Towards a Stochastic Model of Linguistic Competence

Shalom Lappin
King's College London

Université Paris Diderot-Paris 7
December 2, 2010

Outline

Competence and Performance

- Linguistic knowledge can be factored into competence and performance.
- Competence is best represented by a grammar which specifies the set of well-formed strings, and their associated syntactic structures (as well as their morphological, phonological, and semantic properties).
- The principles encoded in such a grammar are applied in interpretation and generation operations.
- The computational and data storage systems through which these operations are implemented provide the elements of performance.

Competence and Performance

- Linguistic knowledge can be factored into competence and performance.
- Competence is best represented by a grammar which specifies the set of well-formed strings, and their associated syntactic structures (as well as their morphological, phonological, and semantic properties).
- The principles encoded in such a grammar are applied in interpretation and generation operations.
- The computational and data storage systems through which these operations are implemented provide the elements of performance.

Competence and Performance

- Linguistic knowledge can be factored into competence and performance.
- Competence is best represented by a grammar which specifies the set of well-formed strings, and their associated syntactic structures (as well as their morphological, phonological, and semantic properties).
- The principles encoded in such a grammar are applied in interpretation and generation operations.
- The computational and data storage systems through which these operations are implemented provide the elements of performance.

Competence and Performance

- Linguistic knowledge can be factored into competence and performance.
- Competence is best represented by a grammar which specifies the set of well-formed strings, and their associated syntactic structures (as well as their morphological, phonological, and semantic properties).
- The principles encoded in such a grammar are applied in interpretation and generation operations.
- The computational and data storage systems through which these operations are implemented provide the elements of performance.

Grammars and Parsers

- The competence-performance distinction runs parallel to the difference between a grammar which recognizes strings and assigns structural analyses to them, and a parsing algorithm that applies the grammar.
- The same grammar can be implemented by a variety of parsers, for example, bottom-up, top-down, CKY, and chart parsers.
- Conversely, a particular parsing algorithm can be used for different grammars and grammar formalisms.

Grammars and Parsers

- The competence-performance distinction runs parallel to the difference between a grammar which recognizes strings and assigns structural analyses to them, and a parsing algorithm that applies the grammar.
- The same grammar can be implemented by a variety of parsers, for example, bottom-up, top-down, CKY, and chart parsers.
- Conversely, a particular parsing algorithm can be used for different grammars and grammar formalisms.

Grammars and Parsers

- The competence-performance distinction runs parallel to the difference between a grammar which recognizes strings and assigns structural analyses to them, and a parsing algorithm that applies the grammar.
- The same grammar can be implemented by a variety of parsers, for example, bottom-up, top-down, CKY, and chart parsers.
- Conversely, a particular parsing algorithm can be used for different grammars and grammar formalisms.

The Classical View of Linguistic Competence

- On the classical view of linguistic competence a formal grammar consists of categorical rules and constraints that define the set of well formed structures for a language.
- Gradience in speakers' acceptability judgements, and frequency effects in interpretation and production are attributed to performance factors.
- The conditions that comprise a grammar are indefeasible.
- Instability in a given speaker's linguistic intuitions and behaviour for a specified set of expressions are taken to be the result of processing mechanisms, such as memory, attentional focus, and perceptual priming.

The Classical View of Linguistic Competence

- On the classical view of linguistic competence a formal grammar consists of categorical rules and constraints that define the set of well formed structures for a language.
- Gradience in speakers' acceptability judgements, and frequency effects in interpretation and production are attributed to performance factors.
- The conditions that comprise a grammar are indefeasible.
- Instability in a given speaker's linguistic intuitions and behaviour for a specified set of expressions are taken to be the result of processing mechanisms, such as memory, attentional focus, and perceptual priming.

The Classical View of Linguistic Competence

- On the classical view of linguistic competence a formal grammar consists of categorical rules and constraints that define the set of well formed structures for a language.
- Gradience in speakers' acceptability judgements, and frequency effects in interpretation and production are attributed to performance factors.
- The conditions that comprise a grammar are indefeasible.
- Instability in a given speaker's linguistic intuitions and behaviour for a specified set of expressions are taken to be the result of processing mechanisms, such as memory, attentional focus, and perceptual priming.

The Classical View of Linguistic Competence

- On the classical view of linguistic competence a formal grammar consists of categorical rules and constraints that define the set of well formed structures for a language.
- Gradience in speakers' acceptability judgements, and frequency effects in interpretation and production are attributed to performance factors.
- The conditions that comprise a grammar are indefeasible.
- Instability in a given speaker's linguistic intuitions and behaviour for a specified set of expressions are taken to be the result of processing mechanisms, such as memory, attentional focus, and perceptual priming.

A Stochastic Approach to Linguistic Competence

- During the past fifteen years the suggestion that linguistic knowledge is best represented stochastically has gained increasing currency among computational linguists, psycholinguists, and even some theoretical linguists.
- Abney (1996), Manning (2003), Jurafsky (2003), Chater and Manning (2006), and Bresnan (2007), *inter alia*, have proposed the use of statistical models to capture gradient effects and soft constraints in syntactic processing, and the role of probabilistic inference in language acquisition.

A Stochastic Approach to Linguistic Competence

- During the past fifteen years the suggestion that linguistic knowledge is best represented stochastically has gained increasing currency among computational linguists, psycholinguists, and even some theoretical linguists.

- Abney (1996), Manning (2003), Jurafsky (2003), Chater and Manning (2006), and Bresnan (2007), *inter alia*, have proposed the use of statistical models to capture gradient effects and soft constraints in syntactic processing, and the role of probabilistic inference in language acquisition.

