

Lecture 9

Sequence Segmentation

Part 1: Definition, Construction

Machine Learning for Structured Data
Vlad Niculae · LTL, UvA · <https://vene.ro/mlsd>

Sequence Segmentation

1 Definition, Construction

2 Algorithm

3 Extensions

4 Evaluation

Sequence Segmentation

The rod cutting problem: We have a rod of length n units, and we can cut it at every marker. What cuts to make to maximize the total value of the resulting pieces?



Sequence Segmentation

The rod cutting problem: We have a rod of length n units, and we can cut it at every marker. What cuts to make to maximize the total value of the resulting pieces?



Sequence Segmentation

The rod cutting problem: We have a rod of length n units, and we can cut it at every marker. What cuts to make to maximize the total value of the resulting pieces?



DNA/RNA:

A C A G A T T A C C

Word segmentation:

私 は 日 本 語 を 学 習

Sequence Segmentation

The rod cutting problem: We have a rod of length n units, and we can cut it at every marker. What cuts to make to maximize the total value of the resulting pieces?



DNA/RNA:

A C A G A T T A C C

Word segmentation:

私は日本語を学習

Entity Extraction:

Mayor Halsema to visit the University of Amsterdam next Friday

Sequence Segmentation

The rod cutting problem: We have a rod of length n units, and we can cut it at every marker. What cuts to make to maximize the total value of the resulting pieces?



DNA/RNA:

A C A G A T T A C C

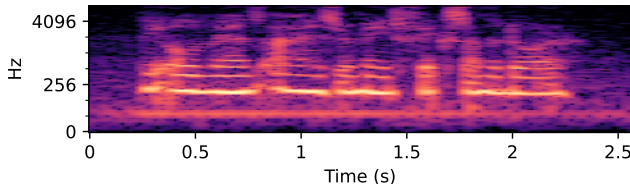
Word segmentation:

私は日本語を学習

Entity Extraction:

Mayor Halsema to visit the University of Amsterdam next Friday

Speech:



Representing and scoring segmentations



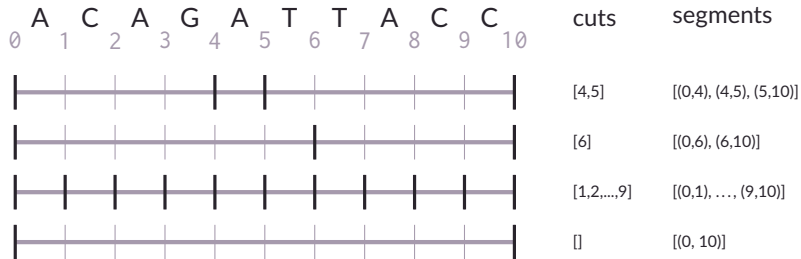
Representing and scoring segmentations



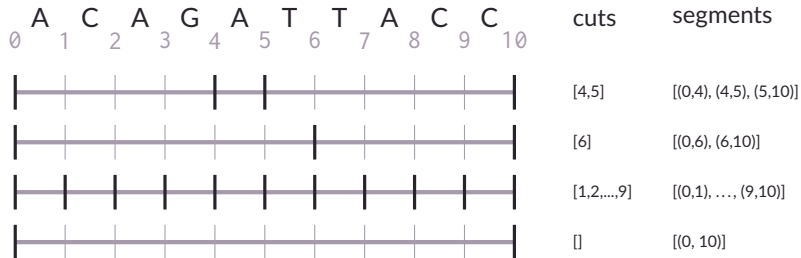
Representing and scoring segmentations



Representing and scoring segmentations

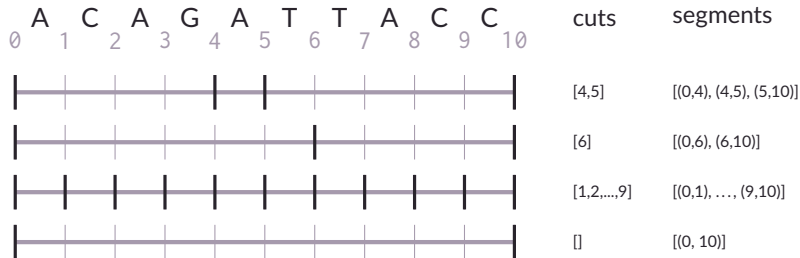


Representing and scoring segmentations







- How many possible segments?

Representing and scoring segmentations



- How many possible segments?
- How many possible *segmentations*?

Representing and scoring segmentations

	A	C	A	G	A	T	T	A	C	C		cuts	segments	score
	0	1	2	3	4	5	6	7	8	9	10			
												[4,5]	[(0,4), (4,5), (5,10)]	$a_{0,4} + a_{4,5} + a_{5,10}$
												[6]	[(0,6), (6,10)]	$a_{0,6} + a_{6,10}$
												[1,2,...,9]	[(0,1), ..., (9,10)]	$a_{0,1} + a_{1,2} + \dots + a_{9,10}$
												[]	[(0, 10)]	$a_{0,10}$

- How many possible segments?
- How many possible *segmentations*?
- Scoring: assign a score to every possible segment (i, j) .

