Universidade Federal de Goiás
Projeto e Análise de Algoritmos

War Story:
Dialing for Documents

Alunos: Jesus José de Oliveira Neto
Marcelo de Castro Cardoso

Universidade Federal de Goias
Mestrado em Ciência da Computação



Agenda

- Por que enviar documentos por telefone?
- O problema: “Dialing for Documents”
- Modelos de teclado telefônico
- Técnicas para decodificação de dados
- Algoritmo de Viterbi
- Considerações finais
- Conclusão

Por que enviar documentos por telefone?

- Dialing for Documents – Documentos por discagem.
- A união da computação e da tele-comunicão criou uma grande variedade de produtos e serviços.
- Aplicações:
  - Comunicação entre computador e telefone
  - Enviar e-mail sem terminal de computador
  - Sistemas de resposta de voz melhorados

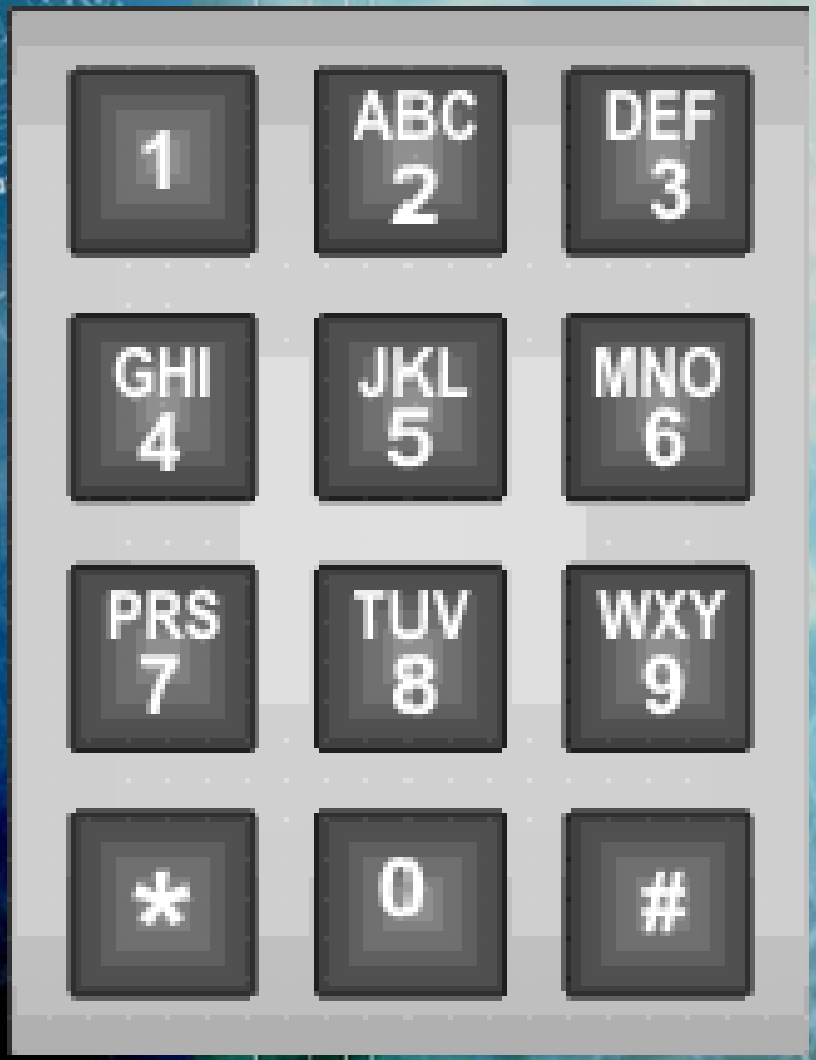
O problema da Periphonics

- Periphonics é uma empresa especializada em sistemas de resposta de voz: “Digite 1 para mais opções, Digite 2 se voçê não digitou 1...”
- Sistemas de reconhecimento de voz:
  - Sistemas de reconhecimento de voz não são precisos para a maioria das aplicações.
  - Os clientes preferem digitar no teclado telefonico do que falar.

O problema da Periphonics

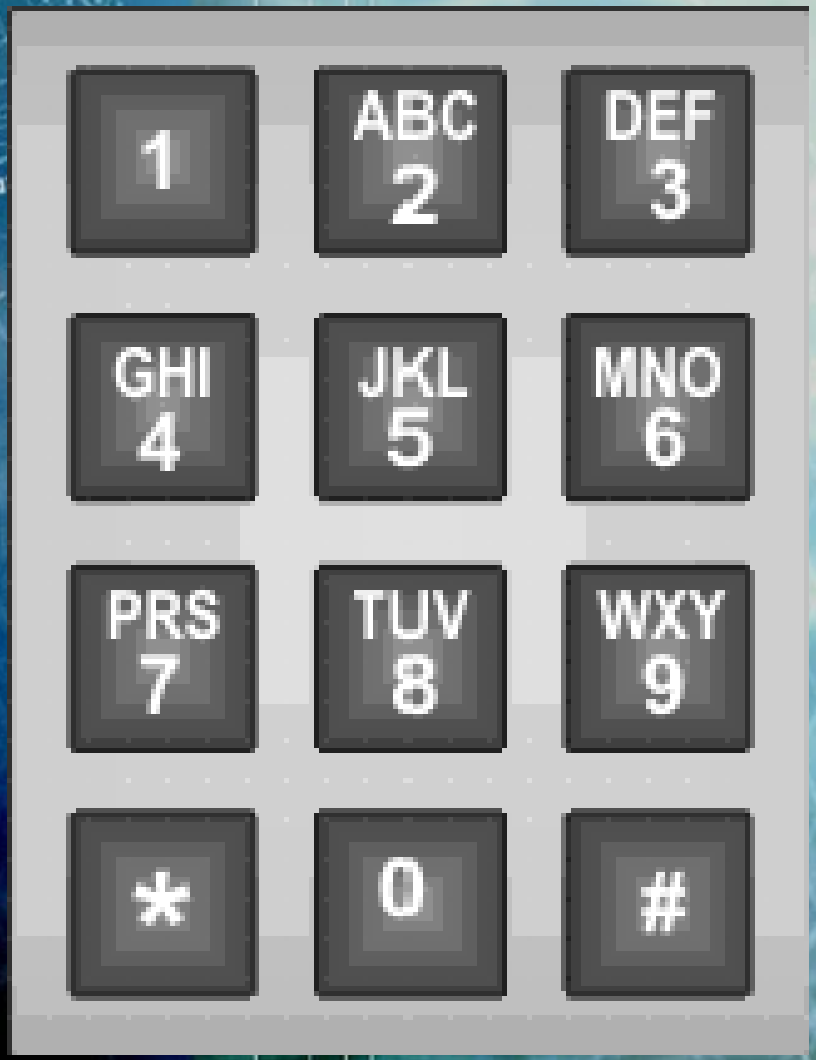
- Sistemas de digitação em teclado telefônico
  - Produz um resultado exato pois o reconhecimento é direto.
  - É ruim porque tem que digitar duas vezes para cada letra.
  - É bastante exaustivo para um texto um pouco mais extenso.