From Natural Language Engineering to Cognitive Modeling

- In many cases these models were originally developed as components of machine learning systems designed to solve engineering tasks in NLP.

- They are now being explored as representations of the cognitive processes involved in human language learning, comprehension, and generation.

From Natural Language Engineering to Cognitive Modeling

- In many cases these models were originally developed as components of machine learning systems designed to solve engineering tasks in NLP.
- They are now being explored as representations of the cognitive processes involved in human language learning, comprehension, and generation.

Probabilistic Context-Free Grammars

- One way of representing linguistic knowledge in stochastic terms is to encode it in a probabilistic grammar, like a Probabilistic Context-Free Grammar (PCFG), which conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- A PCFG 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.
- 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

- One way of representing linguistic knowledge in stochastic terms is to encode it in a probabilistic grammar, like a Probabilistic Context-Free Grammar (PCFG), which conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- A PCFG 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.
- 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

- One way of representing linguistic knowledge in stochastic terms is to encode it in a probabilistic grammar, like a Probabilistic Context-Free Grammar (PCFG), which conditions the probability of a child nonterminal sequence on that of the parent nonterminal.
- A PCFG 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.
- 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.

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

- 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.

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

- 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.

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

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

A PCFG as a Probabilistic Language Model

- When the parameters of a PCFG G are set, it assigns a probability value to every parse $\mathcal{P}$ of a sentence S of L.
- The probability of the parse of a sentence is the product of the probabilities of the rules in the derivation of the parse:
$$p(\mathcal{P}) = \prod_{i=1}^{n} p(pr_{i(pr_i \in P)} \in \mathcal{P}).$$
- The probability of a sentence is the sum of the probability of its parses:
$$p(S) = \sum_{i=1}^{n} p(\mathcal{P}_{i(\mathcal{P}_i \in G(S))}).$$

A PCFG as a Probabilistic Language Model

- When the parameters of a PCFG G are set, it assigns a probability value to every parse $\mathcal{P}$ of a sentence S of L.
- The probability of the parse of a sentence is the product of the probabilities of the rules in the derivation of the parse:

$$p(\mathcal{P}) = \prod_{i=1}^{n} p(pr_{i(pr_i \in P)} \in \mathcal{P}).$$

- The probability of a sentence is the sum of the probability of its parses:

$$p(S) = \sum_{i=1}^{n} p(\mathcal{P}_{i(\mathcal{P}_i \in G(S))}).$$

A PCFG as a Probabilistic Language Model

- When the parameters of a PCFG G are set, it assigns a probability value to every parse $\mathcal{P}$ of a sentence S of L.
- The probability of the parse of a sentence is the product of the probabilities of the rules in the derivation of the parse:

$$p(\mathcal{P}) = \prod_{i=1}^{n} p(pr_{i(pr_i \in P)} \in \mathcal{P}).$$

- The probability of a sentence is the sum of the probability of its parses:

$$p(S) = \sum_{i=1}^{n} p(\mathcal{P}_{i(\mathcal{P}_i \in G(S))}).$$

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 $P(X_1/h_1 \cdots H/h \cdots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.
- These are results for supervised learning from a parse annotated corpus, and so they are not directly relevant for human grammar induction, which is unsupervised.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \cdots H/h \cdots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.
- These are results for supervised learning from a parse annotated corpus, and so they are not directly relevant for human grammar induction, which is unsupervised.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \cdots H/h \cdots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.
- These are results for supervised learning from a parse annotated corpus, and so they are not directly relevant for human grammar induction, which is unsupervised.

Lexicalized Probabilistic Context-Free Grammars

- The probability distributions of the model are of the form $P_s(N/h)$ and $P(X_1/h_1 \cdots H/h \cdots X_n/h_n \mid N/h)$.
- Collins' LPCFG achieves an F-measure performance of approximately 88%.
- Charniak and Johnson (2005) present a LPCFG with an F score of approximately 91%.
- These are results for supervised learning from a parse annotated corpus, and so they are not directly relevant for human grammar induction, which is unsupervised.

Structured Language Models

- Structured language models (SLMs) (Chelba and Jelinek (2000), Chelba (2010)) offer an alternative stochastic model of linguistic competence.
- They use probabilistic push down automata (PPDA) to produce a probability distribution for the strings of a corpus, taken as the yields of CFG parse structures.

Structured Language Models

- Structured language models (SLMs) (Chelba and Jelinek (2000), Chelba (2010)) offer an alternative stochastic model of linguistic competence.
- They use probabilistic push down automata (PPDA) to produce a probability distribution for the strings of a corpus, taken as the yields of CFG parse structures.

Structured Language Models

- Abney et al. (1999) show that while PCFGs and PPDAs are weakly equivalent (generate the same classes of probabilistic languages), they have distinct expressive and learning theoretic properties.
- Both PCFGs and SLMs represent linguistic knowledge as a language model that specifies a probability distribution over the strings of a language through the probability values assigned to their syntactic analyses.

Structured Language Models

- Abney et al. (1999) show that while PCFGs and PPDAs are weakly equivalent (generate the same classes of probabilistic languages), they have distinct expressive and learning theoretic properties.
- Both PCFGs and SLMs represent linguistic knowledge as a language model that specifies a probability distribution over the strings of a language through the probability values assigned to their syntactic analyses.

Arguments for the Language Model View of Competence

Clark and Lappin (2011):

- Language models accommodate the fact that we identify the strings of phonemes, words, and phrases of a language from noisy data containing non-well-formed expressions.
- Recent psycholinguistic research (Saffran et al. (1996), Jurafsky (2003), Thompson and Newport (2007)) indicates that frequency effects and probabilistic inference play a central role in acquisition and processing.
- Grammaticality is an abstract theoretical property, which cannot be clearly and consistently observed in the data, but the probability of strings can be measured directly.

