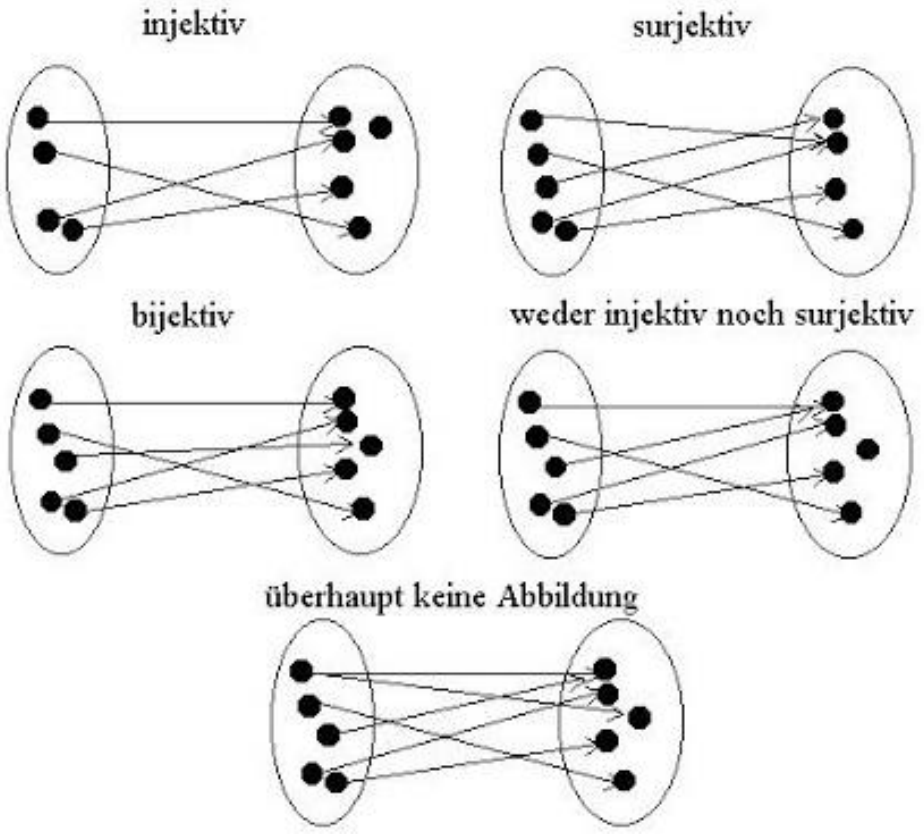
Remark: Text Preprocessing

F(<u>w</u>) := Text Preprocessing

- Tokenization
- Character Set
- Punctuation
- ....

?

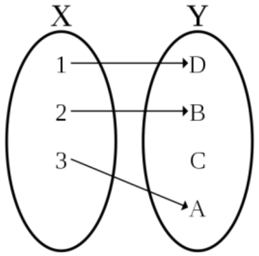


Injective, surjective, bijective - What??

$f: X \rightarrow Y$ is called $\Leftrightarrow$

? injective

Injective, surjective, bijective - What??



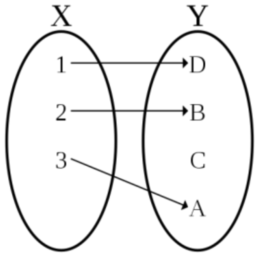
$f: X \rightarrow Y$ is called $\Leftrightarrow$

injective $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$

?

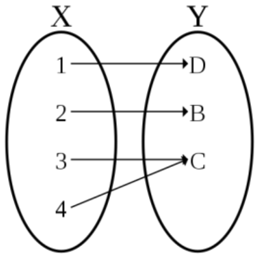
surjective

Injective, surjective, bijective - What??



$f: X \to Y$ is called $\Leftrightarrow$

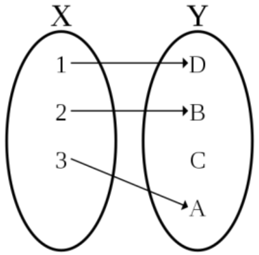
injective $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$



surjective $\forall y \in Y \; \exists x \in X\colon \; f(x) = y$

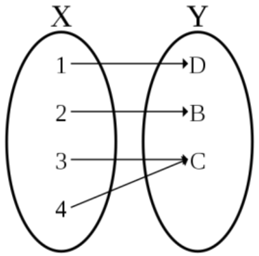
? bijective

Injective, surjective, bijective - What??

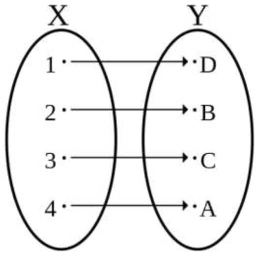


$f: X \to Y$ is called $\Leftrightarrow$

injective $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$



surjective $\forall y \in Y \; \exists x \in X\colon \; f(x) = y$



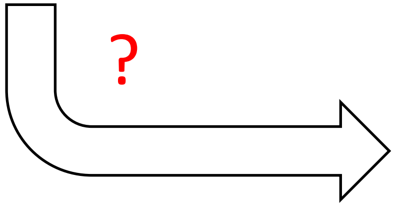
bijective f is injective and surjective

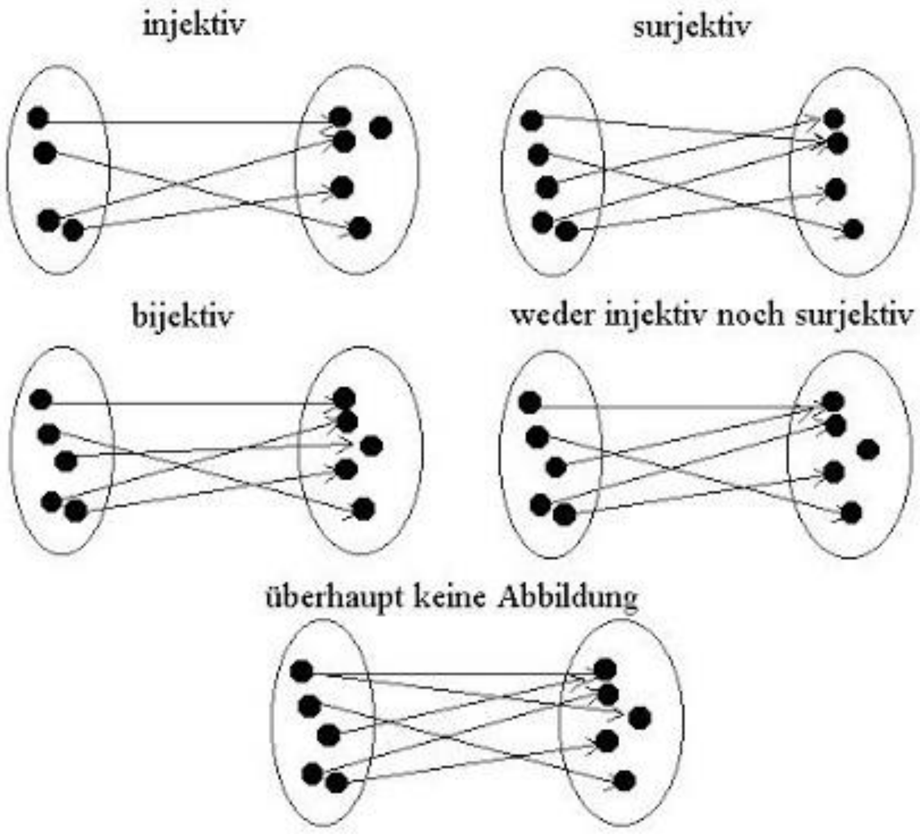
Technische Hochschule Ingolstadt

Remark: Text Preprocessing

F(w) := Text Preprocessing

- Tokenization
- Character Set
- Punctuation
- ....



