Lecture 11: Dynamic Programming and NLP

PCL II, CL, UZH

May 11, 2016

Universität Zürich[UZH]

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

LCS("fear", "fair") bottom-up

Universität Zürich[UZH]

	∅	F	E	A	R
∅	0	0	0	0	0
F	0	f = f 1	e ≠ f 1	a ≠ f 1	r ≠ f 1
A	0	f ≠ a 1	e ≠ a 1	a = a 2	r ≠ a 2
I	0	1	1	2	2
R	0	1	1	2	3

Contents

Universität Zürich[UZH]

- Levenshtein distance
- Tagging - Viterbi
- Sentence alignment

Levenshtein distance

- a metric of how different two strings are
- equals the **minimum** number of edits:
  - character insertions
  - character deletions
  - replacements of one character with another
- required to turn the 1st string into the 2nd

e.g. *lev(*planet, paper*)*: 3

planet → p~~l~~anet → panet → pa~~n~~^{p}et → papet → pape~~t~~^{r} → paper

1: del "l"

2: repl "n" with "p"

3: repl "t" with "r"

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅							
P							
A							
P							
E							
R							

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅	0						
P							
A							
P							
E							
R							

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0						
P	1						
A							
P							
E							
R							

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0						
P	1						
A	2						
P							
E							
R							

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0						
P	1						
A	2						
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1						
A	2						
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	?					
A	2						
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2						
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2	?					
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2	1					
P	3						
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2	1					
P	3	?					
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2	1					
P	3	2					
E	4						
R	5						

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0					
A	2	1					
P	3	2					
E	4	3					
R	5	4					

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1				
A	2	1					
P	3	2					
E	4	3					
R	5	4					

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1				
A	2	1	1				
P	3	2	2				
E	4	3	3				
R	5	4	4				

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2			
A	2	1	1	1			
P	3	2	2	2			
E	4	3	3	3			
R	5	4	4	4			

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3		
A	2	1	1	1	2		
P	3	2	2	2	2		
E	4	3	3	3	3		
R	5	4	4	4	4		

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	
A	2	1	1	1	2	3	
P	3	2	2	2	2	3	
E	4	3	3	3	3	2	
R	5	4	4	4	4	3	

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	5
A	2	1	1	1	2	3	4
P	3	2	2	2	2	3	4
E	4	3	3	3	3	2	3
R	5	4	4	4	4	3	3

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	5
A	2	1	1	1	2	3	4
P	3	2	2	2	2	3	4
E	4	3	3	3	3	2	3
R	5	4	4	4	4	3	**3**

Levenshtein distance
Bottom-up

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	5
A	2	1	1	1	2	3	4
P	3	2	2	2	2	3	4
E	4	3	3	3	3	2	3
R	5	4	4	4	4	3	**3**

= min(3 + 1,
3 + 1,
2 + 1),
2 + **1** because
'T' ≠ 'R'

Levenshtein distance optimal path

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	5
A	2	1	1	1	2	3	4
P	3	2	2	2	2	3	4
E	4	3	3	3	3	**2**	3
R	5	4	4	4	4	3	**3**

Levenshtein distance optimal path

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1	2	3	4	5	6
P	1	0	1	2	3	4	5
A	2	1	1	1	2	3	4
P	3	2	2	2	**2**	3	4
E	4	3	3	3	3	**2**	3
R	5	4	4	4	4	3	**3**

Levenshtein distance optimal path

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	**0**	1	2	3	4	5	6
P	1	**0**	**1**	2	3	4	5
A	2	1	1	**1**	2	3	4
P	3	2	2	2	**2**	3	4
E	4	3	3	3	3	**2**	3
R	5	4	4	4	4	3	**3**

Why fill the entire matrix?

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅	0						
P							
A							
P							
E							
R							

Why fill the entire matrix?

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	0	1					
P	1	0					
A							
P							
E							
R							

Why fill the entire matrix?

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅	0	1					
P	1	0	1				
A		1	1				
P							
E							
R							

Why fill the entire matrix?

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅	**0**	1					
P	1	**0**	1				
A		1	**1**	2			
P			2	**2**			
E							
R							

Why fill the entire matrix?

