Trainable Tree Distance and an application to Question Categorisation

Martin Emms

September 3, 2010

QuestionBank

2755 syntactically analysed and semantically categorised questions

HUM
ENTY
NUM
LOC
2755

Cat	Example
HUM	What is the name of the managing director of Apricot Computer ? (WHNP (WP What))(SQ (VBZ is)(NP (NP (DT the)(NN name))(PP (IN of)(NP (NP (DT the)(JJ managing)(NN director)) (PP (IN of)(NP (NNP Apricot)(NNP Computer)))))))(. ?))
ENTY	What does the Peugeot company manufacture ? (SBARQ (WHNP (WP What))(SQ (VBZ does)(NP (DT the)(NNP Peugeot)(NN company))(VP (VB manufacture)))(. ?))
DESC	What did John Hinckley do to impress Jodie Foster ? (SBARQ (WHNP (WP What))(SQ (VBD did)(NP (NNP John)(NNP Hinckley))(VP (VB do) (S (VP (TO to)(VP (VB impress)(NP (NNP Jodie)(NNP Foster)))))))(. ?))
NUM	When was London 's Docklands Light Railway constructed ? (SBARQ (WHADVP (WRB When))(SQ (VBD was)(NP (NP (NNP London)(POS 's))(NNPS Docklands) (JJ Light)(NN Railway))(VP (VBN constructed)))(. ?))
LOC	What country is the biggest producer of tungsten ? (SBARQ (WHNP (WDT What)(NN country))(SQ (VBZ is)(NP (NP (DT the)(JJS biggest)(NN producer)) (PP (IN of)(NP (NN tungsten)))))(. ?))
ABBR	What is the acronym for the rating system for air conditioner efficiency ? (SBARQ (WHNP (WP What))(SQ (VBZ is)(NP (NP (DT the)(NN acronym))(PP (IN for)(NP (NP (DT the)(NN rating) (NN system))(PP (IN for)(NP (NN air)(NN conditioner)(NN efficiency)))))))(. ?))

Data-driven Syntax ⇒ Semantics

- The QuestionBank data is a finite sample of an infinite target function $f : Syn \mapsto Sem$ function.

HUM
ENTY
NUM
LOC
2755

Data-driven Syntax ⇒ Semantics

- The QuestionBank data
HUM
ENTY
NUM
LOC
2755
is a finite sample of an infinite target function $f : Syn \mapsto Sem$ function.

options to get a representation $\hat{f}$ of f $\Big\{$

Data-driven Syntax ⇒ Semantics

- The QuestionBank data

HUM
ENTY
NUM
LOC
2755

 is a finite sample of an infinite target function $f : Syn \mapsto Sem$ function.

options to get a representation $\hat{f}$ of f $\begin{cases}\text{design by hand} \\ \end{cases}$

Data-driven Syntax ⇒ Semantics

- The QuestionBank data

HUM
ENTY
NUM
LOC
2755

 is a finite sample of an infinite target function $f : Syn \mapsto Sem$ function.

options to get a representation $\hat{f}$ of f $\begin{cases}\text{design by hand}\\ \text{data-driven way, using kNN}\end{cases}$

Data-driven Syntax ⇒ Semantics

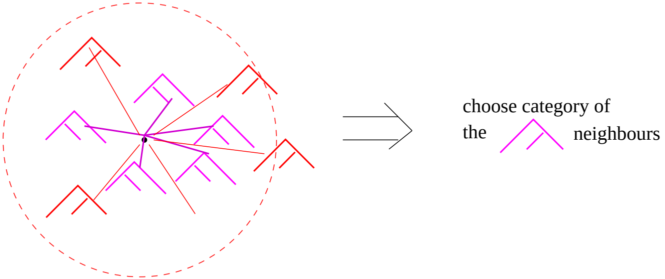
- The QuestionBank data

HUM
ENTY
NUM
LOC
2755

 is a finite sample of an infinite target function $f : \mathit{Syn} \mapsto \mathit{Sem}$ function.

