Probabilistic Semantics for Natural Language

Shalom Lappin and Jan van Eijck

King's College London and CWI Amsterdam

ILLC
September 14, 2011

Outline

Classical Formal Semantic Theories

Gradience and Learning in Semantics

Distributional Models of Meaning

Probabilistic Semantics

Outline

Classical Formal Semantic Theories

Gradience and Learning in Semantics

Distributional Models of Meaning

Probabilistic Semantics

Outline

Classical Formal Semantic Theories

Gradience and Learning in Semantics

Distributional Models of Meaning

Probabilistic Semantics

Outline

Classical Formal Semantic Theories

Gradience and Learning in Semantics

Distributional Models of Meaning

Probabilistic Semantics

Meaning and Denotation in a Model

- A formal semantic theory recursively defines the denotation of an expression in terms of the denotations of its syntactic constituents.
- It computes the semantic values of a sentence as a function of the values of its syntactic constituents.
- Within such a theory the meaning of an expression is identified with a function from indices (the expressions themselves, worlds, situations, times, etc.), to denotations in a model.
- The meaning of a sentence is a function from indices to truth-values.

Meaning and Denotation in a Model

- A formal semantic theory recursively defines the denotation of an expression in terms of the denotations of its syntactic constituents.
- It computes the semantic values of a sentence as a function of the values of its syntactic constituents.
- Within such a theory the meaning of an expression is identified with a function from indices (the expressions themselves, worlds, situations, times, etc.), to denotations in a model.
- The meaning of a sentence is a function from indices to truth-values.

Meaning and Denotation in a Model

- A formal semantic theory recursively defines the denotation of an expression in terms of the denotations of its syntactic constituents.
- It computes the semantic values of a sentence as a function of the values of its syntactic constituents.
- Within such a theory the meaning of an expression is identified with a function from indices (the expressions themselves, worlds, situations, times, etc.), to denotations in a model.
- The meaning of a sentence is a function from indices to truth-values.

Meaning and Denotation in a Model

- A formal semantic theory recursively defines the denotation of an expression in terms of the denotations of its syntactic constituents.
- It computes the semantic values of a sentence as a function of the values of its syntactic constituents.
- Within such a theory the meaning of an expression is identified with a function from indices (the expressions themselves, worlds, situations, times, etc.), to denotations in a model.
- The meaning of a sentence is a function from indices to truth-values.

The Competence-Performance Distinction in Semantics

- Formal semantic theories model both lexical and phrasal meaning through categorical rules and algebraic systems that cannot accommodate gradience effects.
- This approach is common to theories which sustain compositionality and those with employ underspecified representations.
- It effectively invokes the same strong version of the competence-performance distinction that categorical models of syntax assume.
- This view of linguistic knowledge has dominated linguistic theory for the past fifty years.

The Competence-Performance Distinction in Semantics

- Formal semantic theories model both lexical and phrasal meaning through categorical rules and algebraic systems that cannot accommodate gradience effects.
- This approach is common to theories which sustain compositionality and those with employ underspecified representations.
- It effectively invokes the same strong version of the competence-performance distinction that categorical models of syntax assume.
- This view of linguistic knowledge has dominated linguistic theory for the past fifty years.

The Competence-Performance Distinction in Semantics

- Formal semantic theories model both lexical and phrasal meaning through categorical rules and algebraic systems that cannot accommodate gradience effects.
- This approach is common to theories which sustain compositionality and those with employ underspecified representations.
- It effectively invokes the same strong version of the competence-performance distinction that categorical models of syntax assume.
- This view of linguistic knowledge has dominated linguistic theory for the past fifty years.

The Competence-Performance Distinction in Semantics

- Formal semantic theories model both lexical and phrasal meaning through categorical rules and algebraic systems that cannot accommodate gradience effects.
- This approach is common to theories which sustain compositionality and those with employ underspecified representations.
- It effectively invokes the same strong version of the competence-performance distinction that categorical models of syntax assume.
- This view of linguistic knowledge has dominated linguistic theory for the past fifty years.

Explaining Gradience in Linguistic Representation

- Gradient effects in representation are ubiquitous throughout linguistic and other cognitive domains.
- Appeal to performance factors to explain gradience has no explanatory content unless it is supported by a precise account of how the interaction of competence and performance generates these effects in each case.
- By contrast, gradience is intrinsic to the formal models that information theoretic methods use to represent events and processes.

Explaining Gradience in Linguistic Representation

- Gradient effects in representation are ubiquitous throughout linguistic and other cognitive domains.
- Appeal to performance factors to explain gradience has no explanatory content unless it is supported by a precise account of how the interaction of competence and performance generates these effects in each case.
- By contrast, gradience is intrinsic to the formal models that information theoretic methods use to represent events and processes.

Explaining Gradience in Linguistic Representation

- Gradient effects in representation are ubiquitous throughout linguistic and other cognitive domains.
- Appeal to performance factors to explain gradience has no explanatory content unless it is supported by a precise account of how the interaction of competence and performance generates these effects in each case.
- By contrast, gradience is intrinsic to the formal models that information theoretic methods use to represent events and processes.

Three Views of Natural Language

- Bach (1986) identifies two theses on the character of natural language.
  - (a) Chomsky's thesis: natural languages can be described as formal systems.
  - (b) Montague's thesis: natural languages can be described as *interpreted* formal systems.
- Recent work in computational linguistics and cognitive modeling suggests a third proposal.
  - (c) The Harris-Jelinek thesis: natural languages can be described as information theoretic systems, using stochastic models that express the distributional properties of their elements.

Three Views of Natural Language

- Bach (1986) identifies two theses on the character of natural language.
  - (a) Chomsky's thesis: natural languages can be described as formal systems.
  - (b) Montague's thesis: natural languages can be described as *interpreted* formal systems.
- Recent work in computational linguistics and cognitive modeling suggests a third proposal.
  - (c) The Harris-Jelinek thesis: natural languages can be described as information theoretic systems, using stochastic models that express the distributional properties of their elements.

Three Views of Natural Language

- Bach (1986) identifies two theses on the character of natural language.
  - (a) Chomsky's thesis: natural languages can be described as formal systems.
  - (b) Montague's thesis: natural languages can be described as *interpreted* formal systems.
- Recent work in computational linguistics and cognitive modeling suggests a third proposal.
  - (c) The Harris-Jelinek thesis: natural languages can be described as information theoretic systems, using stochastic models that express the distributional properties of their elements.

Three Views of Natural Language

- Bach (1986) identifies two theses on the character of natural language.
  - (a) Chomsky's thesis: natural languages can be described as formal systems.
  - (b) Montague's thesis: natural languages can be described as *interpreted* formal systems.
- Recent work in computational linguistics and cognitive modeling suggests a third proposal.
  - (c) The Harris-Jelinek thesis: natural languages can be described as information theoretic systems, using stochastic models that express the distributional properties of their elements.

Three Views of Natural Language

- Bach (1986) identifies two theses on the character of natural language.
  (a) Chomsky's thesis: natural languages can be described as formal systems.
  (b) Montague's thesis: natural languages can be described as *interpreted* formal systems.
- Recent work in computational linguistics and cognitive modeling suggests a third proposal.
  (c) The Harris-Jelinek thesis: natural languages can be described as information theoretic systems, using stochastic models that express the distributional properties of their elements.

The Language Model Hypothesis

- The Language Model Hypothesis (LMH) for Syntax: Grammatical knowledge is represented as a stochastic language model.
- On the LMH, a speaker acquires a probability distribution $D : \Sigma^* \rightarrow [0, 1]$, over the strings $s \in \Sigma^*$, where Σ is a set of words (phonemes, morphemes, etc.) of the language, and, for any finite subset of Σ^*, $\sum p_D(s) = 1$.
- This distribution is generated by a probabilistic automaton or a probabilistic grammar.
- This distribution represents the probability of a sentence's occurrence in a corpus.
- A probabilistic semantics needs to express the probability of a different property.

The Language Model Hypothesis

- The Language Model Hypothesis (LMH) for Syntax: Grammatical knowledge is represented as a stochastic language model.
- On the LMH, a speaker acquires a probability distribution $D : \Sigma^* \rightarrow [0, 1]$, over the strings $s \in \Sigma^*$, where Σ is a set of words (phonemes, morphemes, etc.) of the language, and, for any finite subset of Σ^*, $\sum p_D(s) = 1$.
- This distribution is generated by a probabilistic automaton or a probabilistic grammar.
- This distribution represents the probability of a sentence's occurrence in a corpus.
- A probabilistic semantics needs to express the probability of a different property.