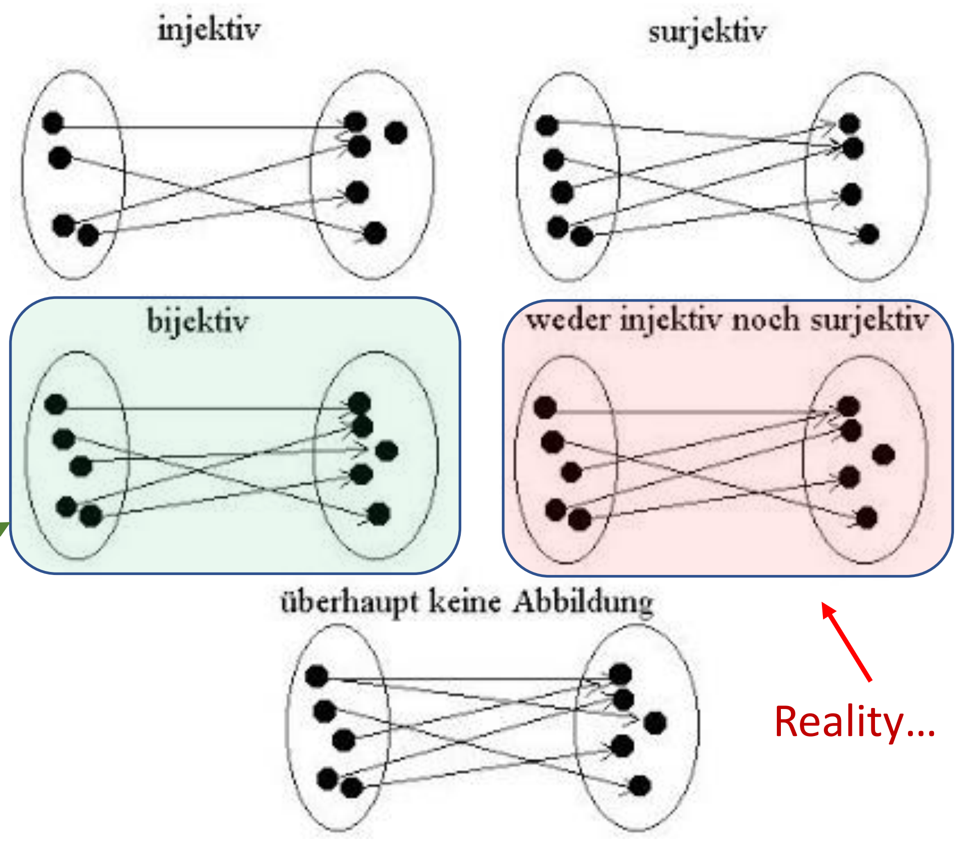


Technische Hochschule Ingolstadt

Remark: Text Preprocessing

F(w) := Text Preprocessing

- Tokenization
- Character Set
- Punctuation
- ....



Desire/Goal...

Reality...

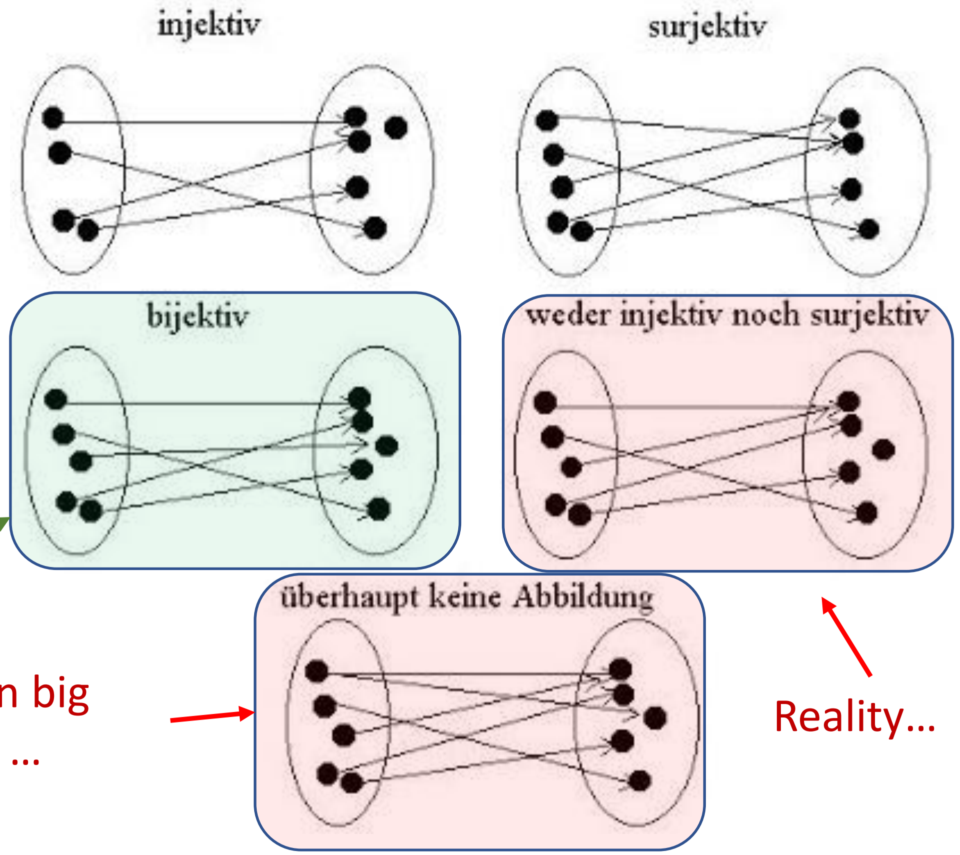
Technische Hochschule Ingolstadt

Remark: Text Preprocessing

F(<u>w</u>) := Text Preprocessing

- Tokenization
- Character Set
- Punctuation
- ....

Desire/Goal...



Reality in big projects ...

Reality...

How to weigh the *association between two words*?

How much more do the two words co-occur in our corpus

than we would have a priori expected them to appear by chance?

Intuition: „How much more do the two words co-occur in our corpus than we would have a priori expected them to appear by chance?"