options to get a representation $\hat{f}$ of f $\begin{cases}\text{design by hand}\\ \text{data-driven way, using kNN}\end{cases}$

$$\hat{f}(S) = \mathit{VOTE}(\{\mathit{categories\ of\ k}\ \textbf{nearest neighbours}\ \mathit{of\ S}\ \})$$



choose category of the neighbours

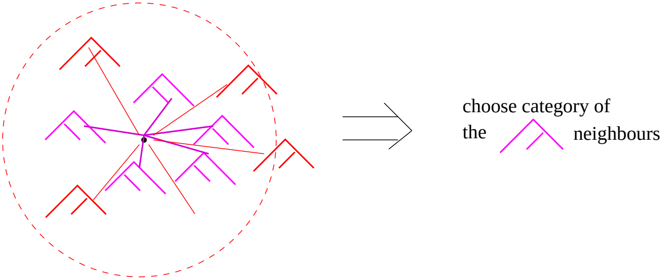
k nearest neighbours for some structure S

Data-driven Syntax ⇒ Semantics

- The QuestionBank data
HUM
ENTY
NUM
LOC
2755
is a finite sample of an infinite target function $f : Syn \mapsto Sem$ function.

options to get a representation $\hat{f}$ of f $\begin{cases}\text{design by hand}\\ \text{data-driven way, using kNN}\end{cases}$

$$\hat{f}(S) = VOTE(\{categories\ of\ k\ \textbf{nearest neighbours}\ of\ S\ \})$$



choose category of the neighbours

k nearest neighbours for some structure S

- So how to compare trees

Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

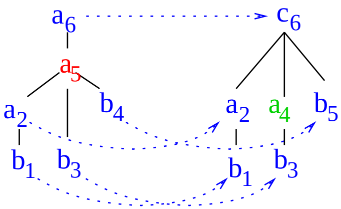
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



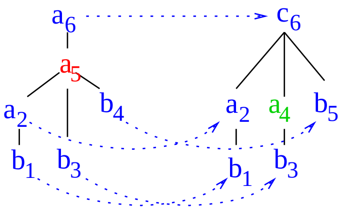
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

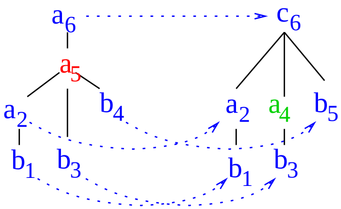
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

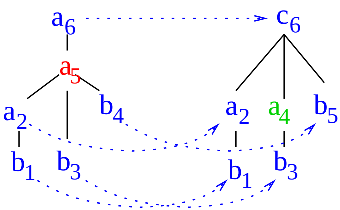
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

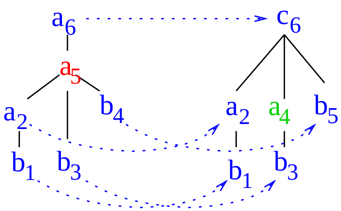
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

match/swaps eg. a_6 goes to c_6

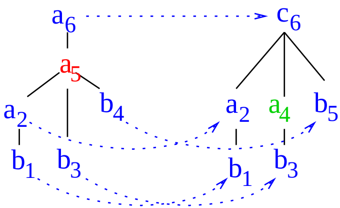
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



given cost table $\mathcal{C}$:

	λ	a	b	c
λ		●	1	●
a	1	0	●	1
b	●	●	0	●
c	●	●	●	●

Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

match/swaps eg. a_6 goes to c_6

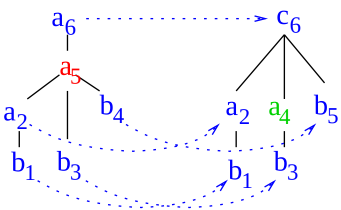
Standard Tree Distance

a partial mapping $\sigma : \mathcal{S} \mapsto \mathcal{T}$ is a Tai mapping iff σ respects left-to-right order and ancestry. Giving costs to mappings leads to

Definition

(*Tree- or Tai-distance*) between $\mathcal{S}$ and $\mathcal{T}$ is the cost of **the least-costly Tai mapping** from $\mathcal{S}$ to $\mathcal{T}$

example Tai mapping σ:



given cost table $\mathcal{C}$:

	λ	*a*	*b*	*c*
λ		●	1	●
a	1	0	●	1
b	●	●	0	●
c	●	●	●	●

Cost of a mapping given by cost of

deletions eg. a_5 has no image

insertions eg. a_4 has no source

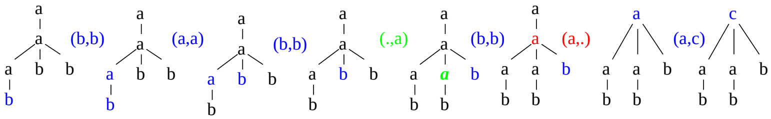
match/swaps eg. a_6 goes to c_6

total cost of σ is sum on non-zero costs

$\mathcal{C}[\lambda][a] + \mathcal{C}[a][\lambda] + \mathcal{C}[a][c]$
= 3
this is also a least cost mapping for this table

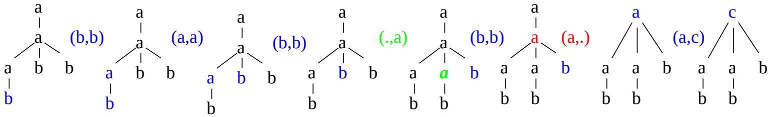
Stochastic version of Tree Distance

- A Tai-mapping can also be serialised in a sequence of edit operations, called an edit-script:



Stochastic version of Tree Distance

- A Tai-mapping can also be serialised in a sequence of edit operations, called an edit-script:

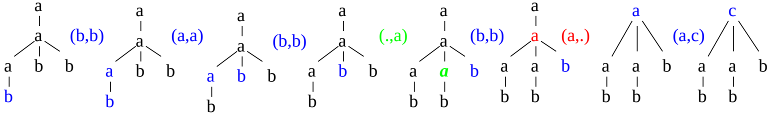


- assuming a prob distribution p on edit-script components $e \in (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})$, can define an overall edit-script probability as

$$P(e_1 \ldots e_n) = p(e_1) \times \ldots \times p(e_n)$$

Stochastic version of Tree Distance

- A Tai-mapping can also be serialised in a sequence of edit operations, called an edit-script:



- assuming a prob distribution p on edit-script components $e \in (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\})$, can define an overall edit-script probability as

$$P(e_1 \ldots e_n) = p(e_1) \times \ldots \times p(e_n)$$

- leading to

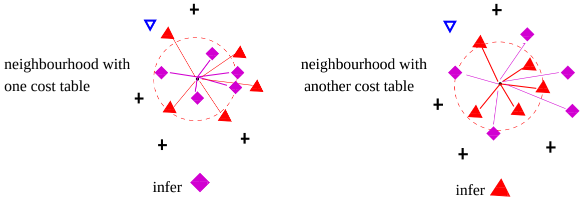
Definition

(*All-paths and Viterbi stochastic Tai distance*)
$\Delta^A(S, T)$ is the sum of the probabilities of all edit-scripts which represent a *Tai*-mapping from S to T;
$\Delta^V(S, T)$ is the probability of the most probable edit-script

Cost adaptation

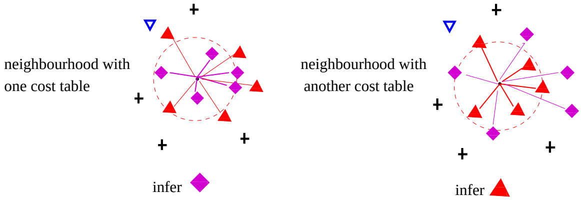
Cost adaptation

- change cost table ⇒ change nearest neighbours ⇒ change categorisation:



Cost adaptation

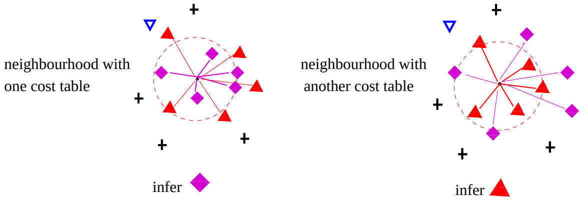
- change cost table ⇒ change nearest neighbours ⇒ change categorisation:



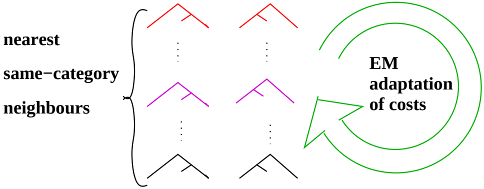
- in scripts between between same-category neighbours *should* have distinctive probs eg. . $P(\textit{who}/\textit{when}) << P(\textit{state}/\textit{country})$.