The Language Model Hypothesis

- The Language Model Hypothesis (LMH) for Syntax: Grammatical knowledge is represented as a stochastic language model.
- On the LMH, a speaker acquires a probability distribution $D : \Sigma^* \rightarrow [0, 1]$, over the strings $s \in \Sigma^*$, where Σ is a set of words (phonemes, morphemes, etc.) of the language, and, for any finite subset of Σ^*, $\sum p_D(s) = 1$.
- This distribution is generated by a probabilistic automaton or a probabilistic grammar.
- This distribution represents the probability of a sentence's occurrence in a corpus.
- A probabilistic semantics needs to express the probability of a different property.

The Language Model Hypothesis

- The Language Model Hypothesis (LMH) for Syntax: Grammatical knowledge is represented as a stochastic language model.
- On the LMH, a speaker acquires a probability distribution $D : \Sigma^* \rightarrow [0, 1]$, over the strings $s \in \Sigma^*$, where Σ is a set of words (phonemes, morphemes, etc.) of the language, and, for any finite subset of Σ^*, $\sum p_D(s) = 1$.
- This distribution is generated by a probabilistic automaton or a probabilistic grammar.
- This distribution represents the probability of a sentence's occurrence in a corpus.
- A probabilistic semantics needs to express the probability of a different property.

The Language Model Hypothesis

- The Language Model Hypothesis (LMH) for Syntax: Grammatical knowledge is represented as a stochastic language model.
- On the LMH, a speaker acquires a probability distribution $D : \Sigma^* \rightarrow [0, 1]$, over the strings $s \in \Sigma^*$, where Σ is a set of words (phonemes, morphemes, etc.) of the language, and, for any finite subset of Σ^*, $\sum p_D(s) = 1$.
- This distribution is generated by a probabilistic automaton or a probabilistic grammar.
- This distribution represents the probability of a sentence's occurrence in a corpus.
- A probabilistic semantics needs to express the probability of a different property.

Reformulating the Competence-Performance Distinction

- 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 gradience of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

Reformulating the Competence-Performance Distinction

- 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 gradience of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

Reformulating the Competence-Performance Distinction

- 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 gradience of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

Reformulating the Competence-Performance Distinction

- 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 gradience of linguistic judgements and the defeasibility of grammatical constraints are now intrinsic to linguistic competence, rather than distorting factors contributed by performance mechanisms.

Gradience in Semantic Properties and Relations

- Lexically mediated relations like synonymy, antinomy, polysemy, and hyponymy are notoriously prone to clustering and overlap effects.
- They hold for pairs of expressions over a continuum of degrees [0,1], rather than Boolean values {1,0}.
- Moreover, the denotations of major semantic types, like the predicates corresponding to Ns, AdjPs, and VPs, can rarely, if ever, be identified as sets with determinate membership.
- The case for abandoning the categorical view of competence and adopting a probabilistic model is at least as strong in semantics as it is in syntax (as well as in other parts of the grammar).

Gradience in Semantic Properties and Relations

- Lexically mediated relations like synonymy, antinomy, polysemy, and hyponymy are notoriously prone to clustering and overlap effects.
- They hold for pairs of expressions over a continuum of degrees [0,1], rather than Boolean values {1,0}.
- Moreover, the denotations of major semantic types, like the predicates corresponding to Ns, AdjPs, and VPs, can rarely, if ever, be identified as sets with determinate membership.
- The case for abandoning the categorical view of competence and adopting a probabilistic model is at least as strong in semantics as it is in syntax (as well as in other parts of the grammar).

Gradience in Semantic Properties and Relations

- Lexically mediated relations like synonymy, antinomy, polysemy, and hyponymy are notoriously prone to clustering and overlap effects.
- They hold for pairs of expressions over a continuum of degrees [0,1], rather than Boolean values {1,0}.
- Moreover, the denotations of major semantic types, like the predicates corresponding to Ns, AdjPs, and VPs, can rarely, if ever, be identified as sets with determinate membership.
- The case for abandoning the categorical view of competence and adopting a probabilistic model is at least as strong in semantics as it is in syntax (as well as in other parts of the grammar).

Gradience in Semantic Properties and Relations

- Lexically mediated relations like synonymy, antinomy, polysemy, and hyponymy are notoriously prone to clustering and overlap effects.
- They hold for pairs of expressions over a continuum of degrees [0,1], rather than Boolean values {1,0}.
- Moreover, the denotations of major semantic types, like the predicates corresponding to Ns, AdjPs, and VPs, can rarely, if ever, be identified as sets with determinate membership.
- The case for abandoning the categorical view of competence and adopting a probabilistic model is at least as strong in semantics as it is in syntax (as well as in other parts of the grammar).

Semantic Learning

- Classical semantic theories characterize a class of representations for the set of meanings of expressions in natural language.
- However, it is unclear how these representations could be learned from the primary linguistic data of language acquisition.
- The problem of developing a plausible account of efficient learnability of appropriate target representations is as important for semantics as it is for other types of linguistic knowledge.
- Most work in formal learning for natural languages has focussed on syntax (grammar induction), morphology, and phonology.

Semantic Learning

- Classical semantic theories characterize a class of representations for the set of meanings of expressions in natural language.
- However, it is unclear how these representations could be learned from the primary linguistic data of language acquisition.
- The problem of developing a plausible account of efficient learnability of appropriate target representations is as important for semantics as it is for other types of linguistic knowledge.
- Most work in formal learning for natural languages has focussed on syntax (grammar induction), morphology, and phonology.

Semantic Learning

- Classical semantic theories characterize a class of representations for the set of meanings of expressions in natural language.
- However, it is unclear how these representations could be learned from the primary linguistic data of language acquisition.
- The problem of developing a plausible account of efficient learnability of appropriate target representations is as important for semantics as it is for other types of linguistic knowledge.
- Most work in formal learning for natural languages has focussed on syntax (grammar induction), morphology, and phonology.

Semantic Learning

- Classical semantic theories characterize a class of representations for the set of meanings of expressions in natural language.
- However, it is unclear how these representations could be learned from the primary linguistic data of language acquisition.
- The problem of developing a plausible account of efficient learnability of appropriate target representations is as important for semantics as it is for other types of linguistic knowledge.
- Most work in formal learning for natural languages has focussed on syntax (grammar induction), morphology, and phonology.

Vector Space Models

- Vector Space Models (VSMs) (Turney and Pantel (2010)) offer a fine-grained distributional method for identifying a range of semantic relations among words and phrases.
- They are constructed from matrices in which words are listed vertically on the left, and the environments in which they appear are given horizontally along the top.
- These environments specify the dimensions of the model, corresponding to words, phrases, documents, units of discourse, or any other objects for tracking the occurrence of words.
- They can also include data structures encoding extra-linguistic elements, like visual scenes and events.

Vector Space Models

- Vector Space Models (VSMs) (Turney and Pantel (2010)) offer a fine-grained distributional method for identifying a range of semantic relations among words and phrases.
- They are constructed from matrices in which words are listed vertically on the left, and the environments in which they appear are given horizontally along the top.
- These environments specify the dimensions of the model, corresponding to words, phrases, documents, units of discourse, or any other objects for tracking the occurrence of words.
- They can also include data structures encoding extra-linguistic elements, like visual scenes and events.

Vector Space Models

- Vector Space Models (VSMs) (Turney and Pantel (2010)) offer a fine-grained distributional method for identifying a range of semantic relations among words and phrases.
- They are constructed from matrices in which words are listed vertically on the left, and the environments in which they appear are given horizontally along the top.
- These environments specify the dimensions of the model, corresponding to words, phrases, documents, units of discourse, or any other objects for tracking the occurrence of words.
- They can also include data structures encoding extra-linguistic elements, like visual scenes and events.

Vector Space Models

- Vector Space Models (VSMs) (Turney and Pantel (2010)) offer a fine-grained distributional method for identifying a range of semantic relations among words and phrases.
- They are constructed from matrices in which words are listed vertically on the left, and the environments in which they appear are given horizontally along the top.
- These environments specify the dimensions of the model, corresponding to words, phrases, documents, units of discourse, or any other objects for tracking the occurrence of words.
- They can also include data structures encoding extra-linguistic elements, like visual scenes and events.

A Word-Context Matrix

	context 1	context 2	context 3	context 4
financial	0	6	4	8
market	1	0	15	9
share	5	0	0	4
economic	0	1	26	12
chip	7	8	0	0
distributed	11	15	0	0
sequential	10	31	0	1
algorithm	14	22	2	1

Matrices and Vectors

- The integers in the cells of the matrix give the frequency of the word in an environment.
- A vector for a word is the row of values across the dimension columns of the matrix.
- The vectors for *chip* and *algorithm* are [7 8 0 0] and [14 22 2 1], respectively.

