UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECONHECIMENTO DE NOMES DE PESSOAS E ORGANIZAÇÕES EM TEXTOS FORENSES USANDO UMA VARIAÇÃO DO MODELO OCULTO DE MARKOV

OSVALDO DALBEN JUNIOR

ORIENTADORA: Profª. Drª. DANIELA BARREIRO CLARO

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO: INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO

PUBLICAÇÃO: PPGENE.DM – 89/2011
BRASÍLIA/DF: DEZEMBRO – 2011

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECONHECIMENTO DE NOMES DE PESSOAS E ORGANIZAÇÕES EM TEXTOS FORENSES USANDO UMA VARIAÇÃO DO MODELO OCULTO DE MARKOV

OSVALDO DALBEN JUNIOR

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISÍTOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:

_____

Daniela Barreiro Claro, PhD (DCC-UFBA)
(Orientadora)

_____

Rafael Timóteo de Sousa Júnior, PhD (ENE-FT-UnB)
(Examinador Interno)

_____

Hélvio Pereira Peixoto, PhD (DITEC-DPF)
(Examinador Externo)

BRASÍLIA/DF, 13 DE DEZEMBRO DE 2011

FICHA CATALOGRÁFICA

DALBEN JR., OSVALDO
Reconhecimento de Nomes de Pessoas e Organizações em Textos Forenses Usando Uma Variação do Modelo Oculto de Markov [Distrito Federal] 2011.
(xiv), (103)p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2011)

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Informática Forense 2. Mineração de Texto
3. Reconhecimento de Entidades Mencionadas 4. Modelo Oculto de Markov
I. ENE/FT/UnB II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DALBEN JR., O. (2011). Reconhecimento de Nomes de Pessoas e Organizações em Textos Forenses usando uma Variação do Modelo Oculto de Markov. Dissertação de Mestrado, Publicação PPGENE.DM – 89/2011, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 103p.

CESSÃO DE DIREITOS

AUTOR: Osvaldo Dalben Junior.

TÍTULO: Reconhecimento de Nomes de Pessoas e Organizações em Textos Forenses usando uma Variação do Modelo Oculto de Markov.

GRAU / ANO: Mestre / 2011

_____

Osvaldo Dalben Junior
Universidade de Brasília
Campus Universitário Darcy Ribeiro – Asa Norte
CEP 70.910-900 Brasília/DF/Brasil.

Dedicado à minha esposa Érica e aos meus filhos, Pedro, Thiago e Osvaldinho.

AGRADECIMENTOS

A Deus.

À minha esposa e meus três filhos, pela compreensão, paciência e coragem ao entrarem comigo nesse projeto.

Aos meus pais, pelo incondicional apoio dado durante todo o período do mestrado.

À minha orientadora, Profª. Drª. Daniela Barreiro Claro, por ter me transmitido o conhecimento e a experiência necessários ao cumprimento desta jornada e por ter acreditado em mim.

Aos colegas do mestrado Polastro, Nassif e Rommel, pelas discussões, incentivos, elogios e críticas que tanto enriqueceram o trabalho.

Aos colegas de trabalho, em especial aos PCF Dourado, Helena, Iracema e Mateus, pela enorme compreensão e incentivo durante o árduo período de estudos.

Ao Sr. Nuno Cardoso, aluno de doutorado do Departamento de Informática da Universidade de Lisboa e criador do Sistema Rembrandt, pelas inúmeras trocas de *emails* que tiveram importante papel no meu entendimento desse Sistema e do REM de forma geral.

Ao PCF Peixoto, por ter lançado a semente deste projeto e me incentivado a fazer parte dele.

Ao DPF, à UnB e ao PRONASCI, por terem tornado viável a execução deste mestrado.

RESUMO

RECONHECIMENTO DE NOMES DE PESSOAS E ORGANIZAÇÕES EM TEXTOS FORENSES USANDO UMA VARIAÇÃO DO MODELO OCULTO DE MARKOV

Autor: Osvaldo Dalben Junior
Orientadora: Profª. Drª. Daniela Barreiro Claro
Programa de Pós-graduação em Engenharia Elétrica
Brasília, dezembro de 2011

Um dos atuais desafios na área da forense computacional está relacionado à análise de mídias computacionais apreendidas em grande quantidade pelas forças policiais. Os arquivos armazenados nessas mídias podem conter nomes de pessoas e organizações suspeitos, porém desconhecidos pelas equipes de investigação. O presente trabalho propõe a criação de um modelo de Reconhecimento de Entidades Mencionadas (REM) baseado no Modelo Oculto de Markov (HMM) para extrair nomes de pessoas e organizações de textos não estruturados contidos em mídias apreendidas. O modelo proposto, denominado ICC-HMM (*Identification-Classification Context HMM*), é dividido em dois submodelos - identificação e classificação de entidades - e utiliza as informações do contexto das palavras e um *gazetteer* como forma de obter melhor desempenho. Experimentos foram realizados aplicados a corpora públicos e forenses e os resultados do ICC-HMM superaram os obtidos por sistemas participantes de avaliações conjuntas específicas para o REM no idioma português, o que sugere que o modelo proposto é aplicável ao cenário forense nacional.

ABSTRACT

RECOGNIZING NAMES OF PEOPLE AND ORGANIZATIONS IN FORENSIC TEXTS USING A HIDDEN MARKOV MODEL VARIATION

Author: Osvaldo Dalben Junior
Advisor: Profª. Drª. Daniela Barreiro Claro
Electrical Engineering Graduate Program
Brasília, December of 2011

One of the current challenges in computer forensics is related to the analysis of computer media seized in large quantities by the police. Files stored in these media may contain names of people and organizations suspected, but unknown by the analysis teams. This paper proposes the creation of a named entity recognition (NER) model based on the Hidden Markov Model (HMM) to extract names of people and organizations contained in unstructured text of seized media. The proposed model, called ICC-HMM (Identification - Classification Context HMM) is divided into two sub-models – identification and classification of entities - and uses the context information of words and a gazetteer in order to obtain better performance. Experiments were carried out on forensic corpora and our results outperformed some of the best NER-based systems in portuguese language. This suggests that the proposed model is applicable in brazilian computer forensics.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIAÇÕES

Back-off – Modelos probabilísticos alternativos que tratam o problema da escassez de treinamento.

CRF – *Conditional Random Fields*, modelo probabilístico usado para predição de sequências.

EI – Extração da Informação.

EM – Entidade Mencionada.

Feature – Qualquer característica associada a uma palavra, direta ou indiretamente.

HMM – *Hidden Markov Model*, modelo probabilístico usado para predição de sequências.

ICC-HMM – *Identification-Classification Context HMM*, nome dado ao modelo proposto no presente trabalho.

IE – *Information Extraction*.

MEMM – *Maximum Entropy Markov Model*, modelo probabilístico usado para predição de sequências.

MT – Mineração de Texto.

NE – *Named Entity*.

NER – *Named Entity Recognition*.

NLP – *Natural Language Processing*.

PLN – Processamento de Linguagem Natural.

REM – Reconhecimento de Entidades Mencionadas.

Smoothing – Ou suavização, técnica usada para minimizar o problema da insuficiência de treinamento através da utilização de modelos probabilísticos auxiliares, pioram a acurácia, porém garantem a predição das sequências.

SVM – *Support Vector Machine*, modelo probabilístico usado para predição de sequências.

Token – No presente trabalho, refere-se à menor unidade textual tratável pelos modelos.

TM – *Text Mining*.

1 - INTRODUÇÃO

Uma operação policial caracteriza-se, em geral, por ser uma ação planejada por uma equipe de investigação, que tem como **meio** a obtenção de informações relevantes sobre os alvos - pessoas acusadas - e como **fim** a comprovação da sua culpa, dolo ou inocência. Tal comprovação pode ser testemunhal, que é baseada em depoimentos de pessoas, ou material, que resulta do exame pericial dos vestígios materiais – coisas ou pessoas - associados à prática criminosa, vestígios estes denominados corpos de delito. Conforme legislado no Código de Processo Penal (CPP) brasileiro[1], a prova testemunhal somente é priorizada quando não é possível o exame do corpo de delito, e este é indispensável nos casos em que a infração deixar vestígios, mesmo quando há a confissão do acusado. Os exames de corpo de delito são realizados por peritos criminais e objetivam definir (i) se o material examinado pode ou não ser considerado uma prova e, sempre que possível, revelar a autoria (quem), materialidade (o quê) e dinâmica (como) do crime sob apuração, e (ii) se, do material periciado, pode-se extrair informações associadas a novos materiais ou fatos suspeitos. No caso (i), quando os exames periciais comprovam a associação de pessoas investigadas com o crime ou a sua inocência, esta prova tem interferência direta no julgamento dos acusados. Já no caso (ii), quanto os exames revelam novas suspeitas, estas informações passam a compor a base de conhecimento da investigação e são utilizadas como ferramenta de apoio à decisão para futuras ações da equipe policial.

Os avanços na área de comunicação interpessoal proporcionados pela inclusão digital, protagonizada principalmente pela popularização da *Internet* a partir da década de 90, resultaram num constante crescimento da casuística de crimes praticados por computador ou com o auxílio deste. Isso, associado ao aumento da capacidade de armazenamento e à diminuição do custo de aquisição de mídias digitais, representou um enorme crescimento do volume de mídias digitais apreendidas em operações policiais nas duas últimas décadas. A maior parte das operações policiais que resultam na apreensão de grande quantidade de mídias está associada a crimes convencionais, de natureza distinta à informática, tais como crimes previdenciários, fazendários ou políticos. Conforme apresentado em (Eleutério e Machado, 2011), cerca de 90% das apreensões de mídias de armazenamento computacional realizadas pela Polícia Federal do Brasil no ano de 2011 ocorreram por

[1] Decreto-Lei nº 3.689 de 3 de outubro de 1941.

suspeita de utilização desses equipamentos como ferramenta de apoio para o cometimento de crimes convencionais, e não como um meio para a sua realização. A apreensão desse material é necessária devido à alta probabilidade de nele serem encontradas informações importantes para a investigação, como, por exemplo, um *email* ou conversa instantânea que reforça uma suspeita de improbidade administrativa ou uma planilha eletrônica que controla determinado repasse ilegal de verba pública. Depois de realizada a apreensão do material, a equipe de investigação solicita à equipe pericial a extração e disponibilização de todos os arquivos potencialmente suspeitos das mídias apreendidas, para que os conteúdos dos mesmos sejam devidamente analisados por equipe especializada da investigação.

Entretanto, como não é possível identificar previamente quais são os arquivos potencialmente suspeitos, em geral são selecionados, extraídos e disponibilizados para análise manual todos os arquivos que indiquem qualquer interação de uma pessoa com um computador, ou seja, são excluídos da seleção somente os arquivos relacionados à instalação do sistema operacional ou de outros aplicativos que possam ser identificados, o que resulta em um grande volume de dados que necessitam de análise manual.

1.1 – MOTIVAÇÃO

O crescente aumento do volume de dados digitais apreendidos nos últimos anos tem tornado improdutiva e ineficaz a tradicional forma de trabalho do investigador, qual seja, formar conclusões baseadas na análise manual e isolada dos arquivos extraídos dessas mídias. Esta forma de trabalho não viabiliza a identificação de nomes de pessoas e organizações mencionadas nos arquivos. Esta informação, se disponível, é relevante para a equipe de investigação, pois pode revelar nomes suspeitos desconhecidos ou inesperados dentro do contexto investigativo.

Além disso, conforme apresentado em (Dalben e Claro, 2011), a identificação de nomes de pessoas e organizações em mídias forenses pode ser usada como um filtro capaz de reduzir mais de 90% dos arquivos comumente analisados manualmente, com risco mínimo de descarte de arquivos relevantes.

1.2 – PROBLEMA

Diante do cenário apresentado, evidencia-se uma carência de métodos e ferramentas que automatizem o processo de identificação de nomes de pessoas e organizações em arquivos digitais, representando um entrave na área da investigação criminal, por exigir maior esforço humano na tarefa de análise do conteúdo de mídias digitais apreendidas e gerar resultados não satisfatórios quanto à revelação de informações latentes presentes nessas mídias.

1.3 – PROPOSTA

Este trabalho propõe o reconhecimento automatizado de nomes de pessoas e organizações contidos em textos não estruturados de arquivos presentes em mídias computacionais apreendidas, como forma de:

- Revelar à equipe de investigação os nomes das pessoas e organizações citados no conteúdo das mídias apreendidas e assim enriquecer a base de conhecimento da investigação;
- Criar um filtro para reduzir o volume de arquivos analisados manualmente e, consequentemente, reduzir o tempo de análise desses arquivos.

1.4 – OBJETIVO

O presente trabalho objetiva a criação de um modelo de Reconhecimento de Entidades Mencionadas (REM) baseado no Modelo Oculto de Markov (Rabiner, 1990), ou HMM (*Hidden Markov Model*), adaptado ao contexto forense, a fim de reconhecer nomes de pessoas e organizações contidos em textos forenses não estruturados e redigidos no idioma português.

O termo *textos forenses* é usado no presente trabalho para representar textos de arquivos contidos em mídias apreendidas. Os textos forenses podem se enquadrar em diferentes gêneros textuais, como contratos, *e-mails*, recibos, etc., por isso são caracterizados como independentes de domínio.

Alguns critérios foram decisórios na comparação dos resultados do presente trabalho. Considerando que (i) a independência de domínio, (ii) o idioma português, (iii) os textos não estruturados e (iv) o foco no reconhecimento das categorias *pessoa* e *organização* são critérios diretamente relacionados ao objetivo apresentado, optou-se pela comparação dos resultados do presente trabalho com os obtidos pelo sistema Rembrandt (Cardoso, 2008), que alcançou o melhor desempenho na avaliação conjunta do segundo HAREM (Mota e Santos, 2008). A escolha do HAREM como evento balizador da avaliação se justifica pelo fato das coleções de dados nele utilizadas conterem entidades dos tipos *pessoa* e *organização*, serem escritos na língua portuguesa e serem extraídos de fontes de gêneros variados (Mota et al., 2008a). Além do Rembrandt, os resultados obtidos foram também comparados com três variações do modelo proposto, com o objetivo de identificar e analisar o impacto provocado pela ausência de alguns recursos no desempenho da solução proposta.

1.5 – PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições esperadas com o presente trabalho são as seguintes:

- Desenvolvimento de um algoritmo de REM baseado no HMM especificamente para a língua portuguesa;
- Socialização de um protótipo representativo do modelo proposto, desenvolvido com o objetivo de possibilitar a realização dos experimentos do presente trabalho. Esse protótipo contribuirá para a realização de futuros experimentos utilizando o modelo proposto, bem como poderá seu utilizado como linha de base para o desenvolvimento de um sistema de REM aplicado ao cenário forense;
- Disponibilização, às equipes de investigação policial, de uma lista contendo os nomes das pessoas e organizações mencionadas no conteúdo de mídias computacionais apreendidas em operações policiais, o que favorecerá a revelação de nomes desconhecidos, ficando a cargo das equipes de investigação as análises cabíveis quanto à verificação do envolvimento desses nomes com o ilícito sob apuração;
- Disponibilização da lista de arquivos que contêm pelo menos um nome de pessoa ou organização mencionado, dentre todos os arquivos de uma mídia apreendida. Segundo os autores em (Dalben e Claro, 2011), em média, os itens dessa lista correspondem a menos de 10% do total de arquivos e devem ser priorizados no

processo de análise manual dos arquivos, uma vez que 99,9% dos arquivos relevantes tendem a conter nomes de pessoa ou organização.

1.6 – ORGANIZAÇÃO DO TRABALHO

O segundo capítulo desta dissertação apresenta os conceitos fundamentais da Mineração de Textos (MT), necessários para o entendimento do funcionamento dos modelos de REM.

No capítulo três é apresentado o REM, que é a tarefa de Extração da Informação (EI) associada ao objetivo do presente trabalho.

O capítulo quatro detalha o HMM, modelo base utilizado para o desenvolvimento da solução proposta.

O capítulo cinco descreve alguns trabalhos correlatos que contêm propostas que foram utilizadas na solução desenvolvida no presente trabalho.

O sexto capítulo apresenta o ICC-HMM (*Identification-Classification Context HMM*), que é o modelo probabilístico proposto para resolver o problema do REM aplicado ao cenário forense.

O capítulo sete descreve os experimentos realizados e apresenta e discute os seus resultados.

Por fim, o oitavo capítulo apresenta as conclusões finais e direciona trabalhos futuros dentro da linha de pesquisa abordada.

2 - MINERAÇÃO DE TEXTOS

Dentro da área da linguística computacional, a tarefa de processar a linguagem natural tem o objetivo de permitir a comunicação entre seres humanos e máquinas da forma mais natural possível, ou seja, sem alterar a forma como o homem se expressa. Quando esta tarefa se aplica a textos redigidos por humanos, atua-se no campo da mineração de textos, uma área que tem sido bastante explorada nas últimas décadas através de métodos e ferramentas especializados em extrair de textos não estruturados informações estruturadas com utilidades específicas.

Conforme apresentado em (Feldman e Sanger, 2007), a Mineração de Textos (MT), assim como a Mineração de Dados (MD), busca identificar determinados padrões que, quando aplicados aos dados de entrada, permitem a obtenção de informações úteis neles contidas. A diferença básica entre a MT e a MD é que na primeira esses dados de entrada são coleções de textos não estruturados e na última são registros de bases de dados estruturados. Enquanto que a MD analisa os relacionamentos e visões a serem criados entre os dados estruturados, a MT concentra-se no pré-processamento das coleções de textos, ou seja, na transformação do dado não estruturado em um dado estruturado intermediário, que será usado como entrada para a segunda etapa da mineração, composta por operações de descoberta do conhecimento.

A MT se aplica também a textos semiestruturados, que se caracterizam por possuir elementos consistentes de formatação que permitem a inferência de informações específicas com pouco ou nenhum esforço, como *emails*, páginas HTML e arquivos no formato PDF.

O Reconhecimento de Entidades Mencionadas (REM), foco principal do presente trabalho, é uma tarefa baseada na etiquetação de sequências textuais e pertence à subárea da MT denominada Extração da Informação (EI).

2.1 – ARQUITETURA

A arquitetura da MT pode ser dividida em quatro tarefas: (i) o pré-processamento, (ii) as operações de mineração, (iii) as técnicas de refinamento e (iv) a camada de apresentação (Feldman e Sanger, 2007). Essas tarefas são representadas na Figura 2.1.

