Information in Spoken Language

A quantitative approach

Rob van Son

Chair of Phonetic Sciences
ACLC
University of Amsterdam

LOT winterschool 2006

Amsterdam Center for Language and Communication

ACLC

1 Symbolic Information in language

- Introduction
- Information in the Lexicon
- Markov models
- Hidden Markov Models
- Predictability in context
- Human word recognition
- Phonemic information
- Bibliography

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Introduction

Language is understood in symbolic form

- Speech is transcribed into phonemes and words
- Somehow, phonemes seem to matter for understanding
- What information is carried by a single phoneme?
- Answer depends on the information in words
- The predictability of words
- Word recognition by humans
- What does a single phoneme contribute?

Information in the Lexicon

Content and function words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Content and *function* words

- A basic distinction
- Content words have an independent, lexical, meaning
- Function words have little, if any, lexical meaning, but chiefly indicate a grammatical relationship
- Function words form a closed class (around 1000 words)
- Content words belong to an open class
- Function words have a high frequency/low information content
- But how to treat the information content of words itself?

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) $= 0.173$ bits/form (average information difference)

http://en.wikipedia.org/wiki/Zipf_distribution

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) = 0.173 bits/form (average information difference)

`http://en.wikipedia.org/wiki/Zipf_distribution`

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) $= 0.173$ bits/form (average information difference)

`http://en.wikipedia.org/wiki/Zipf_distribution`

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) $= 0.173$ bits/form (average information difference)

http://en.wikipedia.org/wiki/Zipf_distribution

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) = 0.173 bits/form (average information difference)

http://en.wikipedia.org/wiki/Zipf_distribution

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Word frequencies follow a Zipf distribution

- 8 Jane Austens novels (1811-1817)
- 14,817 word forms on 801,183 words (tokens)
- Most frequent 10: *the to and of a her I in was it* (180,464 tokens)
- 4,571 words occur only once, 1,992 twice
- $H = 9.306$ bits/form
- Kullback-Leibler(Words, Zipfs dist) $= 0.173$ bits/form (average information difference)

`http://en.wikipedia.org/wiki/Zipf_distribution`

[Dover(2004)][Kawamura and Hatano(2002)][Project Gutenberg(2005)]

Information in the Lexicon

Author	N	Forms	H	KL(Zipf)
Henry Fielding	957,777	20,689	9.516	0.141
Jane Austen	801,183	14,817	9.306	0.173
Brontë sisters	892,779	24,658	9.828	0.108
Plato (transl.)	1,445,372	21,095	9.173	0.129

Works of several authors 1750-1900?

- Large variation in number of distinct word forms (vocabulary)
- Information per word (H) roughly constant ($2^H \approx 890$)
- Difference with Zipf's distribution $\leq 2\%$

Information in the Lexicon

Author	N	Forms	H	KL(Zipf)
Henry Fielding	957,777	20,689	9.516	0.141
Jane Austen	801,183	14,817	9.306	0.173
Brontë sisters	892,779	24,658	9.828	0.108
Plato (transl.)	1,445,372	21,095	9.173	0.129

Works of several authors 1750-1900?

- Large variation in number of distinct word forms (vocabulary)
- Information per word (H) roughly constant ($2^H \approx 890$)
- Difference with Zipf's distribution $\leq 2\%$

Information in the Lexicon

Author	N	Forms	H	KL(Zipf)
Henry Fielding	957,777	20,689	9.516	0.141
Jane Austen	801,183	14,817	9.306	0.173
Brontë sisters	892,779	24,658	9.828	0.108
Plato (transl.)	1,445,372	21,095	9.173	0.129

Works of several authors 1750-1900?

- Large variation in number of distinct word forms (vocabulary)
- Information per word (H) roughly constant ($2^H \approx 890$)
- Difference with Zipf's distribution $\leq 2\%$

Information in the Lexicon

	Fielding	Austen	Brontë	Plato
Henry Fielding	0.141	0.311	0.377	0.627
Jane Austen	0.323	0.173	0.326	0.767
Brontë sisters	0.323	0.270	0.108	0.712
Plato (transl.)	0.581	0.588	0.651	0.129

Kullback-Leibler divergence or Cross Entropy

- Diagonal terms are distance to Zipf distribution
- $KL(q,p) = H(p) - H_X(q,p) \approx$ information difference per word
- Averaged over one of the distributions
- Multi-authored translations of Plato differ from novels
- All works differ more from each other than from Zipf's distr.

Information in the Lexicon

	Fielding	Austen	Brontë	Plato
Henry Fielding	*0.141*	0.311	0.377	0.627
Jane Austen	0.323	*0.173*	0.326	0.767
Brontë sisters	0.323	0.270	*0.108*	0.712
Plato (transl.)	0.581	0.588	0.651	*0.129*

Kullback-Leibler divergence or Cross Entropy

- Diagonal terms are distance to Zipf distribution
- $KL(q,p) = H(p) - H_X(q,p) \approx$ information difference per word
- Averaged over one of the distributions
- Multi-authored translations of Plato differ from novels
- All works differ more from each other than from Zipf's distr.

Information in the Lexicon

	Fielding	Austen	Brontë	Plato
Henry Fielding	*0.141*	0.311	0.377	0.627
Jane Austen	0.323	*0.173*	0.326	0.767
Brontë sisters	0.323	0.270	*0.108*	0.712
Plato (transl.)	0.581	0.588	0.651	*0.129*

Kullback-Leibler divergence or Cross Entropy

- Diagonal terms are distance to Zipf distribution
- $KL(q,p) = H(p) - H_X(q,p) \approx$ information difference per word
- Averaged over one of the distributions
- Multi-authored translations of Plato differ from novels
- All works differ more from each other than from Zipf's distr.