Representing and scoring segmentations

	A	C	A	G	A	T	T	A	C	C										
	0	1	2	3	4	5	6	7	8	9	10									

cuts	segments	score
[4,5]	[(0,4), (4,5), (5,10)]	$a_{0,4} + a_{4,5} + a_{5,10}$
[6]	[(0,6), (6,10)]	$a_{0,6} + a_{6,10}$
[1,2,...,9]	[(0,1), ..., (9,10)]	$a_{0,1} + a_{1,2} + \dots + a_{9,10}$
[]	[(0, 10)]	$a_{0,10}$

- How many possible segments?
- How many possible *segmentations*?
- Scoring: assign a score to every possible segment (i, j) .
- You can visualize this as the “upper triangle” of a $(n+1) \times (n+1)$ matrix:

	0	1	...	n-1	n
0					
1					
...					
n-1					
n					

Lecture 9

Sequence Segmentation

Part 2: Algorithm

Machine Learning for Structured Data
Vlad Niculae · LTL, UvA · <https://vene.ro/mlsd>

Sequence Segmentation

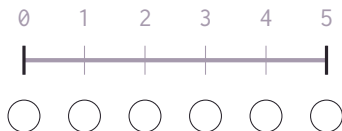
① Definition, Construction

② Algorithm

③ Extensions

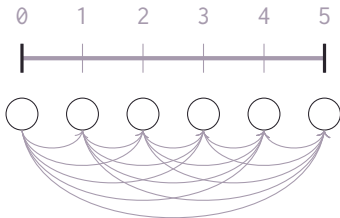
④ Evaluation

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

Dynamic programming: DAG formulation

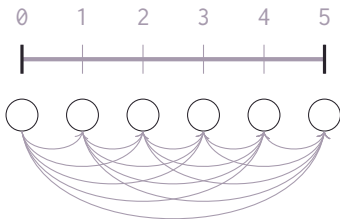


Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Dynamic programming: DAG formulation



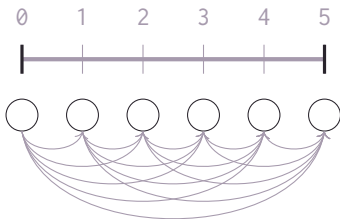
Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

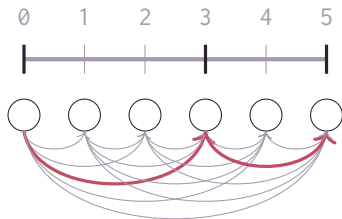
Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Any path from 0 to n corresponds
to a segmentation of the sequence.

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

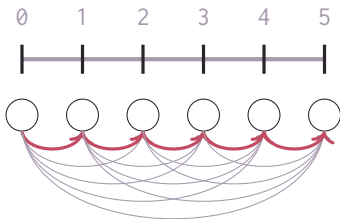
Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Any path from 0 to n corresponds
to a segmentation of the sequence.

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

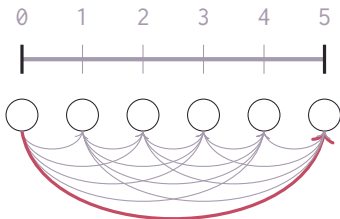
Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Any path from 0 to n corresponds
to a segmentation of the sequence.

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

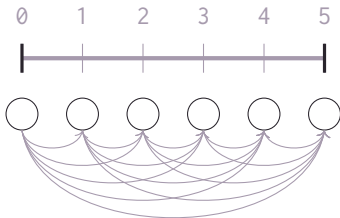
Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Any path from 0 to n corresponds
to a segmentation of the sequence.

Dynamic programming: DAG formulation



Nodes: one per fencepost. $V = \{0, 1, \dots, n\}$.

Edges: one per segment.

$$E = \{(i, j) : 0 \leq i < j \leq n\}.$$

Topologic order?

Any path from 0 to n corresponds to a segmentation of the sequence.

Viterbi for segmentation

input: segment scores $\mathbf{A} \in \mathbb{R}^{n \times n}$

Forward: compute recursively

$$m_1 = a_{01}; \pi_1 = 0$$

for $j = 2$ to n **do**

$$m_j \leftarrow \max_{0 \leq i < j} m_i + a_{ij}$$

$$\pi_j \leftarrow \arg \max_{0 \leq i < j} m_i + a_{ij}$$

$$f^* = m_n$$

Backward: follow backpointers

$$\mathbf{y}^* = []; j \leftarrow n$$

while $j > 0$ **do**

$$\mathbf{y}^* = [(\pi_j, j)] + \mathbf{y}^*$$

$$j = \pi_j$$

Analogously, we can obtain a *Forward* algorithm for $\log Z$: exercise for you.