Definition

The PMI between target word w and context word c is defined as

$$PMI(w,c) := \log_2 \frac{P(w,c)}{P(w)P(c)}.$$

Further Reading: Chapter 6.6, https://web.stanford.edu/~jurafsky/slp3/

Pointwise Mutual Information

Examples

$$PMI(w,c) \coloneqq \log_2 \frac{P(w,c)}{P(w)P(c)}$$

PMI >> 0 := co-occurrence of w_i and w_j
PMI near zero := w_i and w_j are unlikely as word pair in the corpus

Pointwise Mutual Information

Examples

$$PMI(w, c) := \log_2 \frac{P(w, c)}{P(w)P(c)}$$

$$\mathrm{PMI}(\mathrm{santa}, \mathrm{fe}) = 13.74529$$

$$\mathrm{PMI}(\mathrm{hong}, \mathrm{kong}) = 12.54758$$

$$\mathrm{PMI}(\mathrm{las}, \mathrm{vegas}) = 12.53122$$

$$\mathrm{PMI}(\mathrm{los}, \mathrm{angeles}) = 11.16868$$

$$\mathrm{PMI}(\mathrm{tele}, \mathrm{communications}) = 8.13129$$

$$\mathrm{PMI}(\mathrm{multimillion}, \mathrm{dollar}) = 8.03381$$

$$\mathrm{PMI}(\mathrm{U.}, \mathrm{S.}) = 7.21532.$$

Pointwise Mutual Information

Examples

$$PMI(w,c) \coloneqq \log_2 \frac{P(w,c)}{P(w)P(c)}$$

$$\begin{aligned}
\mathrm{PMI}(\mathrm{santa},\mathrm{fe}) &= 13.74529\\
\mathrm{PMI}(\mathrm{hong},\mathrm{kong}) &= 12.54758\\
\mathrm{PMI}(\mathrm{las},\mathrm{vegas}) &= 12.53122\\
\mathrm{PMI}(\mathrm{los},\mathrm{angeles}) &= 11.16868\\
\mathrm{PMI}(\mathrm{tele},\mathrm{communications}) &= 8.13129\\
\mathrm{PMI}(\mathrm{multimillion},\mathrm{dollar}) &= 8.03381\\
\mathrm{PMI}(\mathrm{U.},\mathrm{S.}) &= 7.21532.
\end{aligned}$$

$$\begin{aligned}
\mathrm{PMI}(\mathrm{accounted},\mathrm{for}) &= 5.98041\\
\mathrm{PMI}(\mathrm{according},\mathrm{to}) &= 5.93769\\
\mathrm{PMI}(\mathrm{intends},\mathrm{to}) &= 5.9195\\
\mathrm{PMI}(\mathrm{able},\mathrm{to}) &= 5.91757
\end{aligned}$$

Pointwise Mutual Information

Examples

$$PMI(w,c) \coloneqq \log_2 \frac{P(w,c)}{P(w)P(c)}$$

$$\mathrm{PMI}(\text{manufacturer}, \text{holds}) = 0.00058$$

$$\mathrm{PMI}(\text{stocks}, \text{shows}) = 0.00013$$

$$\mathrm{PMI}(\text{legal}, \text{confrontation}) = -0.00013$$

$$\mathrm{PMI}(\text{pennsylvania}, \text{cleveland}) = -0.00061$$

Pointwise Mutual Information

Examples

$$PMI(w,c) \coloneqq \log_2 \frac{P(w,c)}{P(w)P(c)}$$

$$\mathrm{PMI}(\mathrm{manufacturer}, \mathrm{holds}) = 0.00058$$
$$\mathrm{PMI}(\mathrm{stocks}, \mathrm{shows}) = 0.00013$$
$$\mathrm{PMI}(\mathrm{legal}, \mathrm{confrontation}) = -0.00013$$
$$\mathrm{PMI}(\mathrm{pennsylvania}, \mathrm{cleveland}) = -0.00061$$

$$\mathrm{PMI}(\mathrm{not}, \mathrm{international}) = -4.50398$$
$$\mathrm{PMI}(\mathrm{two}, \mathrm{interests}) = -5.00358$$
$$\mathrm{PMI}(\mathrm{the}, \mathrm{said}) = -8.95582$$

Pointwise Mutual Information

Interpretation

$$PMI(w, c) \coloneqq \log_2 \frac{P(w, c)}{P(w)P(c)}$$

```
PMI >> 0        := co-occurrence of wᵢ and wⱼ
PMI near zero   := wᵢ and wⱼ are unlikely as word pair in the corpus
```

Pointwise Mutual Information

Question

$$PMI(w,c) \stackrel{?}{=} PMI(c,w)$$

Mutual Information

"It quantifies the amount of information obtained about one random variable through observing the other random variable."

What was entropy?

$$H(X) = $$ Who remembers it?

What was entropy?

$$H(X) = -\sum_{w \in X} p(w) \log_2 p(w)$$

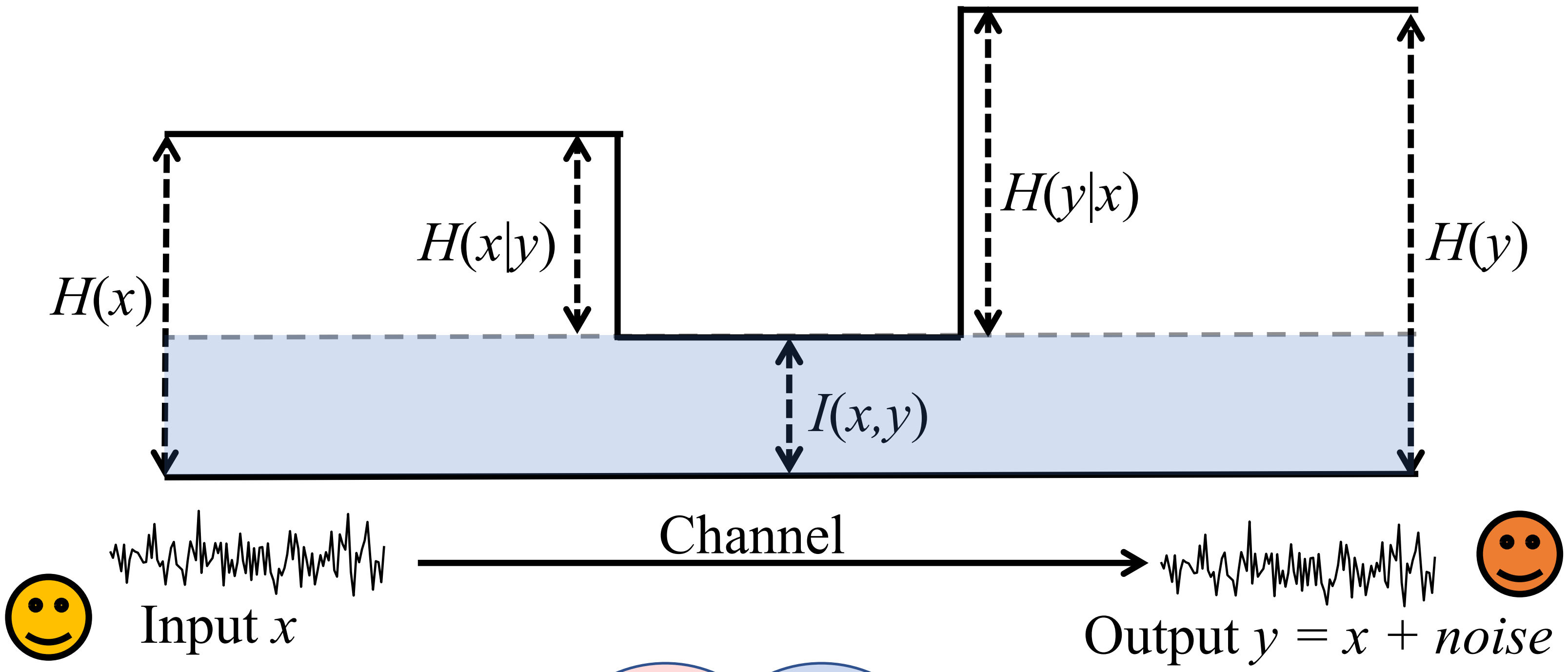
What was conditional entropy?