Information in the Lexicon

	Fielding	Austen	Brontë	Plato
Henry Fielding	*0.141*	0.311	0.377	0.627
Jane Austen	0.323	*0.173*	0.326	0.767
Brontë sisters	0.323	0.270	*0.108*	0.712
Plato (transl.)	0.581	0.588	0.651	*0.129*

Kullback-Leibler divergence or Cross Entropy

- Diagonal terms are distance to Zipf distribution
- $KL(q,p) = H(p) - H_X(q,p) \approx$ information difference per word
- Averaged over one of the distributions
- Multi-authored translations of Plato differ from novels
- All works differ more from each other than from Zipf's distr.

Information in the Lexicon

	Fielding	Austen	Brontë	Plato
Henry Fielding	*0.141*	0.311	0.377	0.627
Jane Austen	0.323	*0.173*	0.326	0.767
Brontë sisters	0.323	0.270	*0.108*	0.712
Plato (transl.)	0.581	0.588	0.651	*0.129*

Kullback-Leibler divergence or Cross Entropy

- Diagonal terms are distance to Zipf distribution
- $KL(q,p) = H(p) - H_X(q,p) \approx$ information difference per word
- Averaged over one of the distributions
- Multi-authored translations of Plato differ from novels
- All works differ more from each other than from Zipf's distr.

Information in the Lexicon

	Austen	Brontë	Plato
Henry Fielding	0.489	0.460	0.447
Jane Austen		0.526	0.536
Brontë sisters			0.470

Normalized Compression Distance (NCD)

- $NCD(x, y) = \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))}$
- $C(x)$ is number of distinct word forms in text
- Half the words in the larger corpus do not appear in smaller corpus
- Jane Austen's works had the smallest "vocabulary" $\Rightarrow$ largest distances

Information in the Lexicon

	Austen	Brontë	Plato
Henry Fielding	0.489	0.460	0.447
Jane Austen		0.526	0.536
Brontë sisters			0.470

Normalized Compression Distance (NCD)

- $NCD(x, y) = \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))}$
- $C(x)$ is number of distinct word forms in text
- Half the words in the larger corpus do not appear in smaller corpus
- Jane Austen's works had the smallest "vocabulary" $\Rightarrow$ largest distances

Information in the Lexicon

	Austen	Brontë	Plato
Henry Fielding	0.489	0.460	0.447
Jane Austen		0.526	0.536
Brontë sisters			0.470

Normalized Compression Distance (NCD)

- $NCD(x, y) = \frac{C(xy) - min(C(x), C(y))}{max(C(x), C(y))}$
- $C(x)$ is number of distinct word forms in text
- Half the words in the larger corpus do not appear in smaller corpus
- Jane Austen's works had the smallest "vocabulary" $\Rightarrow$ largest distances

Information in the Lexicon

	Austen	Brontë	Plato
Henry Fielding	0.489	0.460	0.447
Jane Austen		0.526	0.536
Brontë sisters			0.470

Normalized Compression Distance (NCD)

- $NCD(x,y) = \frac{C(xy) - min(C(x),C(y))}{max(C(x),C(y))}$
- $C(x)$ is number of distinct word forms in text
- Half the words in the larger corpus do not appear in smaller corpus
- Jane Austen's works had the smallest "vocabulary" $\Rightarrow$ largest distances

Markov models: Predictability

$$P(w_{i+1}|w_i) = \frac{P(w_{i+1}, w_i)}{P(w_i)} \tag{1}$$

$$P(w_1, \ldots, w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, \ldots, w_{i-1}) \tag{2}$$

Extremely useful technology

- Probability of the next word, given the previous words
- Probability of a sentence (N-gram)
- Most likely sentence (eg, using Viterbi algorithm)
- Humans do not use anything like it

[Jurafsky and Martin(2000)]

Markov models: Predictability

$$P(w_{i+1}|w_i) = \frac{P(w_{i+1}, w_i)}{P(w_i)} \quad (1)$$

$$P(w_1, \ldots, w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, \ldots, w_{i-1}) \quad (2)$$

Extremely useful technology

- Probability of the next word, given the previous words
- Probability of a sentence (N-gram)
- Most likely sentence (eg, using Viterbi algorithm)
- Humans do not use anything like it

[Jurafsky and Martin(2000)]

Markov models: Predictability

$$P(w_{i+1}|w_i) = \frac{P(w_{i+1}, w_i)}{P(w_i)} \qquad (1)$$

$$P(w_1, \ldots, w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, \ldots, w_{i-1}) \qquad (2)$$

Extremely useful technology

- Probability of the next word, given the previous words
- Probability of a sentence (N-gram)
- Most likely sentence (eg, using Viterbi algorithm)
- Humans do not use anything like it

[Jurafsky and Martin(2000)]

Markov models: Predictability

$$P(w_{i+1}|w_i) = \frac{P(w_{i+1}, w_i)}{P(w_i)} \tag{1}$$

$$P(w_1, \ldots, w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, \ldots, w_{i-1}) \tag{2}$$

Extremely useful technology

- Probability of the next word, given the previous words
- Probability of a sentence (N-gram)
- Most likely sentence (eg, using Viterbi algorithm)
- Humans do not use anything like it

[Jurafsky and Martin(2000)]

Hidden Markov Models

$$P(S_i|O_i, S_{i-1}) = P(O_i|S_i) \cdot P(S_i|S_{i-1})$$

HMM technology is extremely important for language (re-)search

- Split the problem in an Observation part and a Model part
- The "states", S_i, are not visible
- However, they can be estimated using the observation probabilities $P(O_i|S_i)$ and the transition probabilities $P(S_i|S_{i-1})$
- Standard in Automatic Speech Recognition (ASR)

[Jurafsky and Martin(2000)]

Hidden Markov Models

$$P(S_i|O_i, S_{i-1}) = P(O_i|S_i) \cdot P(S_i|S_{i-1})$$

HMM technology is extremely important for language (re-)search

- Split the problem in an Observation part and a Model part
- The "states", S_i, are not visible
- However, they can be estimated using the observation probabilities $P(O_i|S_i)$ and the transition probabilities $P(S_i|S_{i-1})$
- Standard in Automatic Speech Recognition (ASR)

[Jurafsky and Martin(2000)]