Matrices and Vectors

- The integers in the cells of the matrix give the frequency of the word in an environment.
- A vector for a word is the row of values across the dimension columns of the matrix.
- The vectors for *chip* and *algorithm* are [7 8 0 0] and [14 22 2 1], respectively.

Matrices and Vectors

- The integers in the cells of the matrix give the frequency of the word in an environment.
- A vector for a word is the row of values across the dimension columns of the matrix.
- The vectors for *chip* and *algorithm* are [7 8 0 0] and [14 22 2 1], respectively.

Measuring Semantic Distance

- A pair of vectors from a matrix can be projected as lines from a common point on a plane.
- The smaller the angle between the lines, the greater the similarity of the terms, as measured by their co-occurrence across the dimensions of the matrix.
- Computing the *cosine* of this angle is a convenient way of measuring the angles between vector pairs.
- If $\vec{x} = \langle x_1, x_2, ..., x_n \rangle$ and $\vec{y} = \langle y_1, y_2, ..., y_n \rangle$ are two vectors, then

$$cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}}$$

Measuring Semantic Distance

- A pair of vectors from a matrix can be projected as lines from a common point on a plane.
- The smaller the angle between the lines, the greater the similarity of the terms, as measured by their co-occurrence across the dimensions of the matrix.
- Computing the *cosine* of this angle is a convenient way of measuring the angles between vector pairs.
- If $\vec{x} = \langle x_1, x_2, ..., x_n \rangle$ and $\vec{y} = \langle y_1, y_2, ..., y_n \rangle$ are two vectors, then

$$cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}}$$

Measuring Semantic Distance

- A pair of vectors from a matrix can be projected as lines from a common point on a plane.
- The smaller the angle between the lines, the greater the similarity of the terms, as measured by their co-occurrence across the dimensions of the matrix.
- Computing the *cosine* of this angle is a convenient way of measuring the angles between vector pairs.
- If $\vec{x} = \langle x_1, x_2, ..., x_n \rangle$ and $\vec{y} = \langle y_1, y_2, ..., y_n \rangle$ are two vectors, then

$$cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}}$$

Measuring Semantic Distance

- A pair of vectors from a matrix can be projected as lines from a common point on a plane.
- The smaller the angle between the lines, the greater the similarity of the terms, as measured by their co-occurrence across the dimensions of the matrix.
- Computing the *cosine* of this angle is a convenient way of measuring the angles between vector pairs.
- If $\vec{x} = \langle x_1, x_2, ..., x_n \rangle$ and $\vec{y} = \langle y_1, y_2, ..., y_n \rangle$ are two vectors, then

$$cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}}$$

Measuring Semantic Distance

- The cosine of $\vec{x}$ and $\vec{y}$ is their internal product, formed by summing the products of the corresponding elements of the two vectors and normalizing the result relative to the lengths of the vectors.
- In computing $cos(\vec{x}, \vec{y})$ it may be desirable to apply a smoothing function to the raw frequency counts in each vector to compensate for sparse data, or to filter out the effects of high frequency terms.
- A higher value for $cos(\vec{x}, \vec{y})$ correlates with greater semantic relatedness of the terms associated with the $\vec{x}$ and $\vec{y}$ vectors.

Measuring Semantic Distance

- The cosine of $\vec{x}$ and $\vec{y}$ is their internal product, formed by summing the products of the corresponding elements of the two vectors and normalizing the result relative to the lengths of the vectors.
- In computing $cos(\vec{x}, \vec{y})$ it may be desirable to apply a smoothing function to the raw frequency counts in each vector to compensate for sparse data, or to filter out the effects of high frequency terms.
- A higher value for $cos(\vec{x}, \vec{y})$ correlates with greater semantic relatedness of the terms associated with the $\vec{x}$ and $\vec{y}$ vectors.

Measuring Semantic Distance

- The cosine of $\vec{x}$ and $\vec{y}$ is their internal product, formed by summing the products of the corresponding elements of the two vectors and normalizing the result relative to the lengths of the vectors.
- In computing $cos(\vec{x}, \vec{y})$ it may be desirable to apply a smoothing function to the raw frequency counts in each vector to compensate for sparse data, or to filter out the effects of high frequency terms.
- A higher value for $cos(\vec{x}, \vec{y})$ correlates with greater semantic relatedness of the terms associated with the $\vec{x}$ and $\vec{y}$ vectors.

VSMs as Representations of Lexical Meaning and Learning

- VSMs provide highly successful methods for identifying a variety of lexical semantic relations, including synonymy, antinomy, polysemy, and hypernym classes.
- They also perform very well in unsupervised sense disambiguation tasks.
- VSMs offer a distributional view of lexical semantic learning.
- On this approach speakers acquire lexical meaning by estimating the environments (linguistic and non-linguistic) in which the words of their language appear.

VSMs as Representations of Lexical Meaning and Learning

- VSMs provide highly successful methods for identifying a variety of lexical semantic relations, including synonymy, antinomy, polysemy, and hypernym classes.
- They also perform very well in unsupervised sense disambiguation tasks.
- VSMs offer a distributional view of lexical semantic learning.
- On this approach speakers acquire lexical meaning by estimating the environments (linguistic and non-linguistic) in which the words of their language appear.

VSMs as Representations of Lexical Meaning and Learning

- VSMs provide highly successful methods for identifying a variety of lexical semantic relations, including synonymy, antinomy, polysemy, and hypernym classes.
- They also perform very well in unsupervised sense disambiguation tasks.
- VSMs offer a distributional view of lexical semantic learning.
- On this approach speakers acquire lexical meaning by estimating the environments (linguistic and non-linguistic) in which the words of their language appear.

VSMs as Representations of Lexical Meaning and Learning

- VSMs provide highly successful methods for identifying a variety of lexical semantic relations, including synonymy, antinomy, polysemy, and hypernym classes.
- They also perform very well in unsupervised sense disambiguation tasks.
- VSMs offer a distributional view of lexical semantic learning.
- On this approach speakers acquire lexical meaning by estimating the environments (linguistic and non-linguistic) in which the words of their language appear.

Compositional VSMs (CVSMs)

- VSMs measure semantic distances and relations among words independently of syntactic structure (bag of words).
- Recent work has sought both to integrate syntactic information into the dimensions of the vector matrices (Pado and Lapata (2007)), and to extend VSM semantic spaces to the compositional meanings of sentences.
- Mitchell and Lapata (2008) compare additive and multiplicative models for computing the vectors of complex syntactic constituents, and they demonstrate better results with the latter for sentential semantic similarity tasks.
- These models use simple functions for combining constituent vectors, and they do not represent the dependence of composite vectors on syntactic structure.

Compositional VSMs (CVSMs)

- VSMs measure semantic distances and relations among words independently of syntactic structure (bag of words).
- Recent work has sought both to integrate syntactic information into the dimensions of the vector matrices (Pado and Lapata (2007)), and to extend VSM semantic spaces to the compositional meanings of sentences.
- Mitchell and Lapata (2008) compare additive and multiplicative models for computing the vectors of complex syntactic constituents, and they demonstrate better results with the latter for sentential semantic similarity tasks.
- These models use simple functions for combining constituent vectors, and they do not represent the dependence of composite vectors on syntactic structure.

Compositional VSMs (CVSMs)

- VSMs measure semantic distances and relations among words independently of syntactic structure (bag of words).
- Recent work has sought both to integrate syntactic information into the dimensions of the vector matrices (Pado and Lapata (2007)), and to extend VSM semantic spaces to the compositional meanings of sentences.
- Mitchell and Lapata (2008) compare additive and multiplicative models for computing the vectors of complex syntactic constituents, and they demonstrate better results with the latter for sentential semantic similarity tasks.
- These models use simple functions for combining constituent vectors, and they do not represent the dependence of composite vectors on syntactic structure.

Compositional VSMs (CVSMs)

- VSMs measure semantic distances and relations among words independently of syntactic structure (bag of words).
- Recent work has sought both to integrate syntactic information into the dimensions of the vector matrices (Pado and Lapata (2007)), and to extend VSM semantic spaces to the compositional meanings of sentences.
- Mitchell and Lapata (2008) compare additive and multiplicative models for computing the vectors of complex syntactic constituents, and they demonstrate better results with the latter for sentential semantic similarity tasks.
- These models use simple functions for combining constituent vectors, and they do not represent the dependence of composite vectors on syntactic structure.

A Syntactically Driven Compositional VSM

- Coecke et al. (2010) and Grefenstette et al. (2011) propose a procedure for computing vector values for sentences that specifies a correspondence between the vectors and the syntactic structures of their constituents.
- This procedure relies upon a category theoretic representation of the types of a pregroup grammar (PGG, Lambek (2007,2008)), which builds up complex syntactic categories through direction-marked function application in a manner similar to a basic categorial grammar.
- All sentences receive vectors in the same vector space, and so they can be compared for semantic similarity using measures like cosine.