O Teclado Telefônico



Texto:

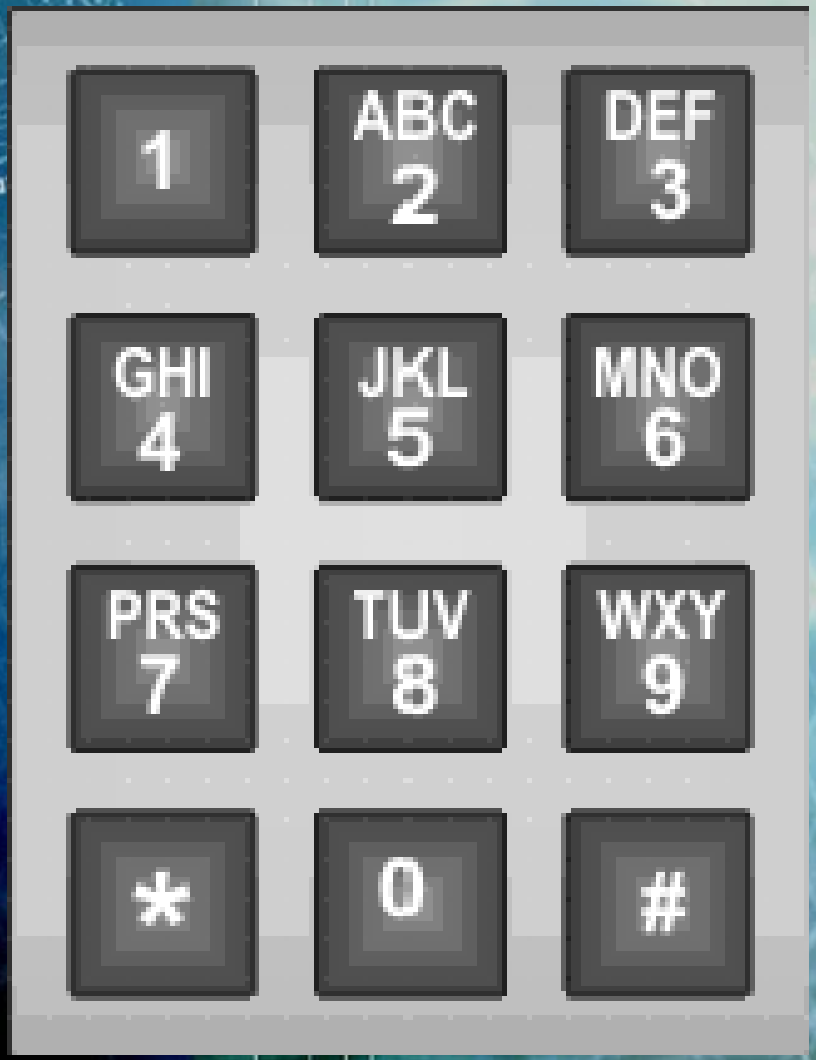
O Teclado Telefônico



Texto:

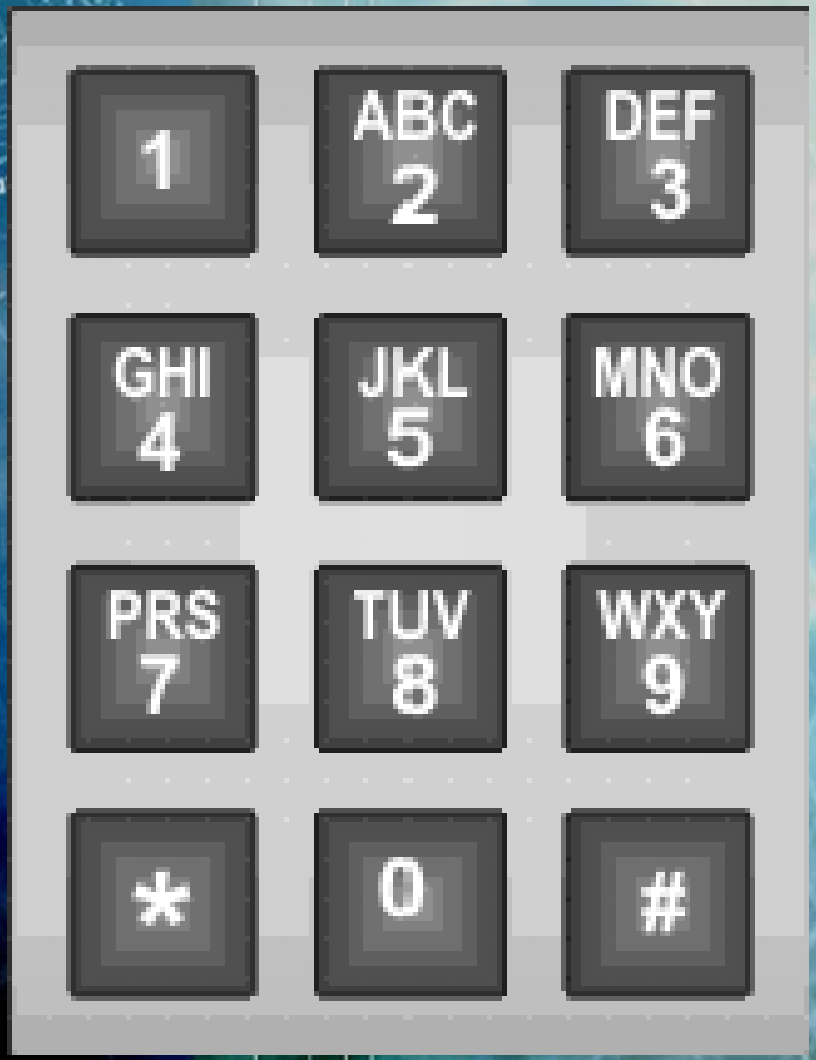
Código 7 (P, R, S)

