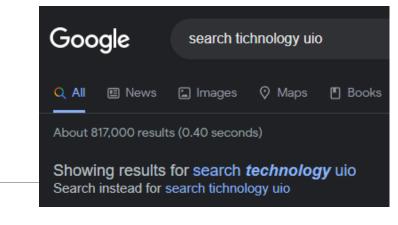
Tries for Approximate String Matching

by H. Shang and T. H. Merrett (1995)

A PRESENTATION BY ANNA AND JONAS





Applications: spell checking, searching and classification

EXACT

exact_match('abc', 'abc') → True exact_match('abc', 'abd') → False

The == operator in Python

```
>>> 'abc' == 'abc'
True
>>> 'abc' == 'abd'
False
```

APPROXIMATE

apprx_match('abc', 'abc', k=1) \rightarrow True apprx_match('abc', 'abd', k=1) \rightarrow True

k determines how approximate the match is

apprx_match('abc', 'abd', k=0) \rightarrow False

			a =	sit	ting			
		1	P =	kit	ten			
	#	K	ı	τ	T	£	N	
#	0	1	2	3	4	5	6	
S	1	1	2	3	4	5	6	
I	2	2	1	2	3	4	5	
7	3	3	2	1	2	3	4	
Τ	4	4	3	2	1	2	3	
I	5	5	4	3	2	2	3	
2	6	6	5	4	3	3	2	
2 5	7	7	6	5	4	4	3	

Edit distance

Number of changes/operations required for one word to become another

"How different are words a and b?"

Operations:

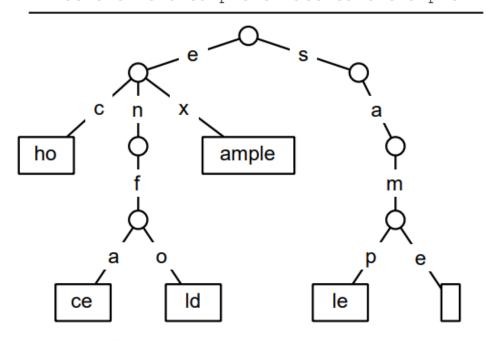
+ Insertion: cot \rightarrow coat

 \Rightarrow Deletion: coat \Rightarrow cot

 $\downarrow \uparrow$ Substitution: coat \rightarrow cost

 \leftarrow Transposition: cost \rightarrow cots

Text:
 echo enfold sample enface same example



Trie

Tree structure

Keys in leaf nodes

Each edge has one character of a key

Compressed because of shared prefixes

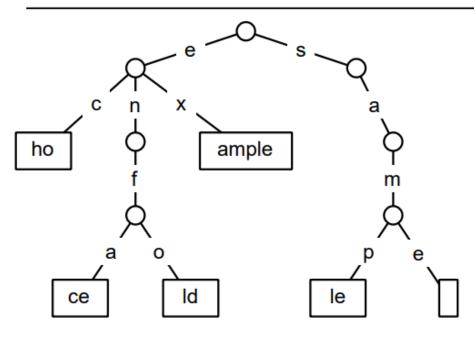
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
đ	4	3	3	2	2	2	2	3

Fαι	t M	licta	nce
Lui	ιu	IISLO	IIICC

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

Traverse trie recursively with DFS (Depth First Search)

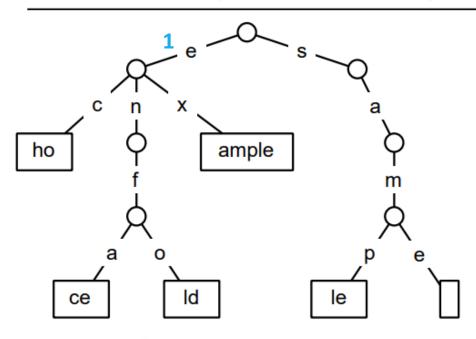
For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:

echo enfold sample enface same example



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	φ	а	С	а	f	В	а	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2 -		

Edit distance

Edit distance with Ukkonen cutoff

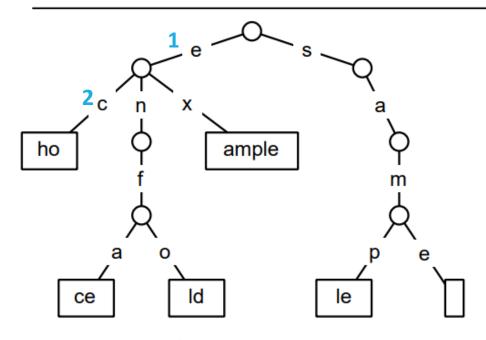
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	φ	а	С	d	f	Ь	đ	f
φ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2 -		