A Syntactically Driven Compositional VSM

- Coecke et al. (2010) and Grefenstette et al. (2011) propose a procedure for computing vector values for sentences that specifies a correspondence between the vectors and the syntactic structures of their constituents.
- This procedure relies upon a category theoretic representation of the types of a pregroup grammar (PGG, Lambek (2007,2008)), which builds up complex syntactic categories through direction-marked function application in a manner similar to a basic categorial grammar.
- All sentences receive vectors in the same vector space, and so they can be compared for semantic similarity using measures like cosine.

A Syntactically Driven Compositional VSM

- Coecke et al. (2010) and Grefenstette et al. (2011) propose a procedure for computing vector values for sentences that specifies a correspondence between the vectors and the syntactic structures of their constituents.
- This procedure relies upon a category theoretic representation of the types of a pregroup grammar (PGG, Lambek (2007,2008)), which builds up complex syntactic categories through direction-marked function application in a manner similar to a basic categorial grammar.
- All sentences receive vectors in the same vector space, and so they can be compared for semantic similarity using measures like cosine.

Tensor Products

- A PGG CVSM computes the values of a complex syntactic structure through a function that computes the tensor product of the vectors of its constituents, while encoding the correspondence between their grammatical types and their semantic vectors.
- For two (finite) vector spaces A, B, their tensor product $A \otimes B$ is constructed from the Cartesian product of the vectors in A and B.
- For any two vectors $v \in A$, $w \in B$, $v \otimes w$ is the vector consisting of all possible products $v_{i \in v} \times w_{j \in w}$.
- Smolensky (1990) uses tensor products of vector spaces to construct representations of complex structures (strings and trees) from the distributed variables and values of the units in a connectionist network.

Tensor Products

- A PGG CVSM computes the values of a complex syntactic structure through a function that computes the tensor product of the vectors of its constituents, while encoding the correspondence between their grammatical types and their semantic vectors.
- For two (finite) vector spaces A, B, their tensor product $A \otimes B$ is constructed from the Cartesian product of the vectors in A and B.
- For any two vectors $v \in A$, $w \in B$, $v \otimes w$ is the vector consisting of all possible products $v_{i \in v} \times w_{j \in w}$.
- Smolensky (1990) uses tensor products of vector spaces to construct representations of complex structures (strings and trees) from the distributed variables and values of the units in a connectionist network.

Tensor Products

- A PGG CVSM computes the values of a complex syntactic structure through a function that computes the tensor product of the vectors of its constituents, while encoding the correspondence between their grammatical types and their semantic vectors.
- For two (finite) vector spaces A, B, their tensor product $A \otimes B$ is constructed from the Cartesian product of the vectors in A and B.
- For any two vectors $v \in A$, $w \in B$, $v \otimes w$ is the vector consisting of all possible products $v_{i \in v} \times w_{j \in w}$.
- Smolensky (1990) uses tensor products of vector spaces to construct representations of complex structures (strings and trees) from the distributed variables and values of the units in a connectionist network.

Tensor Products

- A PGG CVSM computes the values of a complex syntactic structure through a function that computes the tensor product of the vectors of its constituents, while encoding the correspondence between their grammatical types and their semantic vectors.
- For two (finite) vector spaces A, B, their tensor product $A \otimes B$ is constructed from the Cartesian product of the vectors in A and B.
- For any two vectors $v \in A$, $w \in B$, $v \otimes w$ is the vector consisting of all possible products $v_{i \in v} \times w_{j \in w}$.
- Smolensky (1990) uses tensor products of vector spaces to construct representations of complex structures (strings and trees) from the distributed variables and values of the units in a connectionist network.

Computing the Vector of a Sentence

- PGGs are modeled as *compact closed categories*.
- A sentence vector is computed by a linear map f on the tensor product for the vectors of its main constituents, where f stores the type categorial structure of the string determined by its PGG representation.
- The vector for a sentence headed by a transitive verb, for example, is computed according to the equation

$$\overrightarrow{subj\ V_{tr}\ obj} = f(\overrightarrow{subj} \otimes \overrightarrow{V_{tr}} \otimes \overrightarrow{obj})$$

Computing the Vector of a Sentence

- PGGs are modeled as *compact closed categories*.
- A sentence vector is computed by a linear map f on the tensor product for the vectors of its main constituents, where f stores the type categorial structure of the string determined by its PGG representation.
- The vector for a sentence headed by a transitive verb, for example, is computed according to the equation

$$\overrightarrow{subj\ V_{tr}\ obj} = f(\overrightarrow{subj} \otimes \overrightarrow{V_{tr}} \otimes \overrightarrow{obj})$$

Computing the Vector of a Sentence

- PGGs are modeled as *compact closed categories*.
- A sentence vector is computed by a linear map f on the tensor product for the vectors of its main constituents, where f stores the type categorial structure of the string determined by its PGG representation.
- The vector for a sentence headed by a transitive verb, for example, is computed according to the equation

$$\overrightarrow{subj\ V_{tr}\ obj} = f(\overrightarrow{subj} \otimes \overrightarrow{V_{tr}} \otimes \overrightarrow{obj})$$

Computing the Vector of a Sentence

- The vector of a transitive verb V_{tr} could be taken to be an element of the tensor product of the vector spaces for the two noun bases corresponding to its possible subject and object arguments $\overrightarrow{V_{tr}} \in N \otimes N$.
- Then the vector for a sentence headed by a transitive verb could be computed as the point-wise product of the verb's vector, and the tensor product of its subject and its object.

$$\overrightarrow{subj\ V_{tr}\ obj} = \overrightarrow{V_{tr}} \odot (\overrightarrow{subj} \otimes \overrightarrow{obj})$$

Computing the Vector of a Sentence

- The vector of a transitive verb V_{tr} could be taken to be an element of the tensor product of the vector spaces for the two noun bases corresponding to its possible subject and object arguments $\overrightarrow{V_{tr}} \in N \otimes N$.
- Then the vector for a sentence headed by a transitive verb could be computed as the point-wise product of the verb's vector, and the tensor product of its subject and its object.

$$\overrightarrow{subj\ V_{tr}\ obj} = \overrightarrow{V_{tr}} \odot (\overrightarrow{subj} \otimes \overrightarrow{obj})$$

Advantages of PGG Compositional VSMs

- PGG CVSMs offer a formally grounded and computationally efficient method for obtaining vectors for complex expressions from their syntactic constituents.
- They permit the same kind of measurement for relations of semantic similarity among sentences that lexical VSMs give for word pairs.
- They can be trained on a (PGG parsed) corpus, and their performance evaluated against human annotators' semantic judgements for phrases and sentences.

Advantages of PGG Compositional VSMs

- PGG CVSMs offer a formally grounded and computationally efficient method for obtaining vectors for complex expressions from their syntactic constituents.
- They permit the same kind of measurement for relations of semantic similarity among sentences that lexical VSMs give for word pairs.
- They can be trained on a (PGG parsed) corpus, and their performance evaluated against human annotators' semantic judgements for phrases and sentences.

Advantages of PGG Compositional VSMs

- PGG CVSMs offer a formally grounded and computationally efficient method for obtaining vectors for complex expressions from their syntactic constituents.
- They permit the same kind of measurement for relations of semantic similarity among sentences that lexical VSMs give for word pairs.
- They can be trained on a (PGG parsed) corpus, and their performance evaluated against human annotators' semantic judgements for phrases and sentences.

Problems with CVSMs

- Although the vector of a complex expression is the value of a linear map on the vectors of its parts, it is not obvious what independent property this vector represents.
- Sentential vectors do not correspond to the distributional properties of these sentences, as the data in the PLD is too sparse to estimate distributional vectors for all but a few sentences, across most dimensions.
- Coecke et al. 2010 show that it is possible to encode a classical model theoretic semantics in their system by using vectors to express sets, relations, and truth-values.
- But CVSMs are interesting to the extent that the sentential vectors that they assign are derived from lexical vectors that represent the distributional properties of these expressions.

Problems with CVSMs

- Although the vector of a complex expression is the value of a linear map on the vectors of its parts, it is not obvious what independent property this vector represents.
- Sentential vectors do not correspond to the distributional properties of these sentences, as the data in the PLD is too sparse to estimate distributional vectors for all but a few sentences, across most dimensions.
- Coecke et al. 2010 show that it is possible to encode a classical model theoretic semantics in their system by using vectors to express sets, relations, and truth-values.
- But CVSMs are interesting to the extent that the sentential vectors that they assign are derived from lexical vectors that represent the distributional properties of these expressions.