Lecture 9

Sequence Segmentation

Part 3: Extensions

Machine Learning for Structured Data
Vlad Niculae · LTL, UvA · <https://vene.ro/mlsd>

Sequence Segmentation

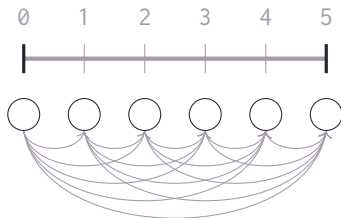
① Definition, Construction

② Algorithm

③ Extensions

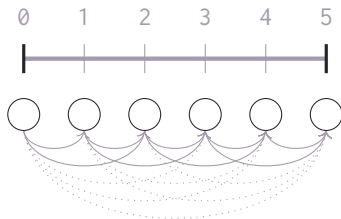
④ Evaluation

Extension 1: bounded segment length



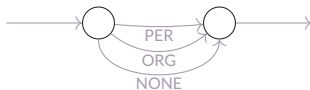
- can be much faster if we limit segment lengths to $L \ll n$.
- in terms of the DAG: discard edges ij where $j - i > L$
- exercise: how does this impact the complexity of Viterbi?

Extension 1: bounded segment length



- can be much faster if we limit segment lengths to $L \ll n$.
- in terms of the DAG: discard edges ij where $j - i > L$
- exercise: how does this impact the complexity of Viterbi?

Extension 2: labeled segments



- each segment also receives a label (e.g., PERSON, ORGANIZATION, NONE...)
- the labels are independent given the cuts: for any two nodes in the DAG, we only need to pick the best edge between them.

Extension 3: labeled + transitions

- drawing inspiration from sequence tagging: what if we want a reward/penalty for consecutive PERSON→ORGANIZATION segments?
- labels no longer independent given cuts.
- still solvable via DP, but must keep track of transitions.
- essentially a combination of the sequence tagging DAG and the segmentation DAG.

Lecture 9

Sequence Segmentation

Part 4: Evaluation

Machine Learning for Structured Data
Vlad Niculae · LTL, UvA · <https://vene.ro/mlsd>

Sequence Segmentation

- 1 Definition, Construction
- 2 Algorithm
- 3 Extensions
- 4 Evaluation**

Evaluation

Well, what would we do in the unstructured case?

- Per-class Precision:

What fraction of the test cases predicted as class c are correctly predicted?

$$P_{(c)} = \frac{\sum_{i=1}^N \mathbb{I}[y^{(i)} = c \& \hat{y}^{(i)} = c]}{\sum_{i=1}^N \mathbb{I}[\hat{y}^{(i)} = c]}$$

- Per-class Recall:

What fraction of the test cases from class c are correctly predicted?

$$R_{(c)} = \frac{\sum_{i=1}^N \mathbb{I}[y^{(i)} = c \& \hat{y}^{(i)} = c]}{\sum_{i=1}^N \mathbb{I}[y^{(i)} = c]}$$

- Per-class F_1 score: $F_{1,(c)} = 2(P_c^{-1} + R_c^{-1})^{-1}$

Balances precision and recall (harmonic mean).

Binary clf.: usual (and intuitive) to only compute P/R/F for the “positive” class.

Evaluation

Another way to think about P/R/F:

For class c ,

TP	TN
FP	FN

- $TP_{(c)}$: true positives: $y^{(i)} = c$ and $\hat{y}^{(i)} = c$.
- $FP_{(c)}$: false positives: $y^{(i)} \neq c$ and $\hat{y}^{(i)} = c$.
- $FN_{(c)}$: false negatives: $y^{(i)} = c$ and $\hat{y}^{(i)} \neq c$.
- $TN_{(c)}$: true negatives: $y^{(i)} \neq c$ and $\hat{y}^{(i)} \neq c$.

Then,

$$P_{(c)} = \frac{TP_{(c)}}{TP_{(c)} + FP_{(c)}} \quad R_{(c)} = \frac{TP_{(c)}}{TP_{(c)} + FN_{(c)}} \quad \text{Acc}_{(c)} = \frac{1}{N} \sum_c TP_{(c)} + TN_{(c)}$$

Evaluation

Macro-average P (or R,F) score over classes

- weighted (by class frequency): denoting $N_c = \sum_{i=1}^N \mathbb{I}[y^{(i)} = c]$,

$$\sum_{c=1}^K \frac{N_c}{N} P_{(c)}$$

- unweighted:

$$\frac{1}{K} \sum_{c=1}^K P_{(c)}$$

Micro-average:

First add up TP, FP, FN, TN over classes.
Then compute P/R/F for this “total” class.