Universität Zürich[UZH]

	ø	P	L	A	N	E	T
ø	**0**	1					
P	1	**0**	1				
A		1	**1**	2			
P			2	**2**	3		
E				3	**3**		
R							

Why fill the entire matrix?

Universität Zürich[UZH]

	∅	**P**	**L**	**A**	**N**	**E**	**T**
∅	**0**	1					
P	1	**0**	1				
A		1	**1**	2			
P			2	**2**	3		
E				3	**3**	4	
R					4	**4**	

Why fill the entire matrix?

Universität Zürich[UZH]

	∅	P	L	A	N	E	T
∅	**0**	1					
P	1	**0**	1				
A		1	**1**	2			
P			2	**2**	3		
E				3	**3**	4	
R					4	**4**	**5**

:-(

Levenshtein distance

Universität Zürich[UZH]

- Why a fail?
  - locally motivated (greedy) optimization runs a danger of local optima
  - every intermediate step has an influence on later steps
  - its influence cannot be assessed locally
- Which is why
  - the global solution is to fill the whole matrix without fixing any decisions
  - at the end it will be clear, which path is optimal overall

Bottom-up

Universität Zürich[UZH]

```
from collections import defaultdict


def lev(xs, ys):
    matrix = defaultdict(lambda: defaultdict(int))

    for i, x in enumerate([None] + list(xs)):
        for j, y in enumerate([None] + list(ys)):
            if i > 0 and j > 0:
                replCost = (0 if x == y else 1) + matrix[i-1][j-1]
                insCost = matrix[i-1][j] + 1
                delCost = matrix[i][j-1] + 1
                matrix[i][j] = min(replCost, insCost, delCost)
            elif i > 0:
                matrix[i][j] = i
            else:
                matrix[i][j] = j
    return matrix[i][j]

print lev("planet", "paper")
```

complexity: $O(n \cdot m)$

Top-down

Universität Zürich[UZH]

```
def lev(xs, ys):
    if (xs and ys):
        replCost = lev(xs[:-1], ys[:-1]) +
                   (0 if xs[-1] == ys[-1] else 1)
        delCost = lev(xs[:-1], ys) + 1
        insCost = lev(xs, ys[:-1]) + 1

        return min(replCost, insCost, delCost)
    elif (xs):
        return len(xs)
    else:
        return len(ys)

print lev("planet", "paper")    # 3
```

- computes intermediate results repeatedly
- **exponential** complexity

Top-down + Memoisation

Universität Zürich[UZH]

```
def lev(xs, ys):
    global matrix

    if (not matrix[xs][ys]): # recompute if needed
        if (xs and ys):
            replCost = lev(xs[:-1], ys[:-1]) + (0  if xs[-1] == ys[-1] else 1)
            delCost = lev(xs[:-1], ys) + 1
            insCost = lev(xs, ys[:-1]) + 1

            matrix[xs][ys] = min(replCost, insCost, delCost)
        elif (xs):
            matrix[xs][ys] = len(xs)
        else:
            matrix[xs][ys] = len(ys)

    return matrix[xs][ys]
```

complexity: $O(n \cdot m)$

```
from collections import defaultdict
matrix = defaultdict( lambda: defaultdict(int))

print lev("planet", "paper") # 3
```

Contents

Universität Zürich[UZH]

- Levenshtein distance
- Tagging - Viterbi
- Sentence alignment

My walk was awesome *(Lecture 7)*

Universität Zürich[UZH]

$p(t \mid \text{“my”}) = \{ \text{‘pron’}: 0.99, \ldots \}$

$p(t \mid \text{“walk”}) = \{ \text{‘verb’}: 0.8, \text{‘noun’}: 0.19, \ldots \}$

$p(t \mid \text{“was”}) = \{ \text{‘verb’}: 0.92, \ldots \}$

$p(t \mid \text{“awesome”}) = \{ \text{‘adj’}: 0.99, \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘<s>’}) = \{ \text{‘noun’}: 0.35, \text{‘pron’}: 0.3 \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘pron’}) = \{ \text{‘verb’}: 0.3, \text{‘noun’}: 0.35, \text{‘adj’}: 0.3 \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘verb’}) = \{ \text{‘adj’}: 0.2, \text{‘noun’}: 0.15, \text{‘verb’}: 0.01, \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘noun’}) = \{ \text{‘verb’}: 0.3, \text{‘noun’}: 0.2, \ldots \}$