Edit distance

Edit distance with Ukkonen cutoff

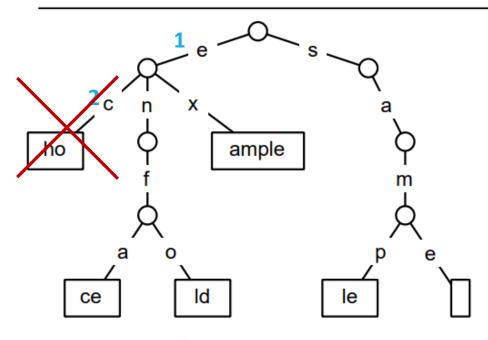
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	٠.	. 1	
$-\alpha$	ıΤ	$\boldsymbol{\alpha}$	istance
LU	ΙL	u	istaiice

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2		

Edit distance with Ukkonen cutoff

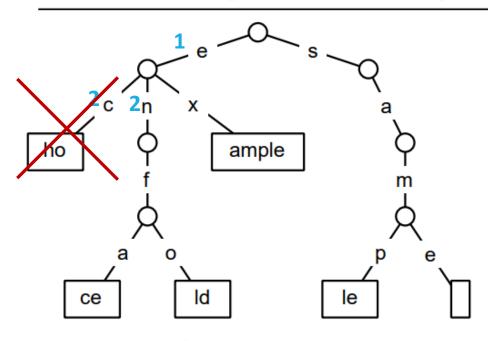
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	٠.				
Hα	ΙŦ	n	ista	n	ഫ
Lu		u	ısta		\sim

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

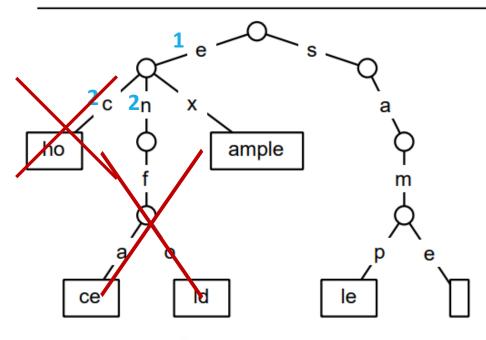
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	b	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
đ	4	3	3	2	2	2	2	3

	.1
FUIT	distance
Luit	uistance

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

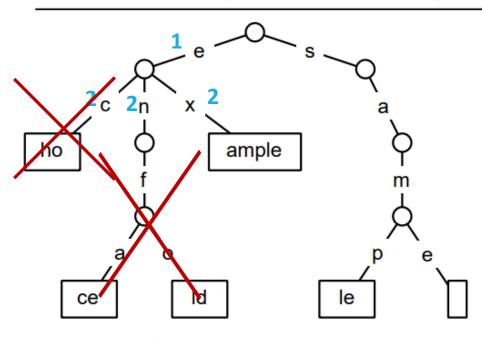
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	b	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

Fαι	t M	licta	nce
Lui	ιu	IISLO	IIICC

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

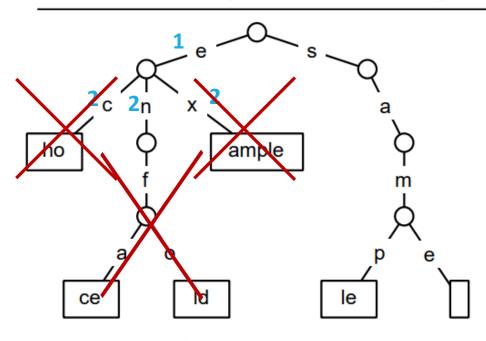
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

Fαι	t M	licta	nce
Lui	ιu	IISLO	IIICC

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

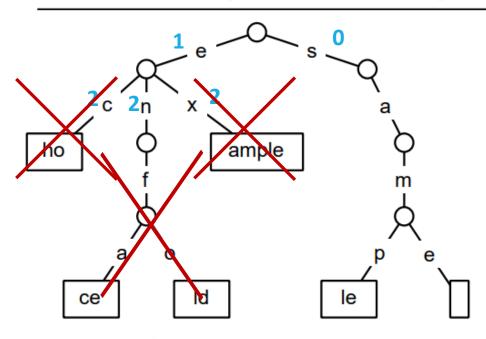
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	φ	а	С	d	f	Ь	đ	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2 -		

Edit distance

Edit distance with Ukkonen cutoff

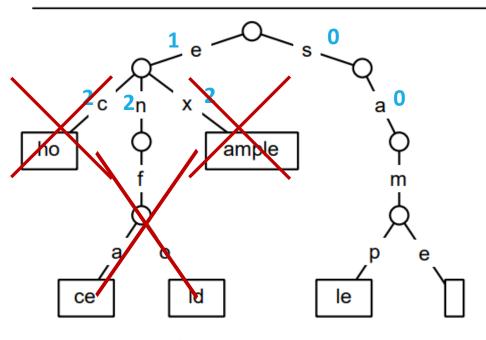
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k = 1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

