervised and semi-supervised learning procedures.
- Very large quantities of raw natural language text are now online and easily accessible at little or no cost.
- Generating the training corpora for supervised learning is expensive and time consuming.
- As the accuracy and coverage of unsupervised systems improve, they become increasingly attractive alternatives to supervised methods for a variety of NLP tasks.

Unsupervised Parsing and NLP

- There are good engineering reasons for investing more research effort in the development of robust unsupervised and semi-supervised learning procedures.
- Very large quantities of raw natural language text are now online and easily accessible at little or no cost.
- Generating the training corpora for supervised learning is expensive and time consuming.
- As the accuracy and coverage of unsupervised systems improve, they become increasingly attractive alternatives to supervised methods for a variety of NLP tasks.

Unsupervised Grammar Induction and the APS

- Recent work on unsupervised parsing indicates that it may be possible to develop efficient machine learning algorithms that acquire accurate and theoretically viable grammars.
- They use only the positive evidence of raw corpora, and they employ weak domain specific learning biases.
- To the extent that such grammar induction procedures are successful, they undermine the APS as an argument for linguistic nativism.
- They show that an ML algorithm can effectively acquire a significant element of human linguistic knowledge relying primarily on generalized information theoretic techniques.

Unsupervised Grammar Induction and the APS

- Recent work on unsupervised parsing indicates that it may be possible to develop efficient machine learning algorithms that acquire accurate and theoretically viable grammars.
- They use only the positive evidence of raw corpora, and they employ weak domain specific learning biases.
- To the extent that such grammar induction procedures are successful, they undermine the APS as an argument for linguistic nativism.
- They show that an ML algorithm can effectively acquire a significant element of human linguistic knowledge relying primarily on generalized information theoretic techniques.

Unsupervised Grammar Induction and the APS

- Recent work on unsupervised parsing indicates that it may be possible to develop efficient machine learning algorithms that acquire accurate and theoretically viable grammars.
- They use only the positive evidence of raw corpora, and they employ weak domain specific learning biases.
- To the extent that such grammar induction procedures are successful, they undermine the APS as an argument for linguistic nativism.
- They show that an ML algorithm can effectively acquire a significant element of human linguistic knowledge relying primarily on generalized information theoretic techniques.

Unsupervised Grammar Induction and the APS

- Recent work on unsupervised parsing indicates that it may be possible to develop efficient machine learning algorithms that acquire accurate and theoretically viable grammars.
- They use only the positive evidence of raw corpora, and they employ weak domain specific learning biases.
- To the extent that such grammar induction procedures are successful, they undermine the APS as an argument for linguistic nativism.
- They show that an ML algorithm can effectively acquire a significant element of human linguistic knowledge relying primarily on generalized information theoretic techniques.

Machine Learning and Human Language Acquisition

- The success of weak bias unsupervised ML in grammar induction (and related NLP tasks) vitiates the APS.
- However, it does not tell us anything about the actual cognitive mechanisms that humans employ in first language acquisition.
- A strong nativist view of UG could, in principle, still turn out to be correct on the basis of independent psychological and biological facts.
- We will see in a later class that recent psycholinguistic work suggests that Bayesian inference of the sort involved in statistical ML methods plays a significant role in central elements of human grammar induction.

Machine Learning and Human Language Acquisition

- The success of weak bias unsupervised ML in grammar induction (and related NLP tasks) vitiates the APS.
- However, it does not tell us anything about the actual cognitive mechanisms that humans employ in first language acquisition.
- A strong nativist view of UG could, in principle, still turn out to be correct on the basis of independent psychological and biological facts.
- We will see in a later class that recent psycholinguistic work suggests that Bayesian inference of the sort involved in statistical ML methods plays a significant role in central elements of human grammar induction.

Machine Learning and Human Language Acquisition

- The success of weak bias unsupervised ML in grammar induction (and related NLP tasks) vitiates the APS.
- However, it does not tell us anything about the actual cognitive mechanisms that humans employ in first language acquisition.
- A strong nativist view of UG could, in principle, still turn out to be correct on the basis of independent psychological and biological facts.
- We will see in a later class that recent psycholinguistic work suggests that Bayesian inference of the sort involved in statistical ML methods plays a significant role in central elements of human grammar induction.

Machine Learning and Human Language Acquisition

- The success of weak bias unsupervised ML in grammar induction (and related NLP tasks) vitiates the APS.
- However, it does not tell us anything about the actual cognitive mechanisms that humans employ in first language acquisition.
- A strong nativist view of UG could, in principle, still turn out to be correct on the basis of independent psychological and biological facts.
- We will see in a later class that recent psycholinguistic work suggests that Bayesian inference of the sort involved in statistical ML methods plays a significant role in central elements of human grammar induction.

Conclusions

- There is increasing interest in unsupervised ML methods in NLP for engineering reasons.
- Recent work on unsupervised morphological analysis, POS tagging, and parsing have yielded encouraging results.
- The increasing accuracy and robustness of these systems have significant implications for the study of human language acquisition.
- They suggest the computational viability of relatively weak bias learning algorithms for grammar induction.

Conclusions

- There is increasing interest in unsupervised ML methods in NLP for engineering reasons.
- Recent work on unsupervised morphological analysis, POS tagging, and parsing have yielded encouraging results.
- The increasing accuracy and robustness of these systems have significant implications for the study of human language acquisition.
- They suggest the computational viability of relatively weak bias learning algorithms for grammar induction.

Conclusions

- There is increasing interest in unsupervised ML methods in NLP for engineering reasons.
- Recent work on unsupervised morphological analysis, POS tagging, and parsing have yielded encouraging results.
- The increasing accuracy and robustness of these systems have significant implications for the study of human language acquisition.
- They suggest the computational viability of relatively weak bias learning algorithms for grammar induction.

Conclusions

- There is increasing interest in unsupervised ML methods in NLP for engineering reasons.
- Recent work on unsupervised morphological analysis, POS tagging, and parsing have yielded encouraging results.
- The increasing accuracy and robustness of these systems have significant implications for the study of human language acquisition.
- They suggest the computational viability of relatively weak bias learning algorithms for grammar induction.