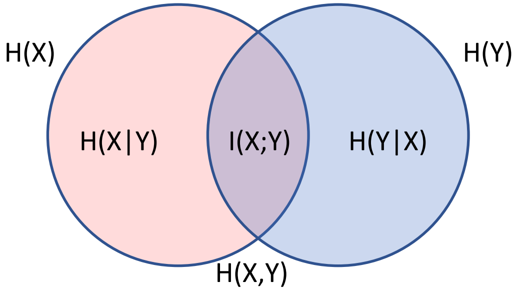
$$
\begin{aligned}
H(Y|X) &= \sum_{w \in X} p(w) H(Y|X = w) \\
&= \sum_{w \in X} p(w) [-\sum_{v \in Y} p(v|w) \log_2 p(v|w)]
\end{aligned}
$$

Technische Hochschule Ingolstadt

Mutual Information

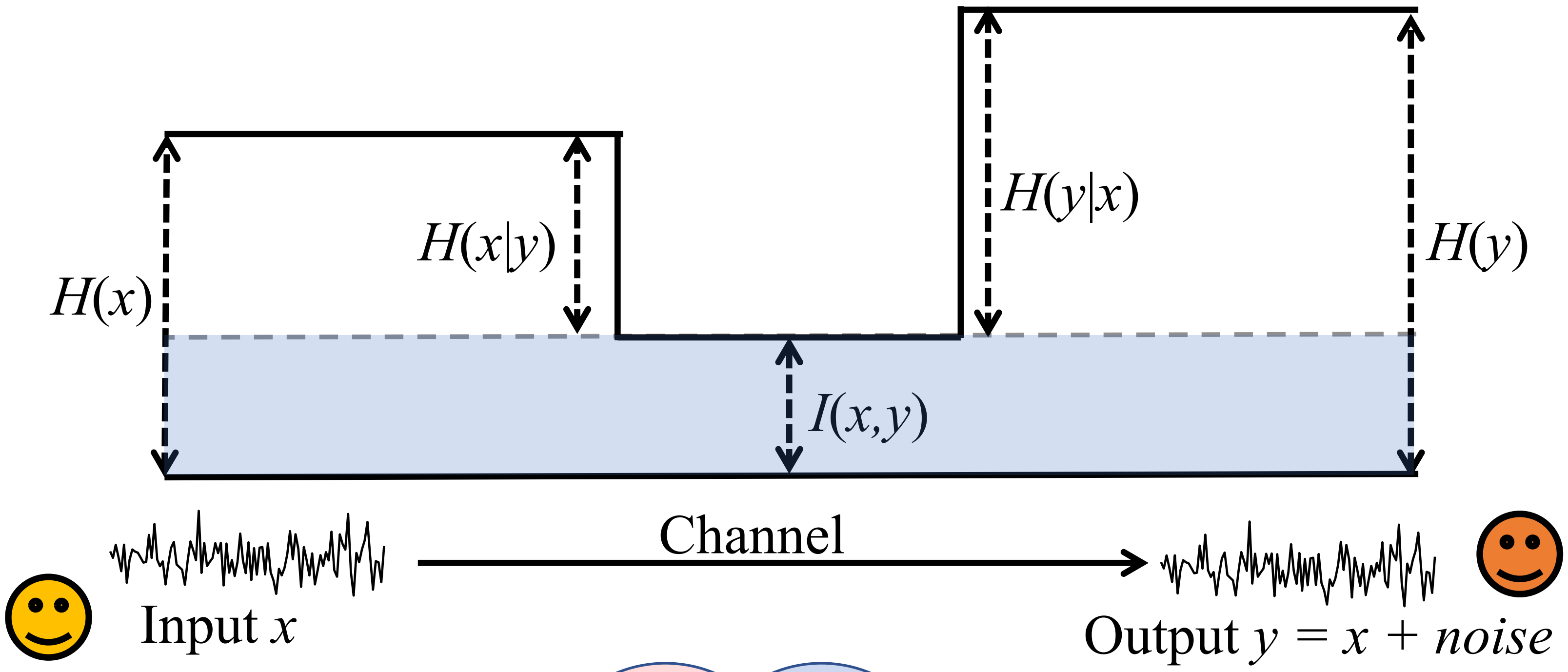
In Deutsch: Transinformation

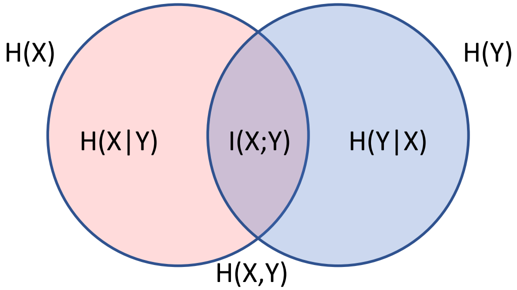




Mutual Information

In Deutsch: Transinformation





$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Mutual Information

In Deutsch: Transinformation

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Technische Hochschule Ingolstadt

Mutual Information

In Deutsch: Transinformation

$$\begin{aligned} H(Y|X) &= \sum_{w \in X} p(w) H(Y|X=w) \\ &= \sum_{w \in X} p(w) [-\sum_{v \in Y} p(v|w) \log_2 p(v|w)] \end{aligned}$$

$$H(X) = -\sum_{w \in X} p(w) \log_2 p(w)$$

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Technische Hochschule Ingolstadt