Cost adaptation

- change cost table ⇒ change nearest neighbours ⇒ change categorisation:


neighbourhood with one cost table
infer ◆
neighbourhood with another cost table
infer ▲

- in scripts between between same-category neighbours *should* have distinctive probs eg. . $P(\textit{who}/\textit{when}) << P(\textit{state}/\textit{country})$.
- IDEA: use Expectation-Maximisation techniques to adapt edit-probs from a corpus of same-category nearest neighbours (cf. HMMs) .



Brute force All-paths EM (infeasible)

Brute force All-paths EM (infeasible)

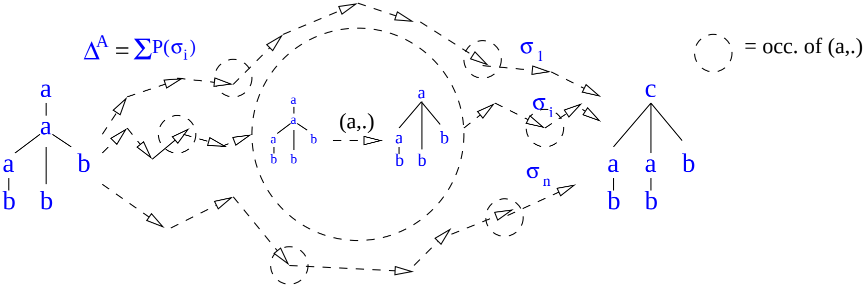
- in principle treat each training pair (S, T) of same-category neighbours as standing for **all** the edit-scripts $\sigma : S \mapsto T$

Brute force All-paths EM (infeasible)

- in principle treat each training pair (S, T) of same-category neighbours as standing for **all** the edit-scripts $\sigma : S \mapsto T$
- derived weighted counts for each edit operation *op* from **all** edit-scipts

Brute force All-paths EM (infeasible)

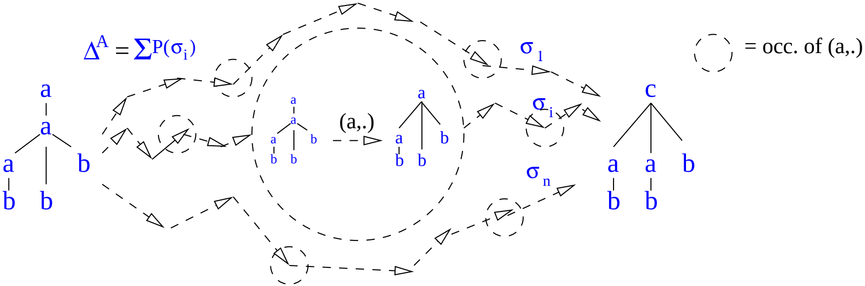
- in principle treat each training pair (S, T) of same-category neighbours as standing for **all** the edit-scripts $\sigma : S \mapsto T$
- derived weighted counts for each edit operation *op* from **all** edit-scipts



$$n_{S,T}(op) = \sum_{\sigma:S\mapsto T} [\frac{P(\sigma)}{\Delta^A(S,T)} \times \#(op \in \sigma)]$$

Brute force All-paths EM (infeasible)

- in principle treat each training pair (S, T) of same-category neighbours as standing for **all** the edit-scripts $\sigma : S \mapsto T$
- derived weighted counts for each edit operation *op* from **all** edit-scipts