Arguments for the Language Model View of Competence

Clark and Lappin (2011):

- Language models accommodate the fact that we identify the strings of phonemes, words, and phrases of a language from noisy data containing non-well-formed expressions.
- Recent psycholinguistic research (Saffran et al. (1996), Jurafsky (2003), Thompson and Newport (2007)) indicates that frequency effects and probabilistic inference play a central role in acquisition and processing.
- Grammaticality is an abstract theoretical property, which cannot be clearly and consistently observed in the data, but the probability of strings can be measured directly.

Arguments for the Language Model View of Competence

Clark and Lappin (2011):

- Language models accommodate the fact that we identify the strings of phonemes, words, and phrases of a language from noisy data containing non-well-formed expressions.
- Recent psycholinguistic research (Saffran et al. (1996), Jurafsky (2003), Thompson and Newport (2007)) indicates that frequency effects and probabilistic inference play a central role in acquisition and processing.
- Grammaticality is an abstract theoretical property, which cannot be clearly and consistently observed in the data, but the probability of strings can be measured directly.

Arguments against the Language Model View of Competence

Clark and Lappin (2011):

- The frequency of expressions often depends on extra-linguistic factors which are not part of linguistic knowledge.
- The acceptability of a string cannot be directly reduced to its probability, given that short ill-formed sentences may have higher probability than long, complex well-formed sentences.

Arguments against the Language Model View of Competence

Clark and Lappin (2011):

- The frequency of expressions often depends on extra-linguistic factors which are not part of linguistic knowledge.
- The acceptability of a string cannot be directly reduced to its probability, given that short ill-formed sentences may have higher probability than long, complex well-formed sentences.

A Non-Argument against the Language Model View

- Chomsky (1957) rejects the use of statistical methods to represent the distinction between grammatical and ungrammatical strings.
    1. Colourless green ideas sleep furiously.
    2. Furiously sleep ideas green colourless.
- (1) and (2) both have a probability approaching nil (in 1957) of appearing in a corpus or actual speech.
- (1) is syntactically well formed, even if semantically anomalous, while (2) is not.

A Non-Argument against the Language Model View

- Chomsky (1957) rejects the use of statistical methods to represent the distinction between grammatical and ungrammatical strings.
    1. Colourless green ideas sleep furiously.
    2. Furiously sleep ideas green colourless.
- (1) and (2) both have a probability approaching nil (in 1957) of appearing in a corpus or actual speech.
- (1) is syntactically well formed, even if semantically anomalous, while (2) is not.

A Non-Argument against the Language Model View

- Chomsky (1957) rejects the use of statistical methods to represent the distinction between grammatical and ungrammatical strings.
    1. Colourless green ideas sleep furiously.
    2. Furiously sleep ideas green colourless.
- (1) and (2) both have a probability approaching nil (in 1957) of appearing in a corpus or actual speech.
- (1) is syntactically well formed, even if semantically anomalous, while (2) is not.

A Non-Argument against the Language Model View

- Chomsky (1957) rejects the use of statistical methods to represent the distinction between grammatical and ungrammatical strings.
    1. Colourless green ideas sleep furiously.
    2. Furiously sleep ideas green colourless.
- (1) and (2) both have a probability approaching nil (in 1957) of appearing in a corpus or actual speech.
- (1) is syntactically well formed, even if semantically anomalous, while (2) is not.

A Smoothed Bigram Model

- Chomsky assumes simple word bigram language models generated by probabilistic finite state automata.
- Pereira (2000) constructs a smoothed bigram model in which the probability of a word depends on the class of the prior word, rather than simply on the preceding word.
- This model computes the conditional probability of a word w_i in a string with the formula

$$P(w_i \mid w_{i-1}) \approx \sum_{c} P(w_i \mid c) P(c \mid w_{i-1})$$

where c is the class of w_{i-1}.

A Smoothed Bigram Model

- Chomsky assumes simple word bigram language models generated by probabilistic finite state automata.
- Pereira (2000) constructs a smoothed bigram model in which the probability of a word depends on the class of the prior word, rather than simply on the preceding word.
- This model computes the conditional probability of a word w_i in a string with the formula

$$P(w_i \mid w_{i-1}) \approx \sum_{c} P(w_i \mid c) P(c \mid w_{i-1})$$

  where c is the class of w_{i-1}.

A Smoothed Bigram Model

- Chomsky assumes simple word bigram language models generated by probabilistic finite state automata.
- Pereira (2000) constructs a smoothed bigram model in which the probability of a word depends on the class of the prior word, rather than simply on the preceding word.
- This model computes the conditional probability of a word w_i in a string with the formula

$$P(w_i \mid w_{i-1}) \approx \sum_c P(w_i \mid c) P(c \mid w_{i-1})$$

where c is the class of w_{i-1}.

A Smoothed Bigram Model

- We can use distributional patterns of words in a corpus to learn their classes from training data.
- Other procedures allow us to compute the values of the parameters $P(w_i \mid c)$ and $P(c \mid w_{i-1})$ from this data.
- When applied to Chomsky's (1957) examples (1) and (2), this model yields a five order of magnitude difference between their probability values for a corpus of newspaper text.

A Smoothed Bigram Model

- We can use distributional patterns of words in a corpus to learn their classes from training data.
- Other procedures allow us to compute the values of the parameters $P(w_i \mid c)$ and $P(c \mid w_{i-1})$ from this data.
- When applied to Chomsky's (1957) examples (1) and (2), this model yields a five order of magnitude difference between their probability values for a corpus of newspaper text.