Fαι	t M	licta	nce
Lui	ιu	IISLO	IIICC

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

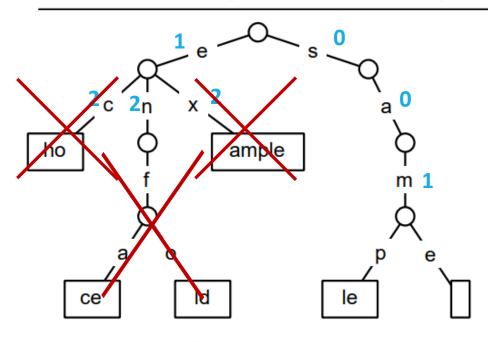
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k=1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
đ	4	3	3	2	2	2	2	3

_							
Ec	И١.	t 1	11	ct	21	ኅɾ	Δ
L	41	ιι	a i	Jι	αı	ı	C

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

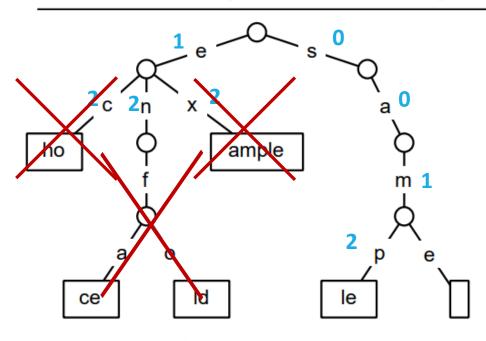
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k=1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	.1
FUIT	distance
Luit	uistance

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

Traverse trie recursively with DFS (Depth First Search)

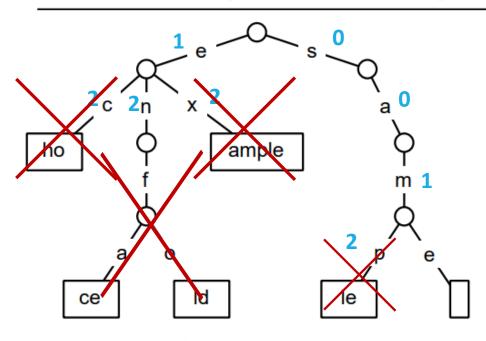
For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k=1

Text:

echo enfold sample enface same example



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	φ	а	С	d	f	Ь	đ	f
φ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2		

Edit distance

Edit distance with Ukkonen cutoff

Traverse trie recursively with DFS (Depth First Search)

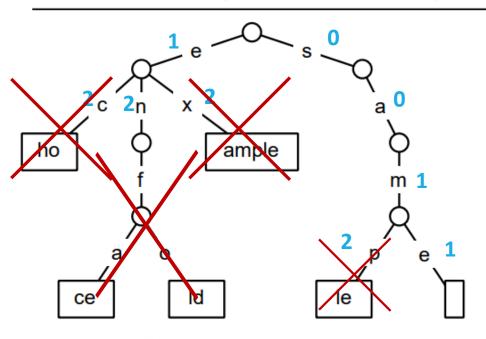
For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k=1

Text:

echo enfold sample enface same example



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

	ϕ	а	С	d	f	Ь	đ	f
φ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d						2		

Edit distance

Edit distance with Ukkonen cutoff

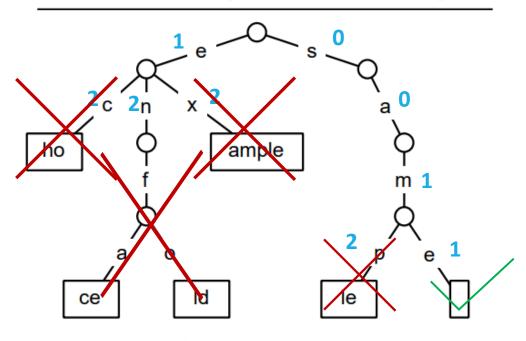
Traverse trie recursively with DFS (Depth First Search)

For each node/character

- Calculate edit distance
- If editDistance > k, skip subtrie

Search: 'sane' k=1

Text:



	φ	а	C	d	f	Ь	d	f
φ	0	1	2	3	4	5	6	7
a	1	0	1	2	3	4	5	6
d	2	1	1	1	2	3	4	5
f	3	. 2	2	2	1	2	3	4
d	4	3	3	2	2	2	2	3

Fai	t A	icta	nce
Lui	ιu	ısta	

	φ	а	С	d	f	Ь	d	f
ϕ	0	1	2	3	4	. 5		
а	1	0	1	2	3	4		
d	2	1	1	1	2	3		
f			2	2	1	2 .		
d			-			2		

Edit distance with Ukkonen cutoff

Performance

Worst case: $O(k|\Sigma|^k)$

Best case: k = 0

Trie

- Grows exponentially
- Better with smaller *k*

Break point: k = 2

agrep

- Linear
- Better with larger *k*

Thanks for your attention

T. Shang & H. T. Merrett, *Tries for Approximate String Matching 1995*