O Teclado Telefônico



Texto: P

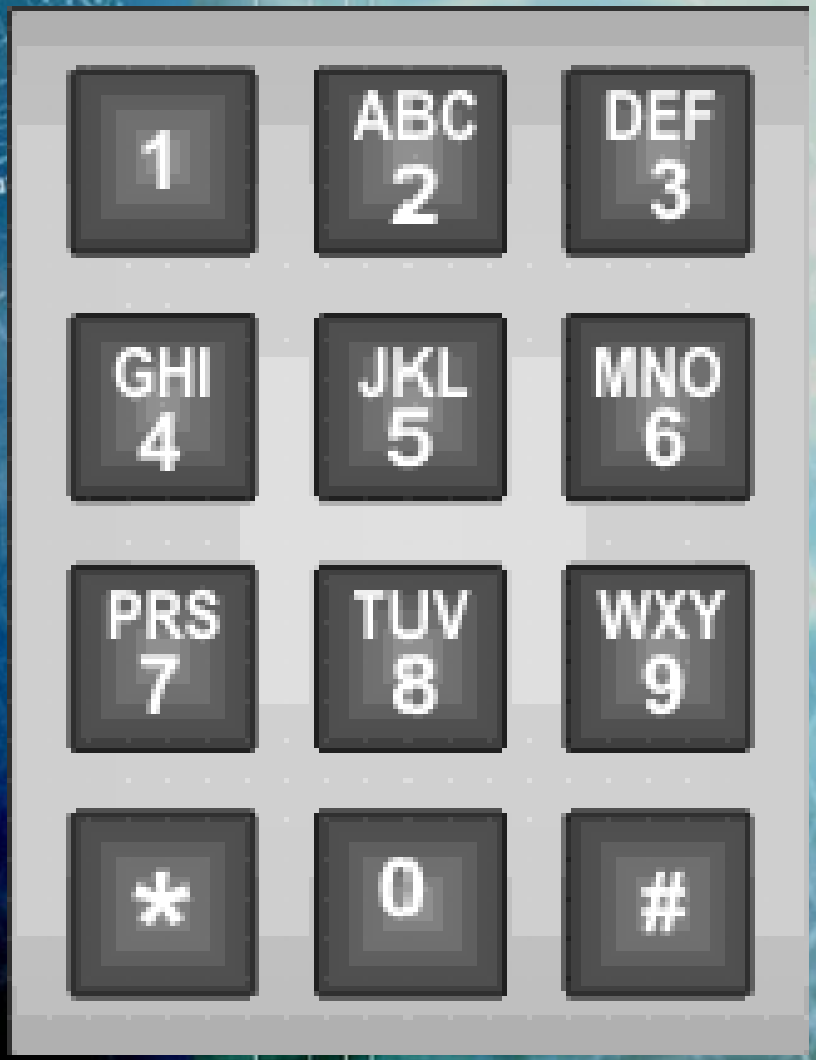
O Teclado Telefônico



Texto: P

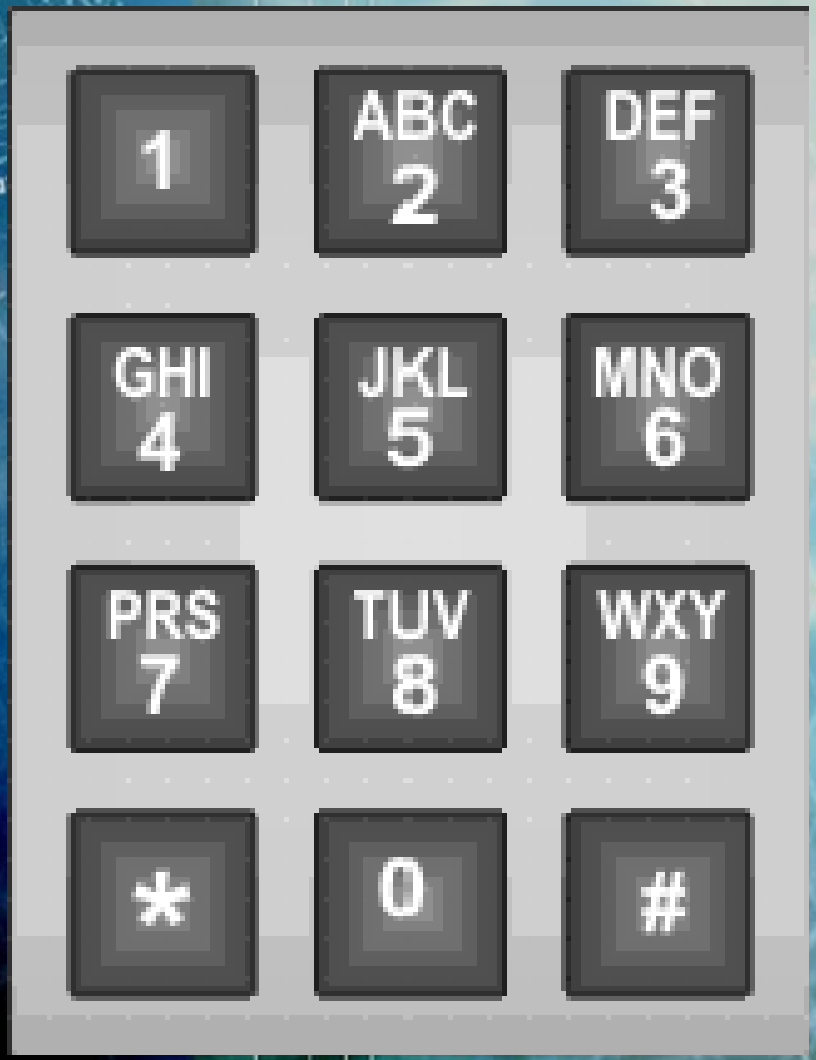
Código 2 (A, B, C)