Hidden Markov Models

$$P(S_i|O_i, S_{i-1}) = P(O_i|S_i) \cdot P(S_i|S_{i-1})$$

HMM technology is extremely important for language (re-)search

- Split the problem in an Observation part and a Model part
- The "states", S_i, are not visible
- However, they can be estimated using the observation probabilities $P(O_i|S_i)$ and the transition probabilities $P(S_i|S_{i-1})$
- Standard in Automatic Speech Recognition (ASR)

[Jurafsky and Martin(2000)]

Hidden Markov Models

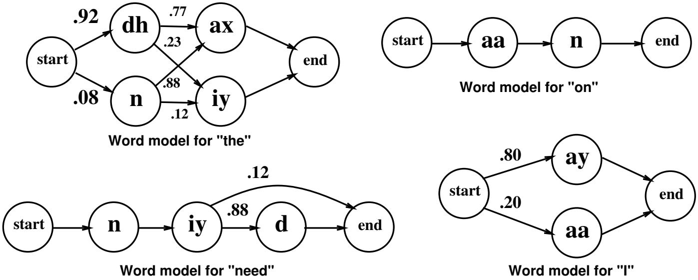
$$P(S_i|O_i, S_{i-1}) = P(O_i|S_i) \cdot P(S_i|S_{i-1})$$

HMM technology is extremely important for language (re-)search

- Split the problem in an Observation part and a Model part
- The "states", S_i, are not visible
- However, they can be estimated using the observation probabilities $P(O_i|S_i)$ and the transition probabilities $P(S_i|S_{i-1})$
- Standard in Automatic Speech Recognition (ASR)

[Jurafsky and Martin(2000)]

Hidden Markov Models: Pronunciation networks

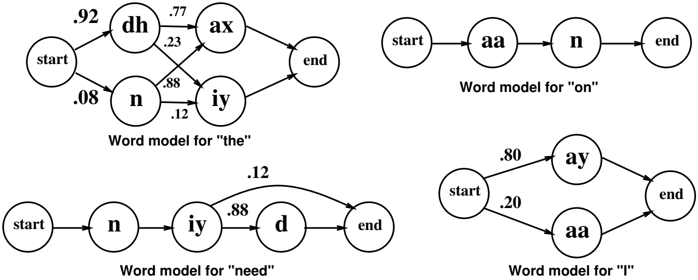


Construct phone state models for each word in the dictionary

- Possible pronunciations for each word have to be encoded in the dictionary
- Transition probabilities are "trained" from the frequency of occurrence of the pronunciation in the corpus

[Jurafsky and Martin(2000)]

Hidden Markov Models: Pronunciation networks

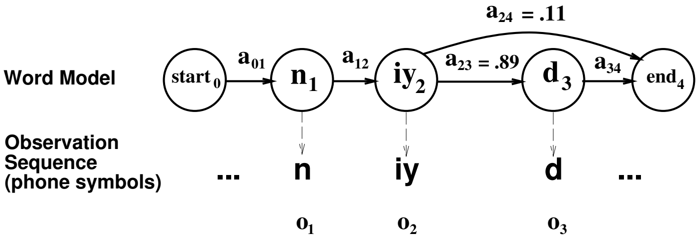


Construct phone state models for each word in the dictionary

- Possible pronunciations for each word have to be encoded in the dictionary
- Transition probabilities are "trained" from the frequency of occurrence of the pronunciation in the corpus

[Jurafsky and Martin(2000)]

Hidden Markov Models: Hidden states

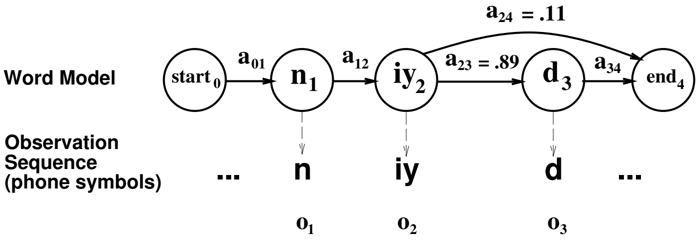


Word models: simple phone state model for *need*

- Each transition has a probability
- start and end are special states
- Each state *or* each transition has associated sound observations with a distinct probability density function (PDF)

[Jurafsky and Martin(2000)]

Hidden Markov Models: Hidden states

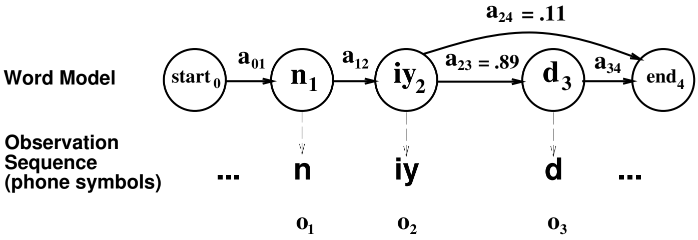


Word models: simple phone state model for *need*

- Each transition has a probability
- start and end are special states
- Each state *or* each transition has associated sound observations with a distinct probability density function (PDF)

[Jurafsky and Martin(2000)]

Hidden Markov Models: Hidden states

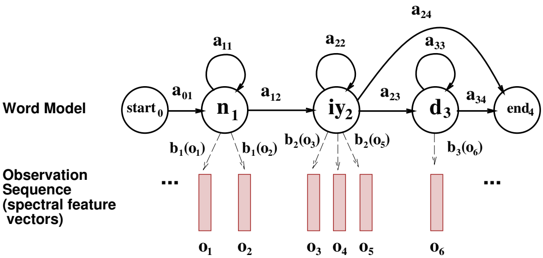


Word models: simple phone state model for *need*

- Each transition has a probability
- start and end are special states
- Each state *or* each transition has associated sound observations with a distinct probability density function (PDF)

[Jurafsky and Martin(2000)]

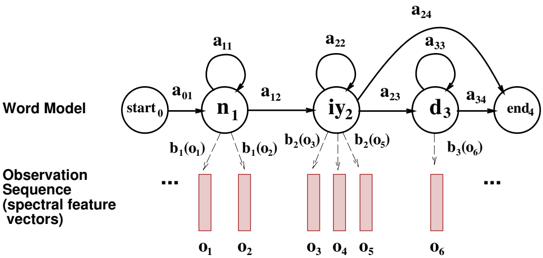
Hidden Markov Models: Observation probabilities