Best tag for ‘walk’:

$p(\text{‘verb’} \mid \text{‘walk’}) \cdot p(\text{‘verb’} \mid \text{‘pron’}) = 0.8 \cdot 0.3 = 0.24$

$p(\text{‘noun’} \mid \text{‘walk’}) \cdot p(\text{‘noun’} \mid \text{‘pron’}) = 0.19 \cdot 0.35 = 0.0665$

My walk was awesome *(Lecture 7)*

Universität Zürich[UZH]

$p(t \mid \text{“my”}) = \{ \text{‘pron’}: 0.99, \ldots \}$

$p(t \mid \text{“walk”}) = \{ \text{‘verb’}: 0.8, \text{‘noun’}: 0.19, \ldots \}$

$p(t \mid \text{“was”}) = \{ \text{‘verb’}: 0.92, \ldots \}$

$p(t \mid \text{“awesome”}) = \{ \text{‘adj’}: 0.99, \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘<s>’}) = \{ \text{‘noun’}: 0.35, \text{‘pron’}: 0.3 \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘pron’}) = \{ \text{‘verb’}: 0.3, \text{‘noun’}: 0.35, \text{‘adj’}: 0.3 \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘verb’}) = \{ \text{‘adj’}: 0.2, \text{‘noun’}: 0.15, \text{‘verb’}: 0.01, \ldots \}$

$p(t_i \mid t_{i-1} = \text{‘noun’}) = \{ \text{‘verb’}: 0.3, \text{‘noun’}: 0.2, \ldots \}$

Best tag for ‘walk’:

$p(\text{‘verb’} \mid \text{‘walk’}) \cdot p(\text{‘verb’} \mid \text{‘pron’}) = 0.8 \cdot 0.3 = 0.24$

$p(\text{‘noun’} \mid \text{‘walk’}) \cdot p(\text{‘noun’} \mid \text{‘pron’}) = 0.19 \cdot 0.35 = 0.0665$

But ‘walk’ as ‘verb’ brings down the likelihood of the whole sequence:

$p(\text{‘pron } \mathbf{verb} \text{ verb adj’} \mid \text{‘my walk was awesome’}) =$

$p(\text{‘pron’} \mid \text{‘my’}) \cdot \mathbf{p(\text{‘verb’} \mid \text{‘walk’})} \cdot p(\text{‘verb’} \mid \text{‘was’}) \cdot p(\text{‘adj’} \mid \text{‘awesome’}) \cdot$
$p(\text{‘pron’} \mid \text{‘<s>’}) \cdot p(\text{‘verb’} \mid \text{‘pron’}) \cdot \mathbf{p(\text{‘verb’} \mid \text{‘verb’})} \cdot p(\text{‘adj’} \mid \text{‘verb’}) =$
$0.99 \cdot \mathbf{0.8} \cdot 0.92 \cdot 0.99 \cdot 0.3 \cdot 0.3 \cdot \mathbf{0.01} \cdot 0.2 = \mathbf{0.00013..}$

$p(\text{‘pron } \mathbf{noun} \text{ verb adj’} \mid \text{‘my walk was awesome’}) =$

$p(\text{‘pron’} \mid \text{‘my’}) \cdot \mathbf{p(\text{‘noun’} \mid \text{‘walk’})} \cdot p(\text{‘verb’} \mid \text{‘was’}) \cdot p(\text{‘adj’} \mid \text{‘awesome’}) \cdot$
$p(\text{‘pron’} \mid \text{‘<s>’}) \cdot p(\text{‘verb’} \mid \text{‘pron’}) \cdot \mathbf{p(\text{‘noun’} \mid \text{‘verb’})} \cdot p(\text{‘adj’} \mid \text{‘verb’}) =$
$0.99 \cdot \mathbf{0.2} \cdot 0.92 \cdot 0.99 \cdot 0.3 \cdot 0.3 \cdot \mathbf{0.15} \cdot 0.2 = \mathbf{0.00049..}$

Tagging

Universität Zürich[UZH]

- Sentence with n words given
- One of m tags can be assigned to each word
- Task: find the most likely tag sequence
- Naive solution:
  - generate all possible tag sequences
  - select the one with the highest probability
- Complexity:
  - there is m^n possible tag sequences
  - $\in O(m^n)$, exponential on sentence length