Problems with CVSMs

- Although the vector of a complex expression is the value of a linear map on the vectors of its parts, it is not obvious what independent property this vector represents.
- Sentential vectors do not correspond to the distributional properties of these sentences, as the data in the PLD is too sparse to estimate distributional vectors for all but a few sentences, across most dimensions.
- Coecke et al. 2010 show that it is possible to encode a classical model theoretic semantics in their system by using vectors to express sets, relations, and truth-values.
- But CVSMs are interesting to the extent that the sentential vectors that they assign are derived from lexical vectors that represent the distributional properties of these expressions.

Problems with CVSMs

- Although the vector of a complex expression is the value of a linear map on the vectors of its parts, it is not obvious what independent property this vector represents.
- Sentential vectors do not correspond to the distributional properties of these sentences, as the data in the PLD is too sparse to estimate distributional vectors for all but a few sentences, across most dimensions.
- Coecke et al. 2010 show that it is possible to encode a classical model theoretic semantics in their system by using vectors to express sets, relations, and truth-values.
- But CVSMs are interesting to the extent that the sentential vectors that they assign are derived from lexical vectors that represent the distributional properties of these expressions.

Classical Formal Semantic Theories vs CVSMs

- In classical formal semantic theories the functions that drive semantic composition are supplied by the type theory, where the type of each expression specifies the formal character of its denotation in a model.
- The sequence of functions that determines the semantic value of a sentence exhibits at each point a value that directly corresponds to an independently motivated semantic property of the expression to which it is assigned.
- Types of denotation provide non-arbitrary formal relations between types of expressions and classes of entities specified relative to a model.
- The sentential vectors obtained from distributional vectors of lexical items lack this sort of independent status.

Classical Formal Semantic Theories vs CVSMs

- In classical formal semantic theories the functions that drive semantic composition are supplied by the type theory, where the type of each expression specifies the formal character of its denotation in a model.
- The sequence of functions that determines the semantic value of a sentence exhibits at each point a value that directly corresponds to an independently motivated semantic property of the expression to which it is assigned.
- Types of denotation provide non-arbitrary formal relations between types of expressions and classes of entities specified relative to a model.
- The sentential vectors obtained from distributional vectors of lexical items lack this sort of independent status.

Classical Formal Semantic Theories vs CVSMs

- In classical formal semantic theories the functions that drive semantic composition are supplied by the type theory, where the type of each expression specifies the formal character of its denotation in a model.
- The sequence of functions that determines the semantic value of a sentence exhibits at each point a value that directly corresponds to an independently motivated semantic property of the expression to which it is assigned.
- Types of denotation provide non-arbitrary formal relations between types of expressions and classes of entities specified relative to a model.
- The sentential vectors obtained from distributional vectors of lexical items lack this sort of independent status.

Classical Formal Semantic Theories vs CVSMs

- In classical formal semantic theories the functions that drive semantic composition are supplied by the type theory, where the type of each expression specifies the formal character of its denotation in a model.
- The sequence of functions that determines the semantic value of a sentence exhibits at each point a value that directly corresponds to an independently motivated semantic property of the expression to which it is assigned.
- Types of denotation provide non-arbitrary formal relations between types of expressions and classes of entities specified relative to a model.
- The sentential vectors obtained from distributional vectors of lexical items lack this sort of independent status.

Truth/Probability Conditions vs Sentential Vectors

- An important part of the interpretation of a sentence involves knowing its truth (more generally, its satisfaction or fulfillment) conditions.
- From a probabilistic perspective, we can exchange truth conditions for probability (or plausibility) conditions, the likelihood of a sentence being true (accepted by competent speakers of the language as true) given certain conditions.
- It is not obvious how we can extract such conditions, expressed in Boolean or probabilistic terms, from sentential vector values, when these are computed from vectors expressing the distributional (rather than the model theoretic or conditional probability) properties of their constituent lexical items.

Truth/Probability Conditions vs Sentential Vectors

- An important part of the interpretation of a sentence involves knowing its truth (more generally, its satisfaction or fulfillment) conditions.
- From a probabilistic perspective, we can exchange truth conditions for probability (or plausibility) conditions, the likelihood of a sentence being true (accepted by competent speakers of the language as true) given certain conditions.
- It is not obvious how we can extract such conditions, expressed in Boolean or probabilistic terms, from sentential vector values, when these are computed from vectors expressing the distributional (rather than the model theoretic or conditional probability) properties of their constituent lexical items.

Truth/Probability Conditions vs Sentential Vectors

- An important part of the interpretation of a sentence involves knowing its truth (more generally, its satisfaction or fulfillment) conditions.
- From a probabilistic perspective, we can exchange truth conditions for probability (or plausibility) conditions, the likelihood of a sentence being true (accepted by competent speakers of the language as true) given certain conditions.
- It is not obvious how we can extract such conditions, expressed in Boolean or probabilistic terms, from sentential vector values, when these are computed from vectors expressing the distributional (rather than the model theoretic or conditional probability) properties of their constituent lexical items.

Probabilistic Logic

- Classical probabilistic logic (Carnap (1950), Nilsson (1986), Fagin and Halpern (1991), Paris (2010)) models uncertainty in our knowledge of the facts about the world.
- Probability distributions are specified over a set of possible states of the world (possible worlds), and the probabilities for the elements of this set sum to 1.
- A proposition ϕ is assigned truth-values across worlds, and ϕ's probability is computed as $\sum p(w)\ where\ w \in W, \|\phi\|^{w} = t$.

Probabilistic Logic

- Classical probabilistic logic (Carnap (1950), Nilsson (1986), Fagin and Halpern (1991), Paris (2010)) models uncertainty in our knowledge of the facts about the world.
- Probability distributions are specified over a set of possible states of the world (possible worlds), and the probabilities for the elements of this set sum to 1.
- A proposition ϕ is assigned truth-values across worlds, and ϕ's probability is computed as $\sum p(w)$ *where* $w \in W, \|\phi\|^w = t$.

Probabilistic Logic

- Classical probabilistic logic (Carnap (1950), Nilsson (1986), Fagin and Halpern (1991), Paris (2010)) models uncertainty in our knowledge of the facts about the world.
- Probability distributions are specified over a set of possible states of the world (possible worlds), and the probabilities for the elements of this set sum to 1.
- A proposition ϕ is assigned truth-values across worlds, and ϕ's probability is computed as $\sum p(w)$ *where* $w \in W, \|\phi\|^{w} = t$.

A Probabilistic Model of Meaning

- In characterizing meaning probabilistically, we can talk of uncertainty about the truth-value of a sentence, given some probability distribution over possible states of affairs.
- The probability of a sentence expresses the likelihood that (semantically) competent speakers of the language assign to the truth of the sentence, given the state of their knowledge about the world.
- We can then represent the meaning of a sentence as a function that maps intensions to functions from knowledge states to probabilities (probability conditions).
- The semantic value of a sentence S is of type $I \rightarrow K \rightarrow [0,1]$, where I is the set of intensions, K is the set of knowledge representations, and $[0,1]$ is the set of reals p with $0 \leq p \leq 1$.

A Probabilistic Model of Meaning

- In characterizing meaning probabilistically, we can talk of uncertainty about the truth-value of a sentence, given some probability distribution over possible states of affairs.
- The probability of a sentence expresses the likelihood that (semantically) competent speakers of the language assign to the truth of the sentence, given the state of their knowledge about the world.
- We can then represent the meaning of a sentence as a function that maps intensions to functions from knowledge states to probabilities (probability conditions).
- The semantic value of a sentence S is of type $I \rightarrow K \rightarrow [0, 1]$, where I is the set of intensions, K is the set of knowledge representations, and $[0, 1]$ is the set of reals p with $0 \leq p \leq 1$.

A Probabilistic Model of Meaning

- In characterizing meaning probabilistically, we can talk of uncertainty about the truth-value of a sentence, given some probability distribution over possible states of affairs.
- The probability of a sentence expresses the likelihood that (semantically) competent speakers of the language assign to the truth of the sentence, given the state of their knowledge about the world.
- We can then represent the meaning of a sentence as a function that maps intensions to functions from knowledge states to probabilities (probability conditions).
- The semantic value of a sentence S is of type $I \rightarrow K \rightarrow [0, 1]$, where I is the set of intensions, K is the set of knowledge representations, and $[0, 1]$ is the set of reals p with $0 \leq p \leq 1$.

A Probabilistic Model of Meaning

- In characterizing meaning probabilistically, we can talk of uncertainty about the truth-value of a sentence, given some probability distribution over possible states of affairs.
- The probability of a sentence expresses the likelihood that (semantically) competent speakers of the language assign to the truth of the sentence, given the state of their knowledge about the world.
- We can then represent the meaning of a sentence as a function that maps intensions to functions from knowledge states to probabilities (probability conditions).
- The semantic value of a sentence S is of type $I \rightarrow K \rightarrow [0, 1]$, where I is the set of intensions, K is the set of knowledge representations, and $[0, 1]$ is the set of reals p with $0 \leq p \leq 1$.