Observed are sound "spectra" for time "frames"

- Observation sequences have a probability
- Calculate this probability for each possible word
- Probabilities of observation O_i calculated from all possible underlying states
- Chose word *sequence* with the highest overall probability

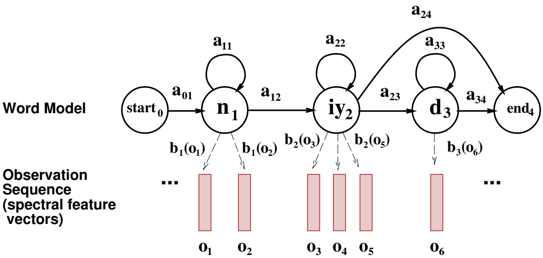
Hidden Markov Models: Observation probabilities



Observed are sound "spectra" for time "frames"

- Observation sequences have a probability
- Calculate this probability for each possible word
- Probabilities of observation O_i calculated from all possible underlying states
- Chose word *sequence* with the highest overall probability

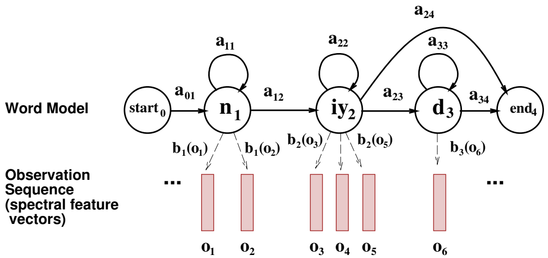
Hidden Markov Models: Observation probabilities



Observed are sound "spectra" for time "frames"

- Observation sequences have a probability
- Calculate this probability for each possible word
- Probabilities of observation O_i calculated from all possible underlying states
- Chose word *sequence* with the highest overall probability

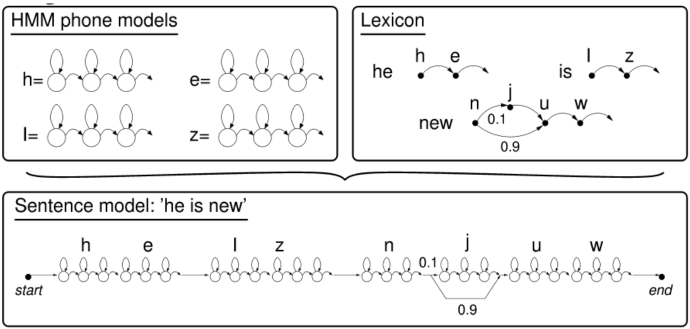
Hidden Markov Models: Observation probabilities



Observed are sound "spectra" for time "frames"

- Observation sequences have a probability
- Calculate this probability for each possible word
- Probabilities of observation O_i calculated from all possible underlying states
- Chose word *sequence* with the highest overall probability

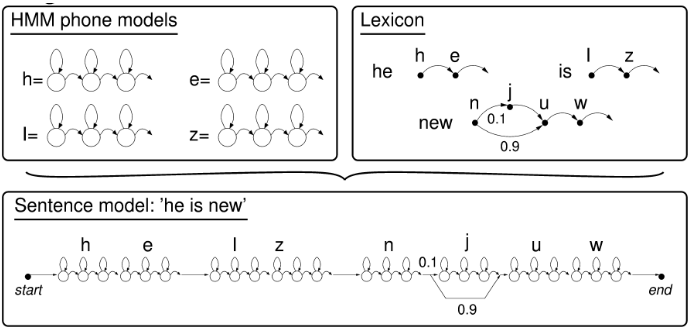
Hidden Markov Models: Phone networks



Phone models are concatenated into utterance networks

- Each word model is itself a Markov finite state network of phone models
- Phones and word are connected through the *start* and *end* states (not shown)

Hidden Markov Models: Phone networks



Phone models are concatenated into utterance networks

- Each word model is itself a Markov finite state network of phone models
- Phones and word are connected through the *start* and *end* states (not shown)

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$

  [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$

  [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$ [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$ [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$

  [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Hidden Markov Models

Hidden Markov Models are very useful for:

- Automatic Speech Recognition
- Automatic POS tagging: $P(POS_i|Word_i, POS_{i-1}) = P(Word_i|POS_i) \cdot P(POS_i|POS_{i-1})$
- Accent assignment [Pan and McKeown(1999)]
- Automatic translation
- Speech Understanding with Semantic HMMs
  $P(Message_i|Words) = P(Words|Message_i) \cdot P(Message_i|Message_{i-1})$

  [Bühler et al.(2005)Bühler, Minker, and Elciyanti]
- Still, humans do not use (Hidden) Markov Models

Predictability in context

Movies. . . pop-corn
Cats and. . . dogs
A stitch in time saves. . . nine
Good morning,. . . how are you

No real model of human grammar but there are regularities in human (re-)cognition

- Priming: people expect words given associates
- Semantic web: words in proximity tend to be cognates
- Words group in "documents"
- Words tend to cluster inside "texts"

Predictability in context

Movies. . . pop-corn
Cats and. . . dogs
A stitch in time saves. . . nine
Good morning,. . . how are you

No real model of human grammar but there are regularities in human (re-)cognition

- Priming: people expect words given associates
- Semantic web: words in proximity tend to be cognates
- Words group in "documents"
- Words tend to cluster inside "texts"

Predictability in context

Movies. . . pop-corn
Cats and. . . dogs
A stitch in time saves. . . nine
Good morning,. . . how are you

No real model of human grammar but there are regularities in human (re-)cognition

- Priming: people expect words given associates
- Semantic web: words in proximity tend to be cognates
- Words group in "documents"
- Words tend to cluster inside "texts"

Predictability in context

Movies. . . pop-corn
Cats and. . . dogs
A stitch in time saves. . . nine
Good morning,. . . how are you