Tagging

Universität Zürich[UZH]

- $n = 20$ words in a sentence
- each word has one of $m = 50$ tags
- number of different ways of tagging the sentence: $m^n = 9.537 \times 10^{33}$
  - 3.2×10^{20} years
  - earth is only $\sim 4.5 \times 10^{9}$ years old

Tagging

Universität Zürich[UZH]

Bi-gram tagger:

- for each word in the sentence
- find the most likely tag:

$$t = \text{argmax}_{t'} \; p(w_i|t') \cdot p(t'|t_{i-1})$$

- starting from the beginning

Tagging

Universität Zürich[UZH]

Bi-gram tagger:

- for each word in the sentence
- find the most likely tag:

$$t = \text{argmax}_{t'} \; p(w_i|t') \cdot p(t'|t_{i-1})$$

- starting from the beginning

- problem: local optima
  - the currently most likely tag can lead to a less likely tag sequence overall
  - we want the algorithm to not fail because of that

Viterbi Algorithm

Universität Zürich[UZH]

$$\boldsymbol{t} = \text{argmax}_{\boldsymbol{t'}}\ p(\boldsymbol{w}|\boldsymbol{t'}) = \text{argmax}_{\boldsymbol{t'}}\ \Pi_i\, p(w_i|t_i') \cdot p(t_i'|t_{i-1})$$

- For calculating $P(t_i|\ t_{i-1})$ only have to go over 1 previous tag
- Dynamic programming:
  - sub-tasks: probabilities for tag sequences from *1* to *i-1*
  - approach: for each candidate of the current tag find optimal predecessor
    - in other words -- let us not decide on a tag, let us evaluate all the options and decide in the end

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det				
Adj				
Verb				
Noun				

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)			
Adj	p(The | Adj) × p(Adj | <s>)			
Verb	p(The | Verb) × p(Verb | <s>)			
Noun	p(The | Noun) × p(Noun | <s>)			

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev		
Adj	p(The | Adj) × p(Adj | <s>)			
Verb	p(The | Verb) × p(Verb | <s>)			
Noun	p(The | Noun) × p(Noun | <s>)			

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev		
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev		
Verb	p(The | Verb) × p(Verb | <s>)			
Noun	p(The | Noun) × p(Noun | <s>)			

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev		
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev		
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev		
Noun	p(The | Noun) × p(Noun | <s>)			

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev		
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev		
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev		
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev		

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev		
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev		
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev		

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev		
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev		

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev		

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	p(soldiers | Verb) × p(Verb | Noun) × prev
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	p(soldiers | Verb) × p(Verb | Noun) × prev
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	p(soldiers | Noun) × p(Noun | Verb) × prev

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	p(soldiers | Verb) × p(Verb | Noun) × prev
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	p(soldiers | Noun) × p(Noun | Verb) × prev

Viterbi Algorithm

Universität Zürich[UZH]

	The	complex	houses	soldiers
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	p(soldiers | Verb) × p(Verb | Noun) × prev
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	p(soldiers | Noun) × p(Noun | Verb) × prev

Viterbi Algorithm

Universität Zürich[UZH]

	The **Det**	complex **Noun**	houses **Verb**	soldiers **Noun**
Det	p(The | Det) × p(Det | <s>)	p(complex | Det) × p(Det | Verb) × prev	p(houses | Det) × p(Det | Verb) × prev	p(soldiers | Det) × p(Det | Verb) × prev
Adj	p(The | Adj) × p(Adj | <s>)	p(complex | Adj) × p(Adj | Det) × prev	p(houses | Adj) × p(Adj | Det) × prev	p(soldiers | Adj) × p(Adj | Adj) × prev
Verb	p(The | Verb) × p(Verb | <s>)	p(complex | Verb) × p(Verb | Noun) × prev	p(houses | Verb) × p(Verb | Noun) × prev	p(soldiers | Verb) × p(Verb | Noun) × prev
Noun	p(The | Noun) × p(Noun | <s>)	p(complex | Noun) × p(Noun | Det) × prev	p(houses | Noun) × p(Noun | Adj) × prev	p(soldiers | Noun) × p(Noun | Verb) × prev