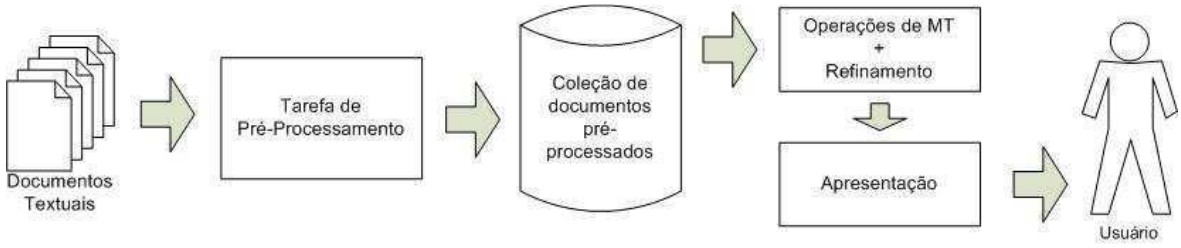


Figura 2.1 - Visão de alto nível da arquitetura da MT (modificado (Feldman e Sanger, 2007))

O **pré-processamento** é a preparação do texto original para as operações de descoberta do conhecimento. Esta etapa envolve a formatação dos dados originais para um padrão pré-definido e a extração de informações associadas aos elementos do texto, com o objetivo de disponibilizá-las como produto final ou como entrada para as tarefas seguintes da MT. É na etapa do pré-processamento que é realizada a tarefa de REM, que produz como saída as informações sobre as categorias de entidade (pessoa, organização) associadas às palavras do texto.

Conforme mostrado na Figura 2.1, as **operações de MT** recebem como entrada as informações produzidas no pré-processamento textual e são consideradas a essência da MT, pois nelas são realizadas as tarefas de mineração propriamente ditas. As principais operações são (i) a análise dos padrões observados nos documentos da coleção, (ii) a análise de tendências e (iii) a descoberta do conhecimento. Como exemplo, se uma equipe de análise periódica de uma coleção de jornais e revistas européias detectar constante redução na quantidade de documentos que fazem referência às palavras "*Bahia*" e "*turismo*", isso pode indicar que a imagem que o europeu possui do turismo na Bahia está piorando. Neste cenário simplificado, ocorreram as três operações retrocitadas: a **análise dos padrões** dos documentos permitiu a extração dos nomes "*Bahia*" e "*turismo*" e dos números a eles associados (quantidade de documentos que os citam); ocorreu a **análise das tendências**, através da associação feita entre a quantidade de ocorrências de palavras e a imagem que o turista tem da Bahia; e a **descoberta do conhecimento**, que é a conclusão

geral da análise, que neste caso inferiu que algo está interferindo negativamente na boa imagem que o turista europeu possui da Bahia.

As **técnicas de refinamento** são uma espécie de filtro para eliminação de informações redundantes ou agrupamento de informações semelhantes, com o objetivo de melhorar o desempenho de um sistema de MT.

Por fim, a última tarefa da arquitetura da MT está associada à **camada de apresentação**, diretamente relacionada à interação que deve existir entre o ser humano interessado na informação e a máquina que a possui. Esta camada se preocupa essencialmente em facilitar o entendimento dos padrões, conceitos e resultados, a parametrização de um sistema e a inserção de consultas personalizadas.

Conforme mencionado previamente, é no pré-processamento que ocorre a tarefa de REM. O principal objetivo do pré-processamento de um texto é prepará-lo para ser submetido às operações de mineração. Diversas tarefas existem com esse objetivo e, segundo (Feldman e Sanger, 2007), essas tarefas podem ser agrupadas em três classes distintas, denominadas: processamento preparatório, tarefas de PLN[2] de propósito geral e tarefas dependentes do problema, conforme representado na Figura 2.2.

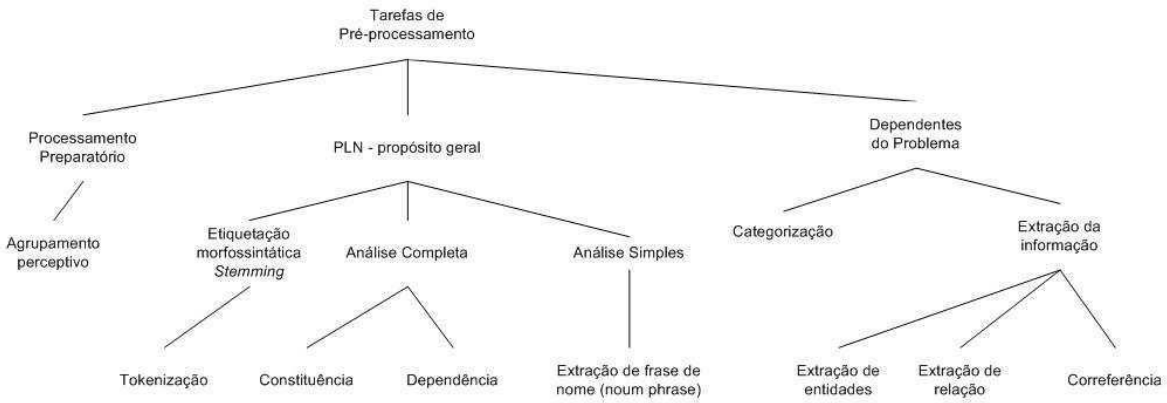


Figura 2.2 - Taxonomia das tarefas de pré-processamento textual (modificado (Feldman e Sanger, 2007))

As **tarefas de processamento preparatório**, representadas no canto esquerdo da Figura 2.2, também conhecidas como padronização do documento (Weiss et al., 2005), convertem

[2] PLN - Processamento de Linguagem Natural.

o documento-alvo do processamento para um formato inteligível para as demais tarefas. Em geral, conversões de arquivos *pdf*, *html, doc*, *rtf*, dentre outros são realizadas para arquivos em texto puro, geralmente no formato *xml*, devido à sua grande aceitação mundial, ou *txt*. Essas tarefas podem também identificar e etiquetar determinadas zonas do texto, como parágrafos ou colunas, e extrair metadados dos documentos, como a data da sua criação ou o nome do seu autor. Além de documentos, pode ser necessária também a preparação de outras fontes de informação, como a fala, a escrita manual ou imagens de texto digitalizado.

A segunda classe é composta pelas **tarefas de PLN de propósito geral** e tem o objetivo de realizar processamentos de conhecimentos não específicos que, em vários casos, são pré-requisitos para as tarefas que resolvem problemas específicos. Dentre as tarefas que compõem esta classe, destacam-se a segmentação em sentenças, a tokenização ou zoneamento, a redução ao radical[3] (do inglês *lemmatization*), a redução ao morfema[4] (do inglês *stemming*) e a análise e etiquetação morfossintática (do inglês *POS-tagging*) das palavras.

O presente trabalho faz uso da segmentação em sentenças, da tokenização e do *stemming*.

A **segmentação em sentenças** é a tarefa de identificar os caracteres que marcam as divisas entre as frases. Os caracteres mais comuns são as pontuações ".", "!", "?" e suas combinações, entretanto em textos menos formais ocorre a separação de sentenças sem a utilização de pontuação (somente com um caractere de controle de avanço de linha, ou *line feed*) ou, em menor quantidade, com a utilização de outros caracteres de pontuação ou especiais. Em (Feldman e Sanger, 2007), os autores afirmam que o maior desafio da tarefa é interpretar o caractere "." (ponto) que, além de representar a separação de duas sentenças, pode representar também a abreviação de uma palavra (por exemplo, *Dr.*, *Sr.*, *Ilmo.*, etc.), a divisão de uma expressão alfanumérica ou de uma data, reticências, etc. A segmentação de sentenças é pré-requisito fundamental para tarefas como a etiquetação morfossintática e o REM, que dependem da interpretação da construção frasal (Weiss et al., 2005). Para ilustrar esta afirmação, supõe-se que um sistema de REM identifique o *token* "*Carlos*"

[3] Elemento estrutural básico da palavra; ex. "puseram" e "posto" (verbo pôr) possuem o mesmo radical.
[4] Menor unidade gramatical que identifica a palavra; ex. "puseram" e "puseste" (verbo pôr) possuem o mesmo morfema "pus", porém "puseram" e "posto" não possuem o mesmo morfema.

como sendo uma Entidade Mencionada (EM) na sequência textual "*Agora vamos falar dos profissionais da empresa. O Carlos é um cara legal.*". Ao tentar identificar o tipo de EM deste *token* (*pessoa*, *organização*, *local*, etc.) não interessa a um sistema de REM analisar o *token* "*empresa*" que precede o termo "*O*", mesmo que esse sistema seja 3-grama, ou seja, examine os 2 *tokens* que precedem o *token* corrente em análise (empresa ⇒ O ⇒ Carlos). Caso isso seja feito, a inclusão do *token* "*empresa*" na análise certamente resultará no aumento da probabilidade do sistema identificar a classe "organização" como associada ao *token* "*Carlos*", o que consequentemente provoca a redução probabilística da etiquetação correta que, neste caso, é da classe "pessoa". Assim, como não há sequência textual padronizada entre sentenças, bem como não é possível precisar o grau de proximidade entre as EM de sentenças adjacentes, sistemas de REM, em geral, tratam as sentenças de forma isolada.

A **tokenização** é pré-requisito para a maioria das tarefas de PLN mais sofisticadas e consiste em dividir as sequências textuais em unidades básicas de processamento, denominadas *tokens*. Nesta tarefa, documentos são divididos em termos, palavras e, menos frequentemente, sentenças, parágrafos e capítulos, a depender da tarefa que fará a utilização do conjunto de *tokens*. Alguns autores, como em (Feldman e Sanger, 2007), consideram a tarefa de segmentação em sentenças como subtarefa da tokenização. Segundo os autores em (Weiss et al., 2005), a tokenização é uma tarefa dependente do idioma, a sua customização nesse sentido é imprescindível para que as determinadas características extraídas dos *tokens* os representem corretamente e assim se evite posterior trabalho de correção desnecessário. Apesar de independer do problema, o desempenho da tarefa de tokenização não é independente do domínio. Em geral, o conhecimento do domínio contribui bastante para a melhora do desempenho das tarefas de propósito geral, principalmente devido a particularidades de domínios envolvendo pontuação, caracteres especiais e *tokens* compostos de mais de uma palavra.

A redução ao morfema (***stemming***) é também uma tarefa de propósito geral que objetiva normalizar palavras, ou seja, agrupar variações de um mesmo radical de modo a reforçar o peso da sua representação no texto. Segundo (Weiss et al., 2005), essa redução pode ocorrer de duas formas, a primeira é a chamada **redução flexível** (*inflectional stemming*), limitada às variações gramaticais de gênero, número e conjugação verbal da palavra com base nos seus prefixos e sufixos, como em "*jogaram*" e "*jogatina*" que se reduzem ao

radical "*jogo*" ou simplesmente "*jog*", e a segunda é a chamada **redução à raiz**, do inglês ***lemmatization***, que busca normalizações mais agressivas do que as realizadas na redução flexível e independe do prefixo e sufixo das palavras. A palavra "*Estrutura*" é um exemplo de raiz associada às palavras "*desestruturação*" e "se*miestruturado*". A indicação quanto ao uso ou não desta tarefa depende da aplicação de mineração em questão. Em geral, dada a dificuldade de previsão de resultados, é indicado o teste da aplicação com e sem a sua utilização como forma de suporte a essa decisão. Há ainda outra variação do *stemming*/*lemmatization*, que é baseada na redução dos *tokens* a um sinônimo e também tem o objetivo de normalizar palavras e assim reforçar o seu peso nas análises textuais.

Por fim, a classe das **tarefas dependentes do problema**, representada na extremidade direita da Figura 2.2, se utiliza de tarefas realizadas nas duas classes previamente apresentadas para resolver problemas específicos de categorização e de extração da informação, como o REM, a detecção de relações e correferências e a categorização de textos.

A próxima seção apresenta em detalhes as principais características da área de EI.

2.2 – EXTRAÇÃO DA INFORMAÇÃO

Extração da informação (EI) é o nome dado ao campo associado diretamente às pesquisas da descoberta do conhecimento que atua na etapa de pré-processamento da MT, mais especificamente como representante de tarefas dependentes do problema, apresentadas previamente na Figura 2.2, dedicadas a extrair informações específicas de textos não estruturados, como os nomes das entidades neles contidos, as possíveis relações existentes entre esses nomes e a categorização na qual determinado texto se enquadra. A utilização de tarefas associadas à EI é condicionada ao tipo de informação que se deseja extrair. Segundo os autores de (Weiss et al., 2005), as principais tarefas associadas à EI são: tokenização e segmentação de sentenças, análise e etiquetação morfossintática, interpretação semântica, interpretação do discurso, preenchimento de *templates* e o próprio REM. Os autores informam também ser comum haver interdependência entre essas tarefas, conforme apresentado na Figura 2.3, onde se pode observar, por exemplo, que sistemas de REM, assim como de respostas a perguntas, podem depender das tarefas de etiquetação morfossintática, de análise sintática e de etiquetação semântica.

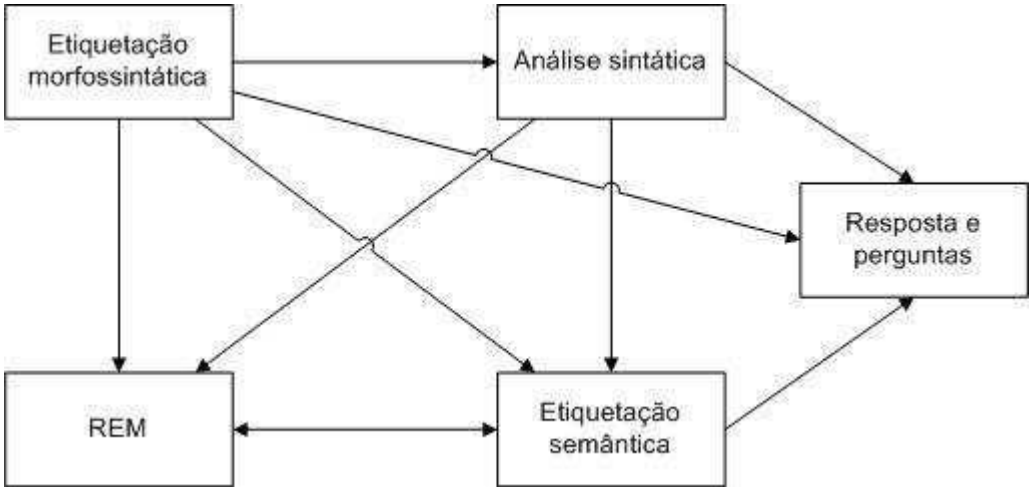


Figura 2.3 - Relações de interdependência entre as tarefas do PLN (Weiss et al., 2005)

Na ótica da autora em (Sarawagi, 2008), a EI objetiva transformar a informação não estruturada em estruturada e, apesar de ser um tópico que surgiu recentemente, associado à tarefa de REM em eventos de avaliação conjunta como o MUC (Grishman e Sundheim, 1996) e o ACE (ACE Group, 2000), em pouco tempo passou a compor diversas áreas acadêmicas, como a inteligência artificial, o aprendizado de máquina, a recuperação da informação, banco de dados, a web e a análise de documentos.

Em (Feldman e Sanger, 2007), os autores afirmaram que as técnicas de EI são indispensáveis durante as operações de pré-processamento textual e ressaltam haver diferenças entre a EI e a RI (recuperação da informação), pois enquanto que a RI retorna documentos com base em determinada busca, o que exige posterior interação humana para leitura e interpretação dos documentos resultantes, a EI retorna informações relevantes em formato estruturado, prontas para serem apresentadas ao usuário ou utilizadas como entrada para outras tarefas de mineração.

Ainda segundo Feldman e Sanger, a EI pode ser vista como uma forma limitada da compreensão completa do texto, por anotar as entidades e seus inter-relacionamentos (fatos ou eventos) e poder, com isso, inferir importantes conceitos semânticos. Eles afirmam que existem quatro tipos básicos de elementos que podem ser extraídos dos textos não estruturados: as entidades, os atributos, os fatos e os eventos. As **entidades** são consideradas os elementos básicos, representam os nomes mencionados nos textos e podem ser classificadas como pessoa, organização, local, gene, etc. Os **atributos** são

características relacionadas às entidades, como a idade, o sexo e a cor da pele de uma pessoa. Os **fatos** são as relações estáveis existentes entre as entidades, como a empresa onde uma pessoa trabalha; e os **eventos** são atividades ou ações associadas às entidades em um determinado momento, como a participação de uma pessoa em uma competição de natação ou em um assalto a banco.

Em última análise, o objetivo principal dos sistemas de EI é a predição, ou seja, dado um conjunto de treinamento, ou um conjunto de regras específicas, e um conjunto de teste, o objetivo é propor uma projeção de sequência textual para o conjunto de teste com base nos conhecimentos adquiridos com o conjunto de treinamento ou com as regras disponíveis, de forma a identificar e classificar automaticamente as informações específicas contidas nesse conjunto. Desse modo, transforma-se o texto não estruturado em uma informação estruturada que pode ser representada na forma de planilha, conforme ilustra o exemplo da Figura 2.4, que apresenta o resultado de uma tarefa de EI que extraiu três informações de um texto não estruturado, identificou as relações existentes entre elas e as representou em formato tabular estruturado.

Figura 2.4 - Representação de um texto não estruturado em forma de planilha

2.3 - *FEATURE*

As *features*[5] são informações sobre as características dos elementos textuais e são usadas na MT como forma de enriquecer o valor informativo que se tem sobre um texto.

[5] No presente trabalho será usado o termo na língua inglesa, a fim de evitar ambiguidades com a tradução do termo para a língua portuguesa.

Segundo os autores em (Feldman e Sanger, 2007), as *features* podem ser divididas de acordo com quatro níveis principais: caracteres, palavras, termos e conceitos. Para exemplificar, na frase "*Ganso (Paulo Henrique) será o melhor jogador da copa de 2014"*, os parêntesis podem ser *features* em nível de **caractere** usadas para indicar quebra da sequência textual; o numeral "*2014*" pode inferir que a **palavra** indica um ano; o **termo** "*copa de 2014*" pode facilitar a interpretação de se tratar de um evento e, apesar da sentença não conter a palavra "*futebol*", rotinas específicas de pré-processamento são capazes de enquadrá-la no **conceito** futebolístico. Os autores de (Feldman e Sanger, 2007) denominam "seleção de *features*" a etapa de pré-processamento de texto que remove palavras irrelevantes e, por outro lado, evidencia as relevantes. Outra forma de interpretar as *features* é como a representação de um texto não estruturado como um conjunto estruturado e bem definido de palavras e suas características, em forma de planilha (Weiss et al., 2005).

Segundo (Sarawagi, 2008), as *features* associadas aos *tokens* são obtidas através das seguintes formas:

- A própria *string* que forma a palavra;
- Significado ortográfico do *token*, como o fato da palavra iniciar com letra maiúscula, ser composta somente por letras maiúsculas, ser um numeral, conter caracteres especiais, ser uma pontuação, etc.;
- Classificação gramatical da palavra, como verbo, substantivo ou complemento, por exemplo;
- Identificação dos dicionários ou listas (*gazetteers*) as quais a palavra pertence, bem como a posição da palavra quando se tratar de entidade composta; exemplos de dicionários e listas são locais geográficos, nomes de pessoas, empresas, proteínas, enzimas, etc.;
- Anotações realizadas em etapas anteriores de processamento do mesmo *token*;
- Lista de *tokens* vizinhos, com o objetivo de contextualizar o *token*.