O Teclado Telefônico



Texto: P A

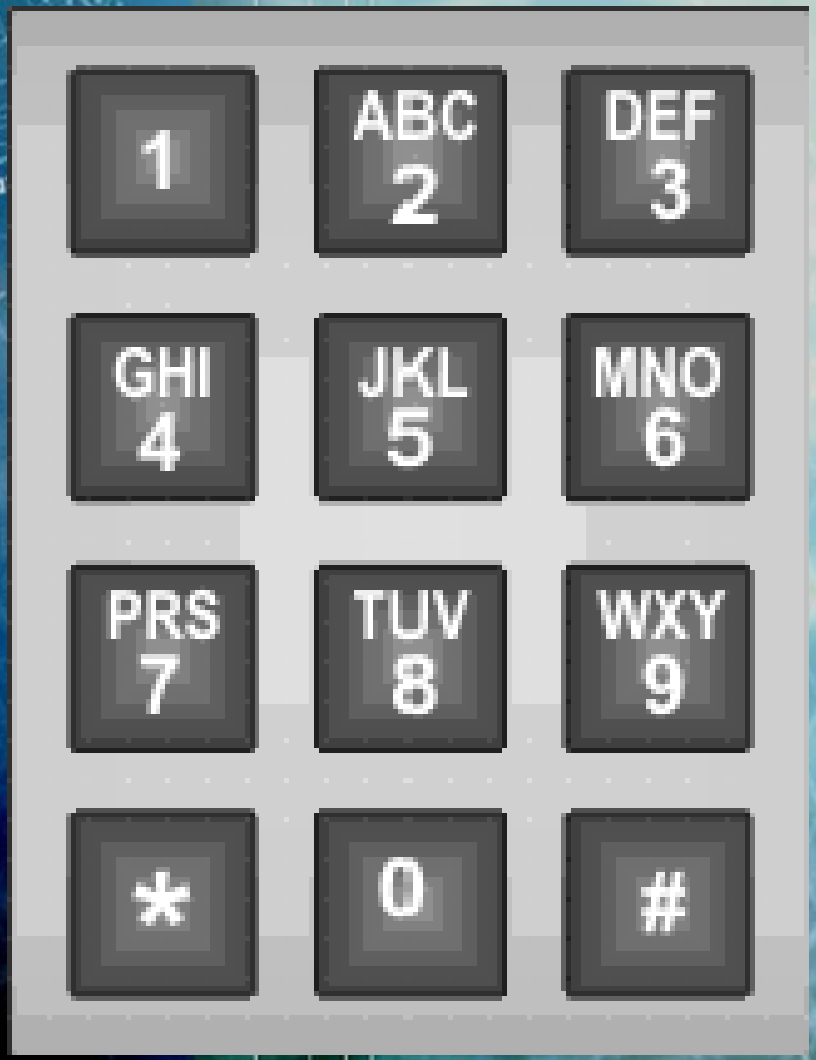
O Teclado Telefônico



Texto: P A

Código 2 (A, B, C)

O Teclado Telefônico



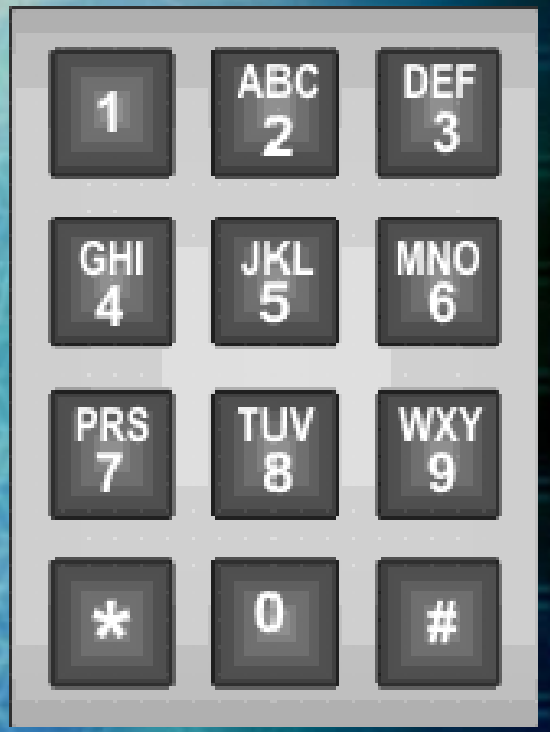
Texto: P A A

Objetivo de Skiena

Construir um sistema que determine uma palavra pelo contexto, apenas precionando uma tecla por vez.

Modelos de Teclado Telefônico

- O padrão americano:
  - Três letras por tecla
  - Q, Z e espaço '&' não aparecem no teclado
- O padrão de Tsakanikaa:
  - Q, Z – “código 1”
  - & – “código 0”
- O padrão de Marics:
  - Q, Z e & – “código *”



Modelo Adotado no Problema

- O padrão de Marics foi adotado por convenção: Q, R e & correspondem a tecla “*”.
- A tecla “#” foi definida como o código delimitador da mensagem.
- As teclas 1 e 0 não tem nenhuma função, e podem ser utilizadas para algum fator adicional.



Primeira Tentativa

- Distribuição irregular da freqüência de letras
  - Tenta adivinhar qual a letra mais provável de ter sido digitada.
  - É escolhida a letra mais utilizada da tecla pressionada.
  - Esta técnica não sabe nada sobre palavras em Inglês.



Primeira Tentativa

- Resultado: totalmente ilegível
  - *enurraore ane reten yeasr ain our ectherr arotght eosti on ugis aootinent a oey oation aoncdivee in licesty ane eedicatee un uhe rrorosition uiat all oen are arectee e ual*



Segunda Tentativa

- Acoplar a técnica anterior com um dicionário de palavras em Inglês
  - Compara o código digitado com as palavras no dicionário.
  - Nos códigos que não batem no dicionário, é utilizada a técnica de adivinhação.
  - Não consegue distinguir entre as diferentes palavras que combinam com o mesmo código.
  - Ex: o código #22737# equivale a 11 palavras diferentes em Inglês (cases, cares, cards, etc.)



Segunda Tentativa

- Resultado: gera muitos erros
  - *eourscore and seven yearr ain our eatherr brought forth on this continent azoey nation conceivee in liberty and dedicatee uo uhe proposition that all men are createe equal*



Terceira Tentativa

- Incorporar restrições gramaticais a partir das palavras identificadas
  - Com a ajuda de Harald Rau, Skiena criou um programa baseado em regras gramaticais.
  - Estas regras foram obtidas de um banco de dados: Brown Corpus.
  - Faz distinção entre as palavras que tem o mesmo código a partir das regras gramaticais.



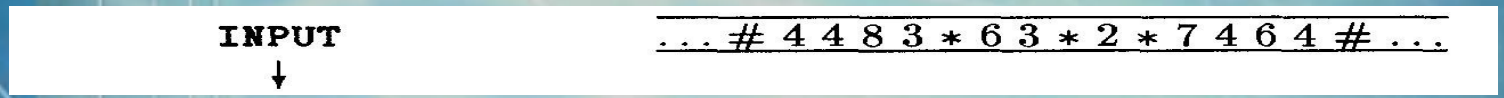
Terceira Tentativa

- Algoritmo de reconstrução: três fases
  - Fase 1 – Identificação de caracteres de espaço (&).
  - Fase 2 – Identificar as palavras cantidatas a partir de um dicionário.
  - Fase 3 – Resolver as palavras ambígüas, ou seja, as palavras diferentes que batem com o mesmo código.