A Probabilistic Model of Meaning

- In characterizing meaning probabilistically, we can talk of uncertainty about the truth-value of a sentence, given some probability distribution over possible states of affairs.
- The probability of a sentence expresses the likelihood that (semantically) competent speakers of the language assign to the truth of the sentence, given the state of their knowledge about the world.
- We can then represent the meaning of a sentence as a function that maps intensions to functions from knowledge states to probabilities (probability conditions).
- The semantic value of a sentence S is of type $I \rightarrow K \rightarrow [0,1]$, where I is the set of intensions, K is the set of knowledge representations, and $[0,1]$ is the set of reals p with $0 \leq p \leq 1$.

Modelling Uncertainty About Basic Predications

- Let a propositional language over a set of basic predications be given, as follows.

$$\begin{aligned} t &::= x \mid a_1 \mid a_2 \mid \cdots \mid a_m \\ Q &::= Q_1 \mid Q_2 \mid \cdots \mid Q_n \\ \phi &::= Qt \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi. \end{aligned}$$

- Here we assume a single variable x, a finite number of proper names $a_1, a_2, \ldots, a_m$ and a finite number of basic unary predicates $Q_1, Q_2, \ldots, Q_n$.
- Any ϕ that contains occurrences of x is called a *predication*. Use $\phi(x)$ for predications, and $\phi(a/x)$ for the result of replacing x by a everywhere in a predication.

Modelling Uncertainty About Basic Predications

- Let a propositional language over a set of basic predications be given, as follows.

$$\begin{array}{rcl} t & ::= & x \mid a_1 \mid a_2 \mid \cdots \mid a_m \\ Q & ::= & Q_1 \mid Q_2 \mid \cdots \mid Q_n \\ \phi & ::= & Qt \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi. \end{array}$$

- Here we assume a single variable x, a finite number of proper names $a_1, a_2, \ldots, a_m$ and a finite number of basic unary predicates $Q_1, Q_2, \ldots, Q_n$.
- Any ϕ that contains occurrences of x is called a *predication*. Use $\phi(x)$ for predications, and $\phi(a/x)$ for the result of replacing x by a everywhere in a predication.

Modelling Uncertainty About Basic Predications

- Let a propositional language over a set of basic predications be given, as follows.

$$\begin{array}{rcl} t & ::= & x \mid a_1 \mid a_2 \mid \cdots \mid a_m \\ Q & ::= & Q_1 \mid Q_2 \mid \cdots \mid Q_n \\ \phi & ::= & Qt \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi. \end{array}$$

- Here we assume a single variable x, a finite number of proper names $a_1, a_2, \ldots, a_m$ and a finite number of basic unary predicates $Q_1, Q_2, \ldots, Q_n$.
- Any ϕ that contains occurrences of x is called a *predication*. Use $\phi(x)$ for predications, and $\phi(a/x)$ for the result of replacing x by a everywhere in a predication.

Modelling Uncertainty About Basic Predications

- Let a propositional language over a set of basic predications be given, as follows.

$$
\begin{array}{rcl}
t & ::= & x \mid a_1 \mid a_2 \mid \cdots \mid a_m \\
Q & ::= & Q_1 \mid Q_2 \mid \cdots \mid Q_n \\
\phi & ::= & Qt \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi.
\end{array}
$$

- Here we assume a single variable x, a finite number of proper names $a_1, a_2, \ldots, a_m$ and a finite number of basic unary predicates $Q_1, Q_2, \ldots, Q_n$.
- Any ϕ that contains occurrences of x is called a *predication*. Use $\phi(x)$ for predications, and $\phi(a/x)$ for the result of replacing x by a everywhere in a predication.

Extending the Language

- Call this language L_n^m.
- If we extend L_n^m with one name a_{m+1}, the new language is called L_n^{m+1}.
- If we extend L_n^m with one new predicate Q_{n+1}, the new language is called L_{n+1}^m

Extending the Language

- Call this language L_n^m.
- If we extend L_n^m with one name a_{m+1}, the new language is called L_n^{m+1}.
- If we extend L_n^m with one new predicate Q_{n+1}, the new language is called L_{n+1}^m

Extending the Language

- Call this language L_n^m.
- If we extend L_n^m with one name a_{m+1}, the new language is called L_n^{m+1}.
- If we extend L_n^m with one new predicate Q_{n+1}, the new language is called L_{n+1}^m

Extending the Language

- Call this language L_n^m.
- If we extend L_n^m with one name a_{m+1}, the new language is called L_n^{m+1}.
- If we extend L_n^m with one new predicate Q_{n+1}, the new language is called L_{n+1}^m

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \rightarrow \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \rightarrow \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \rightarrow \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \to \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \to \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Representing Partial Knowledge

- For convenience, we identify names and objects, so we assume a domain $D_m = \{a_1, a_2, \ldots, a_m\}$.
- The type of a (restricted) world w is given by $w : \{Q_1, \ldots, Q_n\} \rightarrow \mathcal{P}(D_m)$.
- $w(Q_i)$ is the interpretation of Q_i in w.
- A probabilistic model M is a tuple (D, W, P) with D a domain, W a set of worlds for that domain (predicate interpretations in that domain), and P a probability function over W, i.e., for all $w \in W$, $p(w) \in [0, 1]$, and $\sum_{w \in W} p(w) = 1$.
- The probabilities in a model M represent the priors of an idealized semantic learner.

Interpretation

- An interpretation of L_n^m in an L_n^m-model $M = (D, W, P)$ is given in terms of the standard notion $w \models \phi$, as follows:

$$[\![\phi]\!]^M \quad := \quad \sum\{P(w) \mid w \in W, w \models \phi\}$$

- It is straightforward to verify that this yields $[\![\neg\phi]\!]^M = 1 - [\![\phi]\!]^M$.
- Also, if $\phi \models \neg\psi$, i.e., if $W_\phi \cap W_\psi = \emptyset$, then $[\![\phi \vee \psi]\!]^M = \sum_{w \in W_{\phi\vee\psi}} P(w) = \sum_{w \in W_\phi} P(w) + \sum_{w \in W_\psi} P(w) = [\![\phi]\!]^M + [\![\psi]\!]^M$.

Interpretation

- An interpretation of L_n^m in an L_n^m-model $M = (D, W, P)$ is given in terms of the standard notion $w \models \phi$, as follows:

$$[\![\phi]\!]^M \quad := \quad \sum\{P(w) \mid w \in W, w \models \phi\}$$

- It is straightforward to verify that this yields $[\![\neg\phi]\!]^M = 1 - [\![\phi]\!]^M$.
- Also, if $\phi \models \neg\psi$, i.e., if $W_\phi \cap W_\psi = \emptyset$, then $[\![\phi \vee \psi]\!]^M = \sum_{w \in W_{\phi\vee\psi}} P(w) = \sum_{w \in W_\phi} P(w) + \sum_{w \in W_\psi} P(w) = [\![\phi]\!]^M + [\![\psi]\!]^M$.

Interpretation

- An interpretation of L_n^m in an L_n^m-model $M = (D, W, P)$ is given in terms of the standard notion $w \models \phi$, as follows:

$$[\![\phi]\!]^M \quad := \quad \sum\{P(w) \mid w \in W, w \models \phi\}$$

- It is straightforward to verify that this yields $[\![\neg\phi]\!]^M = 1 - [\![\phi]\!]^M$.
- Also, if $\phi \models \neg\psi$, i.e., if $W_\phi \cap W_\psi = \emptyset$, then $[\![\phi \vee \psi]\!]^M = \sum_{w \in W_{\phi\vee\psi}} P(w) = \sum_{w \in W_\phi} P(w) + \sum_{w \in W_\psi} P(w) = [\![\phi]\!]^M + [\![\psi]\!]^M$.

Interpretation

- An interpretation of L_n^m in an L_n^m-model $M = (D, W, P)$ is given in terms of the standard notion $w \models \phi$, as follows:

$$[\![\phi]\!]^M \quad := \quad \sum\{P(w) \mid w \in W, w \models \phi\}$$

- It is straightforward to verify that this yields $[\![\neg\phi]\!]^M = 1 - [\![\phi]\!]^M$.
- Also, if $\phi \models \neg\psi$, i.e., if $W_\phi \cap W_\psi = \emptyset$, then $[\![\phi \vee \psi]\!]^M = \sum_{w \in W_{\phi \vee \psi}} P(w) = \sum_{w \in W_\phi} P(w) + \sum_{w \in W_\psi} P(w) = [\![\phi]\!]^M + [\![\psi]\!]^M$.