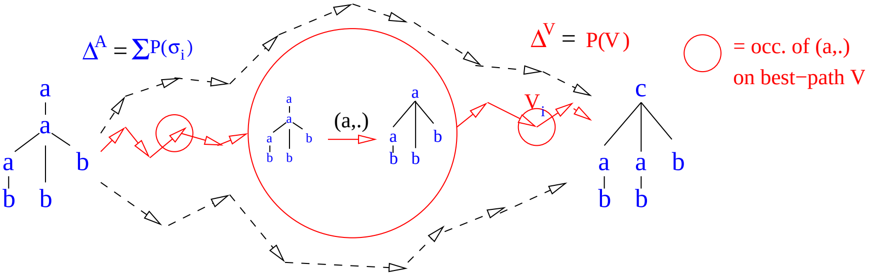


$$n_{S,T}(op) = \sum_{\sigma : S \mapsto T} [\frac{P(\sigma)}{\Delta^A(S,T)} \times \#(op \in \sigma)]$$

- **infeasible**

Viterbi approximation EM^V (feasible)

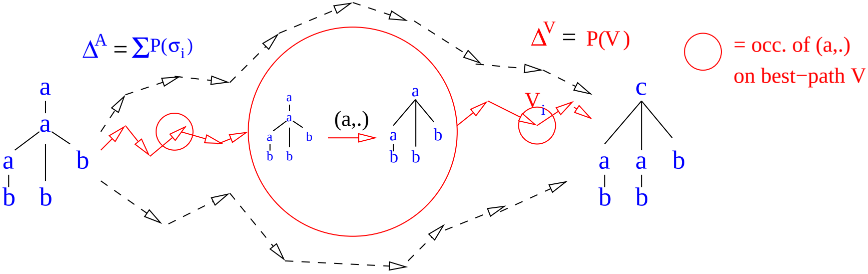
- approximate this by computing counts from only the **best-path** $\mathcal{V}$.



$$n_{(S,T)}(op) = \frac{\Delta^V(S,T)}{\Delta^A(S,T)} \times \#(op \in \mathcal{V})$$

Viterbi approximation EM^V (feasible)

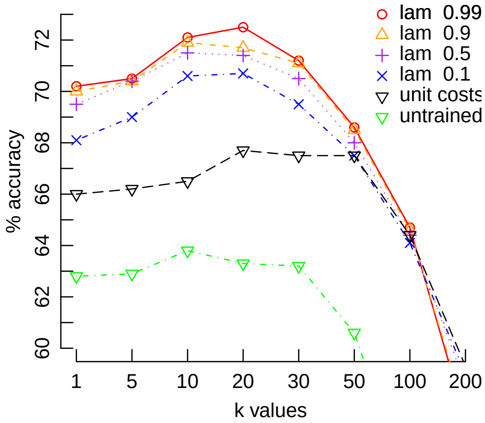
- approximate this by computing counts from only the **best-path** $\mathcal{V}$.



$$n_{(S,T)}(op) = \frac{\Delta^V(S,T)}{\Delta^A(S,T)} \times \#(op \in \mathcal{V})$$

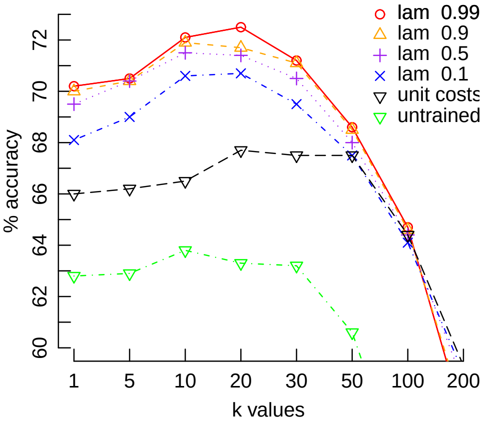
- a dominant best-path has more to say than a weak best-path