A Smoothed Bigram Model

- We can use distributional patterns of words in a corpus to learn their classes from training data.
- Other procedures allow us to compute the values of the parameters $P(w_i \mid c)$ and $P(c \mid w_{i-1})$ from this data.
- When applied to Chomsky's (1957) examples (1) and (2), this model yields a five order of magnitude difference between their probability values for a corpus of newspaper text.

A Second Non-Argument against the Language Model View

- Niyogi (2006) and Yang (2008) argue that PCFGs are not good models of linguistic competence because in the distributions that they produce the probability of a string decreases exponentially in proportion to its length.
- In fact, this is not an unreasonable result.
- The probabilities of strings in natural language corpora do decline rapidly in relation to their length.
- Sigurd et al. (2004) show that the probability distribution for sentence lengths in the Brown corpus is accurately modeled by a function that is bounded by an exponentially decaying function.

A Second Non-Argument against the Language Model View

- Niyogi (2006) and Yang (2008) argue that PCFGs are not good models of linguistic competence because in the distributions that they produce the probability of a string decreases exponentially in proportion to its length.
- In fact, this is not an unreasonable result.
- The probabilities of strings in natural language corpora do decline rapidly in relation to their length.
- Sigurd et al. (2004) show that the probability distribution for sentence lengths in the Brown corpus is accurately modeled by a function that is bounded by an exponentially decaying function.

A Second Non-Argument against the Language Model View

- Niyogi (2006) and Yang (2008) argue that PCFGs are not good models of linguistic competence because in the distributions that they produce the probability of a string decreases exponentially in proportion to its length.
- In fact, this is not an unreasonable result.
- The probabilities of strings in natural language corpora do decline rapidly in relation to their length.
- Sigurd et al. (2004) show that the probability distribution for sentence lengths in the Brown corpus is accurately modeled by a function that is bounded by an exponentially decaying function.

A Second Non-Argument against the Language Model View

- Niyogi (2006) and Yang (2008) argue that PCFGs are not good models of linguistic competence because in the distributions that they produce the probability of a string decreases exponentially in proportion to its length.
- In fact, this is not an unreasonable result.
- The probabilities of strings in natural language corpora do decline rapidly in relation to their length.
- Sigurd et al. (2004) show that the probability distribution for sentence lengths in the Brown corpus is accurately modeled by a function that is bounded by an exponentially decaying function.

Indirect Negative Evidence: Inferring Ungrammaticality from Low Frequency

- Indirect negative evidence has been informally posited in the linguistics and acquisition literature, but no attempt has been made to formalize this concept of evidence in a learning model.
- Clark and Lappin (2009, 2011) (C&L) propose a way of doing this that represents indirect negative evidence stochastically as a two-part inference procedure.
- The learner first infers the low probability of a string from its low frequency in the data.
- He/She then derives the ungrammaticality of a string from its comparatively low probability.

Indirect Negative Evidence: Inferring Ungrammaticality from Low Frequency

- Indirect negative evidence has been informally posited in the linguistics and acquisition literature, but no attempt has been made to formalize this concept of evidence in a learning model.
- Clark and Lappin (2009, 2011) (C&L) propose a way of doing this that represents indirect negative evidence stochastically as a two-part inference procedure.
- The learner first infers the low probability of a string from its low frequency in the data.
- He/She then derives the ungrammaticality of a string from its comparatively low probability.

Indirect Negative Evidence: Inferring Ungrammaticality from Low Frequency

- Indirect negative evidence has been informally posited in the linguistics and acquisition literature, but no attempt has been made to formalize this concept of evidence in a learning model.
- Clark and Lappin (2009, 2011) (C&L) propose a way of doing this that represents indirect negative evidence stochastically as a two-part inference procedure.
- The learner first infers the low probability of a string from its low frequency in the data.
- He/She then derives the ungrammaticality of a string from its comparatively low probability.

Indirect Negative Evidence: Inferring Ungrammaticality from Low Frequency

- Indirect negative evidence has been informally posited in the linguistics and acquisition literature, but no attempt has been made to formalize this concept of evidence in a learning model.
- Clark and Lappin (2009, 2011) (C&L) propose a way of doing this that represents indirect negative evidence stochastically as a two-part inference procedure.
- The learner first infers the low probability of a string from its low frequency in the data.
- He/She then derives the ungrammaticality of a string from its comparatively low probability.

From Low Frequency to Low Probability

- C&L assume each sentence in a presentation is generated independently from the same probability distribution, where this is the Independently and Identically Distributed assumption (IID) common in statistical analysis.
- The IID is an idealizing assumption that abstracts away from the obvious probability dependencies among sentences that are conditioned by semantic, dialogue, discourse, and other factors.
- The hope is that over very large amounts of data the IID converges on an approximation of the facts.
- The inference from the low frequency of a string in a data set to its low probability in the distribution for the language follows from the IID.

From Low Frequency to Low Probability

- C&L assume each sentence in a presentation is generated independently from the same probability distribution, where this is the Independently and Identically Distributed assumption (IID) common in statistical analysis.
- The IID is an idealizing assumption that abstracts away from the obvious probability dependencies among sentences that are conditioned by semantic, dialogue, discourse, and other factors.
- The hope is that over very large amounts of data the IID converges on an approximation of the facts.
- The inference from the low frequency of a string in a data set to its low probability in the distribution for the language follows from the IID.

From Low Frequency to Low Probability

- C&L assume each sentence in a presentation is generated independently from the same probability distribution, where this is the Independently and Identically Distributed assumption (IID) common in statistical analysis.
- The IID is an idealizing assumption that abstracts away from the obvious probability dependencies among sentences that are conditioned by semantic, dialogue, discourse, and other factors.
- The hope is that over very large amounts of data the IID converges on an approximation of the facts.
- The inference from the low frequency of a string in a data set to its low probability in the distribution for the language follows from the IID.