Viterbi Algorithm

Universität Zürich[UZH]

$$\boldsymbol{t} = \text{argmax}_{\boldsymbol{t'}}\ p(\boldsymbol{w}|\boldsymbol{t'}) = \text{argmax}_{\boldsymbol{t'}}\ \Pi_i\, p(w_i|t_i') \cdot p(t_i'|t_{i-1})$$

- For calculating $P(t_i|\ t_{i-1})$ only have to go over 1 previous tag
- total complexity: $O(n)$

Contents

Universität Zürich[UZH]

- Levenshtein distance
- Tagging - Viterbi
- Sentence alignment

Sentence alignment

Universität Zürich[UZH]

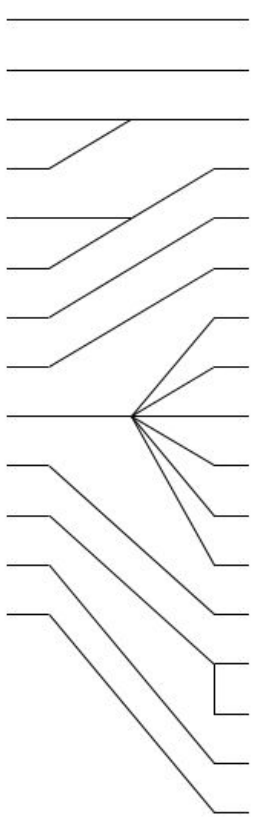
- Given:
  - text (list of sentences) in one (source) language
  - its translation into another (target) language
- Find:
  - for every source text sentence (sentences)

    find the target text sentence (sentences)
    that correspond to it

Example: Text+Berg

Universität Zürich[UZH]

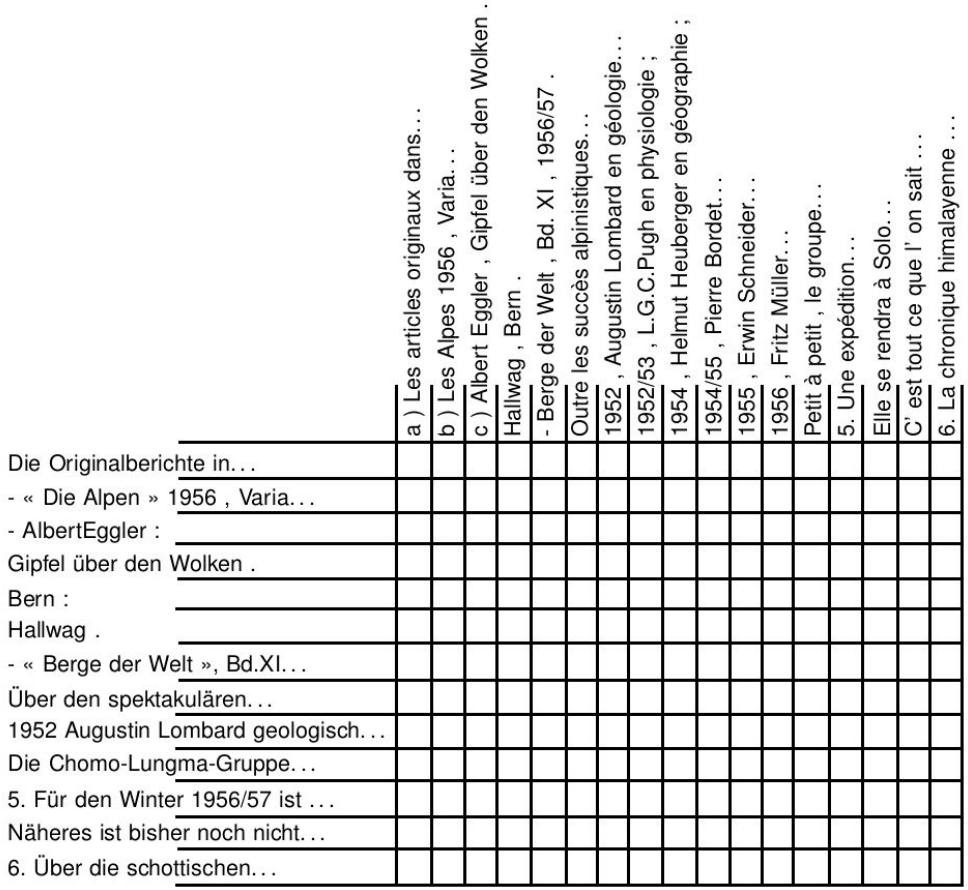
Die Originalberichte in...
- « Die Alpen » 1956 , Varia...
- AlbertEggler :
Gipfel über den Wolken .
Bern :
Hallwag .
- « Berge der Welt », Bd.XI...
Über den spektakulären...
1952 Augustin Lombard geologisch...
Die Chomo-Lungma-Gruppe...
5. Für den Winter 1956/57 ist ...
Näheres ist bisher noch nicht...
6. Über die schottischen...