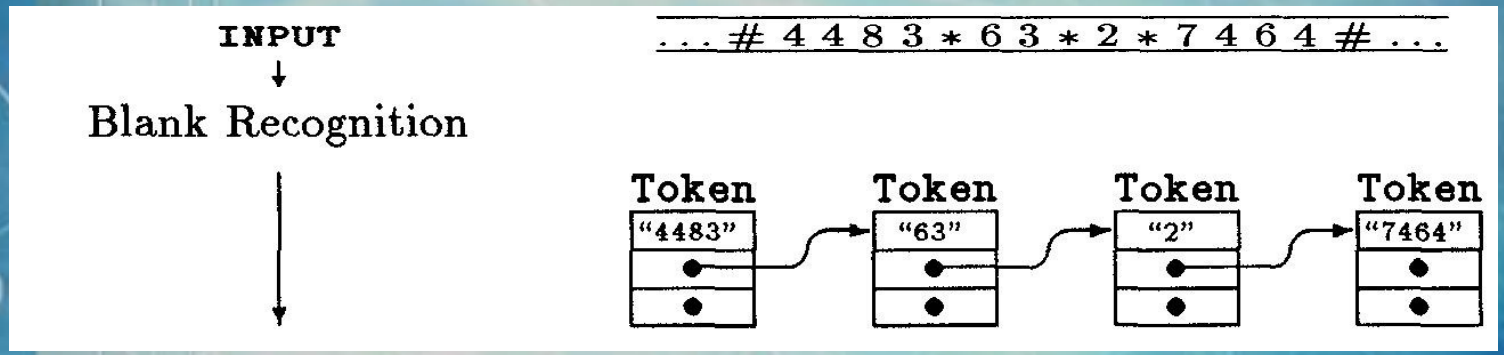
O Algoritmo de Reconstrução

Universidade Federal de Goias
Mestrado em Ciência da Computação

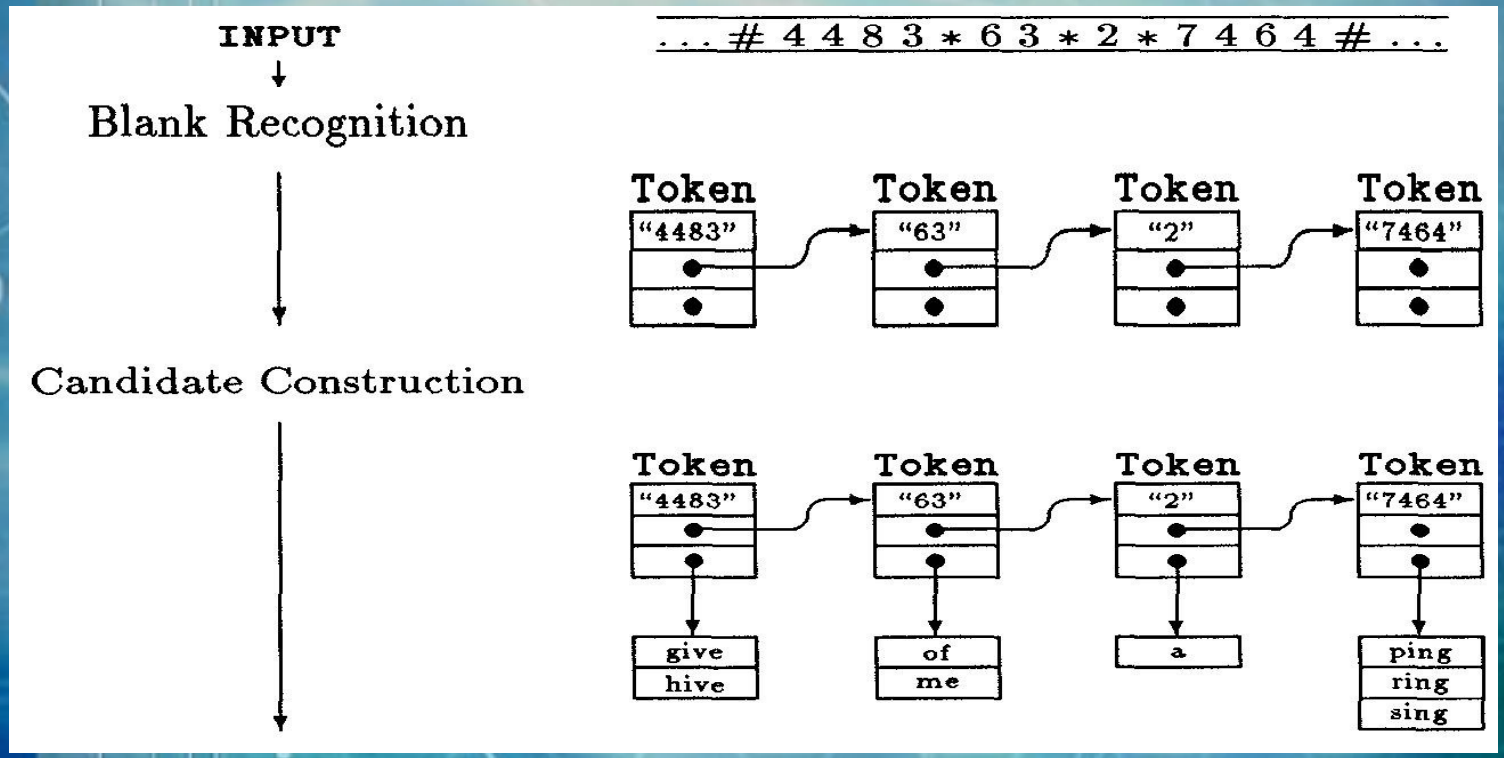
O Algoritmo de Reconstrução



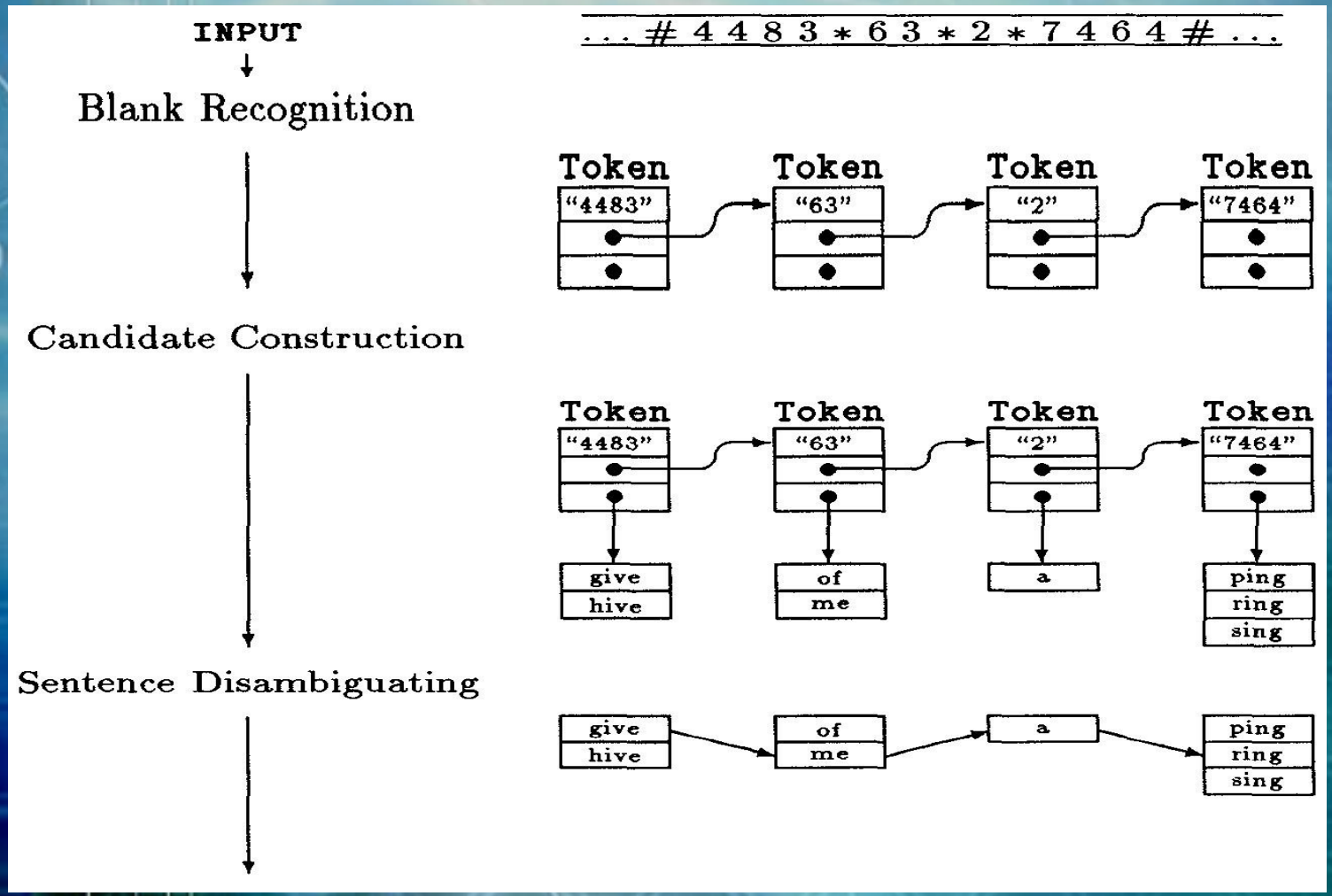
O Algoritmo de Reconstrução



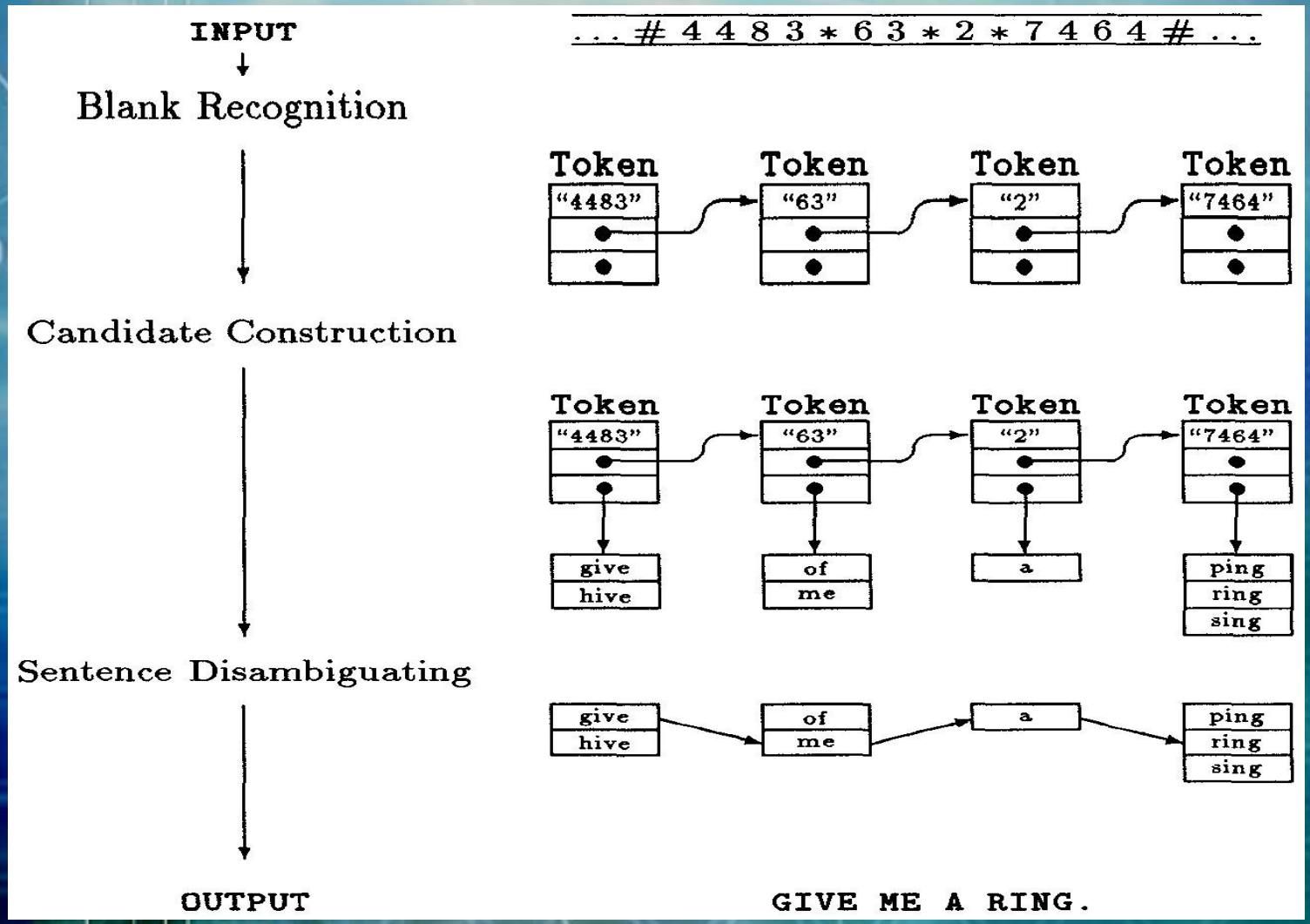