Example 1

- Assume there are just two predicates Q_1 and Q_2, and two objects a, b.
- Complete ignorance about how the predicates are applied is represented by a model with 16 worlds, because for each object x and each predicate Q there are two cases: Q applies to x or not.
- If the likelihood of each of the cases is completely unknown, each of these worlds has probability $\frac{1}{16}$.

Example 1

- Assume there are just two predicates Q_1 and Q_2, and two objects a, b.
- Complete ignorance about how the predicates are applied is represented by a model with 16 worlds, because for each object x and each predicate Q there are two cases: Q applies to x or not.
- If the likelihood of each of the cases is completely unknown, each of these worlds has probability $\frac{1}{16}$.

Example 1

- Assume there are just two predicates Q_1 and Q_2, and two objects a, b.
- Complete ignorance about how the predicates are applied is represented by a model with 16 worlds, because for each object x and each predicate Q there are two cases: Q applies to x or not.
- If the likelihood of each of the cases is completely unknown, each of these worlds has probability $\frac{1}{16}$.

Example 1

- Assume there are just two predicates Q_1 and Q_2, and two objects a, b.
- Complete ignorance about how the predicates are applied is represented by a model with 16 worlds, because for each object x and each predicate Q there are two cases: Q applies to x or not.
- If the likelihood of each of the cases is completely unknown, each of these worlds has probability $\frac{1}{16}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Example 2

- Suppose again there are two objects a, b and two predicates Q_1, Q_2.
- Suppose it is known that a has Q_1, and the probability that b has Q_1 is taken to be $\frac{2}{3}$.
- Suppose it is known that no object has Q_2.
- Then $W = \{w_1, w_2\}$ with $w_1(Q_1) = \{a, b\}$, $w_2(Q_1) = \{a\}$, $w_1(Q_2) = \emptyset$, $w_2(Q_2) = \emptyset$,
- P is given by $P(w_1) = \frac{2}{3}$, $P(w_2) = \frac{1}{3}$.
- In this example $\neg Q_1(b)$ is true in w_2 and not in w_1.
- Therefore $[\![\neg Q_1(b)]\!] = \frac{1}{3}$.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Semantic Concepts: Dictionary Learning

- Learning a new semantic concept Q_{n+1} is learning how (or: to what extent) predicate Q_{n+1} applies to the objects one knows about.
- The simplest way to model such a learning event is as a pair $(Q_{n+1}, \phi(x))$ where $\phi(x)$ is an L_n^m predication.
- The effect of the learning event could then be modelled in a way that is very similar to the manner in which factual change is modelled in epistemic update logic.
- The result of updating a model $M = (D, W, P)$ with concept learning event $(Q_{n+1}, \phi(x))$ is the model that is like M except for the fact that the interpretation in each world of Q_{n+1} is given by

$$w(Q_{n+1}) := \{a \mid a \in D_m, w \models \phi(a/x)\}.$$

- Note that the probability function P of the model does not change.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1x \wedge \neg Q_2x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1 x \wedge \neg Q_2 x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1 x \wedge \neg Q_2 x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1x \wedge \neg Q_2x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1 x \wedge \neg Q_2 x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1 x \wedge \neg Q_2 x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Learning Example

- Let's return to example 1.
- This is the model where there are two objects and two predicates, and nothing is known about the properties of the objects.
- Take the learning event $(Q_3, Q_1x \wedge \neg Q_2x)$.
- This defines Q_3 as the difference of Q_1 and Q_2.
- The resulting model will again have 16 worlds, and in each world w_i, $w_i(Q_3)$ is given by $w_i(Q_1) \cap (D - w_i(Q_2))$.
- The probabilities of the worlds remain unchanged.

Adjustment and Approximation (1)

- To allow adjustment of the meaning of a concept by means of a learning event, we can use probabilistic updating (this follows (Van Benthem, Gerbrandy, Kooi (2009))).
- A concept learning event now is a tuple

$$(Q, \phi, \psi(x), q)$$

where ϕ is a sentence, $\psi(x)$ is a predication, and q is a probability.

- ϕ expresses the observational circumstances of the revision.
- q expresses the observational certainty of the new information.

Adjustment and Approximation (1)

- To allow adjustment of the meaning of a concept by means of a learning event, we can use probabilistic updating (this follows (Van Benthem, Gerbrandy, Kooi (2009))).
- A concept learning event now is a tuple

$$(Q, \phi, \psi(x), q)$$

where ϕ is a sentence, $\psi(x)$ is a predication, and q is a probability.
- ϕ expresses the observational circumstances of the revision.
- q expresses the observational certainty of the new information.

Adjustment and Approximation (1)

- To allow adjustment of the meaning of a concept by means of a learning event, we can use probabilistic updating (this follows (Van Benthem, Gerbrandy, Kooi (2009))).
- A concept learning event now is a tuple

$$(Q, \phi, \psi(x), q)$$

where ϕ is a sentence, $\psi(x)$ is a predication, and q is a probability.

- ϕ expresses the observational circumstances of the revision.
- q expresses the observational certainty of the new information.

Adjustment and Approximation (1)

- To allow adjustment of the meaning of a concept by means of a learning event, we can use probabilistic updating (this follows (Van Benthem, Gerbrandy, Kooi (2009))).
- A concept learning event now is a tuple

$$(Q, \phi, \psi(x), q)$$

  where ϕ is a sentence, $\psi(x)$ is a predication, and q is a probability.
- ϕ expresses the observational circumstances of the revision.
- q expresses the observational certainty of the new information.

Adjustment and Approximation (1)

- To allow adjustment of the meaning of a concept by means of a learning event, we can use probabilistic updating (this follows (Van Benthem, Gerbrandy, Kooi (2009))).
- A concept learning event now is a tuple

  $$(Q, \phi, \psi(x), q)$$

  where ϕ is a sentence, $\psi(x)$ is a predication, and q is a probability.
- ϕ expresses the observational circumstances of the revision.
- q expresses the observational certainty of the new information.

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Adjustment and Approximation (2)

- The result of updating $M = (D, W, P)$ with $(Q, \phi, \psi(x), q)$ is a new model $M = (D, W', P')$.
- W' is given by changing the interpretation of Q in members w of W_ϕ to $\{a \mid w \models \psi(a/x)\}$ while leaving the interpretation of Q in members of $W_{\neg\phi}$ unchanged.
- P' is given by $P'(w) = \frac{P(w) \times q}{X}$ for members of W_ϕ,
- and by $P'(w) = \frac{P(w) \times (1-q)}{X}$ for members of $W_{\neg\phi}$.
- $\frac{1}{X}$ (the normalization factor) is given by

$$X = \sum_{w \in W_\phi} P(w) \times q + \sum_{w \in W_{\neg\phi}} P(w) \times (1 - q).$$

Back to an Example

- Consider again the example with the two objects and the two properties, where nothing is known. A learning event for this could be:

$$(Q_2, \neg Q_1 b, Q_1 x \vee Q_2 x, \frac{2}{3}).$$

- Then the resulting model has again 2 worlds, but now the probability of w_2 has gone up from $\frac{1}{3}$ to

$$\frac{\frac{2}{3} \times \frac{1}{3}}{\frac{1}{3} \times \frac{2}{3} + \frac{2}{3} \times \frac{1}{3}} = \frac{1}{2}$$

- The probability of w_1 has gone down from $\frac{2}{3}$ to

$$\frac{\frac{1}{3} \times \frac{2}{3}}{\frac{4}{9}} = \frac{1}{2}.$$

Back to an Example

- Consider again the example with the two objects and the two properties, where nothing is known. A learning event for this could be:

$$(Q_2, \neg Q_1 b, Q_1 x \vee Q_2 x, \frac{2}{3}).$$

- Then the resulting model has again 2 worlds, but now the probability of w_2 has gone up from $\frac{1}{3}$ to

$$\frac{\frac{2}{3} \times \frac{1}{3}}{\frac{1}{3} \times \frac{2}{3} + \frac{2}{3} \times \frac{1}{3}} = \frac{1}{2}$$

- The probability of w_1 has gone down from $\frac{2}{3}$ to

$$\frac{\frac{1}{3} \times \frac{2}{3}}{\frac{4}{9}} = \frac{1}{2}.$$

Back to an Example

- Consider again the example with the two objects and the two properties, where nothing is known. A learning event for this could be:

$$(Q_2, \neg Q_1 b, Q_1 x \vee Q_2 x, \frac{2}{3}).$$

- Then the resulting model has again 2 worlds, but now the probability of w_2 has gone up from $\frac{1}{3}$ to

$$\frac{\frac{2}{3} \times \frac{1}{3}}{\frac{1}{3} \times \frac{2}{3} + \frac{2}{3} \times \frac{1}{3}} = \frac{1}{2}$$