a) Les articles originaux dans...
b) Les Alpes 1956 , Varia...
c) Albert Eggler , Gipfel über den Wolken .
Hallwag , Bern .
- Berge der Welt , Bd. XI , 1956/57 .
Outre les succès alpinistiques...
1952 , Augustin Lombard en géologie...
1952/53 , L.G.C.Pugh en physiologie ;
1954 , Helmut Heuberger en géographie ;
1954/55 , Pierre Bordet...
1955 , Erwin Schneider...
1956 , Fritz Müller...
Petit à petit , le groupe...
5. Une expédition...
Elle se rendra à Solo...
C' est tout ce que l' on sait ...
6. La chronique himalayenne ...

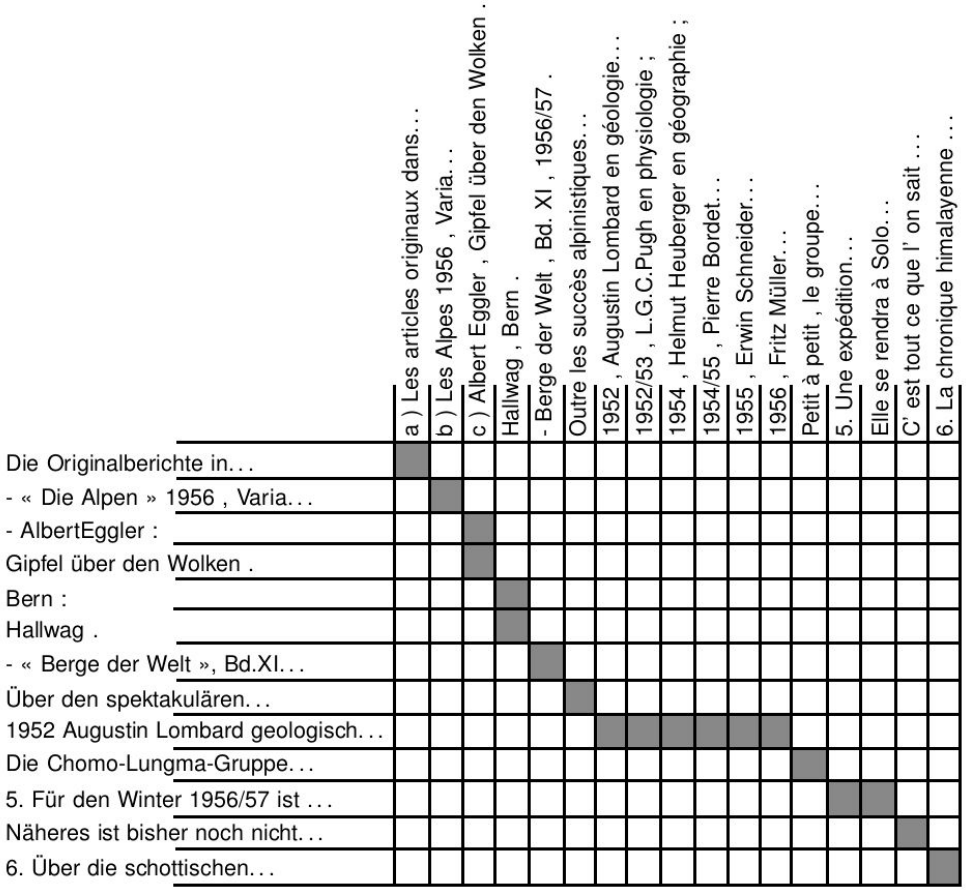
Sentence alignment

- Similar to Lev distance/LCS:



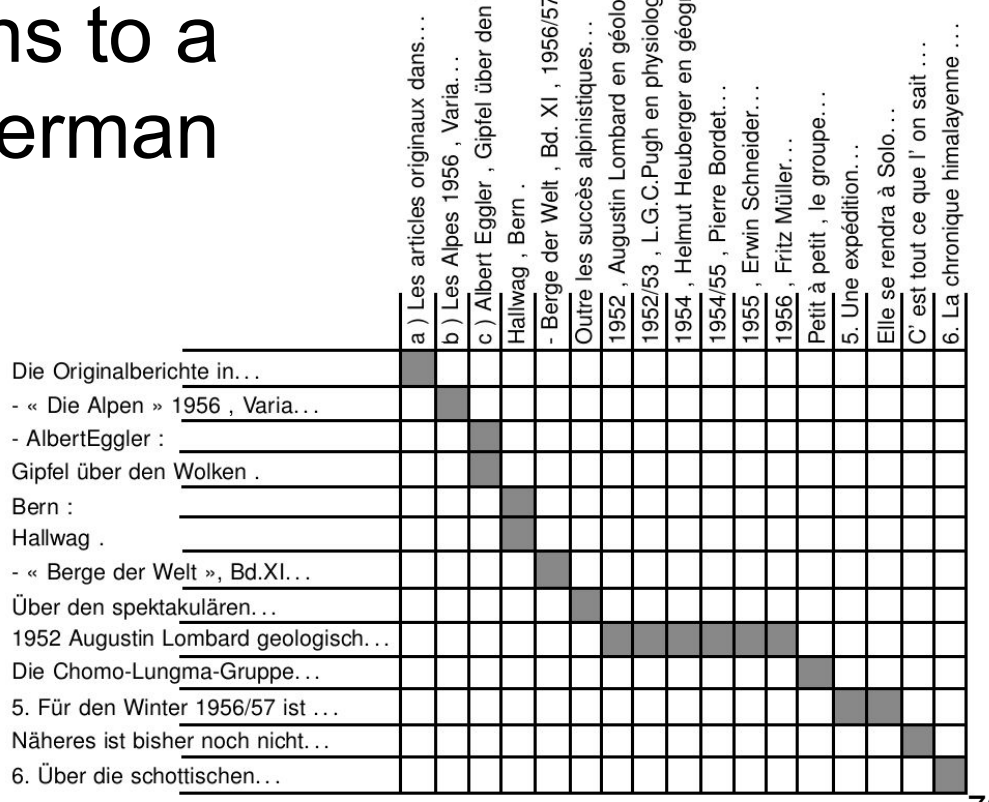
Sentence alignment

- Alignment shown on the matrix:



Sentence alignment

- Alignment shown on the matrix:
- *3 -- (3,4)* here means that French snt. #3 aligns to a concatenation of German snts. #3 & #4



Sentence alignment via Dynamic Programming

Universität Zürich[UZH]

Alignment likelihood can depend on

- sentence length (Gale & Church 1993: **Vanilla aligner**)

  https://www.youtube.com/watch?v=_4InyoC3mtQ
- lexical information (Chen 1993)
- both (Moore 2002; Varga et al. 2005: **HunAlign**)
- advanced (Sennrich & Volk 2010: **BleuAlign**)

Vanilla aligner

Universität Zürich[UZH]

- Principle:
  - longer sentences in one language should correspond to longer sentences in another language

- Likelihood/weight per alignment pair
  - depends on the length ratio of the pair
  - is multiplied by fixed "penalty" values for 1-0/1-1/...

We see...