No presente trabalho, um dos desafios é a adequação das *features* ao contexto forense, principalmente pelo fato de não ser possível prever padronização em textos extraídos de mídias computacionais apreendidas.

2.4 - DOMÍNIO

O domínio ao qual a coleção de textos pertence é outro fator relevante para as tarefas da MT. Um conhecimento prévio do domínio, também citado na literatura como conhecimento de base (do inglês *background knowledge*) (Feldman e Sanger, 2007), permite que haja um melhor direcionamento de *features* e regras de pré-processamento de modo a facilitar a interpretação dos conceitos presentes em uma coleção. Um domínio pode englobar grandes áreas (como a biomedicina ou notícias de jornais) ou áreas mais restritas (como genes e proteínas ou notícias financeiras). A vantagem dos sistemas de MT de domínio específico em relação àqueles independentes do domínio é a possibilidade da utilização de fontes de conhecimento externo para enriquecer o modelo de mineração (Feldman e Sanger, 2007), uma vez que as características linguísticas dos textos não estruturados, em geral, possuem relação com o domínio. O pré-processamento de textos da área biomédica, por exemplo, pode predefinir regras e *features* específicas para a identificação de nomes de genes, enzimas e proteínas, além de facilitar a análise de tendências de mutações com auxílio de conhecimento externo associado ao domínio em questão.

Os tipos de texto que a presente pesquisa objetiva etiquetar não são associados a um domínio específico. Dentro de uma mídia apreendida pode conter qualquer tipo de texto não estruturado, desde um contrato formal de licitação pública contendo diversos termos jurídicos até uma conversa instantânea por aplicativo de bate-papo contendo, por exemplo, símbolos como "*vc*", "*kd*" e "*to*", que representam, respectivamente, as palavras você, cadê e estou do idioma português. A tarefa de REM independente de domínio é ainda pouco explorada na comunidade acadêmica e, segundo os autores em (Louis e Engelbrecht, 2011), isso tem forte relação com a limitação das coleções textuais disponíveis publicamente quanto ao domínio e idioma dos seus textos.

O presente trabalho aborda o Reconhecimento de Entidades Mencionadas (REM), que, conforme já mencionado, pertence à subárea Extração da Informação (EI) da MT. O REM é uma tarefa baseada na etiquetação de sequências textuais que ocorre especificamente na etapa de pré-processamento da MT.

3 - RECONHECIMENTO DE ENTIDADES MENCIONADAS

3.1 – DEFINIÇÃO

Dentre as tarefas pertencentes à área de EI, destaca-se o Reconhecimento de Entidades Mencionadas (REM). O REM objetiva identificar e classificar os nomes das entidades contidas em um texto não estruturado. As principais entidades encontradas nos textos são: pessoa, organização, local, tempo e outros nomes aplicados a domínios específicos, como proteína e enzima no domínio biomédico. Além dessas, é comum também a existência de trabalhos de REM focados em expressões temporais e numéricas, como percentagem e valor monetário. A Figura 3.1 exemplifica a tarefa de REM através da etiquetação de um texto que contém nomes das entidades pessoa (*PES*), organização (*ORG*), local (*LOC*), tempo (*TPO*) e valor monetário (*VAL*).

As **<ORG>**Organizações Pedras Preciosas LTDA**</ORG>** foram vendidas para o **<PES>**Sr. Fulano dos Santos Jr.**</PES>** por **<VAL>**R$200.000,00**</VAL>**, o que levou à mudança da sua sede de **<LOC>**São Paulo**</LOC>** para o **<LOC>**Rio de Janeiro**</LOC>** em **<TPO>**2011**</TPO>**.

Figura 3.1 - Exemplo da tarefa de REM aplicada a um texto

Na Figura 3.1 é possível identificar algumas dificuldades que são inerentes à tarefa de reconhecer entidades em sequências textuais, tais como a identificação, por exemplo, das palavras "*Pedras*" e "*Preciosas*" que compõem a entidade *organização* e não são substantivos; a palavra "*Santos*" se refere à entidade *pessoa* e não a um time de futebol ou substantivo, e "*Paulo*", que deve ser interpretado como cidade e não pessoa.

Os primeiros estudos associados à extração de nomes próprios em textos foram apresentados em (Rau, 1991), entretanto o termo Entidade Mencionada (EM)[6], tornou-se conhecido durante o sexto *Message Understanding Conference* (MUC-6) (Grishman e Sundheim, 1996), que influenciou bastante as pesquisas de EI na década de 90. A partir deste evento, a atividade de reconhecer nos textos nomes de pessoas, organizações, locais,

[6] Tradução livre do termo em inglês *named entity.*

expressões numéricas como tempo, quantia monetária e percentagens foi denominada de Reconhecimento de Entidades Mencionadas (REM)[7].

O REM pode ser caracterizado como um problema de classificação, cujo objetivo é atribuir para cada valor de entrada uma classe, identificada por um nome de EM (Grishman e Sundheim, 1996). Na forma clássica de REM, os valores de entrada são representados pelas palavras ou termos de um texto, denominados *tokens*, e a EM representa a classe ou rótulo associado ao *token*. Por exemplo, no texto da Figura 3.1 a sequência de *tokens* "*São Paulo*" está associada à classe *LOC*. Outra forma de entendimento é tratar o REM como um problema de predição de sequência de estados (Weiss et al., 2005). Neste caso, dada uma sequência *X* de *n tokens* de entrada, o objetivo é inferir a sequência *Y* de *n* estados de saída correspondente, onde y_i é classe de x_i, $0 < i \leq n$. A Figura 3.2 representa as sequências *X* e *Y* associadas ao texto da Figura 3.1, sendo que os elementos da sequência *Y* são representados pelas iniciais *B*, *I* e *O*, referentes à identificação de *tokens* situados na primeira posição de uma EM (*Begin*), situados em qualquer outra posição de uma EM (*Inside*) e não pertencentes a uma EM (*Outside*). As identificações *B* e *I* são acompanhadas da classificação da EM (*PES*, *ORG*, *LOC*, *TPO* ou *VAL*).

X={	As	;Organizações	;Pedras	;Preciosas	;LTDA	;foram	;vendidas	;para	;o	;Sr.	
	;Fulano	;dos	;Santos	;Jr.	;[...]	;Janeiro	;em	;2011	;.	}	
Y={	O	;B-ORG	;I-ORG	;I-ORG	;I-ORG	;O	;O	;O	;O	;B-PES	
	;I-PES	;I-PES	;I-PES	;I-PES	;[...]	;I-LOC	;O	;B-TPO	;O	}	

Figura 3.2 - Sequências *X* e *Y* representando a tarefa de REM associada ao exemplo da Figura 3.1

Analisando a Figura 3.2, observa-se, por exemplo, que *X* contém o valor "*Organizações*", que está associado ao valor *B-ORG* de *Y*, indicando que o *token* "*Organizações*" se encontra na primeira posição (B) de uma EM no texto do tipo *ORG*.

3.2 – MODELOS DETERMINÍSTICOS E PROBABILÍSTICOS

As características predominantes da maioria dos sistemas e modelos de REM já propostos possibilitam a divisão em duas categorias: a determinística, baseada em regras manuais, e a

[7] Tradução livre do termo em inglês *Named Entity Recognition (NER).*

probabilística, através do aprendizado de máquina. Em geral, ambas requerem alto grau de conhecimento linguístico, seja para desenhar as regras manuais ou para modelar os algoritmos de aprendizado (Sarawagi, 2008).

Os modelos **determinísticos** baseados em regras manuais formam a base dos primeiros sistemas de REM (Rau, 1991). A sua concepção, em geral, é baseada na utilização de expressões regulares criadas manualmente que representam regras linguísticas associadas às palavras, como características gramaticais, ortográficas ou de vocabulário. A etiquetação é realizada de forma direta a cada associação existente entre palavras e regras. Na sequência textual "*[...] disse que o senhor Júlio fraudou o documento [...]*", por exemplo, um modelo que contenha a regra "*se a palavra é precedida pelo pronome 'senhor(a)' e é iniciada com letra maiúscula, então é uma EM do tipo 'pessoa'*" etiquetaria o *token* "*Júlio*" com a EM "*pessoa*". Esses modelos possuem a vantagem de não necessitar de coleções de dados etiquetados para treinamento, pois não há qualquer aprendizado de máquina, entretanto requerem maior esforço de desenvolvimento e manutenção das regras, pela sua forte dependência das propriedades linguísticas associadas ao idioma dos textos.

Segundo (Sarawagi, 2008), um sistema baseado em regras clássico é composto por duas partes: um conjunto de regras e um conjunto de políticas para resolver os conflitos entre as regras. O conjunto de regras pode ser definido manualmente ou através de textos de exemplos etiquetados. Em geral, cada regra possui a forma "padrão $\rightarrow$ ação", ou seja, na medida em que determinado texto é processado por um algoritmo de EI baseado em regras, os *tokens* e suas *features* são comparados aos padrões que compõem o conjunto de regras e, caso o resultado dessa comparação seja positivo, a ação da regra é executada. Em geral, o formato de representação das regras é baseado em linguagens formadas por expressões regulares. A Figura 3.3 utiliza o mesmo exemplo citado no parágrafo anterior para ilustrar o processo de EI baseado em regras manuais.

Texto analisado: "*[...] disse que o senhor Júlio fraudou o documento [...]*"

Formato das regras: *padrão* → *ação*

Funcionamento do sistema: a cada palavra, da esquerda para a direita, são testados todos os *padrões* de forma sequencial e, ao primeiro teste que retornar verdadeiro, aplica-se a *ação* correspondente.

No texto do exemplo em tela, quando forem testados os padrões com a palavra "*Júlio*" e chegar a vez do teste da expressão regular

<Sr.|senhor precede X>
E → *(X é EM do tipo PESSOA)*
<X inicia com letra maiúscula>

será retornado *verdadeiro*, pois a palavra "*Júlio*" é precedida pela palavra "*senhor*" e é iniciada com letra maiúscula, e então será aplicada a ação correspondente à regra, ou seja, o sistema predirá que a palavra "*Júlio*" é uma EM do tipo *pessoa*.

Figura 3.3 - Exemplo de aplicação de um sistema de EI baseado em regras manuais

Segundo os autores em (Zhou e Su, 2002), uma vantagem dos modelos baseados em regras manuais em relação aos probabilísticos é o fato da expertise humana ser capaz de capturar evidências (internas e externas) de problemas de REM de forma mais objetiva e eficiente do que as coleções de treinamento o fazem. Assim, justifica-se que alguns modelos determinísticos podem superar alguns modelos probabilísticos, como ocorreu em competições realizadas no MUC-6 (Grishman e Sundheim, 1996) e no MUC-7 (Chinchor, 1998).

Os **algoritmos probabilísticos** baseiam-se no estudo quantitativo dos exemplos positivos e negativos contidos em coleções textuais de treinamento (etiquetadas) para modelar um sistema estocástico que objetiva inferir a identificação e classificação das EM contidas em um texto-alvo (Feldman e Sanger, 2007). A precisão desses modelos está diretamente relacionada à quantidade de palavras, à qualidade da etiquetação, ao idioma e ao domínio das suas coleções de treinamento, que são os conjuntos de textos cujas EM são previamente etiquetadas e preferencialmente revisadas manualmente, usados para o treinamento do modelo.

Segundo autores de (Chang et al., 2006), os modelos probabilísticos maximizam a reusabilidade e minimizam o custo de manutenção. Entretanto, se por um lado essa abordagem tende a reduzir muito o tempo gasto com o desenvolvimento e manutenção do

sistema quando comparada às técnicas determinísticas, por outro requer grande quantidade de textos etiquetados para o treinamento.

Quanto ao aprendizado de máquina, os modelos probabilísticos podem ser **supervisionados**, quando dependem de grandes volumes de textos etiquetados, **semissupervisionados**, quando pouca informação etiquetada é suficiente para iniciar o modelo, ou **não supervisionados**, quanto independem de qualquer etiquetação prévia (Nadeau e Sekine, 2007). Com o intuito de maximizar os níveis de precisão para que sejam próximos aos níveis dos modelos supervisionados, os demais modelos (semissupervisionado e não supervisionado) se utilizam de métodos complementares que objetivam o reconhecimento de padrões, tais como: a exploração do contexto associado às entidades etiquetadas (Riloff e Jones, 1999), a generalização de palavras através de classes semânticas pré-estabelecidas (Pasca et al., 2006), a identificação de padrões de repetição de EM em certos domínios textuais (Shinyama e Sekine, 2004) e a similaridade de contexto entre grupos usando técnicas de agrupamento (do inglês *clustering*) em textos não etiquetados (Miller et al., 2004).

3.3 – PRINCIPAIS MODELOS PROBABILÍSTICOS

Os principais modelos probabilísticos utilizados para a tarefa de REM existentes são o *Hidden Markov Model* (HMM), o *Maximum Entropy Markov Model* (MEMM), o *Conditional Random Fields* (CRF) e o *Support Vector Machine* (SVM).

O HMM (Rabiner, 1990) é o modelo-base utilizado no presente trabalho, caracteriza-se por ser um modelo generativo que usa o teorema de *Bayes* (Russell e Norvig, 2010) para descrever a probabilidade de junção entre a entrada X e a saída Y através da geração probabilística de Y como função de X, da forma $P(Y|X) = P(Y) \cdot P(X|Y)$. No caso da tarefa de REM, X representa a sequência observável (*tokens* e *features*) e Y é a identificação (*EM* ou *não-EM*) e classificação (*PES*, *ORG*, *LOC, VAL, TPO*, etc.) de X. Um exemplo das sequências X e Y pode ser visto na Figura 3.4.

X=(	O	;João	;desviou	;R$50.000,00	;do	;Senado	;em	;2007	)
Y=(	N	;PES	;N	;VAL	;N	;ORG	;N	;TPO	)

Figura 3.4 - Sequências X e Y representando a tarefa de REM

O MEMM (McCallum et al., 2000) e o CRF (Lafferty et al., 2001) são modelos discriminativos, nos quais $P(Y|X)$ é modelado de forma direta (para cada elemento da sequência X, dispõe-se da distribuição probabilística de Y) e a probabilidade de transição entre dois estados consecutivos atinge o máximo global, pois é dependente de toda a sequência X, e não somente da vizinhança. O fato de não ser necessária a modelagem de $P(X)$ implica na possibilidade de inclusão de um grande número de *features* no modelo.

Por fim, o SVM (Vapnik, 1998) é um caso especial de rede neural, cujo classificador é baseado em propriedades geométricas para computar o hiperplano que melhor separa exemplos de treinamento e teste, representados no hiperespaço através de vetores binários. Essa característica possibilita, assim como ocorre no MEMM e CRF, a inclusão de um número muito grande de *features* no modelo.

Segundo os autores em (Ng e Jordan, 2002), o desempenho dos modelos generativos e discriminativos está associado ao tamanho dos corpora de treinamento, de modo que pequenos corpora favorecem os modelos generativos e, à medida que o seu tamanho vai crescendo, a tendência é que os discriminativos se sobressaiam melhor. Os autores em (Yakhenko et al., 2007) propuseram a criação de um modelo probabilístico híbrido (HMM+CRF) em virtude de uma limitação do CRF também associada à necessidade de grande volume de treinamento para alcançar boa acurácia[8]. Ainda, estudos realizados pelos autores de (Milidiú et al., 2007) concluíram que o HMM é capaz de alcançar acurácia superior ao SVM na tarefa de REM em situações de desconhecimento prévio do domínio e de detalhes linguísticos do texto a ser etiquetado.

Assim, devido às características de pouco conhecimento do domínio dos textos contidos em computadores apreendidos e da carência de textos etiquetados para o REM na língua portuguesa, o presente trabalho abordou a utilização de um algoritmo baseado no HMM.

[8] O termo acurácia se refere às medidas de precisão e revocação, utilizadas para avaliar sistemas de REM. Essas medidas são apresentadas no Capítulo 7.

3.4 – ESPECIFICIDADES

Esta seção aborda alguns assuntos que exercem influência direta no desempenho de um sistema de REM, são eles: *features*, conhecimento externo, domínio, idioma, ambiguidade, correferência e não cobertura.

3.4.1 – *Features*

Conforme apresentado no capítulo 2, *features* são informações sobre as características dos elementos textuais e são usadas como forma de enriquecer o valor informativo que se tem sobre um texto. Diversos trabalhos mostram que a escolha das *features* afeta de forma direta o desempenho dos modelos de REM e que não há um padrão pré-determinado para a escolha das *features* utilizadas, pois a sua adequação ao modelo depende do domínio, do idioma e dos tipos de EM que se deseja extrair dos textos.

No ano de 1996, o autor em (McDonald, 1996) descreveu a importância das *features* ao afirmar que a solução para os problemas de ambiguidade, robustez e portabilidade relacionados à tarefa de identificar e categorizar nomes próprios estava associada ao que ele denominou "evidências internas e externas". Estas evidências são, respectivamente, as *features* extraídas do contexto textual das palavras próximas ao *token* corrente e as *features* obtidas de fontes externas ao texto que está sendo etiquetado.

Estudos em (Zhang e Johnson, 2003) concluíram que, para sistemas de REM independentes da língua, é preferível a utilização de *features* simples baseadas em *tokens* à utilização de *features* linguísticas complexas, que envolvem outros elementos textuais como sentenças, parágrafos e informações de contexto, pois estas últimas apresentam maior dificuldade de adaptação em línguas diferentes. O trabalho alerta ainda quanto ao fato da utilização de *features* mais simples, e consequentemente mais genéricas, ocasionar maior dificuldade na obtenção de precisão em sistemas de REM independentes da língua e conclui que, em geral, os resultados obtidos com a utilização de *features* simples são considerados competitivos com os de *features* complexas.

Em trabalho apresentado no CoNLL'09 (Computational Linguistics & Psycholinguistics Research Center, 2009), os autores em (Ratinov e Roth, 2009) alertaram para a importância

do conhecimento externo de informações e da utilização de *features* não locais para melhorar medidas de precisão e revocação dos modelos de REM. A *features* não locais consideram as múltiplas ocorrências de EM no texto avaliado, enquanto que o conhecimento externo permite inferências de informações não contextualizadas no texto, por exemplo, no caso de haver uma citação isolada do *token* "*Garrincha*", sem qualquer referência a times ou esportes, o conhecimento externo pode auxiliar na inferência desta EM ser do tipo pessoa e, mais especificamente, jogador de futebol.

Já os autores em (Mayfield et al., 2003) desenvolveram um método que, com base em 11 tipos de *features* aplicadas às coleções do evento de avaliação CoNLL´03 (Sang e Meulder, 2003), criou um modelo SVM (*Support Vector Machine*) com centenas de milhares de *features*. A ideia principal é incluir um grande número de *features* binárias (para cada *token*, atribui-se à *feature* valor 1 quando existe ou 0 quando não existe) e deixar que o modelo ignore aquelas irrelevantes, através da identificação da não aderência dessas *features* aos *tokens*. Dessa forma, o processo de criação das *features* pode ser genérico a ponto de dispensar o conhecimento da língua a ser treinada e etiquetada, o que torna o sistema independente do idioma. Este modelo apresentou resultados superiores ao do HMM básico e do modelo utilizado como linha de base no CoNLL´03, entretanto foi superado por outros sistemas participantes do evento.