- The probability of w_1 has gone down from $\frac{2}{3}$ to

$$\frac{\frac{1}{3} \times \frac{2}{3}}{\frac{4}{9}} = \frac{1}{2}.$$

Back to an Example

- Consider again the example with the two objects and the two properties, where nothing is known. A learning event for this could be:

$$(Q_2, \neg Q_1 b, Q_1 x \vee Q_2 x, \frac{2}{3}).$$

- Then the resulting model has again 2 worlds, but now the probability of w_2 has gone up from $\frac{1}{3}$ to

$$\frac{\frac{2}{3} \times \frac{1}{3}}{\frac{1}{3} \times \frac{2}{3} + \frac{2}{3} \times \frac{1}{3}} = \frac{1}{2}$$

- The probability of w_1 has gone down from $\frac{2}{3}$ to

$$\frac{\frac{1}{3} \times \frac{2}{3}}{\frac{4}{9}} = \frac{1}{2}.$$

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a “rose”, and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a “rose”, and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a “rose”, and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a "rose", and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a "rose", and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a “rose”, and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (1)

- You are given something of which you are told that it is called a “rose”, and you observe that it is thorny, red and a flower.
- A learning example is an encounter with a new object a_{m+1}.
- Suppose you learn that predicate Q applies to a_{m+1}.
- The properties you observe of a_{m+1} are given by $\theta(a_{m+1})$, where $\theta(a_{m+1})$ is a conjunction of $\pm Q_i(a_{m+1})$ for all known predicates.
- Update event: $(a_{m+1}, Q, \theta(a_{m+1}))$.
- You learn that a_{m+1} is called a Q, and you observe that a_{m+1} satisfies the properties $\theta(a_{m+1})$.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (2)

- Updating a model $M = (D, W, P)$ for L_n^m with this will create a new model $M' = (D \cup \{a_{m+1}\}, W', P)$ for L_n^{m+1}.
- New model has domain $\{a_1, \ldots, a_{m+1}\}$.
- W' is given by assigning, in each w, to a_{m+1} the properties specified by $\theta(a_{m+1})$.
- Interpretation of Q is given by setting

$$w(Q) = \{a \mid w \models \theta(a/a_{m+1})\}.$$

- This resets the interpretation Q on the basis of the new observation.
- Probability distribution remains unchanged.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

Learning Concepts by Example (3)

- Account can be refined for cases where the observation is less precise.
- Learning event:

$$(a_{m+1}, Q, \{(\theta_1(a_{m+1}), q_1), \ldots, (\theta_k(a_{m+1}), q_k)\})$$

- Here q_i gives the observational probability that the new object satisfies θ_i.
- The probabilities should observe $\sum_{i=1}^{k} q_i = 1$.
- The update can be defined in such way that the probability of the new predicate applying to the old objects will get recomputed.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \to s \to t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \to i) \to (e \to i) \to i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \to t) \to (e \to t) \to t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \to s \to t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \to i) \to (e \to i) \to i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) = \lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \to t) \to (e \to t) \to t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$ $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x\ \text{dom}\ w)(\lambda y.q\ y\ \text{dom}\ w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (1)

- Basic types are e (entities), s (worlds), t (truth values), d (domains) and $[0, 1]$ (the space of probabilities).
- Abbreviate $d \rightarrow s \rightarrow t$ as i (intensions).
- Types for S, N, VP, NP, DET get lifted to the level of intensions, by substituting i for t in all types.
- This gives, e.g., $\tau(\text{DET}) = (e \rightarrow i) \rightarrow (e \rightarrow i) \rightarrow i$.
- The lifting rules for the interpretation functions are completely straightforward.
- $I(\text{Some}) =$
  $\lambda p \lambda q \lambda \text{dom} \lambda w.\text{some}(\lambda x.p\ x \text{ dom } w)(\lambda y.q\ y \text{ dom } w)$.
- Here some is the familiar constant function for existential quantification, of type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$.

A Toy Fragment (2)

- This will give sentences interpretation of type i, i.e., $d \to s \to t$.
- Such intensions can be mapped to probabilities by means of a function prob of type $i \to m \to [0.1]$, where m is the type of models with their domains, i.e., objects of the shape (D, W, P).
- The function prob is given by:

$$\text{prob}\ f\ (D, W, P) = \sum\{P(w) \mid w \in W, f\ D\ w\}.$$

- This assigns to every sentence of the fragment a probability, on the basis of the prior probabilities encoded by (D, W, P).

A Toy Fragment (2)

- This will give sentences interpretation of type i, i.e., $d \to s \to t$.
- Such intensions can be mapped to probabilities by means of a function prob of type $i \to m \to [0.1]$, where m is the type of models with their domains, i.e., objects of the shape (D, W, P).
- The function prob is given by:

$$\text{prob}\ f\ (D, W, P) = \sum \{P(w) \mid w \in W, f\ D\ w\}.$$

- This assigns to every sentence of the fragment a probability, on the basis of the prior probabilities encoded by (D, W, P).

A Toy Fragment (2)

- This will give sentences interpretation of type i, i.e., $d \rightarrow s \rightarrow t$.
- Such intensions can be mapped to probabilities by means of a function prob of type $i \rightarrow m \rightarrow [0.1]$, where m is the type of models with their domains, i.e., objects of the shape (D, W, P).
- The function prob is given by:

$$\text{prob } f\ (D, W, P) = \sum \{P(w) \mid w \in W, f\ D\ w\}.$$

- This assigns to every sentence of the fragment a probability, on the basis of the prior probabilities encoded by (D, W, P).

A Toy Fragment (2)

- This will give sentences interpretation of type i, i.e., $d \rightarrow s \rightarrow t$.
- Such intensions can be mapped to probabilities by means of a function prob of type $i \rightarrow m \rightarrow [0.1]$, where m is the type of models with their domains, i.e., objects of the shape (D, W, P).
- The function prob is given by:

$$\text{prob } f\ (D, W, P) = \sum \{P(w) \mid w \in W, f\ D\ w\}.$$

- This assigns to every sentence of the fragment a probability, on the basis of the prior probabilities encoded by (D, W, P).

A Toy Fragment (2)

- This will give sentences interpretation of type i, i.e., $d \rightarrow s \rightarrow t$.
- Such intensions can be mapped to probabilities by means of a function prob of type $i \rightarrow m \rightarrow [0.1]$, where m is the type of models with their domains, i.e., objects of the shape (D, W, P).
- The function prob is given by:

$$\text{prob } f\ (D, W, P) = \sum \{P(w) \mid w \in W, f\ D\ w\}.$$

- This assigns to every sentence of the fragment a probability, on the basis of the prior probabilities encoded by (D, W, P).

Conclusions

- CVSMs can represent gradience in semantic relations among words, phrases, and sentences, and they offer a viable account of lexical semantic learning.
- However, the vectors that CVSMs assign to complex syntactic structures do not have clear interpretations, and they do not express sentential meaning as probability conditions.
- We propose a fragment of a probabilistic semantic theory that uses classical type theory to compute the probability value of a sentence on the basis of a model for the knowledge of an idealized semantic learner.
- Our approach offers a framework for developing a probabilistic account of semantic learning.

Conclusions

- CVSMs can represent gradience in semantic relations among words, phrases, and sentences, and they offer a viable account of lexical semantic learning.
- However, the vectors that CVSMs assign to complex syntactic structures do not have clear interpretations, and they do not express sentential meaning as probability conditions.
- We propose a fragment of a probabilistic semantic theory that uses classical type theory to compute the probability value of a sentence on the basis of a model for the knowledge of an idealized semantic learner.
- Our approach offers a framework for developing a probabilistic account of semantic learning.

Conclusions

- CVSMs can represent gradience in semantic relations among words, phrases, and sentences, and they offer a viable account of lexical semantic learning.
- However, the vectors that CVSMs assign to complex syntactic structures do not have clear interpretations, and they do not express sentential meaning as probability conditions.
- We propose a fragment of a probabilistic semantic theory that uses classical type theory to compute the probability value of a sentence on the basis of a model for the knowledge of an idealized semantic learner.
- Our approach offers a framework for developing a probabilistic account of semantic learning.

Conclusions

- CVSMs can represent gradience in semantic relations among words, phrases, and sentences, and they offer a viable account of lexical semantic learning.
- However, the vectors that CVSMs assign to complex syntactic structures do not have clear interpretations, and they do not express sentential meaning as probability conditions.
- We propose a fragment of a probabilistic semantic theory that uses classical type theory to compute the probability value of a sentence on the basis of a model for the knowledge of an idealized semantic learner.
- Our approach offers a framework for developing a probabilistic account of semantic learning.