From Low Frequency to Low Probability

- C&L assume each sentence in a presentation is generated independently from the same probability distribution, where this is the Independently and Identically Distributed assumption (IID) common in statistical analysis.
- The IID is an idealizing assumption that abstracts away from the obvious probability dependencies among sentences that are conditioned by semantic, dialogue, discourse, and other factors.
- The hope is that over very large amounts of data the IID converges on an approximation of the facts.
- The inference from the low frequency of a string in a data set to its low probability in the distribution for the language follows from the IID.

From Low Probability to Ungrammaticality

- Grammaticality does not reduce to a high probability value for a string.
- Some grammatical strings in a language have vanishingly rare frequency, and so they have low probability
- We also cannot identify ungrammaticality with 0 probability, as some ungrammatical strings do occur in the primary linguistic data.
- We need to specify a suitable lower bound on probability to distinguish grammatical from ungrammatical strings.

From Low Probability to Ungrammaticality

- Grammaticality does not reduce to a high probability value for a string.
- Some grammatical strings in a language have vanishingly rare frequency, and so they have low probability
- We also cannot identify ungrammaticality with 0 probability, as some ungrammatical strings do occur in the primary linguistic data.
- We need to specify a suitable lower bound on probability to distinguish grammatical from ungrammatical strings.

From Low Probability to Ungrammaticality

- Grammaticality does not reduce to a high probability value for a string.
- Some grammatical strings in a language have vanishingly rare frequency, and so they have low probability
- We also cannot identify ungrammaticality with 0 probability, as some ungrammatical strings do occur in the primary linguistic data.
- We need to specify a suitable lower bound on probability to distinguish grammatical from ungrammatical strings.

From Low Probability to Ungrammaticality

- Grammaticality does not reduce to a high probability value for a string.
- Some grammatical strings in a language have vanishingly rare frequency, and so they have low probability
- We also cannot identify ungrammaticality with 0 probability, as some ungrammatical strings do occur in the primary linguistic data.
- We need to specify a suitable lower bound on probability to distinguish grammatical from ungrammatical strings.

A Lower Probability Bound for Grammatical Strings

- Given that the learner learns from unlabelled data, there must be a function from the set of distributions for a language $\mathcal{D}(L)$ to that language.
- This condition entails the Disjoint Distribution Assumption (DDA):
  If $L \neq L'$ then $\mathcal{D}(L) \cap \mathcal{D}(L') = \emptyset$.
- If g is a function that maps a string into a lower bound probability value for grammaticality, relative to a distribution, then we can specify the restricted set of possible distributions for a language as $\mathcal{D}(L, g) = \{D : p_D(s) > g_D(s) \Leftrightarrow s \in L\}$.

A Lower Probability Bound for Grammatical Strings

- Given that the learner learns from unlabelled data, there must be a function from the set of distributions for a language $\mathcal{D}(L)$ to that language.
- This condition entails the Disjoint Distribution Assumption (DDA):
  If $L \neq L'$ then $\mathcal{D}(L) \cap \mathcal{D}(L') = \emptyset$.
- If g is a function that maps a string into a lower bound probability value for grammaticality, relative to a distribution, then we can specify the restricted set of possible distributions for a language as $\mathcal{D}(L, g) = \{D : p_D(s) > g_D(s) \Leftrightarrow s \in L\}$.

A Lower Probability Bound for Grammatical Strings

- Given that the learner learns from unlabelled data, there must be a function from the set of distributions for a language $\mathcal{D}(L)$ to that language.
- This condition entails the Disjoint Distribution Assumption (DDA):
  If $L \neq L'$ then $\mathcal{D}(L) \cap \mathcal{D}(L') = \emptyset$.
- If g is a function that maps a string into a lower bound probability value for grammaticality, relative to a distribution, then we can specify the restricted set of possible distributions for a language as $\mathcal{D}(L, g) = \{D : p_D(s) > g_D(s) \Leftrightarrow s \in L\}$.

Specifying the Threshold Function

- Defining the restricted set of possible distributions in terms of the lower bound function *g* satisfies the DDA.
- To have content this definition must be supplemented with a characterization of *g*.
- It is useful to render *g* dependent on properties of the string (such as its length), and the distribution.
- One way of specifying *g* that is dependent on the distribution is to make it sensitive to the conditional probabilities of a class-based *n*-gram language model of the kind described in Pereira (2000).
- When *g* depends on properties of *D*, the learner will need to estimate these properties in order to determine *g*.

Specifying the Threshold Function

- Defining the restricted set of possible distributions in terms of the lower bound function g satisfies the DDA.
- To have content this definition must be supplemented with a characterization of g.
- It is useful to render g dependent on properties of the string (such as its length), and the distribution.
- One way of specifying g that is dependent on the distribution is to make it sensitive to the conditional probabilities of a class-based n-gram language model of the kind described in Pereira (2000).
- When g depends on properties of D, the learner will need to estimate these properties in order to determine g.

Specifying the Threshold Function

- Defining the restricted set of possible distributions in terms of the lower bound function *g* satisfies the DDA.
- To have content this definition must be supplemented with a characterization of *g*.
- It is useful to render *g* dependent on properties of the string (such as its length), and the distribution.
- One way of specifying *g* that is dependent on the distribution is to make it sensitive to the conditional probabilities of a class-based *n*-gram language model of the kind described in Pereira (2000).
- When *g* depends on properties of *D*, the learner will need to estimate these properties in order to determine *g*.