Em (Sang e Meulder, 2003), ao analisarem os sistemas participantes do CoNLL-2003, bem como os resultados por eles obtidos, os autores concluíram que, na tarefa compartilhada de REM independente do idioma, a escolha das *features* teve tanta importância quanto a escolha do modelo de aprendizado de máquina. Apesar de não existir uma regra pré-determinada quanto à escolha das *features* para os sistemas de REM, observou-se que nos sistemas participantes do CoNLL´03 algumas *features* costumavam ser mais escolhidas que outras. A Tabela 3.1 mostra as seis *features* mais utilizadas nesses sistemas. Vale ressaltar que a tarefa compartilhada avaliada neste caso é referente ao REM independente do idioma, com etiquetação de textos nas línguas inglesa e alemã.

Tabela 3.1 - As seis *features* mais utilizadas na tarefa de REM independente do idioma no CoNLL´03 (total de sistemas participantes: 16)

Feature	**Significado**	**Quantidade de sistemas que utilizaram a *feature***
Características léxicas	Presença do *token* em dicionários	15
Características gramaticais	Classificação gramatical do *token*	14
N-gramas de caracteres (afixos)	Primeiros ou últimos *N* caracteres do *token*	13
Classe de EM inferida ao *token* da posição anterior	Classe do *token* que precede o *token* corrente (*PES*, *ORG*, *LOC*, etc.)	12
Características ortográficas	Token iniciado em letra maiúscula, contendo número, contendo pontuação, etc.	12
Gazetteers	Token presente em *gazetteers*	11

3.4.2 – Conhecimento externo

Nesta seção são abordadas as formas de utilização de três fontes de conhecimento externo como fator favorável à melhora do desempenho em modelos de REM. As fontes são: os *gazetteers*, a Wikipédia e as coleções de textos não etiquetados.

Uma das técnicas utilizada para tratar problemas de cobertura e ambiguidade em tarefas de REM é a utilização de *gazetteers*, que são listas externas contendo nomes de entidades e são usadas para comparação com os *tokens* do texto. Listas estáticas, obtidas na web ou em bases de dados específicas, são comumente usadas como *gazetteers* pelo método da simples correspondência com *tokens*, que é a identificação quanto à presença ou ausência do *token* na lista.

Além do método da simples correspondência com *tokens*, tem se tornado frequente o uso de *gazetteers* como *features* em modelos de REM baseados em aprendizado de máquina. Autores em (Cohen e Sarawagi, 2004), (Florian et al., 2003), (Toral e Muñoz, 2006) e (Kazama e Torisawa, 2007) obtiveram melhores resultados com essa abordagem, que alia a flexibilidade dos modelos estatísticos à precisão dos *gazetteers*.

Há também pesquisas associadas à utilização da Wikipédia como fonte de informações para a construção de *gazetteers*. Os trabalhos de (Toral e Muñoz, 2006) e (Kazama e Torisawa, 2007), mencionados no parágrafo anterior, reportam resultados com melhor revocação através da utilização da Wikipédia. Já os autores em (Ratinov e Roth, 2009) desenvolveram um modelo que utilizou 16 gazetteers baseados na Wikipédia e outros 14 baseados em listas de nomes comuns, e atribuiu pesos para esses *gazetteers* tornando-os *features* do sistema. Os experimentos concluíram que a incorporação de *gazetteers* como features nos modelos tendem a melhorar a acurácia na tarefa de REM.

O sistema determinístico Rembrandt (Cardoso, 2008), que obteve o melhor desempenho na tarefa de REM para os tipos *pessoa* e *organização* na avaliação conjunta do segundo HAREM (Mota e Santos, 2008), é baseado em regras manuais e na Wikipédia. Segundo o seu autor, a Wikipédia possui algumas propriedades que favorecem a sua utilização em *gazetteers*. Dentre estas propriedades, destacam-se o fato de ser uma enciclopédia digital aberta e colaborativa, de possuir atualização frequente de EM e de conter páginas de redirecionamento, o que gera um mapeamento $1 \rightarrow n$ entre determinado *token* e as suas possíveis variações. Por exemplo, o *token* "*Sena*" pode ser redirecionado para a página da Wikipédia "*Ayrton Senna*", entidade do tipo *pessoa* (ex-corredor automobilístico brasileiro), ou para a página "*Rio Sena*", entidade do tipo local (rio que banha Paris), a depender do contexto associado ao *token*.

Entretanto, vale ressaltar que, como o trabalho proposto objetiva o reconhecimento de nomes de pessoas e organizações contidos em mídias forenses, a utilização da Wikipédia não é indicada, pois a maior parte dos nomes considerados relevantes para o trabalho não são referentes a pessoas ou organizações amplamente conhecidas, assim não possuem entrada publicada na Wikipédia.

Por fim, os autores em (Miller et al., 2004) apresentaram melhor desempenho de sistemas de REM através da utilização de técnicas semissupervisionadas de agrupamento (*clustering*) aplicadas a textos externos não anotados. A técnica utilizada é baseada em modelos de classes de palavras e se resume à criação de grupos (*clusters*) de *tokens* representados por uma árvore binária hierárquica, sendo que os *tokens* se situam nas folhas e cada nível de profundidade da árvore agrupa um nível de abstração de *tokens*. O quarto nível de profundidade da árvore, por exemplo, indica um agrupamento similar ao por

classe gramatical, e o fato desses agrupamentos enriquecerem o contexto dos *tokens* e das *features* implica de forma direta em uma melhor precisão do modelo de REM.

3.4.3 – Domínio e Idioma

Os principais fatores que impactam no desempenho de um sistema de REM estão associados ao domínio, ao idioma textual e a problemas intrínsecos da escrita, como a ambiguidade, a não cobertura e a correferência. A ambiguidade está relacionada ao fato de um mesmo token poder ser associado a mais de uma classe de EM, a não cobertura representa a ausência do *token* analisado no conjunto de regras ou no corpus de treinamento do modelo de REM e a correferência se refere à identificação de *tokens* que não são EM, mas fazem referências a outros *tokens* classificados como EM, como é frequente ocorrer com os pronomes pessoais (*ele*, *nós*, etc.) e relativos (*dele*, *nosso*, *aquele*, etc.), por exemplo.

3.4.3.1 – Domínio

A criação de sistemas de REM independentes de domínio é um dos desafios enfrentados por especialistas da área. Segundo os autores em (Weiss et al., 2005), quanto mais específico for o domínio da coleção de documentos, maior será a acurácia de um sistema de extração da informação, uma vez que o ambiente se torna mais controlado e tanto os algoritmos determinísticos quanto os probabilísticos conseguem prevenir mais facilmente a ocorrência de problemas como a não cobertura e a ambiguidade.

Segundo os autores em (Louis e Engelbrecht, 2011), as coleções textuais públicas, etiquetadas ou não, são limitadas ao domínio e ao idioma dos seus textos, e isso contribui para o estado incipiente no qual os trabalhos na área de EI independente do domínio se encontram. No cenário prático, as ferramentas de processamento de sequências textuais em geral requerem interação de usuário quando há necessidade de mudanças associadas ao domínio. Alguns trabalhos estão sendo desenvolvidos na área de adaptação de domínio, como em (Chiticariu et al., 2010) e (Guo et al., 2009), cujo objetivo é, dado um modelo de REM treinado para etiquetar textos-alvo pertencentes a determinado domínio, adaptá-lo para a etiquetação em outro domínio. Estas propostas são, portanto, aplicáveis a textos

pertencentes a domínios específicos, por isso não resolvem o problema da etiquetação independente do domínio.

Autores em (Edward et al., 2008) propuseram um sistema de REM baseado no modelo probabilístico MEMM utilizando um algoritmo genético para a otimização das *features* locais e globais de acordo com os diferentes domínios dos textos-alvo. O sistema foi avaliado usando o *corpus* de notícias disponível na tarefa compartilhada do CoNLL'03 (Sang e Meulder, 2003) para treinamento e um corpus jurídico para testes, e vice-versa. Os resultados mostraram que a utilização das *features* otimizadas representou um ganho da ordem de 1% a 2% em relação à sua não utilização, porém o sistema alcançou somente 70% de precisão nos testes treinados com o corpus jurídico, enquanto que sistemas participantes do CoNLL'03 alcançaram precisão superior a 88% (Sang e Meulder, 2003). Esses números comprovam que a etiquetação de REM independente de domínio é um desafio ainda em aberto para a área de EI.

3.4.3.2 – Idioma

O problema da carência de coleções textuais públicas etiquetadas na língua portuguesa (Louis e Engelbrecht, 2011) aliada às dificuldades linguísticas inerentes do idioma são as causas principais da reduzida contribuição científica existente na área de REM para o português.

Com vistas a suprir essa deficiência, em 1998 surge em Portugal o projeto "Processamento Computacional do Português" (Santos, 2000), que futuramente viria a se chamar projeto Linguateca (Santos et al., 2004). Desde então, este grupo fomentou duas avaliações conjuntas para REM na língua portuguesa, denominadas primeiro HAREM (Santos et al., 2006) e segundo HAREM (Carvalho et al., 2008), para as quais foram criados dois corpus etiquetados para treinamentos e testes, denominados Coleção Dourada 1 (Rocha e Santos, 2007) e Coleção Dourada 2 (Mota e Santos, 2008). Esses corpora, os métodos de avaliação usados e os resultados obtidos nos eventos são considerados referência mundial para o REM na língua portuguesa e por isso foram utilizados como balizadores da avaliação da solução proposta do presente trabalho.

Além dos dois corpora publicados pelo projeto Linguateca, há um terceiro corpus produzido no idioma português e etiquetado com EM, denominado *TagShare* (Barreto et al., 2006), entretanto o mesmo não é disponibilizado de forma gratuita, por isso não pôde ser avaliado junto à solução proposta.

Além da disponibilização dos corpora etiquetados especificamente para a tarefa de REM, as avaliações conjuntas do primeiro e segundo HAREM contribuíram para o surgimento de sistemas de REM na língua portuguesa. As propostas que obtiveram os melhores desempenhos nesses eventos são baseadas em regras gramaticais determinísticas, a exemplo dos sistemas Priberam (Amaral et al., 2008), que obteve o melhor desempenho geral, e Rembrandt (Cardoso, 2008), que obteve os melhores desempenhos específicos para o REM das categorias de EM *pessoa* e *organização*, que são as categorias relevantes para o presente trabalho.

3.4.3.3 – Ambiguidade, correferência e não cobertura

Conforme apresentado no início da Seção 3.4.3, o problema da ambiguidade está relacionado ao fato de um mesmo *token* poder ser associado a mais de uma classe de EM, a correferência está associada à identificação de *tokens* que não são EM mas fazem referências a outros *tokens* classificados como EM, e a não cobertura representa a ausência do *token* analisado no conjunto de regras ou no corpus de treinamento do modelo de REM, o que dificulta a sua classificação.

As **<ORG>**Organizações Pedras Preciosas LTDA**</ORG>** foram vendidas para o **<PES>**Sr. Fulano dos Santos Jr.**</PES>** por **<VAL>**R$200.000,00**</VAL>**, o que levou à mudança da sua sede de **<LOC>**São Paulo**</LOC>** para o **<LOC>**Rio de Janeiro **</LOC>** em **<TPO>**2011**</TPO>**.

Figura 3.5 - Exemplo de texto etiquetado por um sistema de REM

Na Figura 3.5, observa-se exemplos de ambiguidade nos *tokens* "*Santos*", "*Paulo*" e "*Janeiro*", pois os mesmos poderiam representar, respectivamente, um time de futebol, uma pessoa e um mês do ano. Há também correferência associada ao pronome relativo "*sua*". Se for necessário que o sistema de REM classifique esse pronome, o sistema deve ser capaz de identificar que ele é referente à organização "Organizações Pedras Preciosas LTDA", e não à pessoa "Sr. Fulano dos Santos Jr.". Por fim, supondo-se a utilização de um

modelo probabilístico para etiquetar o texto da Figura 3.5, os exemplos de treinamento poderiam não conter a "*Fulano*", o que dificultaria a sua classificação como EM *pessoa*.

Alguns trabalhos, como em (Todorovic et al., 2008) e (Riloff e Jones, 1999), propõem a exploração do contexto do *token* como forma de ampliar as informações que o modelo possui sobre o *token* e as palavras próximas a ele. Essa abordagem contribui para a desambiguação e a solução do problema da correferência do *token*, uma vez que as informações do seu contexto contêm elementos que podem associar o *token* a uma classe de EM ou a outro token, pertencente ao contexto, que tenha sido classificado previamente.

Outra proposta para se minimizar o problema da ambiguidade é o uso de informações não locais, apresentado em (Ratinov e Roth, 2009), uma vez que a classificação de um *token* pode contribuir para a desambiguação de novas ocorrências do mesmo *token* no texto.

O problema da não cobertura está diretamente relacionado à insuficiência de regras criadas, no caso de modelos determinísticos, ou de exemplos de treinamento, no caso dos probabilísticos. A insuficiência de regras requer a atualização do conjunto de regras do modelo, o que é um problema enfrentado pelos modelos determinísticos, especialmente nos casos de adaptação a novos domínios ou idiomas (Sarawagi, 2008). Já os modelos probabilísticos de REM, para resolver problemas de insuficiência de treinamento, usam técnicas conhecidas como *back-off*, a exemplo dos trabalhos apresentados em (Bikel et al., 1999) e (Todorovic et al., 2008). Modelos de *back-off* são utilizados para, nos casos de não cobertura, criar mecanismos que classifiquem o *token* não coberto com base em um novo modelo de probabilidades, que é menos preciso, porém garante a sua classificação. Os modelos de *back-off* são apresentados no Capítulo 5.

O presente trabalho utiliza uma adaptação do modelo oculto de Markov (HMM) para reconhecer entidades mencionadas em textos forenses. O HMM é apresentado no capítulo seguinte.

4 - MODELO OCULTO DE MARKOV

O presente capítulo apresenta o modelo oculto de Markov segundo proposto em (Rabiner, 1990) com o foco direcionado à tarefa de etiquetação de sequências textuais. Este modelo é utilizado como base do algoritmo desenvolvido no trabalho proposto.

Os estudos originários dos modelos estatísticos de Markov datam dos anos 60 (Baum e Petrie, 1966), entretanto a sua utilização como solução para problemas computacionais ocorreu muito tempo depois. Esta lacuna ocorreu principalmente devido ao fato dos antigos artigos estarem restritos à comunidade acadêmica matemática, distante dos engenheiros e linguistas que trabalhavam com a fala e a escrita, onde se concentram a maior parte das aplicações associadas aos modelos de Markov (Rabiner, 1990). Os primeiros trabalhos associados à aplicação prática do HMM ocorreram em 1983 para resolver o problema do reconhecimento da fala (Levinson et al., 1983) e o embasamento matemático do modelo estimulou a sua posterior aplicação a diversos outros problemas, como a etiquetação de sequências textuais, o reconhecimento de caracteres e o mapeamento do genoma humano.

A forma de se solucionar determinado problema com o HMM é através da representação dos sinais observáveis do mundo real em um modelo conceitual de sinais, que podem ser discretos, como é o caso das sequências textuais e de dados meteorológicos, ou contínuos, a exemplo da fala e escrita manual cursiva.

4.1 – Modelo de Markov

Um modelo de Markov descreve uma sequência de estados no tempo de forma que no instante t (t = 1, 2, ...,T) o modelo seja representado por um dos N estados distintos S_1, S_2, ..., S_N e que a probabilidade de transição entre S_i e S_j seja representada por a_{ij}.

Este modelo pode ser representado por um grafo direcionado denominado cadeia de Markov, cujos vértices são os N estados distintos e cujas arestas são as probabilidades não nulas de transições entre os estados. A Figura 4.1 ilustra uma cadeia de Markov com quatro estados (N=4),

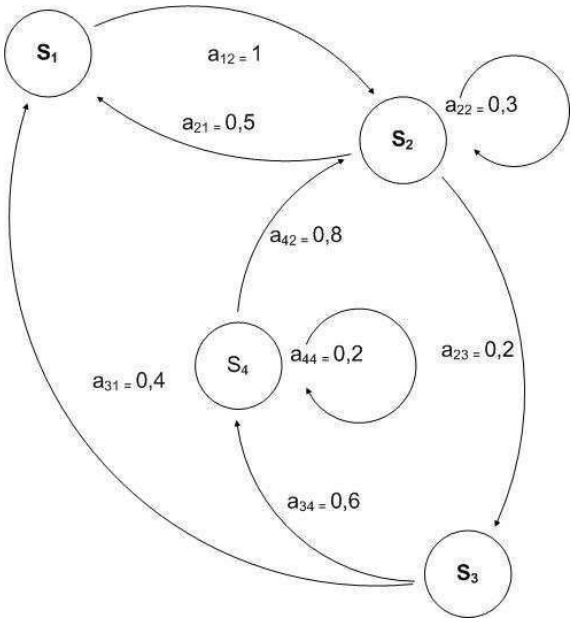


Figura 4.1 - Exemplo de cadeia de Markov

Na cadeia de Markov apresentada na Figura 4.1, os vértices S_1 a S_4 representam os 4 estados que compõem o modelo, enquanto que as arestas a_{ij} representam as probabilidades de transição entre os estados *Si* e *Sj*, ou seja, as probabilidades do estado *Sj* ocorrer no tempo *t* e o estado *Si* ocorrer no tempo *t*-1. Por exemplo, a probabilidade de transição entre os estados S_3 e S_1 é de 40% (a_{31} = 0,4) e a probabilidade do estado S2 ocorrer nos instantes *t*-1 e *t* é de 30%. Observa-se ainda que há probabilidade nula de transição entre alguns estados, como entre S_4 e S_3 e entre S_2 e S_4, pois não há aresta conectando esses estados, e que há probabilidade 1 de transição entre S_1 e S_2, o que significa que a única possibilidade de transição partindo do estado S_1 é para o estado S_2.

Denota-se por q_t o estado representante do modelo no tempo *t*. Por exemplo, se $q_3 = S_2$, então é correto afirmar que no tempo *t*=3 o modelo é representado pelo estado S_2. Supondo-se $q_t = S_i$, a cada incremento discreto de *t*, pode haver alteração de estado ou não, a depender das probabilidades de transição de S_i para S_j, denotadas por $a_{ij} = P(q_t=S_j \mid q_{t-1}=S_i)$.

Por exemplo, na Figura 4.1, se no tempo *t-1* o estado representante do modelo é S_4 ($q_{t-1} = S_4$), então, no tempo *t*,

$$q_t = S_j \Leftrightarrow a_{4j} = max_{(1 \le k \le N)} [a_{4k}], \qquad 1 \le j \le N \qquad \text{Eq. (4.1)}$$

ou seja, a transição $S_4 \rightarrow S_j$ será definida pela maior probabilidade de transição a_{4j}.