Mutual Information

In Deutsch: Transinformation

$$\mathrm{I}(w_I, w_J) := \sum_{i,j} P(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

$$\begin{aligned} H(Y|X) &= \sum_{w \in X} p(w) H(Y|X = w) \\ &= \sum_{w \in X} p(w) [-\sum_{v \in Y} p(v|w) \log_2 p(v|w)] \end{aligned}$$

$$H(X) = -\sum_{w \in X} p(w) \log_2 p(w)$$

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

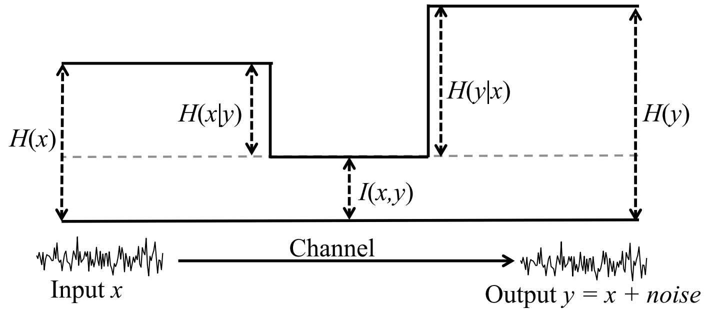
Technische Hochschule Ingolstadt

Remarks

Pointwise Mutual Information

$$\mathrm{I}(w_I, w_J) := \sum_{i,j} P(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

- "In many applications, one wants to **maximize mutual information** (thus increasing dependencies), which is often equivalent to **minimizing** [conditional entropy]."



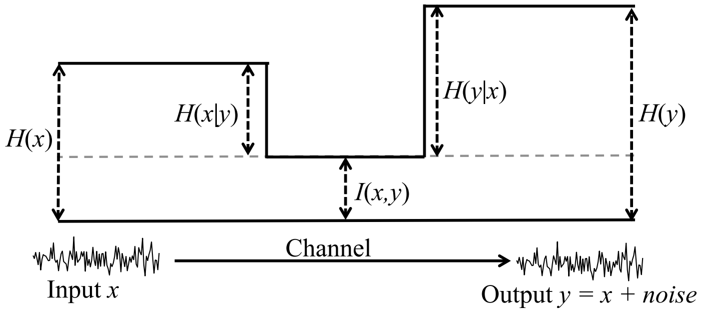
Further Reading: https://en.wikipedia.org/wiki/Mutual_information

Remarks

Pointwise Mutual Information

$$\mathrm{I}(w_I, w_J) := \sum_{i,j} P(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

- “In many applications, one wants to **maximize mutual information** (thus increasing dependencies), which is often equivalent to **minimizing** conditional entropy.”



Question: Relation to Language Models or ML in general?

- LMs: minimize perplexity ⇔ maximize probability of test set ⇔ minimize cross entropy (-> Jurafsky, Section 3.8)
- ML: goal is often: minimize cross entropy loss

Further Reading: https://en.wikipedia.org/wiki/Mutual_information

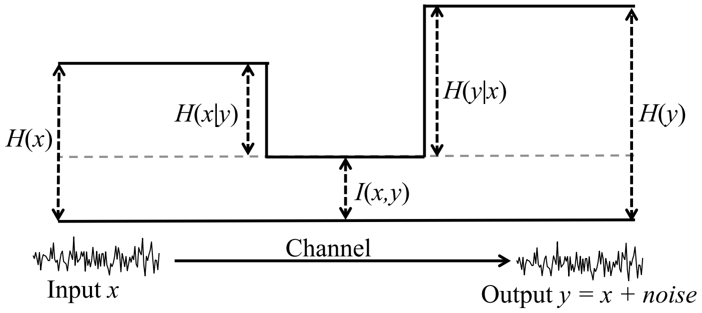
Technische Hochschule Ingolstadt

Remarks

Pointwise Mutual Information

$$\mathrm{I}(w_I, w_J) := \sum_{i,j} P(w_i, w_j) \log \boxed{\frac{p(w_i, w_j)}{p(w_i)p(w_j)}}$$

- “In many applications, one wants to **maximize mutual information** (thus increasing dependencies), which is often equivalent to **minimizing conditional entropy**.”
- “mutual information between phrases and contexts is used as a feature for k-means clustering to discover semantic clusters (concepts)”



Further Reading: https://en.wikipedia.org/wiki/Mutual_information

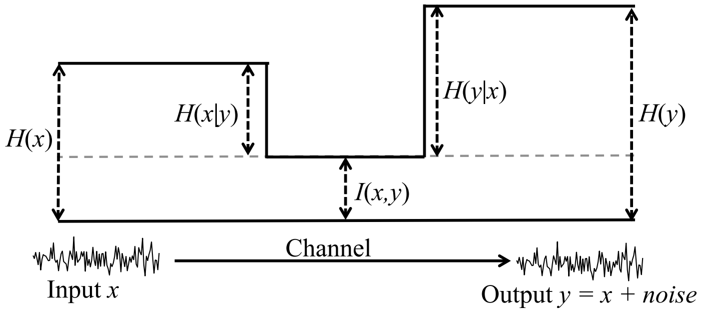
Remarks

Pointwise Mutual Information

$$\mathrm{I}(w_I, w_J) := \sum_{i,j} P(w_i, w_j) \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

- "In many applications, one wants to **maximize mutual information** (thus increasing dependencies), which is often equivalent to **minimizing conditional entropy**."

- "mutual information between phrases and contexts is used as a feature for k-means clustering to discover semantic clusters (concepts)"
- Can be used for feature selection in Machine Learning: minimum Redundancy Maximum Relevance (mRMR) Algorithm



Further Reading: https://en.wikipedia.org/wiki/Mutual_information

Further Watching: https://youtu.be/YKDZHPJ-pQ0

Text correction

Technische Hochschule Ingolstadt

Text correction

Input	To check	Correction
I don't know **whether** I want to	**whether**	whether
The **weather** is pretty bad today	**weather**	weather
Whether he likes me or not I can't say	**whether**	whether
I like sunny **weather**	**weather**	weather

Any Idea?

https://en.wikipedia.org/wiki/Levenshtein_distance

Text correction

Input	To check	Correction
I don’t know **whether** I want to	**whether**	whether
The **weather** is pretty bad today	**weather**	weather
Whether he likes me or not I can’t say	**whether**	whether
I like sunny **weather**	**weather**	weather

Levenshtein distance(“whether”, “weather”)

…

https://en.wikipedia.org/wiki/Levenshtein_distance

Text correction

Input	To check	Correction
I don’t know **whether** I want to	**whether**	whether
The **weather** is pretty bad today	**weather**	weather
Whether he likes me or not I can’t say	**whether**	whether
I like sunny **weather**	**weather**	weather

Idea?

https://www.cl.uni-heidelberg.de/courses/ws19/ecl/material/ngram_handout.pdf

Technische Hochschule Ingolstadt

Text correction

Input	To check	Correction
I don't know **whether** I want to	**whether**	whether
The **weather** is pretty bad today	**weather**	weather
Whether he likes me or not I can't say	**whether**	whether
I like sunny **weather**	**weather**	weather

P(know whether) vs. P(know weather)

P(whether I) vs. P(weather I)

…

https://www.cl.uni-heidelberg.de/courses/ws19/ecl/material/ngram_handout.pdf

Code-Switch Detection

„Wenn Sie dort sind, please do not talk about our product. Ist das klar?“

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Code-Switch Detection

„**Wenn Sie dort sind**, please do not talk about our product. **Ist das klar?**“

German English **German**

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Anglicism Detection

„Mach ein **mind map**“ → English

vs.