Be explicit and thoughtful!

For instance:

many rare classes that are very easy to recognize -> unweighted F_1 would give an overly optimistic summary close to 1.

class proportions will change at test time or performance should be equally good on all classes, unweighted can make more sense!

Structured evaluation: Segmentations



Gold segments: $y = [(0, 3), (3, 5), (5, 6), (6, 11)]$

Predicted: $\hat{y} = [(0, 4), (4, 5), (5, 11)]$

The number of pred. and gold segments differ.

We could interpret this as binary clf of cuts, and evaluate cut accuracy or P/R/F.

Not a great idea:

above, we correctly got the positive cut at 5.
(and correctly said no cut at 1,2,...)

But no correct segments were returned!

Structured evaluation: Segmentations



Gold segments: $y = [(0, 3), (3, 5), (5, 6), (6, 11)]$

Predicted: $\hat{y} = [(0, 4), (4, 5), (5, 11)]$

The number of pred. and gold segments differ.

We could interpret this as binary clf of cuts, and evaluate cut accuracy or P/R/F.

Not a great idea:

above, we correctly got the positive cut at 5.
(and correctly said no cut at 1,2,...)

But no correct segments were returned!

Segment-level P/R/F (Sproat and Emerson, 2003):

True positive segments (appearing both in y and \hat{y}).

False positive segments (in \hat{y} but not in y)

False negative segments (in y but not in \hat{y})

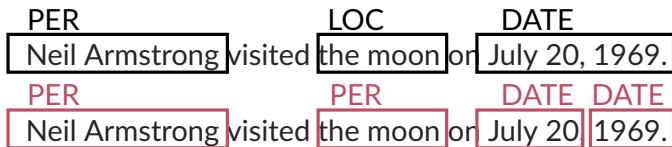
$$P = \frac{TP}{TP+FP} = \frac{\text{n. correctly predicted segments}}{\text{n. predicted segments}}$$

$$R = \frac{TP}{TP+FN} = \frac{\text{n. correctly predicted segments}}{\text{n. gold segments}}$$

For this prediction, both are zero.

More advanced metrics: overlap-aware, or
“out-of-vocabulary” rates on held-out data.

Structured evaluation: Labeled segmentations



Gold segments:

{ (PER, 0, 2), (LOC, 3, 5), (DATE, 6, 11)}

Pred segments:

{ (PER, 0, 2), (PER, 3, 5), (DATE, 6, 9),
(DATE, 9, 11)}

TP = {(PER, 0, 2)}

FP = {(PER, 3, 5), (DATE, 6, 9), (DATE, 9, 11)}

FN = {(LOC, 3, 5), (DATE, 6, 11)}

$$P = 1/1 + 3 = .25 \quad R = 1/1 + 2 = .33 \quad F_1 = .2845$$

This is the standard way to evaluate chunking/NER (Tjong Kim Sang and Buchholz, 2000; Tjong Kim Sang, 2002)

Per-class P/R/F, and adding "unlabeled P/R/F" possible, but not standard.

Note: segment accuracy is not useful: the set TN would contain almost all possible segments.





Historical Notes and References

- The segmentation model is technically also a CRF, often called semi-Markov CRF or semi-CRF attributed (in this form) to Sarawagi and Cohen (2004).
- To the best of my knowledge the first attestation of the Viterbi algorithm in this model is due to Bridle and Sedgwick (1977) (also the person who coined the word “softmax”!) However this conference paper is garbled in the IEEE online archive and can only be found uncorrupted in libraries.
- The segmentation DP is also (unreferenced) one of the teaching examples of DP in the third edition of Cormen et al. (2009) (“rod cutting”).



Summary

- Segmentations of a length- n sequence: $O(2^n)$ possible segmentations, $O(n^2)$ possible segments.
- Dynamic programming gives polynomial-time probabilistic segmentation models.
- Extensions can accommodate maximum lengths, labels, transitions.

References I

-  Bridle, J. and N. Sedgwick (1977). "A method for segmenting acoustic patterns, with applications to automatic speech recognition". In: *ICASSP '77. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2, pp. 656–659.
-  Cormen, Thomas H et al. (2009). *Introduction to algorithms (third edition)*. MIT press.
-  Sarawagi, Sunita and William W Cohen (2004). "Semi-Markov Conditional Random Fields for Information Extraction". In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press.
-  Sproat, Richard and Thomas Emerson (July 2003). "The First International Chinese Word Segmentation Bakeoff". In: *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*. Sapporo, Japan: Association for Computational Linguistics, pp. 133–143.

References II

-  Tjong Kim Sang, Erik F. (2002). "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.
-  Tjong Kim Sang, Erik F. and Sabine Buchholz (2000). "Introduction to the CoNLL-2000 Shared Task Chunking". In: *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.