Considerando-se a cadeia de Markov da Figura 4.1, tem-se que

$$a_{41} = 0,\ a_{42} = 0{,}8,\ a_{43} = 0 \text{ e } a_{44} = 0{,}2\ ,$$

logo, a maior probabilidade de transição é $a_{42} = 0{,}8$, o que significa que o estado representante do modelo no tempo t é S_2, ou

$$q_t = S_2.$$

O modelo de Markov requer especificação acerca da dependência entre o estado corrente q_t e os estados prévios q_{t-1}, q_{t-2}, ..., sendo que, no modelo de primeira ordem a predição de q_t depende exclusivamente de q_{t-1}, no de segunda ordem depende de q_{t-1}, q_{t-2}, e assim sucessivamente. Desta forma, um modelo de Markov de primeira ordem pode ser representado da seguinte forma:

$$P[q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, ...] = P[q_t = S_j \mid q_{t-1} = S_i], \qquad \text{Eq. (4.2)}$$

ou seja, o estado representante do modelo no tempo t depende exclusivamente do estado representante no tempo t-1, consequentemente se assume independência de q_t em relação a $q_{t-2}, q_{t-3}, ..., q_1$.

Sendo assim, o conjunto de probabilidades de transição de estados a_{ij} em um modelo de primeira ordem é definido por

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i], \qquad 1 \le i, j \le N \qquad \text{Eq. (4.3)}$$

e, por se tratar de um modelo estocástico matematicamente válido, tem-se que

$$a_{ij} \ge 0, \qquad 1 \le i, j \le N \qquad \text{Eq. (4.4)}$$

$$\sum_{j=1}^{N} a_{ij} = 1. \qquad 1 \le i \le N \qquad \text{Eq. (4.5)}$$

Aos modelos de primeira ordem é associado o termo **bigrama**, pelo fato das suas distribuições probabilísticas considerarem dois *tokens*, o corrente e o seu predecessor. Em geral, associa-se o termo n-grama a um modelo de ordem n-1.

Este modelo pode ser usado, por exemplo, para estimar o próximo número resultante de uma sequência de lançamentos de um dado viciado ou a quantidade de carros que serão roubados no próximo mês em determinado bairro, nesses casos as saídas geradas (números do dado lançado ou números de roubos) são uma sequência de estados e cada um deles corresponde a um evento físico observável (número da face virada para cima de um dado arremessado ou quantidade mensal de roubos registrados). A esse tipo de processo dá-se o nome de modelo observável de Markov.

4.2 – Modelo Oculto de Markov

Há diversos casos de aplicações não adaptáveis aos modelos observáveis de Markov e o REM é uma delas, pois é uma tarefa de etiquetação textual, onde se tem acesso aos *tokens* (sinais observáveis), entretanto não se possui informação direta sobre as etiquetas (estados), que podem servir para identificar um *token* (Entidade Mencionada ou NÃO Entidade Mencionada) ou para classificá-lo (*pessoa*, *organização*, *local*, etc.). Em cenários como esse, nos quais se dispõe dos sinais observáveis gerados pelos estados, mas não de informações diretas dos estados, que permanecem "ocultos", pode-se usar o Modelo Oculto de Markov, ou HMM. O modelo gráfico do HMM é apresentado na Figura 4.2, que mostra a sequência de estados q_1 q_2 ... q_T, a sequência observável O_1 O_2 ... O_T, linha superior tracejada representa a transição de estados $q_{t-1} \rightarrow q_t$ inferida pelo modelo na linha do tempo, e a coluna tracejada representa os sinais observáveis sendo gerados pelos estados. Devido a essa geração dos sinais observáveis, o HMM é categorizado como modelo generativo.

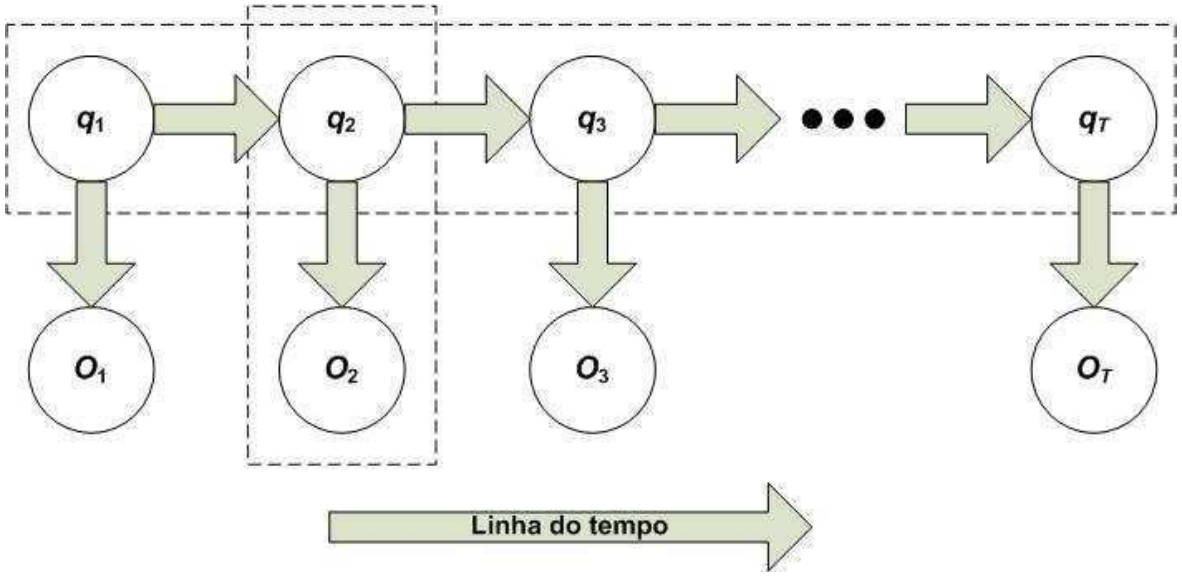


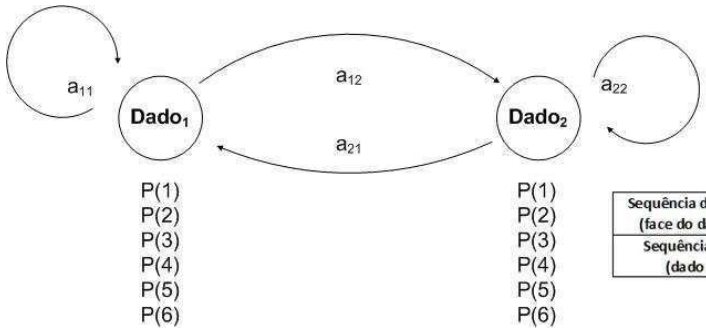
Figura 4.2 - Modelo gráfico do HMM

O HMM pode então ser definido como "um autômato[9] de estados finitos baseado em um modelo estocástico de transição de estados e emissões de símbolos" (Rabiner, 1990).

[9] Máquina, aparelho ou dispositivo que executa determinadas tarefas ou funções.

Conforme apresentado na Seção 4.1, o modelo observável de Markov pode ser exemplificado com uma sequência de lançamentos de um dado viciado para a qual se deseja inferir o resultado do próximo arremesso. Outro objetivo no mesmo cenário seria determinar a probabilidade de uma sequência de lançamentos ocorrer, como a sequência 5-3-4-3-3-2-1-6. Nesses dois casos, o conjunto de estados do modelo (S_1 a S_6 - um para cada número do dado) é o próprio conjunto observável.

Para efeito de comparação entre o modelo observável e o modelo oculto de Markov, altera-se o exemplo de lançamento de um dado para dois dados viciados. Neste caso, o modelo tem dois estados ocultos, um para cada dado, cada estado tem a si associado uma distribuição probabilística de ocorrências dos números 1 a 6, e a transição entre os dois estados seria caracterizada por uma matriz de transição de estados. Deseja-se identificar a sequência de estados mais provável, dado uma sequência observável. A Figura 4.3 ilustra como seria esse modelo.



Tempo →	1	2	3	4	5	...	T
Sequência de observações (face do dado lançado)	3	5	4	4	6	...	3
Sequência de estados (dado lançado)	Dado2	Dado2	Dado1	Dado2	Dado1	...	Dado1

Figura 4.3 - Modelo gráfico do HMM para lançamentos aleatórios de dois dados

Observa-se na figura que o modelo calcula as probabilidades de transição de estados a_{ij} e calcula também, para cada um dos estados ocultos ($Dado_1$ e $Dado_2$), as probabilidades de emissão do sinal observável $P(1)$ a $P(6)$. Esses cálculos são realizados com base nos exemplos de treinamento do modelo. Com essas probabilidades calculadas, o HMM é capaz de solucionar o que se deseja neste exemplo, ou seja, considerando-se dois dados, predizer a sequência mais provável de dados lançados nos instantes 1 a T uma vez que se conhece uma sequência de T números gerada por T lançamentos desses dois dados. A tabela situada à direita da Figura 4.3 mostra que, no exemplo em comento, para a sequência de lançamentos 3, 5, 4, 4, 6, ..., 3, a sequência de estados mais provável é: dado 2, dado 2, dado 1, dado 2, dado 1, ..., dado 1.

O HMM é um modelo generativo (a sequência de observações é gerada pela sequência de estados), e a probabilidade de ocorrência de uma sequência de estados Q, dada uma sequência de observações O, é calculada através do teorema da probabilidade condicional de *Bayes* (Bikel et al., 1999):

$$P(Q\,|O) = \frac{P\,(O,Q)}{P\,(O)}. \qquad \text{Eq. (4.6)}$$

Como o denominador $P(O)$ da equação 4.6 é constante, pois a sequência observável permanece inalterada no tempo, tem-se que a maximização probabilística da sequência de estados Q dada a sequência de observações O é dada pela maximização do numerador $P(O,Q)$, que é a probabilidade de junção entre O e Q, expressa por

$$P(O,Q) = P(O \mid Q) \cdot P(Q), \qquad \text{Eq. (4.7)}$$

ou, quando aplicada ao HMM,

$$P(O,Q \mid \lambda) = P(O \mid Q,\lambda) \cdot P(Q \mid \lambda), \qquad \text{Eq. (4.8)}$$

onde λ representa o modelo HMM.

Conclui-se, com isso, que o HMM é baseado na probabilidade de junção entre O e Q, e que o seu funcionamento requer a construção de três modelos de distribuição probabilística, um associado a $P(O \mid Q)$, que é a distribuição das observações para determinado estado, e dois associados a $P(Q)$, que são as probabilidades dos estados iniciais (em t=1) e as probabilidades de transição de estados.

Esses três modelos compõem a descrição formal do HMM, que é dada pelos seguintes elementos:

- O conjunto V de símbolos observáveis discretos ou contínuos $V = \{v_1, v_2, ..., v_M\}$, de tamanho M, que corresponde ao dicionário de observações físicas do sistema sendo modelado; neste trabalho será utilizada a notação $O = O_1\ O_2\ ..\ O_T$. para representar as sequências observáveis no tempo, no intervalo 1 ... T;
- O conjunto S de estados ocultos discretos $S = \{S_1, S_2, ..., S_N\}$, de tamanho N, que, em aplicações práticas, carrega algum significado associado aos símbolos observáveis; neste trabalho será utilizada a notação $Q = q_1\ q_2\ ..\ q_T$ para tratar de sequências de estados no tempo, no intervalo 1 ... T;
- O vetor B de probabilidades de geração de símbolos observáveis no estado S_j, $B = \{b_j(k)\}$, que corresponde à probabilidade de geração de $v_1, v_2, ..., v_M$ por S_j;

$b_j(k) = P[v_k \text{ no instante } t \mid q_t = S_j], \quad 1 <= j <= N ; 1 <= k <= M$ Eq. (4.9)

- A matriz de transição de estados $A = \{a_{ij}\}$, correspondente às probabilidades de transição entre S_i e S_j;

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i], \quad 1 <= i, j <= N \quad \text{Eq. (4.10)}$$

- A distribuição probabilística inicial dos estados, $\pi = \{\pi_i\}$, que corresponde às probabilidades de ocorrência de $S_1, S_2, ..., S_N$ no instante $t = 1$;

$$\pi_i = P[q_1 = S_i], \quad 1 <= i <= N. \quad \text{Eq. (4.11)}$$

O HMM é definido pela tripla $\lambda = \{A, B, \pi\}$ e a geração da sequência de sinais observáveis ocorre da seguinte forma:

1. Através de π_i, define-se o estado inicial $q_1=S_i$;
2. Define-se tempo $t = 1$;
3. Obtém-se o símbolo observável v_t através de B;
4. Faz-se a transição para o estado seguinte q_{t+1} através de A;
5. Atribui-se $t=t+1$;
6. Se $t = T$, FIM; caso contrário, retorna ao passo 3.

4.3 - Os problemas clássicos do HMM

Os três problemas que o HMM é capaz de solucionar são:

1. **Problema da avaliação**: dada uma sequência de observações $O = O_1 O_2 ... O_T$ e um HMM $\lambda = \{A, B, \pi\}$, como computar $P(O \mid \lambda)$? Ou seja, dada um modelo λ e uma sequência de observações, como computar a probabilidade desta sequência ter sido gerada pelo modelo? A solução para este problema é útil principalmente no caso de existirem vários modelos HMM para determinada aplicação e ser necessário identificar qual o que melhor representa determinada sequência observável;
2. **Problema da sequência ótima de estados**: dada uma sequência de observações $O = O_1 O_2 ... O_T$ e um HMM $\lambda = \{A, B, \pi\}$, como escolher a sequência de estados $Q = q_1 q_2 ... q_T$ com maior probabilidade de ter gerado O? Este é o principal problema a ser resolvido no presente trabalho, ou seja, dada uma sequência textual, qual a sequência de etiquetas (classes de EM) que a representa melhor?
3. **Problema da otimização de parâmetros**: como ajustar os parâmetros A, B e π de forma a maximizar $P(O \mid \lambda)$? Ou seja, como adaptar os parâmetros do modelo ao corpus de treinamento? A solução para este problema é desejável no caso de haver

ruído considerável no corpus de treinamento do modelo ou quando se opta pela abordagem de múltiplos modelos para determinada aplicação.

Os problemas 1 e 3 não se aplicam ao presente trabalho, por isso o detalhamento das suas soluções não será apresentado nesta dissertação. A não aplicação desses problemas à tarefa de etiquetação de EM se justifica pelo fato de não ser necessária a criação de mais de um modelo HMM na solução proposta, uma vez que os corpus de treinamento utilizados são integralmente etiquetados manualmente, o que garante a correta parametrização do modelo. Ademais, ainda que fosse optado pela criação de vários modelos, o tempo investido na comparação entre modelos e na parametrização dos mesmos para cada sequência avaliada tornaria a solução inviável do ponto de vista prático.

Para resolver o problema 2, que é encontrar a sequência de estados ótima associada à sequência observada, deve-se maximizar $P(Q \mid O,\lambda)$, que é igual a $P(Q,O \mid \lambda) = P(Q \mid \lambda) \cdot P(O \mid Q,\lambda)$ (equação 4.8), e a técnica para se chegar a esse objetivo é baseada na programação dinâmica e denominada **algoritmo de Viterbi** (Viterbi, 1967).

De acordo com esse algoritmo, para se determinar a melhor sequência de estados $Q = q_1\ q_2\ ...\ q_T$ para uma sequência de observações $O = O_1\ O_2\ ...\ O_T$, é necessária a obtenção do valor de

$$\delta_t(i) = \max P[\ q_1\ q_2\ ...\ q_t = i\ ,\ O_1\ O_2\ ...\ O_t \mid \lambda\ ], \qquad \text{Eq. (4.12)}$$

que é a máxima probabilidade obtida por uma sequência de estados no tempo t, associada às primeiras t observações e cujo estado final é S_i.

Por indução, tem-se que, em t+1

$$\delta_{t+1}(j) = [\max \delta_t(i) \cdot a_{ij}]\ \cdot b_j(O_{t+1})\ . \qquad \text{Eq. (4.13)}$$

Para que o algoritmo consiga recuperar a sequência de estados ótima, é necessário "memorizar" o estado que maximizou $\delta_{t+1}(j)$ para cada t e cada j. Esses estados são então armazenados no vetor $\psi_t(j)$.

O algoritmo de Viterbi completo para a obtenção da melhor sequência de estados é apresentado no pseudocódigo da Figura 4.4.

1 – Inicialização ($t = 1$)

$$\delta_t(i) = \pi_i \cdot b_i(O_1) , \qquad 1 \leq i \leq N$$

$$\psi_1(i) = 0.$$

2 – Recursão[10] ($2 \leq t \leq T$)

$$\delta_t(j) = max_{(1 \leq i \leq N)} [\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(O_t) , \qquad 1 \leq j \leq N$$

$$\psi_t(j) = argmax_{(1 \leq i \leq N)} [\delta_{t-1}(i) \cdot a_{ij}] , \qquad 1 \leq j \leq N$$

3 – Finalização ($t = T$)

$$q_T = argmax_{(1 \leq i \leq N)} [\delta_T(i)] .$$

4 – Recuperação da sequência (*backtracking*)

$$q_t = \psi_{t+1} (q_{t+1}), \qquad t = T - 1, T - 2, ..., 1 .$$

Figura 4.4 - Pseudocódigo do algoritmo de Viterbi (modificado - (Rabiner, 1990))

Na Figura 4.4, percebe-se que as etapas 1 e 2 percorrem a sequência temporal t, de 1 a T, com o objetivo de memorizar as maiores probabilidades de junção entre observações e estados para cada t e j, enquanto que a recuperação efetiva da melhor sequência de estados se dá nas etapas 3 e 4.

Este algoritmo executa a tarefa da obtenção da sequência ótima de estados com complexidade $\theta(TN^2)$, pois realiza N iterações de multiplicação na etapa 1, mais N^2 iterações de multiplicação na etapa 2 executadas t-1 vezes, o que resulta em $N + (T\text{-}1)\ N^2$ multiplicações. Vale ressaltar que se essa tarefa fosse executada através do cálculo de probabilidade de todas as sequências de estados candidatas do modelo, a complexidade seria $\theta(2TN^T)$ (Rabiner, 1990).

A implementação deste algoritmo aplicado à solução proposta é apresentada no Capítulo 6.

Alguns trabalhos propõem modificações no HMM de modo a torná-lo aplicável à tarefa de REM. Essas propostas são discutidas no próximo capítulo.

[10] A função *max* retorna a maior probabilidade e a função *argmax* retorna o argumento (estado) que maximizou a probabilidade.

5 - TRABALHOS CORRELATOS

Este capítulo apresenta os trabalhos que são diretamente relacionados ao desenvolvimento do presente trabalho. Primeiramente é apresentado o Rembrandt (Cardoso, 2008), um sistema de REM determinístico especializado no idioma português que obteve os melhores resultados do segundo HAREM (Mota et al., 2008) quanto ao REM dos tipos pessoa e organização. Em seguida, são detalhados os trabalhos que propuseram adaptações ao HMM como forma de torná-lo aplicável à tarefa de REM, principalmente com o uso de *features* e contextos de palavras, e por fim são apresentadas pesquisas que registraram melhores resultados com a utilização de *gazetteers* como recursos extras em sistemas de REM.