Specifying the Threshold Function

- Defining the restricted set of possible distributions in terms of the lower bound function g satisfies the DDA.
- To have content this definition must be supplemented with a characterization of g.
- It is useful to render g dependent on properties of the string (such as its length), and the distribution.
- One way of specifying g that is dependent on the distribution is to make it sensitive to the conditional probabilities of a class-based n-gram language model of the kind described in Pereira (2000).
- When g depends on properties of D, the learner will need to estimate these properties in order to determine g.

Specifying the Threshold Function

- Defining the restricted set of possible distributions in terms of the lower bound function *g* satisfies the DDA.
- To have content this definition must be supplemented with a characterization of *g*.
- It is useful to render *g* dependent on properties of the string (such as its length), and the distribution.
- One way of specifying *g* that is dependent on the distribution is to make it sensitive to the conditional probabilities of a class-based *n*-gram language model of the kind described in Pereira (2000).
- When *g* depends on properties of *D*, the learner will need to estimate these properties in order to determine *g*.

Revising PAC Learning with Indirect Evidence

- Given g it is possible to model indirect negative evidence through membership queries on large samples of data.
- The learner can test a number of strings polynomial in the sample for grammaticality by computing the probability of each string s from its frequency, and then comparing its probability to the threshold value $g(s)$.

Revising PAC Learning with Indirect Evidence

- Given g it is possible to model indirect negative evidence through membership queries on large samples of data.
- The learner can test a number of strings polynomial in the sample for grammaticality by computing the probability of each string s from its frequency, and then comparing its probability to the threshold value $g(s)$.

Revising PAC Learning with Indirect Evidence

- C&L revise the definition of PAC learning (Valiant (1984)) so that an algorithm effectively learns a class $\mathcal{L}$ not for every distribution $D \in \mathcal{D}$, but for every distribution $D \in \mathcal{D}(L, g)$.
- In this revised PAC learning paradigm the data set is not labeled, and the set of possible distributions on the data is restricted by a function giving a lower probability bound for membership in the language.

Revising PAC Learning with Indirect Evidence

- C&L revise the definition of PAC learning (Valiant (1984)) so that an algorithm effectively learns a class $\mathcal{L}$ not for every distribution $D \in \mathcal{D}$, but for every distribution $D \in \mathcal{D}(L, g)$.
- In this revised PAC learning paradigm the data set is not labeled, and the set of possible distributions on the data is restricted by a function giving a lower probability bound for membership in the language.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function *g* was originally designed to restrict the set of possible distributions on which learning is required in a probabilistic learning model.
- It can be adapted to a stochastic model of competence in order to identify the set of strings in a language by means of a lower probability bound.
- The function characterizes this bound not simply in terms of the frequency of a string, but its probability as conditioned by a set of features identifiable from the distribution, such as its prefix and suffix patterns, constituent lexical classes, length, etc.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function *g* was originally designed to restrict the set of possible distributions on which learning is required in a probabilistic learning model.
- It can be adapted to a stochastic model of competence in order to identify the set of strings in a language by means of a lower probability bound.
- The function characterizes this bound not simply in terms of the frequency of a string, but its probability as conditioned by a set of features identifiable from the distribution, such as its prefix and suffix patterns, constituent lexical classes, length, etc.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function *g* was originally designed to restrict the set of possible distributions on which learning is required in a probabilistic learning model.
- It can be adapted to a stochastic model of competence in order to identify the set of strings in a language by means of a lower probability bound.
- The function characterizes this bound not simply in terms of the frequency of a string, but its probability as conditioned by a set of features identifiable from the distribution, such as its prefix and suffix patterns, constituent lexical classes, length, etc.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function allows us to characterize competence stochastically without reducing grammaticality to frequency.
- It also permits us to filter out the influence of non-linguistic elements in specifying the lower probability bound for membership in the language.
- Therefore, a language model supplemented with a suitable threshold function avoids the two main problems that Clark and Lappin (2011) cite for stochastic models of competence.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function allows us to characterize competence stochastically without reducing grammaticality to frequency.
- It also permits us to filter out the influence of non-linguistic elements in specifying the lower probability bound for membership in the language.
- Therefore, a language model supplemented with a suitable threshold function avoids the two main problems that Clark and Lappin (2011) cite for stochastic models of competence.

Incorporating the Grammaticality Threshold Function into a Language Model

- The threshold function allows us to characterize competence stochastically without reducing grammaticality to frequency.
- It also permits us to filter out the influence of non-linguistic elements in specifying the lower probability bound for membership in the language.
- Therefore, a language model supplemented with a suitable threshold function avoids the two main problems that Clark and Lappin (2011) cite for stochastic models of competence.

The Relation between Probabilistic Learning and Probabilistic Competence

- In principle, a probabilistic learning algorithm can identify a class of non-probabilistic grammars.
- So, for example, if one specifies a class of FSAs or CFGs that are appropriately bounded in size, then these classes will have finite VC dimensionality, and they will be uniformly PAC learnable (Nowak et al. (2002)).
- However, given the substantial evidence both for probabilistic inference in acquisition, and stochastic effects in linguistic processing and production, it is plausible to seek a theory that connects the probabilistic nature of learning with the design of competence.

The Relation between Probabilistic Learning and Probabilistic Competence

- In principle, a probabilistic learning algorithm can identify a class of non-probabilistic grammars.
- So, for example, if one specifies a class of FSAs or CFGs that are appropriately bounded in size, then these classes will have finite VC dimensionality, and they will be uniformly PAC learnable (Nowak et al. (2002)).
- However, given the substantial evidence both for probabilistic inference in acquisition, and stochastic effects in linguistic processing and production, it is plausible to seek a theory that connects the probabilistic nature of learning with the design of competence.