No real model of human grammar but there are regularities in human (re-)cognition

- Priming: people expect words given associates
- Semantic web: words in proximity tend to be cognates
- Words group in "documents"
- Words tend to cluster inside "texts"

Predictability in context

Movies. . . pop-corn
Cats and. . . dogs
A stitch in time saves. . . nine
Good morning,. . . how are you

No real model of human grammar but there are regularities in human (re-)cognition

- Priming: people expect words given associates
- Semantic web: words in proximity tend to be cognates
- Words group in "documents"
- Words tend to cluster inside "texts"

Predictability in context: In-document frequency

Distinguish global and local word frequencies: Term Frequency times Inverse Document Frequency (TF*IDF)

- Humans expect certain words based on topic, form, and style
- Break down corpus into small units, eg, articles, of roughly equal size
- Determine global word frequency
- Divide it by the fraction of documents the word is used in
- In-document frequency accounts statistically for both *priming* and *semantic webs*

[Pan and McKeown(1999)]

Predictability in context: In-document frequency

Distinguish global and local word frequencies: Term Frequency times Inverse Document Frequency (TF*IDF)

- Humans expect certain words based on topic, form, and style
- Break down corpus into small units, eg, articles, of roughly equal size
- Determine global word frequency
- Divide it by the fraction of documents the word is used in
- In-document frequency accounts statistically for both *priming* and *semantic webs*

[Pan and McKeown(1999)]

Predictability in context: In-document frequency

Distinguish global and local word frequencies: Term Frequency times Inverse Document Frequency (TF*IDF)

- Humans expect certain words based on topic, form, and style
- Break down corpus into small units, eg, articles, of roughly equal size
- Determine global word frequency
- Divide it by the fraction of documents the word is used in
- In-document frequency accounts statistically for both *priming* and *semantic webs*

[Pan and McKeown(1999)]

Predictability in context: In-document frequency

Distinguish global and local word frequencies: Term Frequency times Inverse Document Frequency (TF*IDF)

- Humans expect certain words based on topic, form, and style
- Break down corpus into small units, eg, articles, of roughly equal size
- Determine global word frequency
- Divide it by the fraction of documents the word is used in
- In-document frequency accounts statistically for both *priming* and *semantic webs*

[Pan and McKeown(1999)]

Predictability in context: In-document frequency

Distinguish global and local word frequencies: Term Frequency times Inverse Document Frequency (TF*IDF)

- Humans expect certain words based on topic, form, and style
- Break down corpus into small units, eg, articles, of roughly equal size
- Determine global word frequency
- Divide it by the fraction of documents the word is used in
- In-document frequency accounts statistically for both *priming* and *semantic webs*

[Pan and McKeown(1999)]

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts
  [Sproat and van Santen(1998)]
- Difference in distribution around a word predicts the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- ⇒ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts [Sproat and van Santen(1998)]
- Difference in distribution around a word predicts the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts [Sproat and van Santen(1998)]
- Difference in distribution around a word predicts the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts
  [Sproat and van Santen(1998)]
- Difference in distribution around a word predicts the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution
[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts
  [Sproat and van Santen(1998)]
- Difference in distribution around a word predicts the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts
  [Sproat and van Santen(1998)]
- Difference in distribution around a word *predicts* the word
- Subtract context divergence from word information
- Context increases "perceived" frequency

Predictability in context: Context distinctiveness

CD = Kullback-Leibler distance between local context distribution and global word distribution

[McDonald and Shillcock(2001)]

Predictability from direct context

- Use a statistical predictor
- Determine information about a word in its context
- $\Rightarrow$ Kulback-Leibler divergence
- Bag-of-words technique, eg, 10 word contexts

  [Sproat and van Santen(1998)]
- Difference in distribution around a word *predicts* the word
- Subtract context divergence from word *information*
- Context increases “perceived” frequency

Human word recognition

Basic aspects of word recognition

- Prelexical: Sound $\longrightarrow$ Phonemic Symbols
- Postlexical:
- $\longrightarrow$ Activation
- $\longrightarrow$ Competition
- $\longrightarrow$ Selection

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition

Basic aspects of word recognition

- Prelexical: Sound $\longrightarrow$ Phonemic Symbols
- Postlexical:
- $\longrightarrow$ Activation
- $\longrightarrow$ Competition
- $\longrightarrow$ Selection

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition

Basic aspects of word recognition

- Prelexical: Sound $\longrightarrow$ Phonemic Symbols
- Postlexical:
- $\longrightarrow$ Activation
- $\longrightarrow$ Competition
- $\longrightarrow$ Selection

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition

Basic aspects of word recognition

- Prelexical: Sound $\longrightarrow$ Phonemic Symbols
- Postlexical:
- $\longrightarrow$ Activation
- $\longrightarrow$ Competition
- $\longrightarrow$ Selection

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler]

[McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition

Basic aspects of word recognition

- Prelexical: Sound $\longrightarrow$ Phonemic Symbols
- Postlexical:
- $\longrightarrow$ Activation
- $\longrightarrow$ Competition
- $\longrightarrow$ Selection

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler]

[McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- All matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- *All* matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- *All* matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- *All* matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler]
[McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- *All* matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler]
[McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Activation

Phon.	ML Word	I(W)	Matching N	*Dutch*
b	bij	7.882	122,629	terms
boː	bovendien	11.658	5772	terms
boːm	bomen	14.831	798	terms

After recognizing a new phoneme, words are activated

- *All* matching words are activated
- Frequent words are activated more than rare words
- Non-matching phonemes decrease or end the activation of a word
- A word end is identified after an impossible continuation
- Word boundaries are also indicated by prosodic markers
- Simple model ignores morphological structure

[Cutler(1997)] [Norris et al.(2000)Norris, McQueen, and Cutler] [McQueen et al.(2003)McQueen, Cutler, and Norris]

Human word recognition: Postlexical stage - Competition

Words "compete" for activation

- Recognition is blocked as long as there are multiple candidates left
- Only when there is a single word left, is it recognized
- Words can occur inside other words
- $\Rightarrow$ wait for word boundary [Cutler(1997)]