5.1 – O sistema Rembrandt

O Rembrandt (Cardoso, 2008) é um sistema de REM determinístico, baseado em regras gramaticais manuais e em informações extraídas da Wikipédia para a língua portuguesa. Foi o sistema que obteve os melhores resultados para a etiquetação das entidades pessoa e organização na tarefa de avaliação conjunta de REM do segundo HAREM (Mota et al., 2008). O HAREM é um evento referenciado mundialmente na área de reconhecimento de nomes em textos na língua portuguesa e as publicações dele originadas representam pesquisas relevantes para o presente trabalho, pois estão associadas ao REM em textos independentes de domínio onde é avaliado o reconhecimento das entidades mencionadas *pessoa* e *organização*.

Com o intuito de melhorar a acurácia do sistema Rembrandt, o autor propôs a Wikipédia[11] como base do conhecimento para a classificação das EM. Para interagir com a Wikipédia, foi desenvolvida uma interface denominada Saskia (Cardoso, 2008), que facilita a navegação pelas estruturas de categorias, ligações e redirecionamentos, possibilitando a extração do conhecimento.

Cada documento de texto lido pelo Rembrandt é submetido a uma sequência de processos de etiquetação sucessivos até alcançar a versão final, que é o texto etiquetado com as entidades reconhecidas. O funcionamento do Rembrandt é apresentado na Figura 5.1.

[11] http://pt.wikipedia.org.

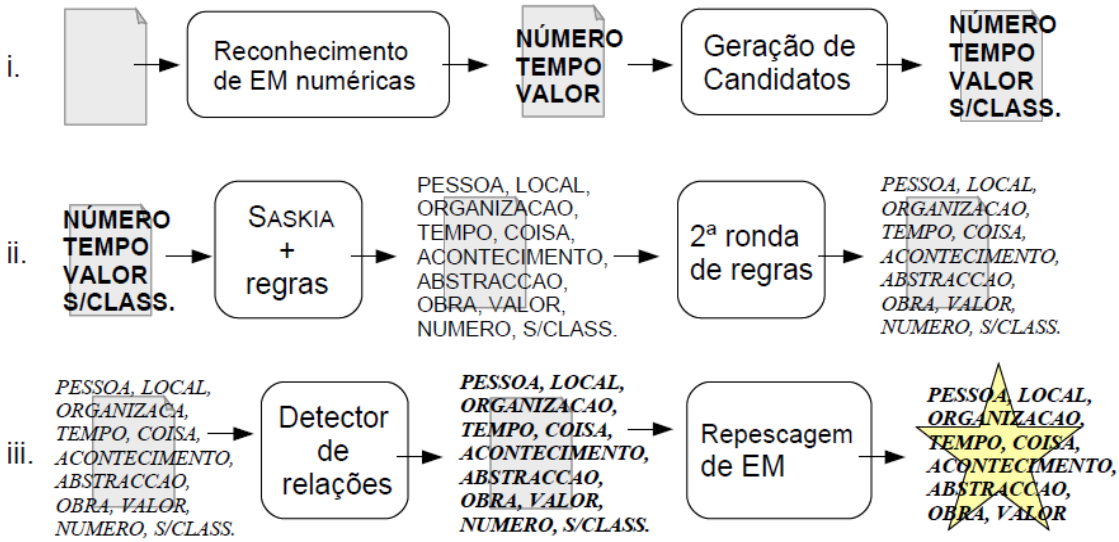


Figura 5.1 - O funcionamento do Rembrandt (extraído de (Cardoso, 2008))

Conforme mostra a figura, o funcionamento do Rembrandt pode ser dividido em três grupos, a seguir apresentados:

i. Identificação das palavras candidatas e EM. Este grupo é formado pelas seguintes tarefas:
   a. Divisão dos textos em sentenças e palavras;
   b. Reconhecimento de expressões numéricas;
   c. Identificação de palavras candidatas a EM;
   d. Geração de entidades candidatas aos tipos expressões temporais e valores.

   Na tarefa *c*, a regra de identificação de palavras candidatas a EM busca por sequências de palavras contendo pelo menos uma letra maiúscula e/ou algarismo, com ocorrência facultativa dos termos *de*, *da*, *do*, *das*, *dos* e *e* (da expressão regular "d[aeo]s?|e", também conhecida por "termos *daeose*"), exceto no início ou no final da sequência;

ii. Classificação das entidades candidatas resultantes da etapa anterior. Este processo é realizado primeiro pela Wikipédia (através da interface Saskia), que relaciona todos os significados que a EM pode ter, e depois pelas regras gramaticais, que se utilizam das características internas e externas da EM para tentar a desambiguação. Em seguida, considerando as classificações obtidas, ocorre uma segunda ronda de aplicação de regras gramaticais com o objetivo de classificação das EM compostas, com ou sem os termos *daeose*, utilizando-se novamente da Saskia e de regras de classificação;

iii. Repescagem das EM sem classificação. Nessa etapa, são aplicadas regras específicas para a detecção de relações entre EM, com o objetivo de identificar relações entre EM com e sem classificação e assim classificar as que não foram previamente classificadas. Em seguida, uma última tentativa de classificação é realizada através da comparação das entidades mencionadas com uma lista de nomes comuns e, por fim, as EM não classificadas são eliminadas.

A estratégia de classificação de EM usando a Wikipédia, em linhas gerais, é dividida em três etapas: (1) o emparelhamento das EM, ou seja, cada entidade mencionada no texto deve estar associada a pelo menos uma página na Wikipédia, seja diretamente ou através da utilização de âncoras e redirecionamentos, (2) a memorização das categorias às quais a página da Wikipédia está associada e (3) a classificação dessas categorias através da aplicação de regras gramaticais específicas. Por exemplo, a entidade mencionada "Porto" está associada às seguintes páginas da Wikipédia portuguesa: "*a segunda maior cidade portuguesa*", "*cidade no Piauí, Brasil*", "*cidade em Zamora, Espanha*", "*aldeia na freguesia de Troviscal*", "*área localizada à beira d'água destinada à atracação de embarcações*", "*Futebol Clube do Porto*", etc.. A etapa (1) faz esta associação entre a EM e as páginas, a etapa (2) memoriza as categorias associadas às páginas encontradas, como "*Cidades de Portugal*", "*Municípios de Portugal*", "*Clubes de futebol de Portugal*", etc., e a etapa (3) faz a associação entre essas categorias e as classes de EM, como *local* e *organização*.

Em seguida, cabe às regras gramaticais a desambiguação das classificações sugeridas pela Wikipédia. As regras gramaticais são desenhadas manualmente e buscam por padrões que revelem a existência de EM nas sentenças e a execução de determinadas ações quando EM forem encontradas. A sua atuação ocorre através de cláusulas aplicadas ordenadamente, de modo que a regra só é considerada bem sucedida no caso de todas as cláusulas a ela associadas retornarem o valor verdade. Além da geração de novas EM, as regras podem disparar também outras duas ações: a detecção de relações entre entidades e a geração de mais de uma entidade associada a uma mesma palavra.

A aplicação das regras é sequencial, da esquerda para a direita, tanto para as palavras de uma sentença, quanto para as sentenças de um texto. As regras bem sucedidas são

imediatamente executadas, de modo que as novas EM identificadas ou classificadas passam a ser consideradas pelas próximas regras aplicadas.

O sistema Rembrandt obteve os melhores resultados na avaliação conjunta do segundo HAREM (Mota et al., 2008) para as EM *pessoa* e *organização*. Os números desses resultados são apresentados no capítulo 7, onde são discutidos comparados com os resultados obtidos pela solução proposta.

Diferentemente do Rembrandt, o presente trabalho propõe um modelo de REM através de algoritmos probabilísticos, baseado no HMM.

5.2 – O HMM aplicado ao REM

O desenvolvimento da solução proposta se baseou na utilização do modelo oculto de Markov para a tarefa de REM. Dentre os trabalhos pesquisados, os principais são o *IdentiFinder* (Bikel et al., 1999) e o *Context-HMM* (Todorovic et al., 2008).

5.2.1 – O sistema *IdentiFinder*

Autores em (Bikel et al., 1999) desenvolveram um sistema denominado *IdentiFinder*, que propõe a utilização de uma variação do HMM para identificar e classificar nomes, datas, horas e quantidades numéricas em sequências textuais. O HMM foi escolhido devido ao fato de ter apresentado bons resultados associados à tarefa de etiquetação morfossintática (*POS-tagging*), e também devido à relativa facilidade de se identificar fenômenos textuais locais indicativos da presença de nomes (exemplo: a presença do *token* "*Ltda*" em um texto sugere que o *token* anterior é uma EM do tipo organização).

O modelo proposto pelo *IdentiFinder* objetiva a junção de cada palavra no texto a uma classe de nome, que pode ser um dos tipos de EM (*pessoa*, *organização*, *local*, etc.) ou o rótulo "*não-EM*". Os estados do modelo são organizados por regiões, essas são representadas pelas classes de nome (*pessoa*, *organização*, *local*, etc.) e o HMM gera um modelo para cada região. Por fim, existem dois estados especiais representando o início e o fim de cada sentença no texto.

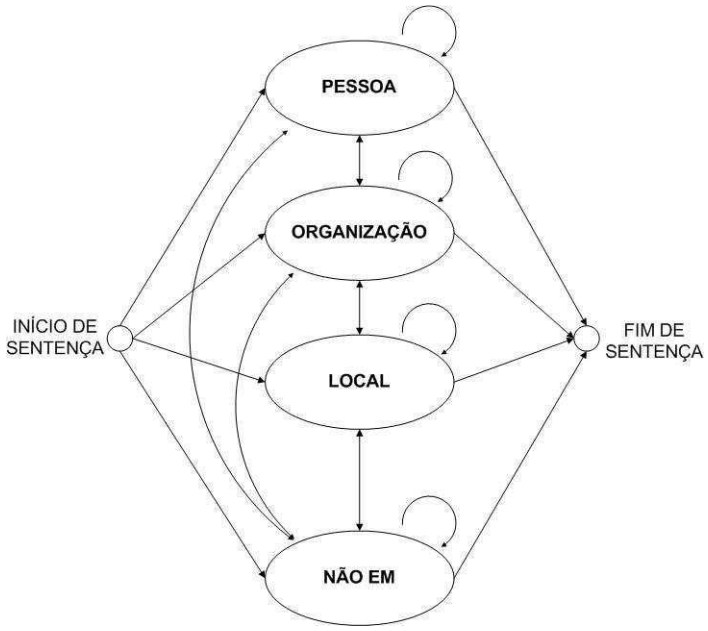


Figura 5.2 - Visão geral do modelo do *IdentiFinder* considerando 4 regiões (modificado (Bikel et al., 1999))

A Figura 5.2 apresenta uma visão geral de um modelo composto por 4 regiões representadas pelas classes de nome *pessoa*, *organização*, *local* e *não-EM*. Cada uma dessas regiões é representada por um modelo denominado "modelo de bigrama[12]" (Bikel et al., 1999).

O modelo de bigrama é usado para computar a verossimilhança da sequência de palavras dentro de cada classe de nome e pode ser representado por uma cadeia de Markov, na qual a probabilidade de ocorrência de cada palavra O_t depende somente da palavra anterior O_{t-1}, ou seja, cada palavra é representada por um estado no modelo de bigrama e a probabilidade de ocorrência de uma sequência de palavras é expressa por

$$\prod_{i=1}^{T} P(O_i \mid O_{i-1}), \qquad O_0 \text{ representa o início da sequência.} \qquad \text{Eq. (5.1)}$$

Os autores afirmam que o fato de haver um modelo para cada classe de nome melhora o desempenho do sistema, pois em geral as EM possuem evidências internas que ajudam na predição de idiomas e domínios distintos, principalmente pela ocorrência de nomes mais comuns. Além disso, evidências externas à classe de nome facilitam a identificação das fronteiras das EM, através de *tokens* como preposições, pronomes ou títulos[13] frequentes ("*Sr.*", "*Dr.*", "*Organizações*", '*Presidente*", etc.).

[12] O termo *bigrama* representa dois itens consecutivos dentro de uma sequência.
[13] O título de um nome é uma palavra que o precede com certa frequência.

Uma diferença do modelo do sistema *IdentiFinder* em relação ao HMM puro apresentado no capítulo anterior é a utilização de *features* (associadas às palavras) embutidas na sequência de observações O. Dessa forma, ao invés de expressar essa sequência por $O = O_1 O_2 \ldots O_T$, como no HMM puro, a mesma é expressa por $<O,F> = <O,F>_1 <O,F>_2 \ldots <O,F>_T$, sendo que F_t representa a *feature* associada a O_t. Isso exige uma adaptação no modelo, pois o HMM passa a considerar também a distribuição probabilística de F, além da distribuição de O. A Tabela 5.1 apresenta as *features* consideradas no *IdentiFinder*.

Tabela 5.1 - *Features* associadas às palavras (modificado – (Bikel et al., 1999))

Feature	**Exemplo**	**Significado intuitivo**
twoDigitNum	90	Ano, 2 dígitos
fourDigitNum	1990	Ano, 4 dígitos
containsDigitAndAlpha	A8956-67	Código de produto
containsDigitAndDash	09-96	Data
containsDigitAndSlash	11/9/89	Data
containsDigitAndComma	23,000.00	Valor monetário
containsDigitAndPeriod	1.00	Valor monetário, percentual
otherNum	456789	Outros números
allCaps	BBN	Organização
capPeriod	M.	Iniciais de nome de pessoa
firstWord	Primeira palavra da sentença	O fato da palavra ser iniciada em maiúscula não interfere no modelo
initCap	Sally	Iniciais em maiúscula
lowerCase	can	Não possui maiúsculas
Other	,	Pontuação, ou qualquer outro termo não categorizado

As *features* são computadas na ordem em que estão listadas na Tabela 5.1. Essa precedência é importante pois a cada palavra será associada somente a primeira *feature* que caracterizá-la. Por exemplo, se a palavra “Sempre” iniciar uma sentença, a ela será associada a *feature firstWord*, e não a *feature initCap*.

Os autores do referido trabalho relatam que optaram pelo uso de *features* diretamente associadas às palavras, em detrimento da etiquetação morfossintática da palavra, pelo fato

das características associadas às letras maiúsculas serem essenciais para a tarefa de REM e a etiquetação morfossintática não "garantir" esta informação.

O modelo formal do *IdentiFinder* divide-se em dois: o modelo de alto nível e o modelo de *back-off*. O modelo de alto nível é o primeiro a ser utilizado em fase de etiquetação e, quando um *token* não é encontrado no treinamento ou o treinamento dos bigramas é considerado insuficiente, o sistema recorre ao modelo de *back-off*, que gera submodelos com menor poder de acurácia em relação ao primeiro, porém é capaz de etiquetar *tokens* desconhecidos ou "mal treinados".

O modelo de alto nível é composto por três submodelos, usados para a geração (i) da classe de nome, (ii) da primeira palavra em uma classe de nome e (iii) das demais palavras em uma classe de nome.

Com base nesses três modelos, são definidas as probabilidades usadas no sistema de REM, descritas a seguir:

- Probabilidade de geração da primeira palavra de uma classe de nome (que coincide com a probabilidade de geração da classe de nome, por isso resulta do produto entre elas):

  $$P(CN_t \mid CN_{t\text{-}1}, O_{t\text{-}1}) \cdot P(\langle O,F\rangle_{\text{primeira}} \mid CN_t, CN_{t\text{-}1}),$$ CN = classe de nome.

  O primeiro fator desse produto não usa *feature*, pois a intuição dos autores é que as *features* (como iniciais em maiúsculas) das palavras que antecedem nomes ("*sr.*", "*dr.*", "*organizações*", etc.) em geral são irrelevantes.
- Probabilidade de geração das demais palavras de uma classe de nome:

  $$P(\langle O,F\rangle_t \mid \langle O,F\rangle_{t\text{-}1}, CN_t). \qquad \text{Eq. (5.2)}$$
- Probabilidade de geração da última palavra de uma classe de nome:

  $$P(\langle O,F\rangle_{\text{última}} \mid \langle O,F\rangle_t, CN_t). \qquad \text{Eq. (5.3)}$$

O treinamento do modelo é realizado através da estimativa das probabilidades com base nos exemplos de treinamento, da seguinte forma:

$$P(CN_t \mid CN_{t\text{-}1}, O_{t\text{-}1}) = \frac{c\,(CN_t, CN_{t\text{-}1}, O_{t\text{-}1})}{c\,(CN_{t\text{-}1}, O_{t\text{-}1})}, \qquad \text{Eq. (5.4)}$$

$$P(<O,F>_{\text{primeira}} \mid CN_t, CN_{t\text{-}1}) = \frac{c\,(<O,F>_{\text{primeira}}, CN_t, CN_{t\text{-}1})}{c\,(CN_t, CN_{t\text{-}1})} \quad \text{e} \qquad \text{Eq. (5.5)}$$

$$P(<O,F>_{\text{t}} \mid <O,F>_{t\text{-}1}, CN_t) = \frac{c\,(<O,F>_{\text{t}}, <O,F>_{\text{t-1}}, CN_t)}{c\,(<O,F>_{\text{t-1}}, CN_t)}, \qquad \text{Eq. (5.6)}$$

sendo que *c*(*evento*) representa o número de vezes (contagem) que o *evento* ocorreu na coleção de treinamento.

Conforme apresentado previamente, quando um *token* não é encontrado no treinamento ou o treinamento dos bigramas é considerado insuficiente, o *IdentiFinder* recorre ao modelo de *back-off*,

O primeiro modelo de *back-off* do *IdentiFinder* é o modelo de palavras desconhecidas, usado para calcular a probabilidade de transição de estados associada a *tokens* não vistos no treinamento do modelo de alto nível (problema da não cobertura). A modelagem é realizada através da divisão do corpus de treinamento original em duas partes de mesmo tamanho. Utiliza-se uma parte para treinar e a outra para testar, de modo que todas as palavras do corpus de teste não existentes no corpus de treinamento são substituídas pelo termo "_DESCONHECIDA_". Faz-se o mesmo processo invertendo as duas partes do corpus original (treinamento e teste) e, ao final, unem-se os dois modelos de palavras desconhecidas e se tem um modelo bem parecido com o principal, exceto pelo fato de não conter as palavras desconhecidas, mas sim o termo "_DESCONHECIDA_" no seu lugar. Assim como ocorre no modelo de alto nível, o treinamento do modelo de palavras desconhecidas é realizado com base nos exemplos de treinamento. Este modelo é usado sempre que, em tempo de etiquetação (teste), pelo menos uma palavra do bigrama é desconhecida.