The Relation between Probabilistic Learning and Probabilistic Competence

- In principle, a probabilistic learning algorithm can identify a class of non-probabilistic grammars.
- So, for example, if one specifies a class of FSAs or CFGs that are appropriately bounded in size, then these classes will have finite VC dimensionality, and they will be uniformly PAC learnable (Nowak et al. (2002)).
- However, given the substantial evidence both for probabilistic inference in acquisition, and stochastic effects in linguistic processing and production, it is plausible to seek a theory that connects the probabilistic nature of learning with the design of competence.

Modifying the Distribution Free Learning Assumption

- The PAC learning paradigm requires that if a class of languages is learnable, then it is uniformly learnable for all probability distributions on data samples from that class.
- By modifying this assumption and restricting the set of possible distributions available for PAC learning in a specified hypothesis space $\mathcal{H}$, it is possible to significantly alter the class of learnable languages.
- This approach uses properties of the probability distributions for a class of languages to facilitate learning of that class, and this can solve computational complexity problems.

Modifying the Distribution Free Learning Assumption

- The PAC learning paradigm requires that if a class of languages is learnable, then it is uniformly learnable for all probability distributions on data samples from that class.
- By modifying this assumption and restricting the set of possible distributions available for PAC learning in a specified hypothesis space $\mathcal{H}$, it is possible to significantly alter the class of learnable languages.
- This approach uses properties of the probability distributions for a class of languages to facilitate learning of that class, and this can solve computational complexity problems.

Modifying the Distribution Free Learning Assumption

- The PAC learning paradigm requires that if a class of languages is learnable, then it is uniformly learnable for all probability distributions on data samples from that class.
- By modifying this assumption and restricting the set of possible distributions available for PAC learning in a specified hypothesis space $\mathcal{H}$, it is possible to significantly alter the class of learnable languages.
- This approach uses properties of the probability distributions for a class of languages to facilitate learning of that class, and this can solve computational complexity problems.

PAC Learning PDFAs

- Clark and Thollard (2004) (C&T) define a set of probabilistic deterministic FSAs (PDFAs), which generate stochastic regular languages (a set of strings in a regular language to which the PDFA assigns probability values).
- C&T show that if we restrict the set of possible distributions for a PAC model to those generated by PDFAs, the class of regular languages that these automata define is PAC learnable, on the basis of positive evidence only.
- Their result depends on a confidence theshold for distinguishing the distinct states of a PDFA on the basis of the distributional properties of the strings in a data set.

PAC Learning PDFAs

- Clark and Thollard (2004) (C&T) define a set of probabilistic deterministic FSAs (PDFAs), which generate stochastic regular languages (a set of strings in a regular language to which the PDFA assigns probability values).
- C&T show that if we restrict the set of possible distributions for a PAC model to those generated by PDFAs, the class of regular languages that these automata define is PAC learnable, on the basis of positive evidence only.
- Their result depends on a confidence theshold for distinguishing the distinct states of a PDFA on the basis of the distributional properties of the strings in a data set.

PAC Learning PDFAs

- Clark and Thollard (2004) (C&T) define a set of probabilistic deterministic FSAs (PDFAs), which generate stochastic regular languages (a set of strings in a regular language to which the PDFA assigns probability values).
- C&T show that if we restrict the set of possible distributions for a PAC model to those generated by PDFAs, the class of regular languages that these automata define is PAC learnable, on the basis of positive evidence only.
- Their result depends on a confidence theshold for distinguishing the distinct states of a PDFA on the basis of the distributional properties of the strings in a data set.

NTS Languages

- A CFG is non-terminally distinct (NTS) if for any two non-terminals *A*, *C* in the grammar, the string sets derivable from *A* and *C* are disjoint.
- This property entails that the phrases of distinct syntactic categories do not overlap.
- Clark (2006) shows that a subclass of CF languages, generated by a restricted set of NTS PCFGs, is PAC learnable from positive evidence only, with restrictions on the probability distributions for these grammars.
- The learning algorithm for an NTS PCFG applies distributionally specified measures of distinctness for identifying its non-terminals from strings in a data set.

NTS Languages

- A CFG is non-terminally distinct (NTS) if for any two non-terminals A, C in the grammar, the string sets derivable from A and C are disjoint.
- This property entails that the phrases of distinct syntactic categories do not overlap.
- Clark (2006) shows that a subclass of CF languages, generated by a restricted set of NTS PCFGs, is PAC learnable from positive evidence only, with restrictions on the probability distributions for these grammars.
- The learning algorithm for an NTS PCFG applies distributionally specified measures of distinctness for identifying its non-terminals from strings in a data set.

NTS Languages

- A CFG is non-terminally distinct (NTS) if for any two non-terminals A, C in the grammar, the string sets derivable from A and C are disjoint.
- This property entails that the phrases of distinct syntactic categories do not overlap.
- Clark (2006) shows that a subclass of CF languages, generated by a restricted set of NTS PCFGs, is PAC learnable from positive evidence only, with restrictions on the probability distributions for these grammars.
- The learning algorithm for an NTS PCFG applies distributionally specified measures of distinctness for identifying its non-terminals from strings in a data set.

NTS Languages

- A CFG is non-terminally distinct (NTS) if for any two non-terminals *A*, *C* in the grammar, the string sets derivable from *A* and *C* are disjoint.
- This property entails that the phrases of distinct syntactic categories do not overlap.
- Clark (2006) shows that a subclass of CF languages, generated by a restricted set of NTS PCFGs, is PAC learnable from positive evidence only, with restrictions on the probability distributions for these grammars.
- The learning algorithm for an NTS PCFG applies distributionally specified measures of distinctness for identifying its non-terminals from strings in a data set.