„Mach eine **Gedächtniskarte**” → German

„ **Cancel** mal die Bestellung“ → English

vs.

„**Storniere** mal die Bestellung“ → German (from Latin)

vs.

„**Mach** mal die Bestellung **rückgängig**“ → German

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Technische Hochschule Ingolstadt

Problems

Pseudo-Anglicism

German: Handy vs. English: Mobile Phone

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Technische Hochschule Ingolstadt

Problems

Pseudo-Anglicism

German: Handy vs. English: Mobile Phone

German: Beamer vs. English: Projector

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Problems

Pseudo-Anglicism

German: Handy vs. English: Mobile Phone

German: Beamer vs. English: Projector

Hybrid word

Aquaphobia: Latin ”aqua” (water), Greek ”phobia”(fear)

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Problems

Pseudo-Anglicism

German: Handy vs. English: Mobile Phone

German: Beamer vs. English: Projector

Hybrid word

Aquaphobia: Latin ”aqua” (water), Greek ”phobia”(fear)

Schadsoftware: German “*Schad*en” (damage)

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Problems

Pseudo-Anglicism

German: Handy vs. English: Mobile Phone

German: Beamer vs. English: Projector

Hybrid word

Aquaphobia: Latin ”aqua” (water), Greek ”phobia”(fear)

Schadsoftware: German “*Schad*en” (damage)

English: malware

…

https://www.csl.uni-bremen.de/cms/images/documents/publications/BA_SebastianLeidig.pdf

Technische Hochschule Ingolstadt

Naive Bayes Classifiers

Naive Bayes for text classification

What does a NB classifier model look like?

$$\hat{c} = \text{argmax}_{c \in C} P(c|d)$$
$$= \text{argmax}_{c \in C} P(d|c)P(c)$$

Technische Hochschule Ingolstadt

What does a NB classifier model look like?

$$\begin{aligned}\hat{c} &= \mathrm{argmax}_{c \in C} P(c|d) \\ &= \mathrm{argmax}_{c \in C} P(d|c)P(c)\end{aligned}$$

Compute most probable class, given some document d by

choosing the class which has the highest product of two probabilities

What does a NB classifier model look like?

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c|d)$$
$$= \operatorname{argmax}_{c \in C} P(d|c) P(c)$$

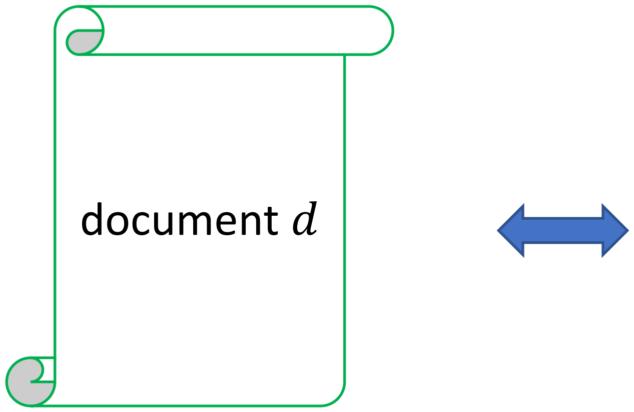
Compute most probable class, given some document d by

choosing the class which has the highest product of two probabilities:

likelihood of document and class prior probability

How to represent a document?

Remember: $\hat{c} = \text{argmax}_{c \in C} P(d|c)P(c)$



set of features $\mathrm{f}_1, \dots, f_n$

$$\hat{c} = \text{argmax}_{c \in C} P(\mathrm{f}_1, \dots, f_n|c)P(c)$$

Two simplifying assumptions

Technische Hochschule Ingolstadt

Two simplifying assumptions

1. A word’s position doesn’t matter

Two simplifying assumptions

1. A word’s position doesn’t matter (Bag-of-Words assumption)

Two simplifying assumptions

1. A word's position doesn't matter (Bag-of-Words assumption)
2. Features $\{\mathrm{f}_1, \ldots, f_n\} = F$ only encode word identity, not its position

Technische Hochschule Ingolstadt

Two simplifying assumptions

1. A word's position doesn't matter (Bag-of-Words assumption)
2. Features $\{\mathrm{f}_1, \ldots, f_n\} = F$ only encode word identity, not its position

2nd assumption ⇔ ***Naive Bayes Assumption***

Technische Hochschule Ingolstadt

Two simplifying assumptions

1. A word's position doesn't matter (Bag-of-Words assumption)
2. Features $\{\mathrm{f}_1, \ldots, f_n\} = F$ only encode word identity, not its position

2nd assumption ⇔ ***Naive Bayes Assumption***

$$P(\mathrm{f}_1, \ldots, f_n | c) = P(f_1 | c) \cdot P(f_2 | c) \cdot \ldots \cdot P(f_n | c)$$

Technische Hochschule Ingolstadt

Two simplifying assumptions

1. A word's position doesn't matter (Bag-of-Words assumption)
2. Features $\{\mathrm{f}_1, \ldots, f_n\} = F$ only encode word identity, not its position

2nd assumption ⇔ ***Naive Bayes Assumption***
= conditional independence assumption

$$P(\mathrm{f}_1, \ldots, f_n | c) = P(f_1 | c) \cdot P(f_2 | c) \cdot \ldots \cdot P(f_n | c)$$

Two simplifying assumptions

1. A word's position doesn't matter (Bag-of-Words assumption)
2. Features $\{\mathrm{f}_1, \ldots, f_n\} = F$ only encode word identity, not its position

2nd assumption ⇔ ***Naive Bayes Assumption***
= conditional independence assumption

$$P(\mathrm{f}_1, \ldots, f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot \ldots \cdot P(f_n|c)$$

$$\Rightarrow \; \hat{c} = \mathrm{argmax}_{c \in C} \, P(c) \cdot \prod_{f \in F} P(f|c)$$

Calculating NB document classifier

Remember: $\hat{c} = \operatorname{argmax}_{c \in C} P(c) \cdot \prod_{f \in F} P(f|c)$

Technische Hochschule Ingolstadt

Calculating NB document classifier

Remember: $\hat{c} = \operatorname{argmax}_{c \in C} P(c) \cdot \prod_{f \in F} P(f|c)$

Let X be the set of word positions in test document $d = (w_1, \ldots, w_{|X|})$.