O Algoritmo de Reconstrução



O Algoritmo de Reconstrução



O Algoritmo de Reconstrução

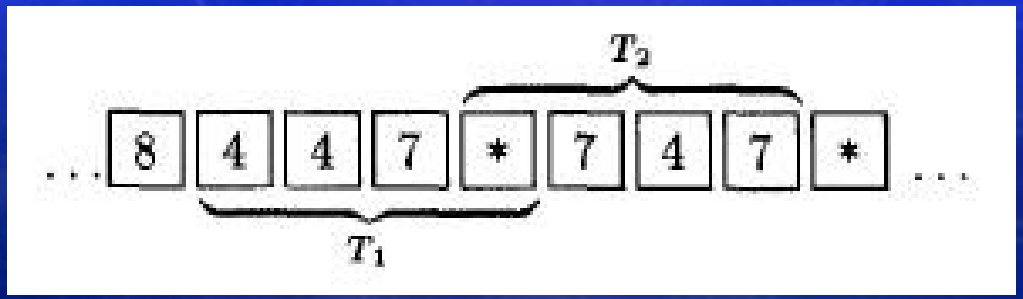


Fase 1
Identificar Caracteres de espaço

- Não é trivial – o código “*” corresponde ao Q e Z, além do espaço.
- Para 98,99% das vezes o código “*” é realmente espaço.
- Há técnicas para melhorar esta aproximação.