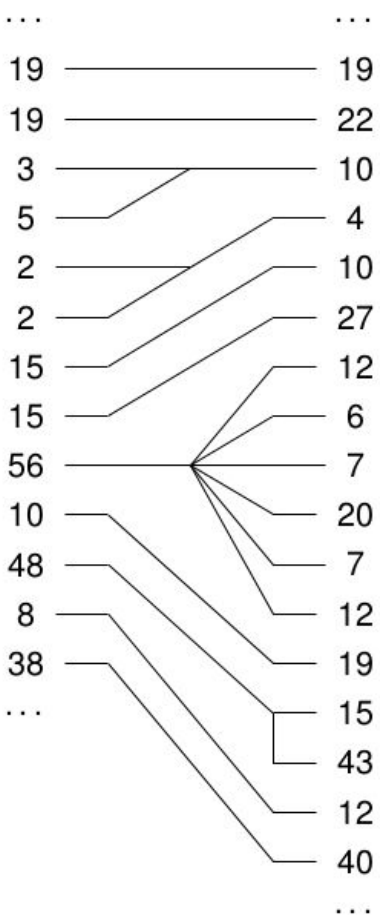
...		...	
Die Originalberichte in...	19	19	a) Les articles originaux dans...
- « Die Alpen » 1956 , Varia...	19	22	b) Les Alpes 1956 , Varia...
- AlbertEggler :	3	10	c) Albert Eggler , Gipfel über den Wolken .
Gipfel über den Wolken .	5	4	Hallwag , Bern .
Bern :	2	10	- Berge der Welt , Bd. XI , 1956/57 .
Hallwag .	2	27	Outre les succès alpinistiques...
- « Berge der Welt », Bd.XI...	15	12	1952 , Augustin Lombard en géologie...
Über den spektakulären...	15	6	1952/53 , L.G.C.Pugh en physiologie ;
1952 Augustin Lombard geologisch...	56	7	1954 , Helmut Heuberger en géographie ;
Die Chomo-Lungma-Gruppe...	10	20	1954/55 , Pierre Bordet...
5. Für den Winter 1956/57 ist ...	48	7	1955 , Erwin Schneider...
Näheres ist bisher noch nicht...	8	12	1956 , Fritz Müller...
6. Über die schottischen...	38	19	Petit à petit , le groupe...
...		15	5. Une expédition...
		43	Elle se rendra à Solo...
		12	C' est tout ce que l' on sait ...
		40	6. La chronique himalayenne ...
		...	

Vanilla aligner sees...

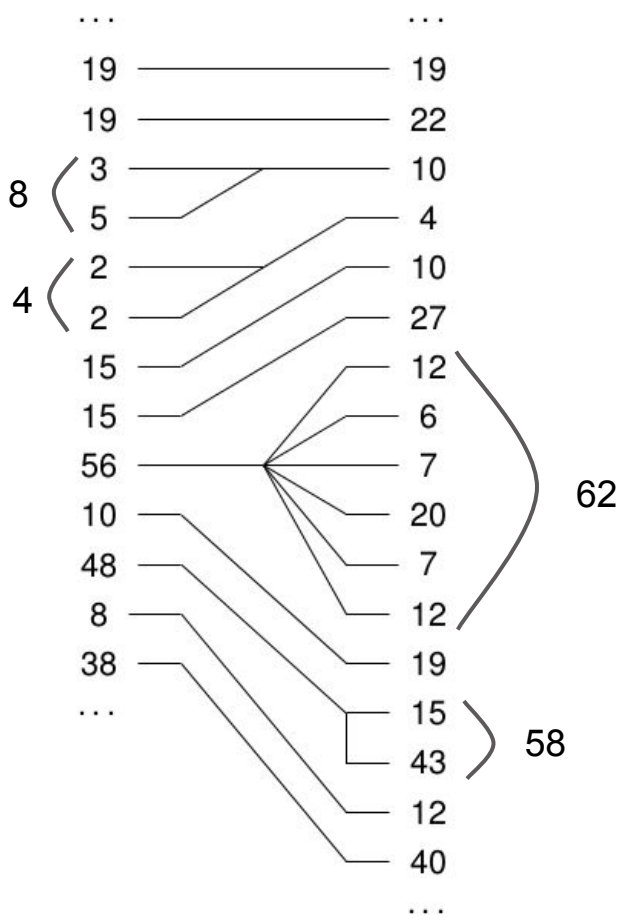
Universität Zürich[UZH]

...	...
19	19
19	22
3	10
5	4
2	10
2	27
15	12
15	6
56	7
10	20
48	7
8	12
38	19
...	15
	43
	12
	40
	...

Vanilla aligner aligns:



Vanilla aligner aligns:



Lecture 11: Dynamic Programming and NLP

PCL II, CL, UZH

May 11, 2016

Universität Zürich[UZH]