Human word recognition: Postlexical stage - Competition

Words "compete" for activation

- Recognition is blocked as long as there are multiple candidates left
- Only when there is a single word left, is it recognized
- Words can occur inside other words
- $\Rightarrow$ wait for word boundary [Cutler(1997)]

Human word recognition: Postlexical stage - Competition

Words "compete" for activation

- Recognition is blocked as long as there are multiple candidates left
- Only when there is a single word left, is it recognized
- Words can occur inside other words
- ⇒ wait for word boundary [Cutler(1997)]

Human word recognition: Postlexical stage - Competition

Words "compete" for activation

- Recognition is blocked as long as there are multiple candidates left
- Only when there is a single word left, is it recognized
- Words can occur inside other words
- $\Rightarrow$ wait for word boundary [Cutler(1997)]

Human word recognition: Postlexical stage - Selection

Competition blocks selection

- If there is only a single candidate left, chose it
- If there is no single perfect candidate, chose the best (eg, *cigaret* for /ʃigərɛt/)
- Ganong effect: All candidates that fit an incomplete phoneme remain activated
  (eg, /(g/k)ot/ $\rightarrow$ *Goat* or *Coat*)
- Without enough phonetic "evidence", the most likely word is chosen $\rightarrow$ phonemic restoration
- People are only rarely aware of what *phonemes* have actually been spoken

Human word recognition: Postlexical stage - Selection

Competition blocks selection

- If there is only a single candidate left, chose it
- If there is no single perfect candidate, chose the best (eg, *cigaret* for /ʃɪgərɛt/)
- Ganong effect: All candidates that fit an incomplete phoneme remain activated (eg, /(g/k)ot/ $\rightarrow$ *Goat* or *Coat*)
- Without enough phonetic "evidence", the most likely word is chosen $\rightarrow$ phonemic restoration
- People are only rarely aware of what *phonemes* have actually been spoken

Human word recognition: Postlexical stage - Selection

Competition blocks selection

- If there is only a single candidate left, chose it
- If there is no single perfect candidate, chose the best (eg, *cigaret* for /ʃigərɛt/)
- Ganong effect: All candidates that fit an incomplete phoneme remain activated (eg, /(g/k)ot/ → *Goat* or *Coat*)
- Without enough phonetic "evidence", the most likely word is chosen → phonemic restoration
- People are only rarely aware of what *phonemes* have actually been spoken

Human word recognition: Postlexical stage - Selection

Competition blocks selection

- If there is only a single candidate left, chose it
- If there is no single perfect candidate, chose the best (eg, *cigaret* for /ʃɪgərɛt/)
- Ganong effect: All candidates that fit an incomplete phoneme remain activated (eg, /(g/k)ot/ $\rightarrow$ *Goat* or *Coat*)
- Without enough phonetic "evidence", the most likely word is chosen $\rightarrow$ phonemic restoration
- People are only rarely aware of what *phonemes* have actually been spoken

Human word recognition: Postlexical stage - Selection

Competition blocks selection

- If there is only a single candidate left, chose it
- If there is no single perfect candidate, chose the best (eg, *cigaret* for /ʃıgərɛt/)
- Ganong effect: All candidates that fit an incomplete phoneme remain activated (eg, /(g/k)ot/ → *Goat* or *Coat*)
- Without enough phonetic “evidence”, the most likely word is chosen → phonemic restoration
- People are only rarely aware of what *phonemes* have actually been spoken

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a large (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a large (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a large (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a large (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a large (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a *large* (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, >1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a *large* (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, >1 million word-forms?
- No data available on this measure

Phemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a *large* (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information

What difference a phoneme makes?

- Combine Information theory with Human word recognition
- Words matching a phoneme onset with and without the new phoneme added
- $I(phon|onset) = H(W|onset + \star) - H(W|onset + phon)$
- $H(W|S)$: Entropy of words matching S
- Use a *large* (automatically) transcribed corpus, eg, 350*Mword Twente News Corpus* [Ordelman(2002)]
- Sensitive to vocabulary structure
- But: Psychologically implausible?
- Compound words, $>$ 1 million word-forms?
- No data available on this measure

Phonemic information: Alternative

$$
\begin{aligned}
I(phon|onset) &= -\log_2 P(phon|onset) \\
&= -\log_2 \frac{TokenCount(onset + phon)}{TokenCount(onset + \star)}
\end{aligned}
$$

Information in phoneme given preceding onset

- Insensitive to vocabulary structure
- Can use smaller ($\approx 30Mw$) corpora
- Adaptable
- Psychological plausibility???
- Data are available

Phonemic information: Alternative

$$
\begin{aligned}
I(phon|onset) &= -\log_2 P(phon|onset) \\
&= -\log_2 \frac{TokenCount(onset + phon)}{TokenCount(onset + \star)}
\end{aligned}
$$

Information in phoneme given preceding onset

- Insensitive to vocabulary structure
- Can use smaller ($\approx 30Mw$) corpora
- Adaptable
- Psychological plausibility???
- Data are available

Phonemic information: Alternative

$$
\begin{aligned}
I(phon|onset) &= -\log_2 P(phon|onset) \\
&= -\log_2 \frac{TokenCount(onset + phon)}{TokenCount(onset + \star)}
\end{aligned}
$$

Information in phoneme given preceding onset

- Insensitive to vocabulary structure
- Can use smaller ($\approx 30Mw$) corpora
- Adaptable
- Psychological plausibility???
- Data are available

Phemic information: Alternative

$$
\begin{aligned}
I(phon|onset) &= -\log_2 P(phon|onset) \\
&= -\log_2 \frac{TokenCount(onset + phon)}{TokenCount(onset + \star)}
\end{aligned}
$$

Information in phoneme given preceding onset

- Insensitive to vocabulary structure
- Can use smaller ($\approx 30Mw$) corpora
- Adaptable
- Psychological plausibility???
- Data are available