A segunda abordagem de *back-off* do sistema é usada para a suavização (*smoothing*) do modelo em casos de bigramas não vistos ou insuficientes no treinamento. Primeiramente são definidos os níveis de *back-off* associados aos três modelos de bigramas representados nas equações 5.4 a 5.6. Esses níveis são apresentados na Tabela 5.2, onde se pode perceber que a cada descida de nível o modelo se torna menos específico. É calculado então o peso

ϕ, que passa a ser o coeficiente da probabilidade do bigrama, no nível em que está sendo computada (assim como o peso 1-ϕ é o coeficiente do nível de *back-off* desse bigrama). Ressalta-se que quanto maior o valor de ϕ, mais representativo é o nível do bigrama sendo computado e, consequentemente, menor é a necessidade de se recorrer ao *smoothing*. Considerando-se que um bigrama modelado é representado por $P(X|Y)$, o cálculo de ϕ é dado por:

$$\Phi = \left(1 - \frac{anterior\ c(Y)}{c(Y)}\right) \cdot \frac{1}{1+\frac{saídas\ únicas\ de\ Y}{c(Y)}}, \qquad \text{Eq. (5.7)}$$

sendo que $c(Y)$ representa o número de vezes (contagem) que o evento Y ocorreu na coleção de treinamento.

Tabela 5.2 - Níveis de *back-off* para cada bigrama modelado (modificado – (Bikel et al., 1999))

Bigramas de classe de nome	Bigramas de primeira palavra	Bigrama de demais palavras
$P(CN_t \mid CN_{t-1}, O_{t-1})$	$P(<O,F>_{\text{primeira}} \mid CN_t, CN_{t-1})$	$P(<O,F>_t \mid <O,F>_{t-1}, CN_t)$
↓	↓	↓
$P(CN_t \mid CN_{t-1})$	$P(<O,F>_{\text{primeira}} \mid$ +início+, $CN_t)$	$P(<O,F>_t \mid CN_t)$
↓	↓	↓
$P(CN_t)$	$P(<O,F>_{\text{primeira}} \mid CN_t)$	$P(O \mid CN_t) \cdot P(F \mid CN_t)$
↓	↓	↓
$\frac{1}{número\ de\ classes\ de\ nome}$	$P(O \mid CN_t) \cdot P(F \mid CN_t)$	$\frac{1}{V} \cdot \frac{1}{número\ de\ features}$,
	↓	sendo que V representa o número de palavras do modelo
	$\frac{1}{V} \cdot \frac{1}{número\ de\ features}$	

A Figura 5.3 apresenta um exemplo de aplicação da segunda abordagem de *back-off/smoothing* do *IdentiFinder*.

- Exemplo de sentença: "*O Sr. Carlos come muito*.";
- Token corrente: "*Carlos*";
- Bigrama sendo computado: *P*(PESSOA | Não-EM, "*Sr.*"), associado a $P(CN_t \mid CN_{t\text{-}1}, O_{t\text{-}1})$;
- Primeiro nível de *back-off* do bigrama acima: *P*(PESSOA | Não-EM) (vide Tabela 5.2);
- Cálculo de Φ:
  - Primeira análise: fator à esquerda da equação de Φ: $(1 - \frac{anterior\ c(Y)}{c(Y)})$:
    - No nível de *back-off*, *c*(*Y*) = c(Não-EM), e no nível de bigrama, *anterior c*(*Y*) = *c*(Não-EM, "*Sr.*");
    - Percebe-se que, se o valor de "anterior *c*(*Y*)" for próximo ao de *c*(*Y*), então Φ tende a zero;
    - Ou seja, se a quantidade de exemplos de treinamento de (Não-EM, "*Sr.*") for parecida com a de (Não-EM), significa que o bigrama sendo computado não é representativo, então há necessidade de se aplicar o *smoothing*, e isso ocorre através da atribuição do peso Φ próximo a zero ao bigrama *P*(PESSOA | Não-EM, "*Sr.*"), o que desloca quase toda a massa probabilística para um nível abaixo no modelo de *back-off*, ou seja, para o modelo *P*(PESSOA), que será mais influente do que *P*(PESSOA | Não-EM, "*Sr.*") no cenário exemplificado;
  - Segunda análise: fator à direita da equação de Φ: $\frac{1}{1+\frac{saídas\ únicas\ de\ Y}{c(Y)}}$.
    - Supõe-se que, na sequência do exemplo ("*O Sr. Carlos come muito.*"), a palavra "*Sr.*" seja vista 20 vezes como primeira palavra em bigramas, sendo 1 vez com a saída "*Carlos*" e 19 vezes com a saída "*Raul*", e sempre associada à classe *Não-EM*. Neste caso, como a palavra "*Sr.*" Gerou somente 2 saídas únicas ("*Carlos*" e "*Raul*") o peso do fator à direita de Φ será

$$\frac{1}{1+\frac{2}{20}} = \frac{10}{11};$$

    - Ou seja, 10/11 da massa probabilística será aplicada ao modelo do bigrama *P*(PESSOA | Não-EM, "*Sr.*"), e somente (1 - 10/11) ao modelo correspondente de *back-off P*(PESSOA | Não-EM). É um exemplo de não necessidade de *smoothing*.

Figura 5.3 - Exemplo de aplicação da segunda abordagem de *back-off*/*smoothing*

Ao contrário do HMM puro, no sistema *IdentiFinder* os modelos de *back-off* garantem 100% de probabilidade de geração de uma palavra para todos os estados que compõem o modelo. A decodificação do modelo usa programação dinâmica através do algoritmo de Viterbi, assim como ocorre no HMM puro. Por fim, trata-se de um modelo

matematicamente válido, uma vez que a soma de todas as probabilidades de transição partindo de qualquer estado é igual um.

Durante os experimentos do referido trabalho, o modelo foi treinado e avaliado utilizando os corpora do MUC-6 (Grishman e Sundheim, 1996) e MET-1 (Merchant et al., 1996), respectivamente nos idiomas inglês e espanhol. Segundo os autores, os resultados superaram todos os sistemas de REM baseados em aprendizado de máquina existentes à época e alcançaram desempenho compatível aos dos melhores sistemas baseados em regras manuais. Quanto ao tamanho do corpus de treinamento, foram realizados vários experimentos e se concluiu que o desempenho foi bastante prejudicado com a utilização de corpus de tamanho inferior a 100.000 *tokens* e que, por outro lado, a melhora na acurácia observada entre 100.000 e 650.000 *tokens* é inferior a 2%. Por fim, ressaltam os autores que os resultados associados ao MUC-6 não são boas referências para pesquisas de REM independente de domínio, pois suas coleções textuais são bastante específicas.

Como sugestões de trabalhos futuros, os autores priorizam o aumento da distância do contexto modelado, passando de um modelo de bigramas para trigramas, a fim de cobrir erros detectados associados à utilização de caracteres de pontuação (principalmente a vírgula) nas fronteiras ou dentro das EM. Além disso, é sugerida também a utilização de um modelo hierárquico para capturas nomes aninhados (como “Banco do Brasil”) e de uma heurística de treinamento para complementar o conjunto de corpora etiquetados através de coleções não etiquetadas.

5.2.2 – HMM com contexto

Em outro importante trabalho associado ao HMM para a tarefa de REM (Todorovic et al., 2008), os autores demonstraram que a utilização de informações associadas ao contexto de um *token* no HMM contribui para melhorar a acurácia e o tempo de etiquetação de um sistema de REM. O modelo proposto no trabalho, chamado pelos autores de *Context-HMM* (ou *CHMM*), foi baseado no *IdentiFinder* (Bikel et al., 1999) e as principais características que o diferem deste foram a grande relevância atribuída ao contexto dos *tokens*, ou seja, às palavras e *features* localizadas próximas a ele, e a criação de um componente gramatical determinístico para a identificação de EM dos tipos data, hora, valor monetário e percentual, baseado na utilização de dicionários e regras gramaticais.

Em nível de modelo, a principal alteração em relação ao *IdentiFinder* está no fato do *CHMM* considerar que a probabilidade de transição entre estados sucessivos é dependente do contexto, representado pelas palavras e respectivas *features* próximas ao *token* corrente. Assim, o modelo proposto no *CHMM* é dado por

$$P(CN_t \mid CN_{t\text{-}1}, \langle O,F\rangle_{t\text{-}k\,:\,t+k}), \qquad \text{Eq. (5.8)}$$

onde *CN* é a classe de nome e *K* é o tamanho do contexto.

O treinamento do modelo com base nos exemplos de treinamento, é expresso por:

$$P(CN_t \mid CN_{t\text{-}1}, \langle O,F\rangle_{t\text{-}k\,:\,t+k}) = \frac{c\,(CN_t, CN_{t\text{-}1}, \langle O,F\rangle_{t\text{-}k\,:\,t+k})}{c\,(CN_{t\text{-}1}, \langle O,F\rangle_{t\text{-}k\,:\,t+k})}, \qquad \text{Eq. (5.9)}$$

sendo que *c*(*evento*) representa o número de vezes (contagem) que o *evento* ocorreu na coleção de treinamento.

No trabalho em comento, foi denominada *contexto* a condição da probabilidade representativa do modelo (equação 5.8), ou seja, contexto = ($CN_{t\text{-}1}$, $\langle O,F\rangle_{t\text{-}k\,:\,t+k}$). As combinações com que as variáveis $CN_{t\text{-}1}$, $O_{\,t\text{-}k\,:\,t+k}$ e $F_{\,t\text{-}k\,:\,t+k}$ são usadas são determinadas pelos modelos de *back-off*. Exemplos de combinações são: $CN_{t\text{-}1}\ O_t\ O_{t\text{-}1}$, $CN_{t\text{-}1}\ F_t\ O_t$, $CN_{t\text{-}1}\ O_t\ O_{t\text{-}1}\ F_{t\text{-}1}$, etc.. Quanto maior for o número de contextos considerados, mais informação é obtida sobre a probabilidade dos estados, porém maior também é o número de cálculos dos pesos Φ associados ao modelo de *back-off*. O *CHMM* resolve esse problema através de uma estratégia de agrupamento de contextos que têm números de ocorrências similares nos exemplos de treinamento e associando cada grupo a somente um peso Φ. Os experimentos mostraram bons resultados com a utilização de 17 contextos (no intervalo *t*-2 a *t*+2) agrupados em 5 grupos. A probabilidade resultante do modelo de *back-off* é dada pela combinação linear das probabilidades dos grupos, da seguinte forma:

$$P(CN_t \mid C) = \sum_b \Phi_b P_b(CN_t \mid C_b), \qquad C = (CN_{t\text{-}1}, \langle O,F\rangle_{t\text{-}k\,:\,t+k}) \qquad \text{Eq. (5.10)}$$

onde *b* representa os grupos do modelo de *back-off* e Φ_b os pesos dos grupos.

Os autores do referido trabalho desenvolveram um protótipo do *CHMM* e outro do *IdentiFinder*, ambos foram treinados e avaliados utilizando o corpus do MUC-7 (Chinchor, 1998) e comparados entre si quanto à acurácia e velocidade na etiquetação de EM. Os resultados mostraram que o *CHMM* executou o treinamento e a etiquetação duas vezes mais rápido que o *IdentiFinder*, além de ter apresentado acurácia superior na ordem de 3%

em relação ao *IdentiFinder* e resultados satisfatórios com o uso do componente gramatical determinístico proposto.

5.3 – O uso de *gazetteers*

No contexto do REM, *gazetteers* são listas externas que contêm nomes de EM e podem ser usadas para comparação com os *tokens* de um texto durante o seu treinamento ou etiquetação.

Segundo os autores em (Ratinov e Roth, 2009), *gazetteers* facilitam a inferência de informações não contextualizadas no texto, ajudando a resolver problemas de ambiguidade e não cobertura. Nesse sentido, trabalhos como (Cohen e Sarawagi, 2004), (Florian et al., 2003), (Toral e Muñoz, 2006) e (Kazama e Torisawa, 2007) relataram resultados satisfatórios com a utilização de *gazetteers* representados por *features* em modelos de REM probabilísticos, ao invés da sua utilização por simples correspondência de palavras.

Os autores de (Zhou e Su, 2002) relataram melhora nas medidas de precisão e revocação do modelo por eles proposto com a utilização de *gazetteers* dos tipos pessoa, organização, local e data.

Em (Sang e Meulder, 2003), os autores analisaram os sistemas participantes do evento CoNLL-2003 e constaram que os *gazetteers* estavam entre as seis *features* mais usadas pelos sistemas participantes.

O projeto Linguateca (Santos et al., 2004) disponibiliza publicamente o *gazetteer* REPENTINO[14] ((Sarmento et al., 2006) e (Sarmento, 2005)), que, segundo os seus autores, foi especialmente desenvolvido para dar suporte a sistemas de REM na língua portuguesa. O REPENTINO contém cerca de 450.000 instâncias de nomes (divididas em 11 categorias e 97 subcategorias), que foram extraídas de corpora manualmente etiquetados do projeto Linguateca ou através de métodos de busca semiautomatizada na web. A Figura 5.4 apresenta a distribuição das instâncias de EM de acordo com as categorias definidas no REPENTINO.

[14] O acrônimo REPENTINO significa "REPositório para o reconhecimento de ENTIdades com Nome".

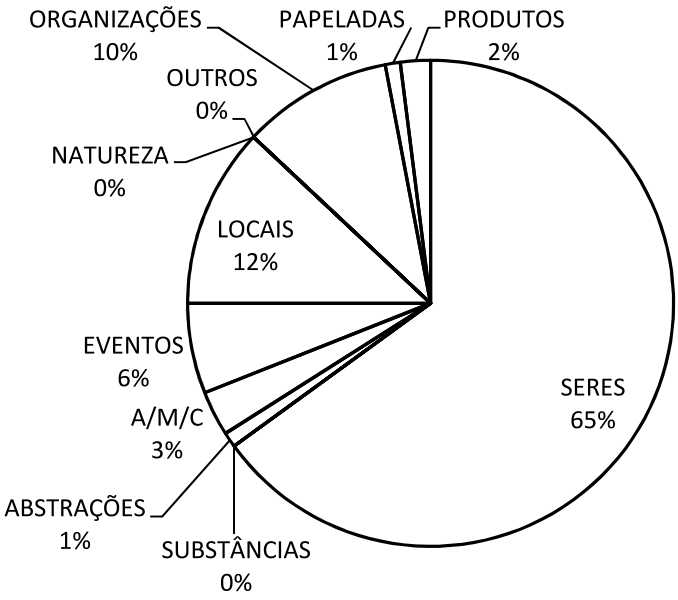


Figura 5.4 - Distribuição das instâncias do REPENTINO de acordo com as categorias (Sarmento, 2005)

Na Figura 5.4, a categoria "seres" representa nomes de pessoas, "papeladas" representa documentos de legislação e normatização e "A/M/C" representa documentos sobre arte, mídia e comunicação.

O sistema de REM denominado SIEMÊS (Sarmento, 2006) é baseado na combinação de regras de análise de contexto com a consulta a almanaques e utiliza o REPENTINO como única fonte de conhecimento externo. A sua participação na avaliação conjunta do primeiro HAREM (Santos et al., 2006) resultou na terceira colocação nas avaliações das EM *pessoa* e *organização*.

Por fim, embora a popularização e profissionalização da Wikipédia propulsione cada vez mais o interesse acadêmico por pesquisas associadas à sua utilização como fonte de dados para a construção de *gazetteers*, no presente trabalho se optou pela não utilização da Wikipédia, devido ao fato dos nomes de pessoas e organizações contidos em textos forenses serem, em geral, desconhecidos (pessoas comuns, pequenas empresas, etc.), e por isso comumente não encontrados nas páginas da Wikipédia.

6 - SOLUÇÃO PROPOSTA

Neste capitulo é apresentado o modelo desenvolvido denominado ICC-HMM, que propõe uma adaptação do HMM para reconhecer entidades mencionadas em textos forenses.

6.1 – CARACTERÍSTICAS

O presente trabalho propõe o ICC-HMM (*Identification-Classification-Context*-HMM), um modelo de REM baseado no HMM, especializado no reconhecimento das entidades *pessoa* e *organização* em textos forenses, que possui como principais características:

- A criação de dois HMM distintos, o modelo de identificação, para identificar se um token é EM ou não-EM, e o modelo de classificação, para classificar os tokens identificados como EM;
- A exploração do contexto das palavras (palavras próximas ao token corrente) como forma de prover melhor acurácia associada ao modelo de classificação;
- A utilização de um *gazetteer* criado com base no REPENTINO (Sarmento et al., 2006);
- A utilização de um conjunto de *features* selecionado para o REM em textos forenses.

Neste capítulo, a apresentação do ICC-HMM é iniciada com a descrição das *features* utilizadas. Em seguida, são apresentados o modelo formal e as técnicas empregadas nos processos de decodificação e de treinamento do modelo proposto.

6.2 – *FEATURES*

As sequências textuais tratadas pelo ICC-HMM são representadas por pares ordenados contendo as sequências de *tokens observáveis O* e as suas respectivas *features F*, na forma <*O*, *F*>.

O conjunto das *features* utilizadas é a única parte do modelo que é dependente do domínio e dependente do idioma. Os textos forenses, que são aqueles contidos em arquivos de mídias apreendidas, não possuem um padrão que permita a sua associação a um domínio predominante específico. Esses textos podem representar contratos, editais, procurações, cartas, *emails*, mensagens instantâneas, etc., não sendo possível prever o tipo, teor, dialeto

ou nível de formalidade da sua escrita. Esta independência do domínio observada nos textos forenses interfere diretamente na escolha das *features* do modelo. *Features* de etiquetação morfossintática, por exemplo, frequentemente usadas em sistemas de REM (Feldman e Sanger, 2007), não se aplicam aos textos forenses, principalmente devido aos constantes erros gramaticais e à ausência de um padrão de escrita nesses textos. Por outro lado, o fato do ICC-HMM priorizar o reconhecimento somente dos nomes de *pessoas* e *organizações* facilita a utilização de *features* específicas para essas entidades, como os identificadores CPF, RG, CNPJ e inscrição social, e também permite a generalização de outras *features* na tentativa de elevar o nível de padronização do texto, é o caso de *features* como valor monetário, percentual, data, hora e CEP, que não têm relação direta com as entidades *pessoa* e *organização* e, por isso, podem ser generalizadas como uma única *feature* sem prejuízo à acurácia do modelo.

A seleção das *features* utilizadas no ICC-HMM resultou da análise da frequência da sua ocorrência em textos forenses reais e nos corpora utilizados nos experimentos, assim como da análise da sua interferência no correto reconhecimento das EM, usando como linha de base as *features* propostas nos trabalhos apresentados em (Sang e Meulder, 2003) e (Bikel et al., 1999).

As *features* utilizadas no ICC-HMM são apresentadas na Tabela 6.1.

Assim como ocorre no modelo descrito em (Bikel et al., 1999), no ICC-HMM é associada uma *feature* a cada *token*, sendo que tal associação é feita pela ordem de precedência listada na Tabela 6.1. Por exemplo, na sentença "*Anunciaram Carla como uma das suspeitas*", é atribuída ao *token* "*suspeitas*" a *feature f_other*, ao *token* "*Carla*" a *feature f_first_cap* e ao *token* "*Anunciaram*" a *feature f_first_word*. Observa-se que, apesar do *token* "*Anunciaram*" iniciar com letra maiúscula, a *feature* atribuída é *f_first_word*, e não *f_first_cap*, pois a primeira precede a segunda.