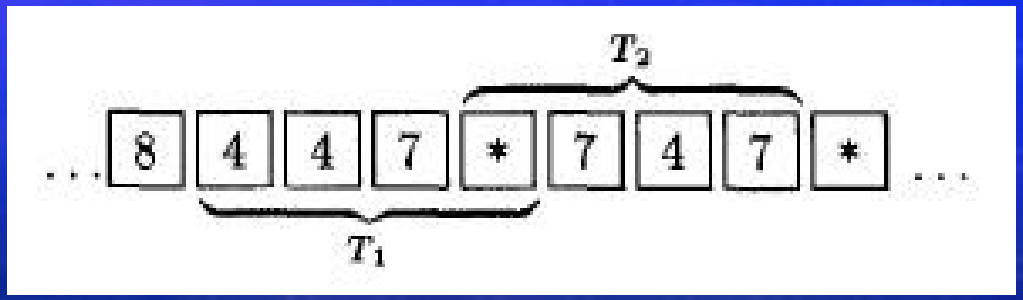
Fase 1
Identificar Caracteres de espaço

- A base da técnica é definir o caractere como espaço a partir do contexto do código “*”.
- O contexto do código “*” é definido como dois trigramas: pré-asterisco e pós-asterisco.
- O código “*” é definido como espaço a partir da análise da freqüência do código “*” no quarto caractere do pré-trigrama e no primeiro caractere do pós-trigrama.



Fase 1
Identificar Caracteres de espaço

- A análise é feita através do dicionário.
- Se a freqüência for alta, ou seja, a maioria dos códigos não tinha um Q ou Z a frente ou atrás nos trigramas, então o “*” é colocado como espaço.



Fase 2
Identificação das Palavras

- Dicionário:
  - 53.324 palavras em Inglês em um *hash* de 44.474 códigos de entrada.
  - A tabela mostra a proporção de palavras com o mesmo código.

Number of coinciding codes	*Words*	*Code Strings*
1	41122	41122
2	4896	2448
3	1476	492
4	588	147
5	300	60
6	174	58
7	42	6
8	48	6
9	9	1
10	0	0
11	11	1

Fase 2
Identificação das Palavras

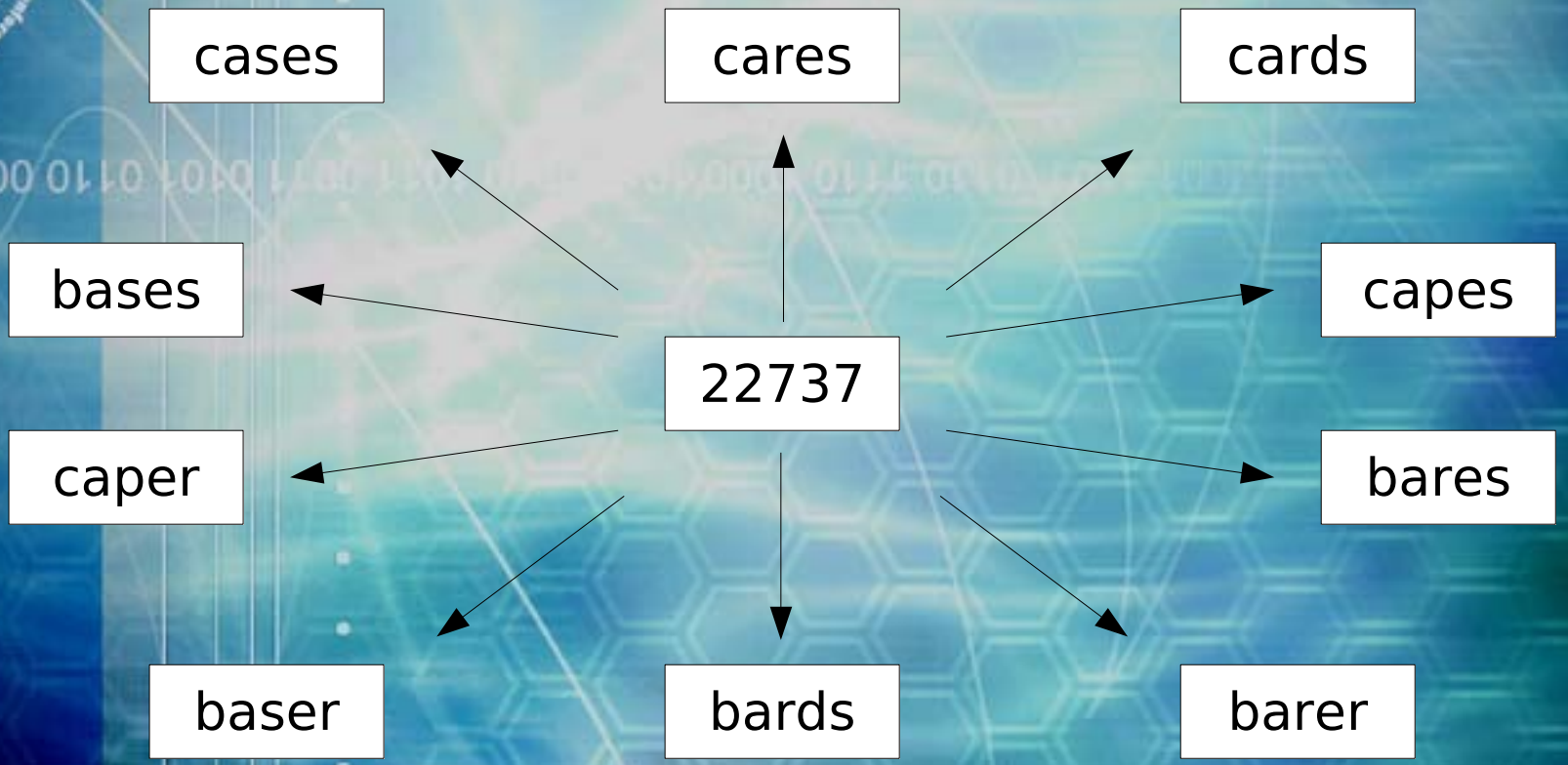
Universidade Federal de Goias
Mestrado em Ciência da Computação

Fase 2
Identificação das Palavras

22737

Universidade Federal de Goias
Mestrado em Ciência da Computação

Fase 2
Identificação das Palavras



Fase 3
Resolver Ambigüidades

- Resolve as ambigüidades através do contexto das palavras.
- Baseia-se em escolher a seqüência certa de palavras através do uso de grafos.
- Este problema é resolvido pelo Algoritmo de Viterbi.

Fase 3
Algoritmo de Viterbi

- Para o Algoritmo de Viterbi temos as seguintes considerações:
  - $X = X_0X_1....X_NX_{N-1}$, onde X é uma seqüência de tokens, em que X_0 e X_{N-1} são delimitadores de sentença.
  - $Z = Z_0Z_1....Z_NZ_{N-1}$, onde Z é uma sentença consistente com X, estas sentenças são obtidas do Brown Corpus.

Fase 3
Algoritmo de Viterbi

- Temos então as seguintes probabilidades:
  - $P(Z_i/X_i)$ – a probabilidade de que a palavra Z_i seja a palavra encontrada quando o código X_i for analisado (vértices).
  - $P(Z_i/Z_{i+1})$ – a probabilidade transitória de que a palavra Z_i seja a palavra encontrada quando a palavra anterior for Z_{i+1} (arestas).