Connecting Learning and Competence

- Neither PDFAs nor NTS PCFGs are expressively adequate for natural language syntax.
- However, the Clark and Thollard (2004) and Clark (2006) results are important in showing how probabilistic learning can depend upon a stochastic representation of the target class.
- When the set of distributions is restricted to those generated by the target stochastic grammar, uniform learning in the PAC framework is possible.
- This constraint imposes the requirement that the primary linguistic data available to the learner directly reflects the probability structure that adult linguistic competence specifies for the strings of the language.

Connecting Learning and Competence

- Neither PDFAs nor NTS PCFGs are expressively adequate for natural language syntax.
- However, the Clark and Thollard (2004) and Clark (2006) results are important in showing how probabilistic learning can depend upon a stochastic representation of the target class.
- When the set of distributions is restricted to those generated by the target stochastic grammar, uniform learning in the PAC framework is possible.
- This constraint imposes the requirement that the primary linguistic data available to the learner directly reflects the probability structure that adult linguistic competence specifies for the strings of the language.

Connecting Learning and Competence

- Neither PDFAs nor NTS PCFGs are expressively adequate for natural language syntax.
- However, the Clark and Thollard (2004) and Clark (2006) results are important in showing how probabilistic learning can depend upon a stochastic representation of the target class.
- When the set of distributions is restricted to those generated by the target stochastic grammar, uniform learning in the PAC framework is possible.
- This constraint imposes the requirement that the primary linguistic data available to the learner directly reflects the probability structure that adult linguistic competence specifies for the strings of the language.

Connecting Learning and Competence

- Neither PDFAs nor NTS PCFGs are expressively adequate for natural language syntax.
- However, the Clark and Thollard (2004) and Clark (2006) results are important in showing how probabilistic learning can depend upon a stochastic representation of the target class.
- When the set of distributions is restricted to those generated by the target stochastic grammar, uniform learning in the PAC framework is possible.
- This constraint imposes the requirement that the primary linguistic data available to the learner directly reflects the probability structure that adult linguistic competence specifies for the strings of the language.

The Competence-Performance Distinction Revisted

- Representing linguistic knowledge stochastically does not eliminate the competence-performance distinction.
- It is still necessary to distinguish between a probabilistic grammar or automaton that generates a language model, and the parsing algorithm that implements it.
- However, a probabilistic characterization of linguistic knowledge does alter the nature of this distinction.
- The probabilistic properties of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

The Competence-Performance Distinction Revisted

- Representing linguistic knowledge stochastically does not eliminate the competence-performance distinction.
- It is still necessary to distinguish between a probabilistic grammar or automaton that generates a language model, and the parsing algorithm that implements it.
- However, a probabilistic characterization of linguistic knowledge does alter the nature of this distinction.
- The probabilistic properties of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

The Competence-Performance Distinction Revisted

- Representing linguistic knowledge stochastically does not eliminate the competence-performance distinction.
- It is still necessary to distinguish between a probabilistic grammar or automaton that generates a language model, and the parsing algorithm that implements it.
- However, a probabilistic characterization of linguistic knowledge does alter the nature of this distinction.
- The probabilistic properties of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

The Competence-Performance Distinction Revisted

- Representing linguistic knowledge stochastically does not eliminate the competence-performance distinction.
- It is still necessary to distinguish between a probabilistic grammar or automaton that generates a language model, and the parsing algorithm that implements it.
- However, a probabilistic characterization of linguistic knowledge does alter the nature of this distinction.
- The probabilistic properties of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

Conclusions

- Representing linguistic competence stochastically consists in identifying it with a language model, which specifies a distribution over the strings of a language.
- In order for such a theory of competence to be viable, it must incorporate a threshold function that specifies a minimal conditional probability value for recognizing a string as an element of the language.
- This function will depend on properties of the distribution and on a variety of features of the string.

Conclusions

- Representing linguistic competence stochastically consists in identifying it with a language model, which specifies a distribution over the strings of a language.
- In order for such a theory of competence to be viable, it must incorporate a threshold function that specifies a minimal conditional probability value for recognizing a string as an element of the language.
- This function will depend on properties of the distribution and on a variety of features of the string.

Conclusions

- Representing linguistic competence stochastically consists in identifying it with a language model, which specifies a distribution over the strings of a language.
- In order for such a theory of competence to be viable, it must incorporate a threshold function that specifies a minimal conditional probability value for recognizing a string as an element of the language.
- This function will depend on properties of the distribution and on a variety of features of the string.

Conclusions

- Restricting the set of distributions over the data to those generated by particular types of probabilistic grammars yields positive learnability results for those classes of grammars.
- This dependency of probabilistic learning on a probabilistic target representation expresses the condition that grammar induction requires a distribution on the data from which the properties of the target can be effectively recovered.
- Adopting a stochastic model of linguistic knowledge does not dispose of the competence-performance distinction, but it imports gradience effects and probabilistic inference into the theory of competence.

Conclusions

- Restricting the set of distributions over the data to those generated by particular types of probabilistic grammars yields positive learnability results for those classes of grammars.
- This dependency of probabilistic learning on a probabilistic target representation expresses the condition that grammar induction requires a distribution on the data from which the properties of the target can be effectively recovered.
- Adopting a stochastic model of linguistic knowledge does not dispose of the competence-performance distinction, but it imports gradience effects and probabilistic inference into the theory of competence.

Conclusions

- Restricting the set of distributions over the data to those generated by particular types of probabilistic grammars yields positive learnability results for those classes of grammars.
- This dependency of probabilistic learning on a probabilistic target representation expresses the condition that grammar induction requires a distribution on the data from which the properties of the target can be effectively recovered.
- Adopting a stochastic model of linguistic knowledge does not dispose of the competence-performance distinction, but it imports gradience effects and probabilistic inference into the theory of competence.