Technische Hochschule Ingolstadt

Calculating NB document classifier

Remember: $\hat{c} = \text{argmax}_{c \in C} P(c) \cdot \prod_{f \in F} P(f|c)$

Let X be the set of word positions in test document $d = (w_1, \dots, w_{|X|})$.
Then compute

$$\hat{c} = \text{argmax}_{c \in C} P(c) \cdot \prod_{i \in X} P(w_i|c)$$

Technische Hochschule Ingolstadt

Calculating NB document classifier

Remember: $\hat{c} = \text{argmax}_{c \in C} \, P(c) \cdot \prod_{f \in F} P(f|c)$

Let X be the set of word positions in test document $d = (w_1, \dots, w_{|X|})$.
Then compute

$$\hat{c} = \text{argmax}_{c \in C} \, P(c) \cdot \prod_{i \in X} P(w_i|c)$$

Question: How to increase speed and avoid underflow?

Calculating NB document classifier

Remember: $\hat{c} = \text{argmax}_{c \in C}\, P(c) \cdot \prod_{f \in F} P(f|c)$

Let X be the set of word positions in test document $d = (w_1, \ldots, w_{|X|})$.
Then compute

$$\hat{c} = \text{argmax}_{c \in C}\, P(c) \cdot \prod_{i \in X} P(w_i|c)$$

Question: How to increase speed and avoid underflow?

$$\hat{c} = \text{argmax}_{c \in C} \log P(c) + \sum_{i \in X} \log P(w_i|c)$$

Technische Hochschule Ingolstadt

Calculating NB document classifier

Remember: $\hat{c} = \operatorname{argmax}_{c \in C} P(c) \cdot \prod_{f \in F} P(f|c)$

Let X be the set of word positions in test document $d = (w_1, \dots, w_{|X|})$.
Then compute

$$\hat{c} = \operatorname{argmax}_{c \in C} P(c) \cdot \prod_{i \in X} P(w_i|c)$$

Question: How to increase speed and avoid underflow?

$$\hat{c} = \operatorname{argmax}_{c \in C} \log P(c) + \sum_{i \in X} \log P(w_i|c)$$

$$\hat{P}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

Remark on Naive Bayes Classification

- NB classifier is a supervised machine learning method $f: X \rightarrow Y$

Technische Hochschule Ingolstadt

Remark on Naive Bayes Classification

- NB classifier is a supervised machine learning method $\mathrm{f}: \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \dots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$

Remark on Naive Bayes Classification

Remember: $\mathrm{f(d)} = \mathrm{argmax_c}\, P(d|c)P(c)$

- NB classifier is a supervised machine learning method $\mathrm{f}: \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \dots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$

Remark on Naive Bayes Classification

Remember: $\mathrm{f(d)} = \ \mathrm{argmax_c}\, P(d|c)P(c)$

- NB classifier is a supervised machine learning method $\mathrm{f}: \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \dots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$
- Given an observation, a NB classifier returns the class most likely to have generated the observation ⇔ **generative model**

Technische Hochschule Ingolstadt

Remark on Naive Bayes Classification

Remember: $\mathrm{f(d)} = \ \mathrm{argmax_c}\, P(d|c)P(c)$

- NB classifier is a supervised machine learning method $\mathrm{f}\colon \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \dots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$
- Given an observation, a NB classifier returns the class most likely to have generated the observation ⇔ **generative model**
  - first a class is sampled from P(c)

Technische Hochschule Ingolstadt

Remark on Naive Bayes Classification

Remember: $\mathrm{f(d)} = \mathrm{argmax}_{\mathrm{c}}\, P(d|c)P(c)$

- NB classifier is a supervised machine learning method $\mathrm{f}\colon \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \ldots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$
- Given an observation, a NB classifier returns the class most likely to have generated the observation ⇔ **generative model**
  - first a class is sampled from P(c)
  - then generate document/words/features… by sampling from P(d|c)

Technische Hochschule Ingolstadt

Remark on Naive Bayes Classification

Remember: $\mathrm{f(d)} = \mathrm{argmax_c}\, P(d|c)P(c)$

- NB classifier is a supervised machine learning method $\mathrm{f}: \mathrm{X} \rightarrow \mathrm{Y}$
  - Given: input $x \in X$ and a fixed set of output classes $\mathrm{Y} = \{c_1, \dots, c_N\}$
  - Goal: return predicted class $\hat{c} \in Y$
- Given an observation, a NB classifier returns the class most likely to have generated the observation ⇔ **generative model**
  - first a class is sampled from P(c)
  - then generate document/words/features… by sampling from P(d|c)

Question: how to generate text in this way?

Digression

Sample sentence from a *unigram* LM

Assume we have access to words in $V \cup \{< EOS >\}$

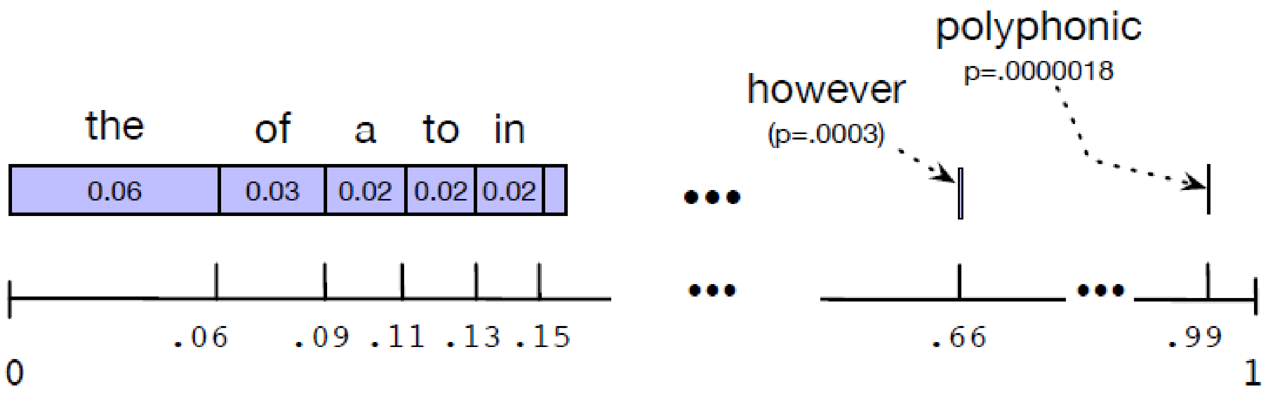
Assume sorted word probabilites: $P(w_1) > P(w_2) > \ldots > P(w_{|V|+1})$