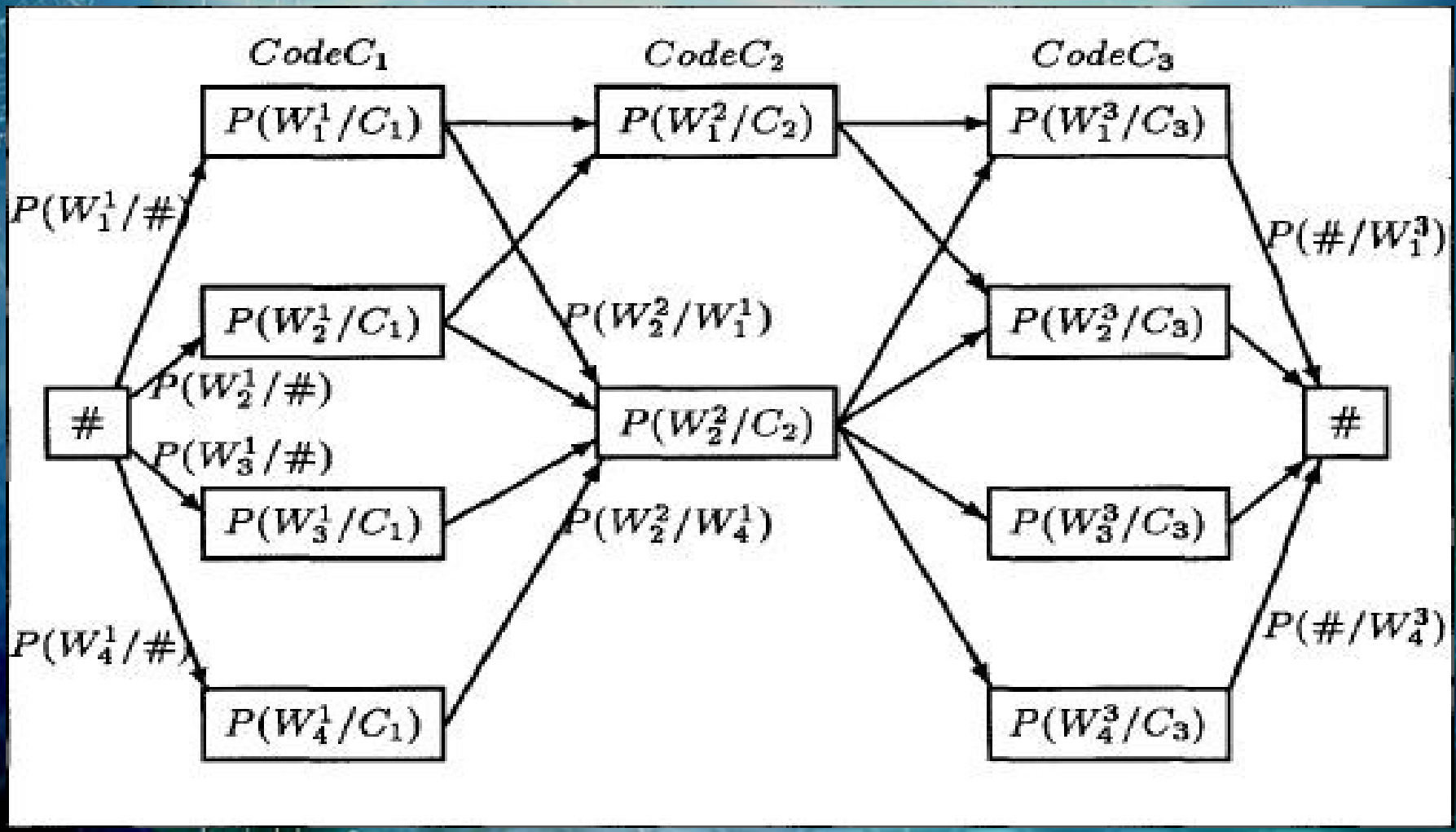
Fase 3
Algoritmo de Viterbi

- O cálculo da probabilidade transitória é feito através da seguintes fórmulas:
  - Caso o par de palavras $W_{i+1}W_i$ não ocorra na base de dados do Brown Corpus
    - $P(W_{i+1}/W_i) := 1/count_{wi}^2$
  - Caso contrário
    - $P(W'_{i+1}/W_i) := 1/count_{wi}$
  - Onde $count_{wi}$ é a freqüência de vezes que a palavra W_i ocorreu no Corpus.

Fase 3
Algoritmo de Viterbi

- Através do grafo gerado G(V, E):
  - V – todas as palavras candidatas (tokens)
  - E – todas as transições possiveis
- O peso do conjunto (V, E) são dados pelas probabilidades calculadas.
- O cálculo da melhor seqüência (ou melhor caminho no grafo) é dado pelo caminho de maior custo (maior somatório de probabilidades).

Fase 3
Grafo de Probabilidades



Considerações Finais

- Primeira Tentativa
  - *enurraore ane reten yeasr ain our ectherr arotght eosti on ugis aootinent a oey oation aoncdivee in licesty ane eedicatee un uhe rrorosition uiat all oen are arectee e ual*
- *Segunda Tentativa*
  - *eourscore and seven yearr ain our eatherr brought forth on this continent azoey nation conceivee in liberty and dedicatee uo uhe proposition that all men are createe equal*
- *Terceira Tentativa*
  - *FOURSCORE AND SEVEN YEARS AGO OUR FATHERS BROUGHT FORTH ON THIS CONTINENT A NEW NATION CONCEIVED IN LIBERTY AND DEDICATED TO THE PROPOSITION THAT ALL MEN ARE CREATED EQUAL*

Considerações Finais

Text	characters	characters correct	non-blanks correct	words correct	time per character
Clinton Speeches	1073593	99.04%	98.86%	97.67%	0.97ms
Herland	278670	98.24%	97.89%	97.02%	0.97ms
Moby Dick	1123581	96.85%	96.25%	94.75%	1.14ms
Bible	3961684	96.20%	95.39%	95.39%	1.33ms
Shakespeare	4558202	95.20%	94.21%	92.86%	0.99ms

Conclusão

- As restrições associadas com os vários e diferentes problemas de reconhecimento de padrões podem ser formuladas como problemas de menor caminho em grafos.
- Procurar uma formulação de grafos para algum problema é sempre uma boa forma de proceder.



Bibliografia

- [1] Skiena, Steven S. “The Algorithm Design Manual”. Springer-Verlag, New York, 1997.
- [2] Rau, Harald; Skiena, Steven S. ”Dialing for Documents: An Experiment in Information Theory”. ACM, 1994.

Perguntas



Obrigado!!

Universidade Federal de Goias
Mestrado em Ciência da Computação