Phonemic information: Alternative

$$
\begin{aligned}
I(phon|onset) &= -\log_2 P(phon|onset) \\
&= -\log_2 \frac{TokenCount(onset + phon)}{TokenCount(onset + \star)}
\end{aligned}
$$

Information in phoneme given preceding onset

- Insensitive to vocabulary structure
- Can use smaller ($\approx 30Mw$) corpora
- Adaptable
- Psychological plausibility???
- Data are available

Phonemic information: In-document frequency

$$I'(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot IDF(w) - TF(w)) \cdot TotalCount$ $TF(w)$: Term frequency of w

Include statistical predictability of in-document frequency

- In the text, the correct word will be "predictable"
- Perceived "frequency" of the correct word, w, is the in-document frequency, $TF(w) \cdot IDF(w)$
- Assume only the correct word, w, is boosted
- Replace global frequency of w by in-document frequency

Phonemic information: In-document frequency

$$I'(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot IDF(w) - TF(w)) \cdot TotalCount$ $TF(w)$: Term frequency of w

Include statistical predictability of in-document frequency

- In the text, the correct word will be "predictable"
- Perceived "frequency" of the correct word, w, is the in-document frequency, $TF(w) \cdot IDF(w)$
- Assume only the correct word, w, is boosted
- Replace global frequency of w by in-document frequency

Phonemic information: In-document frequency

$$I'(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot IDF(w) - TF(w)) \cdot TotalCount$ $TF(w)$: Term frequency of w

Include statistical predictability of in-document frequency

- In the text, the correct word will be "predictable"
- Perceived "frequency" of the correct word, w, is the in-document frequency, $TF(w) \cdot IDF(w)$
- Assume only the correct word, w, is boosted
- Replace global frequency of w by in-document frequency

Phonemic information: In-document frequency

$$I'(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot IDF(w) - TF(w)) \cdot TotalCount$ $TF(w)$: Term frequency of w

Include statistical predictability of in-document frequency

- In the text, the correct word will be "predictable"
- Perceived "frequency" of the correct word, w, is the in-document frequency, $TF(w) \cdot IDF(w)$
- Assume only the correct word, w, is boosted
- Replace global frequency of w by in-document frequency

Phonemic information: Context distinctiveness

$$I''(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot 2^{CD(w)} - TF(w)) \cdot TotalCount$
$CD(w) = KL(LocalDistr(w), GlobalDistr)$

$TF(w)$: Term frequency of w

Focus on direct context of w: $LocalDistr(w)$

- $CD(w)$: Kullback-Leibler distance between local and global distribution
- Perceived frequency is $TF(w) \cdot 2^{CD(w)}$
- Data *are* available

Phonemic information: Context distinctiveness

$$I''(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot 2^{CD(w)} - TF(w)) \cdot TotalCount$
$CD(w) = KL(LocalDistr(w), GlobalDistr)$

$TF(w)$: Term frequency of w

Focus on direct context of w: $LocalDistr(w)$

- $CD(w)$: Kullback-Leibler distance between local and global distribution
- Perceived frequency is $TF(w) \cdot 2^{CD(w)}$
- Data *are* available

Phonemic information: Context distinctiveness

$$I''(phon|onset, w) = -\log_2 \frac{TokenCount(onset + phon) + D(w)}{TokenCount(onset + \star) + D(w)}$$

$D(w) = (TF(w) \cdot 2^{CD(w)} - TF(w)) \cdot TotalCount$
$CD(w) = KL(LocalDistr(w), GlobalDistr)$
$TF(w)$: Term frequency of w

Focus on direct context of w: $LocalDistr(w)$

- $CD(w)$: Kullback-Leibler distance between local and global distribution
- Perceived frequency is $TF(w) \cdot 2^{CD(w)}$
- Data *are* available

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ ($= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6}$)
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^6)$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

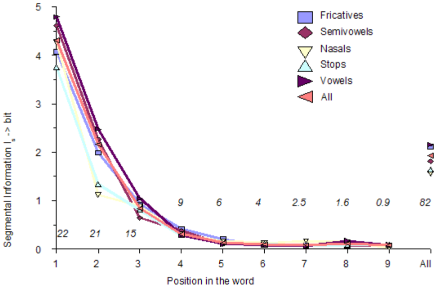
- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ ($= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^6$)
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information: Context distinctiveness

Example: /o/ in Dutch "boom" (*tree*) using CELEX/CGN

- Word tokens starting with /bo/: 67,710 (1,172 CELEX entries)
- The same for /b./: 1,544,483 (26,186 CELEX entries)
- $I = -\log_2(\frac{67710}{1544483}) = 4.51$ bit
- Relative CGN frequency of *boom*: $5.05 \cdot 10^{-5}$
- Context Distinctiveness: $CD(boom) = 4.53$ bit
- Relative frequency in context: $2^{CD(boom)} \cdot 5.05 \cdot 10^{-5} = 1.2 \cdot 10^{-3}$
- CELEX word count of *boom*: 2,226 (smoothed count)
- Context-corrected CELEX count: $45,402$ $(= 1.2 \cdot 10^{-3} \cdot 39 \cdot 10^{6})$
- Correction term: $D(boom) = 45,402 - 2,226 = 43,176$
- $I'' = -\log_2([67710 + 43176]/[1544483 + 43176]) = 3.84$
- That is, $I'' < I$, so context reduces lexical uncertainty.