Digression

Sample sentence from a *unigram* LM

Assume we have access to words in $V \cup \{< EOS >\}$

Assume sorted word probabilites: $P(w_1) > P(w_2) > \ldots > P(w_{|V|+1})$



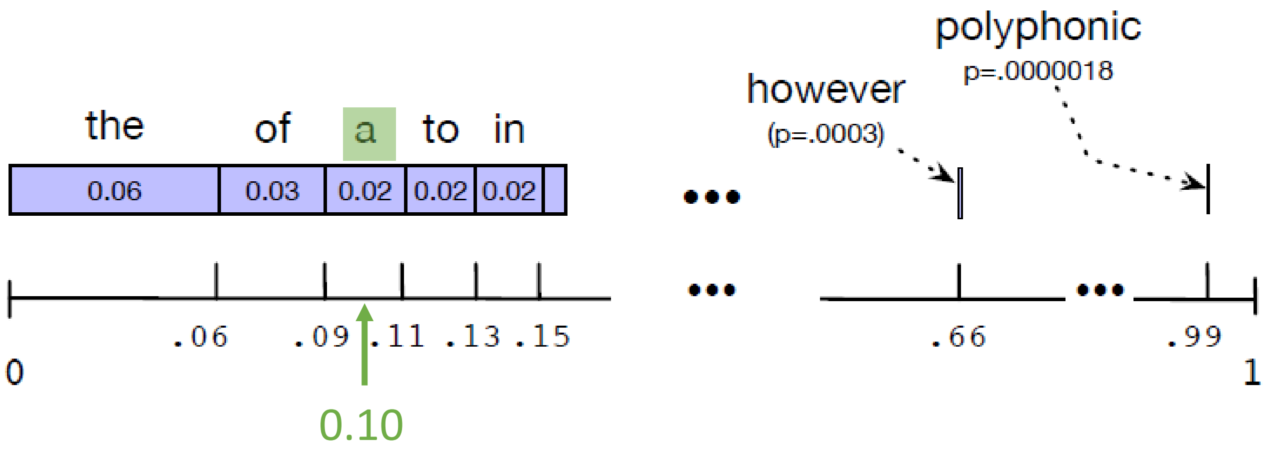
Note: $1 = \sum_i P(w_i)$

Technische Hochschule Ingolstadt

Digression
Sample sentence from a *unigram* LM

Assume we have access to words in $V \cup \{< EOS >\}$

Assume sorted word probabilites: $P(w_1) > P(w_2) > \ldots > P(w_{|V|+1})$



Note: $1 = \sum_i P(w_i)$

Randomly select value between 0 and 1: e.g. 0.10

GENERATED SEQUENCE => a

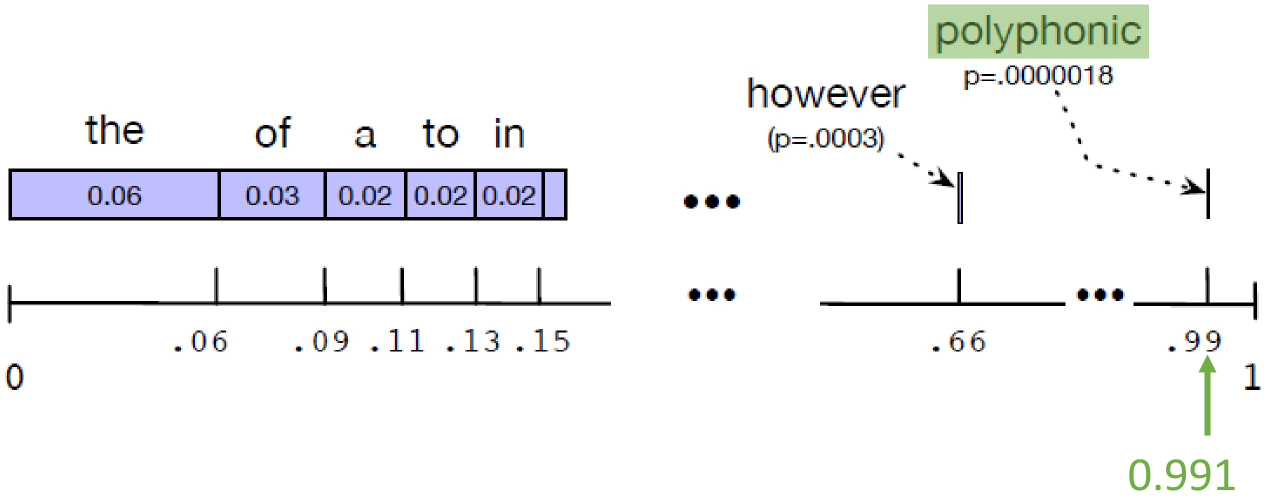
Technische Hochschule Ingolstadt

Digression

Sample sentence from a *unigram* LM

Assume we have access to words in $V \cup \{< EOS >\}$

Assume sorted word probabilites: $P(w_1) > P(w_2) > \ldots > P\left(w_{|V|+1}\right)$



Note: $1 = \sum_i P(w_i)$

Randomly select value between 0 and 1: e.g. 0.991

GENERATED SEQUENCE => a polyphonic

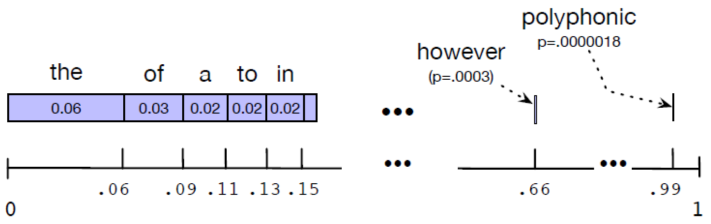
Technische Hochschule Ingolstadt

Digression

Sample sentence from a *unigram* LM

Assume we have access to words in $V \cup \{< EOS >\}$

Assume sorted word probabilites: $P(w_1) > P(w_2) > \ldots > P(w_{|V|+1})$



Note: $1 = \sum_i P(w_i)$

Algorithm

Repeat below steps 1-3, until you generate <EOS> token

1. Randomly select a value $x \in [0,1]$
2. Find that point x on the line
3. Print word whose interval includes chosen value x

Further reading: Section 3.3, https://web.stanford.edu/~jurafsky/slp3/3.pdf

Further questions

[1] https://web.stanford.edu/~jurafsky/slp3/4.pdf

Technische Hochschule Ingolstadt

Further questions

- How to compute $P(c)$? (see Section 4.2 in [1])

$$\hat{c} = \operatorname{argmax}_{c \in C} \log P(c) + \sum_{i \in X} \log P(w_i | c)$$

[1] https://web.stanford.edu/~jurafsky/slp3/4.pdf

Technische Hochschule Ingolstadt

Further questions

- How to compute $P(c)$? (see Section 4.2 in [1])
- What about words in test data which were not present in training documents for a class? (see Section 4.2 in [1])

$$\hat{c} = \operatorname{argmax}_{c \in C} \log P(c) + \sum_{i \in X} \log \boxed{P(w_i|c)}$$

$$\boxed{\hat{P}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}}$$

[1] https://web.stanford.edu/~jurafsky/slp3/4.pdf

Further questions

- How to compute $P(c)$? (see Section 4.2 in [1])
- What about words in test data which were not present in training documents for a class? (see Section 4.2 in [1])
- How to evaluate the classification algorithm?
  (next topic or Section 4.7 in [1])

[1] https://web.stanford.edu/~jurafsky/slp3/4.pdf