Categorisation Results on QuestionBank



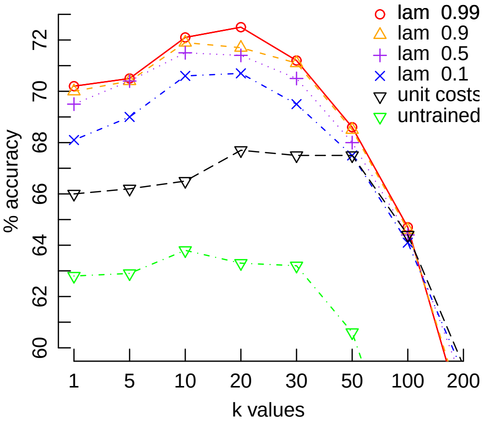
Categorisation Results on QuestionBank

- standard unit-costs ▽, max. 67.7%



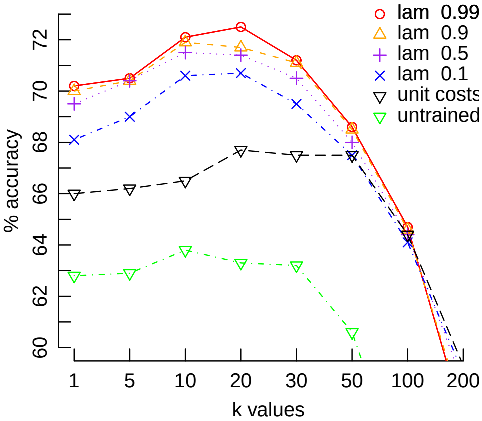
Categorisation Results on QuestionBank

- standard unit-costs ▽, max. 67.7%
- initial stochastic costs ▽ max. 63.8% worse than unit costs



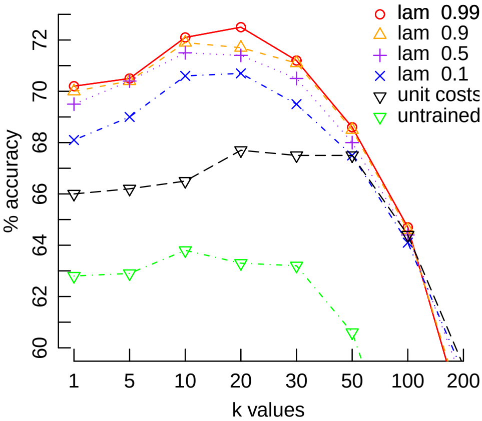
Categorisation Results on QuestionBank

- standard unit-costs
  ▽, max. 67.7%
- initial stochastic costs
  ▽ max. 63.8%
  worse than unit costs
- best *EM*^{V}-adapted costs
  ○, max. 72.5%
  about 5% better than unit-costs
  (▽, max. 67.7%)



Categorisation Results on QuestionBank

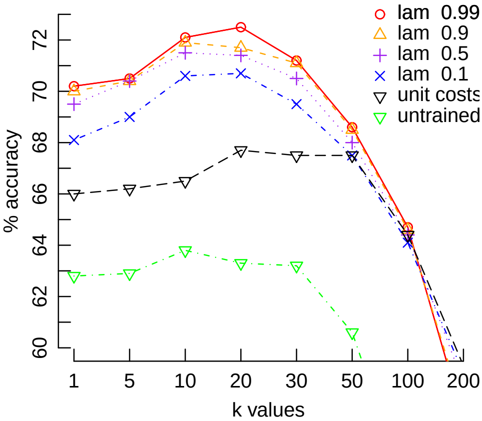
- standard unit-costs
  ▽, max. 67.7%
- initial stochastic costs
  ▽ max. 63.8%
  worse than unit costs
- best EM^V-adapted costs
  ○, max. 72.5%
  about 5% better than unit-costs
  (▽, max. 67.7%)



- Smoothing: learned costs are smoothed by interpolation with a prior $\mathcal{C}_u(d)$ making diag = $d \times$ non-diag:
  $2^{-\mathcal{C}_\lambda[x][y]} = \lambda(2^{-\mathcal{C}[x][y]}) + (1-\lambda)(2^{-\mathcal{C}_u(d)[x][y]})$

Categorisation Results on QuestionBank

- standard unit-costs
  ▽, max. 67.7%
- initial stochastic costs
  ▽ max. 63.8%
  worse than unit costs
- best EM^V-adapted costs
  ○, max. 72.5%
  about 5% better than unit-costs
  (▽, max. 67.7%)



- Smoothing: learned costs are smoothed by interpolation with a prior $\mathcal{C}_u(d)$ making diag = $d \times$ non-diag:
  $2^{-\mathcal{C}_\lambda[x][y]} = \lambda(2^{-\mathcal{C}[x][y]}) + (1-\lambda)(2^{-\mathcal{C}_u(d)[x][y]})$
- Zeroing the diagonal: a final steps zeros the diagonal – a move standardly made in related work on adpative *string distance*