Phonemic information

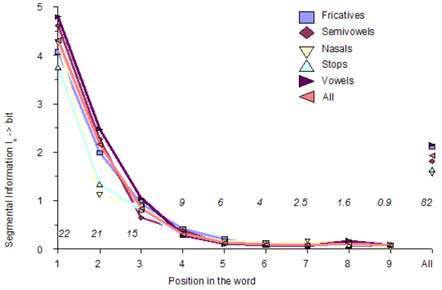


Segments (x1000)

$I''(phon|contex, w)$ versus the position in the word

- Only syllables without a /ə/
- Strong decline after a few positions part of model
- All manners of articulation carry the same information

Phonemic information

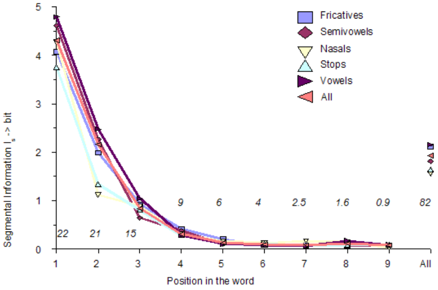


Segments (x1000)

$I''(phon|contex, w)$ versus the position in the word

- Only syllables without a /ə/
- Strong decline after a few positions part of model
- All manners of articulation carry the same information

Phemic information



Segments (x1000)

$I''(phon|contex, w)$ versus the position in the word

- Only syllables without a /ə/
- Strong decline after a few positions part of model
- All manners of articulation carry the same information

Bibliography I

Dirk Bühler, Wolfgang Minker, and Artha Elciyanti.
Using Language Modelling to Integrate Speech Recognition with a Flat Semantic Analysis.
In *Proceedings of the 6th SIGdialWorkshop on Discourse and Dialogue, Lisbon*, pages 212–216, 2 September 2005.
URL http://www.sigdial.org/workshops/workshop6/proceedings/pdf/86-paper.pdf.

A. Cutler.
The comparative perspective on spoken-language processing.
Speech Communication, 21(1-2):3–16, February 1997.
URL http://www.cog.jhu.edu/faculty/smolensky/050.823/Readings/Cutler%2097%20Speech%20Communication.pdf.

Yaniv Dover.
A short account of a connection of power laws to the information entropy.
PHYSICA A, 334:591–597, 2004.
URL http://arxiv.org/pdf/cond-mat/0309383.

Rakesh Dugad and U.B. Desai.
A tutorial on Hidden Markov Models.
Technical Report SPANN-96.1, Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology - Bombay, India, May 1996.
URL http://www.ling.gu.se/~leifg/stat01/doc/newhmmtut.ps.

FSF.
GNU General Public License.
Web, June 1991.
URL http://www.gnu.org/licenses/gpl.html.

Bibliography II

Daniel Jurafsky and James H. Martin.
Speech and Language Processing.
Prentice-Hall, 2000.
ISBN 0-13-095069-6.
URL http://www.cs.colorado.edu/~martin/slp.html.

Kenji Kawamura and Naomichi Hatano.
Universality of zipf's law.
Preprint archive, 2002.
URL http://arxiv.org/pdf/cond-mat/0203455.

S.A. McDonald and R.C. Shillcock.
Rethinking the word frequency effect: the neglected role of distributional information in lexical processing.
Language and Speech, 44(3):295–323, 9 2001.

James M. McQueen, Anne Cutler, and Dennis Norris.
Flow of information in the spoken word recognition system.
Speech Communication, 41(1):257–270, August 2003.

Dennis Norris, James M. McQueen, and Anne Cutler.
Merging information in speech recognition: Feedback is never necessary.
Behavioral and Brain Sciences, 23(03):299–325, June 2000.

Roeland Ordelman.
Twente Nieuws Corpus (TwNC).
Corpus, 2002.
URL http://wwwhome.cs.utwente.nl/~druid/TwNC/TwNC-main.html.

Bibliography III

M. Owens, P. O'Boyle, J. McMahon, J. Ming, and F.J. Smith.
A comparison of human and statistical language model performance using missing-word tests.
Language and Speech, 40:377–389, 1997.

S. Pan and K. McKeown.
Word informativeness and automatic pitch accent modeling.
Proceedings of EMNLP/VLC'99, New Brunswick, NJ, USA, 1999.
URL http://acl.ldc.upenn.edu/W/W99/W99-0619.pdf.

M. I. Posner.
Timing the brain: Mental chronometry as a tool in neuroscience.
PLoS Biology, 3(2):e51, 2005.
URL http://biology.plosjournals.org/archive/1545-7885/3/2/pdf/10.1371_journal.pbio.0030051-L.pdf.

Project Gutenberg.
Project gutenberg free ebook library.
Web, 2005.
URL http://www.gutenberg.org/.

Rebecca Sacra.
Reaction Time and Neural Circuitry.
In Mary Louise Bellamy and Kathy Frame, editors, *Neuroscience laboratory and classroom activities*.
National association of biology teachers society for neuroscience, 1996.
URL http://www.nabt.org/sup/publications/nlca/nlcapdf/13NLCAchp11.pdf.

Bibliography IV

M. Sigman and S. Dehaene.
Parsing a cognitive task: A characterization of the mind's bottleneck.
PLoS Biology, 3(2):e37, 2005.
URL http://biology.plosjournals.org/archive/1545-7885/3/2/pdf/10.1371_journal.pbio.0030037-L.pdf.

R. Sproat and J. van Santen.
Automatic ambiguity detection.
Proceedings of ICSLP 98, Sydney, Australia, 1998.
URL http://compling.ai.uiuc.edu/rws/newindex/a41.pdf.

R.J.J.H. Van Son and L.C.W. Pols.
How efficient is speech?
In *Proceedings of the Institute of Phonetic Sciences*, volume 25, pages 171–184, 2003.
URL http://www.fon.hum.uva.nl/rob/Publications/RvSProceedings25.pdf.

R.J.J.H. Van Son and J.P.H. Van Santen.
Duration and spectral balance of intervocalic consonants: A case for efficient communication.
Speech Communication, 47:100–123, 2005.
URL
http://www.fon.hum.uva.nl/rob/Publications/SPECOM-D-05-00016UncorrectedProof.pdf.

Bibliography

Bibliography III

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".
   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   1. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   2. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

3. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   1. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

2. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
3. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

Bibliography VI

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.
   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.
   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

No Warranty

Bibliography VIII

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *one line to give the program's name and a brief idea of what it does.*
> *Copyright (C) yyyy name of author*
> *This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.*
>
> *This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.*
>
> *You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.*

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> *Gnomovision version 69, Copyright (C) yyyy name of author*
> *Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.*
> *This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.*