Tabela 6.1 - *Features* utilizadas no ICC-HMM

Feature	**Significado intuitivo**	**Exemplo**
f_id_pes	• CPF • RG	• *900.128.524-20* • *6459878-40*
f_id_org	• CNPJ • IE/IM (inscrição estadual/municipal)	• *78.747.136/0003-89* • *125.454.23-1* ou *234.232.566*
f_num	• Número • Valor monetário ou percentual • Data • Hora • CEP • Códigos, identificações ou outras sequências numéricas	• *520* • *12.000,50* ou *12,000.50* ou *5,7* ou *5.7* • *10/01/2012* ou *10.01.2012* ou *10-01-12* • *12:30* ou *12h30* ou *12:30:55* ou *12h30m55* • *41.820-000* ou *41820-000* • *AC001/8-7* ou *50\K70:2*
f_all_cap	Todas as letras do *token* são maiúsculas e *token* formado por mais de uma letra Intuição: EM organização	*CEF* ou *DPF* ou *MPF*
f_first_word	Primeira palavra da sentença Intuição: o fato de iniciar com letra maiúscula não deve interferir na predição de EM	***Os*** *suspeitos confirmaram.*
f_first_cap	Primeira letra do *token* é maiúscula e *token* formado por mais de uma letra Intuição: nome próprio, EM não numérica	*Osvaldo* ou *Polícia* ou *Salvador*
f_punct	Pontuação	*,* ou *;* ou *:*
f_other	Qualquer *token* que não contenha nenhuma das demais *features*	*furto* ou *licitação* ou *suborno* ou *E*

Além das *features* listadas na Tabela 6.1, que representam a sequência *F* do par <*O*, *F*>, o ICC-HMM utiliza no seu modelo de *back-off* uma *feature* especial para representar a sequência observável *O*, que é o *stemming* do *token*, expresso por *f_stem*. O uso do *stemming* se justifica pela sua capacidade de agrupar variações de um mesmo morfema e assim reforçar o peso da sua representação no texto de uma forma que não prejudique a sua carga semântica, o que pode resultar em uma melhor acurácia do modelo. O modelo de

back-off do ICC-HMM é usado quando um *token* não é encontrado no treinamento ou o treinamento é considerado insuficiente. Este modelo é apresentado na Seção 6.3.3.

6.3 – MODELO FORMAL

O ICC-HMM é dividido em dois modelos, o de identificação e o de classificação de EM.

Durante o processo de etiquetação, para cada sentença de um texto, o ICC-HMM percorre a sequência de *tokens* em duas etapas, a primeira para identificá-los como EM ou não-EM e a segunda para classificar (como pessoa, organização, local, etc.) somente os *tokens* identificados como EM na primeira etapa.

A Figura 6.1 apresenta, nos seus quadros tracejados, os diagramas de transição de estados dos modelos de identificação e classificação propostos, bem como as sequências produzidas por cada modelo. Conforme ilustrado na Figura 6.1, o modelo de identificação recebe uma sequência de *tokens* observáveis *O* como entrada e gera a sequência de junções (*O*, *Id*), identificada na figura por *junção_1*, que representa a identificação de cada *token* quanto a ser ou não ser EM, uma vez que $id_t \in S\_Id = \{EM, não\text{-}EM\}$.

Em seguida, esta sequência de junções alimenta o modelo de classificação, que utiliza dois conjuntos de estados, o *S_Cl_naoEM*, formado somente pelo estado *não-EM*, e o conjunto *S_Cl_EM*, formado pelos estados que representam as classes de EM (*pessoa*, *organização*, *local*, etc.). O modelo de classificação produz como saída a sequência de junções (*O*, *Cl*), identificada na Figura 6.1 por *junção_2*, que são os *tokens* e as classes de EM a eles associadas. Os estados "*Início*" e "*Fim*" presentes nos diagramas da Figura 6.1 são utilizados para representar as fronteiras do modelo, que são o início e o fim de uma sentença, e não para identificar ou classificar os *tokens*. As seções seguintes apresentam os modelos de identificação e classificação do ICC-HMM.

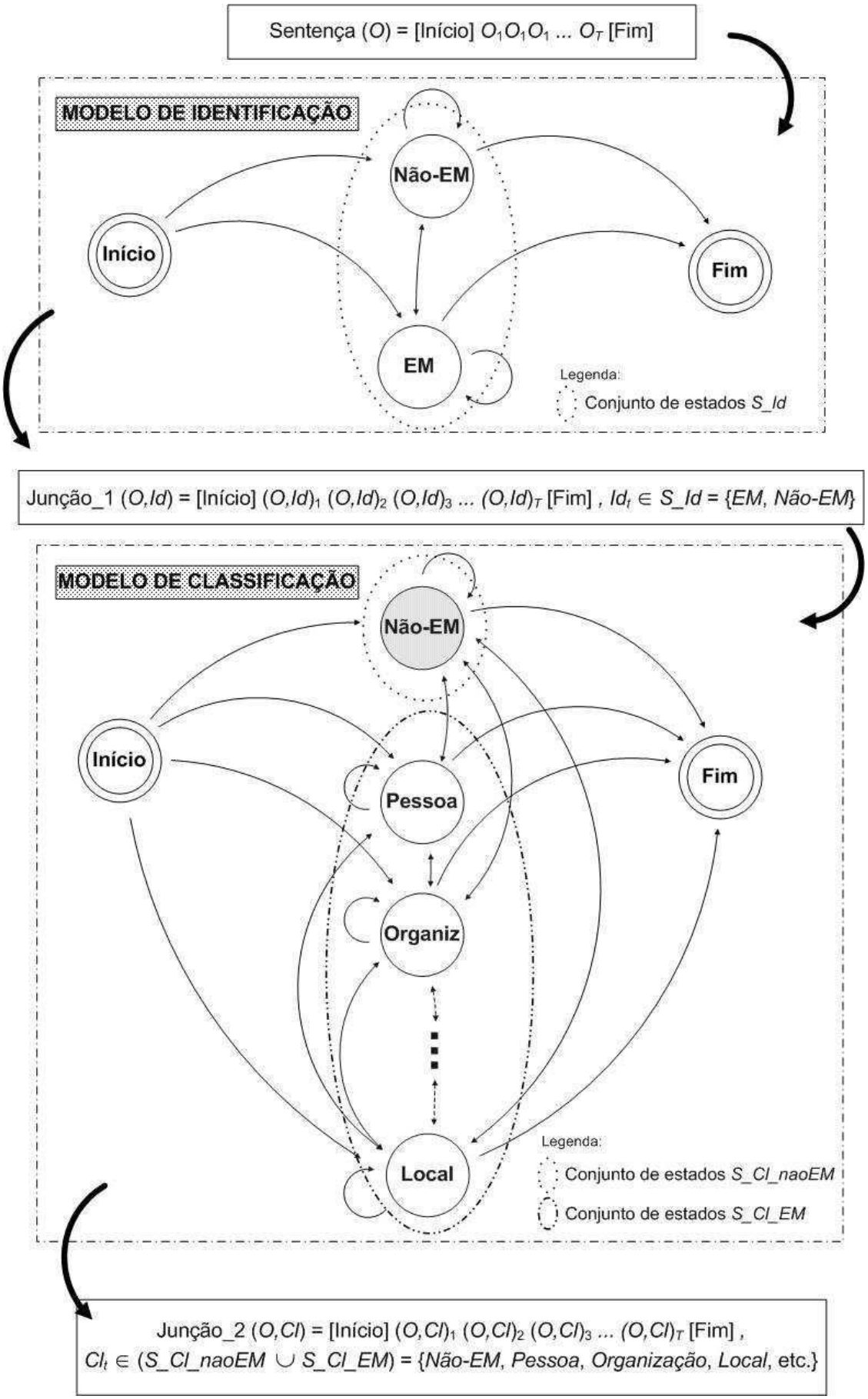


Figura 6.1 - Diagramas de transição de estados do ICC-HMM

6.3.1 – Modelo de identificação

A exemplo do modelo de REM proposto em (Todorovic et al., 2008), com o objetivo de explorar o contexto associado ao *token* corrente, optou-se no ICC-HMM pela utilização de cálculos de probabilidade de geração de estados, dado o contexto, do tipo *P*(*ESTADO*|*CONTEXTO*), e pela não utilização da probabilidade de emissão de *tokens*, do tipo P(*TOKEN*|*ESTADO*). Com a abordagem proposta, observou-se que haveria prejuízo à tarefa de identificação de um *token* ao se comparar o estado *não-EM* com o estado *PESSOA*, *ORGANIZAÇÃO*, ou qualquer outro que represente uma classe de EM, uma vez que, para a tarefa de identificação, a comparação entre dois identificadores (*não-EM* e *EM*) é mais apropriada do que a comparação entre um identificador (*não-EM*) e uma classe (*PESSOA*, por exemplo).

Este problema pode ser ilustrado da seguinte forma: supõe-se que o modelo de REM considera cada estado como uma classe de EM, além do estado *não-EM*. Se um corpus de treinamento contém 16 exemplos do token "*Passos*", sendo 8 associados ao estado *não-EM*, 2 ao estado *PESSOA*, 2 ao estado *ORGANIZAÇÃO*, 2 ao estado *LOCAL* e 2 ao estado *OBRA*, então *P*(*não-EM* | "*Passos*") = $8/16$, que representa o quádruplo de *P*(*PESSOA* | "*Passos*") = $2/16$. Em contrapartida, se forem agrupados todos os estados, exceto o *não-EM*, em um estado único, denominado *EM*, então *P*(*não-EM* | "*Passos*") = *P*(*EM* | "*Passos*") = $(2+2+2+2)/16$, o que torna o processo de identificação mais equilibrado.

Dada essa constatação, optou-se pela separação do ICC-HMM em dois modelos, um de identificação, que agrupa todas as classes de EM em um único estado, e outro de classificação.

O quadro tracejado superior da Figura 6.1 representa o diagrama de transição de estados do modelo de identificação. Esse modelo é um HMM composto por somente dois estados, *S_Id* = {*EM*, *não-EM*}, e é expresso em duas parcelas, da seguinte forma:

$$P(Id_t \mid Id_{t\text{-}1}, \langle O,F\rangle_{t\text{-}1:\,t+1}) + P(Id_t \mid O_t), \qquad \text{Eq. (6.1)}$$

onde $O = O_1\ O_2\ ...\ O_T$ é a sequência de *tokens* observáveis no tempo, $F = F_1\ F_2\ ...\ F_T$ representa a sequência das *features* associadas a *O*, $Id = Id_1\ Id_2\ ...\ Id_T$ é a sequência de estados no tempo.

Este modelo procura explorar a utilização do contexto do *token* corrente na transição de estados, $P(Id_t \mid Id_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1})$, sugerida em (Todorovic et al., 2008), em conjunto com a probabilidade de geração do estado associado ao *token* corrente, $P(Id_t \mid O_t)$, onde $t \in T$.

6.3.2 – Modelo de classificação

Conforme ilustrado na Figura 6.1, o modelo de identificação transforma a sequência $O = O_1\, O_2 \ldots O_T$ na sequência $(O, id) = (O, id)_1\, (O, id)_2 \ldots (O, id)_T$, onde $id_t \in S\_id = \{EM, não\text{-}EM\}$. Essa sequência gerada é posteriormente tratada pelo modelo de classificação, cujo diagrama de transição de estados está representado no quadro tracejado inferior da Figura 6.1 e cujo objetivo é calcular as distribuições probabilísticas que mais agregam informação para a inferência da classe (EM) de um *token*.

O modelo de classificação é expresso pela soma de quatro parcelas, aqui denominadas PI, PT, PC e PG, a seguir apresentadas (os próximos parágrafos detalham os elementos que compõem cada parcela):

1. PI - Probabilidade de identificação:

$$P(Cl_t \mid Cl_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1}) + \quad \text{Eq. (6.2)}$$

$$P(Cl_t \mid O_t) \quad \text{Eq. (6.3)}$$

2. PT - Probabilidade do título[15] gerar a EM:

$$P(Cl_t \mid O_{t\text{-}1}) \quad \text{Eq. (6.4)}$$

3. PC - Probabilidade do contexto gerar a EM:

$$\sum_{u=t-2}^{t+2} P(Cl_t \mid O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2})/4,\ u \neq t \quad \text{Eq. (6.5)}$$

4. PG - Probabilidade do *gazetteer* gerar a EM:

$$P_{\text{gazetteer}}(Cl_t^* \mid O_t) \quad \text{Eq. (6.6)}$$

Nas equações 6.2 a 6.6, $O = O_1\, O_2 \ldots O_T$ é a sequência de *tokens* observáveis no tempo, F representa a sequência das *features* associadas a O, $Cl = Cl_1\, Cl_2 \ldots Cl_T$ é a sequência de estados no tempo e Cl_t^* representa o estado (classe) gerado pelo *gazetteer* REPENTINO dado o *token* O_t. O modelo de classificação utiliza três conjuntos de estados, o *S_Cl_GAZ*, que é independente dos demais, formado pelos 11 estados que compõem o *gazetteer*, o *S_Cl_naoEM*, que é formado somente pelo estado *não-EM*, e o conjunto *S_Cl_EM*,

[15] Título é o nome dado a palavras que precedem entidades mencionadas, como "*Sr.*", "*Dr.*", "*empresa*", etc.

composto por 10 estados associados aos tipos de EM existentes no corpus *Coleção Dourada do Primeiro HAREM*[16] (Rocha e Santos, 2007), uma vez que este corpus foi utilizado para o treinamento do ICC-HMM durante a realização dos experimentos. Esses conjuntos são apresentados nas equações 6.7 a 6.9.

$$S\_Cl\_GAZ = \{pessoa, organização, local, abstração, evento, arte/mídia/comunicação, produto, papelada, substância, natureza, outros\} \quad \text{Eq. (6.7)}$$

$$S\_Cl\_naoEM = \{não\text{-}EM\} \quad \text{Eq. (6.8)}$$

$$S\_Cl\_EM = \{pessoa, organização, local, abstração, tempo, valor, obra, coisa, acontecimento, variado\} \quad \text{Eq. (6.9)}$$

Vale ressaltar que, apesar do ICC-HMM ter sido desenvolvido utilizando os conjuntos de estados representados nas equações 6.7 a 6.9, o modelo suporta o uso de quaisquer *gazetteers* ou corpus de treinamento que impliquem em alterações dos estados contidos, respectivamente, nos conjuntos *S_Cl_GAZ* e *S_Cl_EM*, desde que essas coleções contenham exemplos associados às categorias de EM *pessoa* e *organização*.

As fórmulas dos cálculos de probabilidade das equações 6.2 e 6.3 do modelo de classificação são as mesmas utilizadas no modelo de identificação (exceto pelo fato de *Cl* diferir de *Id*).

A equação 6.4, $P(Cl_t \mid O_{t\text{-}1})$, é referente à probabilidade de geração do estado pelo *token* que precede o *token* corrente. Este cálculo interfere na classificação do primeiro *token* de uma EM, através dos títulos ("*Sr.*", "*Dr.*", "*empresa*", etc.) que costumam precedê-lo, bem como na classificação dos demais *tokens* pertencentes a uma EM. Por exemplo, na sequência "*O Sr. josé Carlos Pacheco fez...*", o título "Sr." auxilia na predição da classe *pessoa* associada ao próximo *token* ("*josé*"), assim como o *token* "*josé*" influencia na inferência da classe *pessoa* associada ao *token* seguinte. A não utilização da *feature F* neste cálculo possibilita a inferência de EM atípicas, como, por exemplo, nomes de pessoas ou organização iniciados em letra minúscula.

A equação 6.5 do modelo de classificação foi denominada probabilidade de contexto do ICC-HMM e tem o objetivo de calcular a probabilidade de geração de um estado, dadas as informações do contexto do *token* corrente. O termo "contexto", no presente trabalho, é

[16] Os corpora usados neste trabalho são detalhados nos experimentos (capítulo 7).

referente às informações contidas nos *tokens* localizados próximos ao *token* corrente. A probabilidade de contexto considera as combinações (produto cartesiano) de todos os *tokens* do intervalo $O_{t\text{-}2:t+2}$, exceto O_t, nos exemplos de treinamento, e constitui um importante recurso para atenuar os problemas de não cobertura e ambiguidade durante o processo de etiquetação de sequências. A Figura 6.2 exemplifica a utilização deste recurso.

Tarefa: Classificação de EM – etiquetação de sequências.

Sentença: "*Pede pro Paulo apagar as evidências.*".

Token corrente: O_t = "*Paulo*".

Tokens do contexto: $O_{t\text{-}2}$ = "*Pede*", $O_{t\text{-}1}$ = "*pro*", O_{t+1} = "*apagar*", O_{t+2} = "*as*".

Funcionamento da probabilidade de contexto:

1. Calcula-se $P(Cl_t \mid O_{t\text{-}2} \subset O_{t\text{-}2,t\text{-}1,t+1,t+2})$, a probabilidade do *token* "*pede*" ocorrer nos exemplos de treinamento dentre as posições t-2, t-1, t+1 ou t+2 para cada estado de Cl_t treinado (*pessoa*, *organização*, *local*, etc.).
   O resultado de $P(Cl_t \mid O_{t\text{-}2} \subset O_{t\text{-}2,t\text{-}1,t+1,t+2})$ será algo parecido com:
   P(PESSOA | $O_{t\text{-}2}$ ou $O_{t\text{-}1}$ ou O_{t+1} ou O_{t+2}="*Pede*") = 0,60
   P(ORG | $O_{t\text{-}2}$ ou $O_{t\text{-}1}$ ou O_{t+1} ou O_{t+2}="*Pede*") = 0,25
   P(LOCAL | $O_{t\text{-}2}$ ou $O_{t\text{-}1}$ ou O_{t+1} ou O_{t+2}="*Pede*") = 0,15
2. Em seguida, repete-se o cálculo do passo (1), dessa vez aplicado ao *tokens* "*pro*" ($O_{t\text{-}1}$), "*apagar*" (O_{t+1}) e "as" (O_{t+2}).
3. Por fim, para cada estado (*pessoa*, *organização*, *local*, etc.), calcula-se a média aritmética simples das quatro probabilidades obtidas nos passos (1) e (2). Estes são os valores de

$$\sum_{u=t-2}^{t+2} P(Cl_t \mid O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2})/4,\ u \neq t.$$

Figura 6.2 - Exemplo de funcionamento da probabilidade de contexto do ICC-HMM

O presente trabalho faz uso do gazetteer REPENTINO (Sarmento et al., 2006), cujo modelo está representado na equação 6.6, $P_{\text{gazetteer}}(Cl_t* \mid O_t)$, que calcula a probabilidade de geração dos estados que compõem o *gazetteer*, dado o *token* corrente. Em (Sarmento, 2005), um dos autores do REPENTINO sugere que sejam realizadas correções quando do

seu uso em sistemas de REM em virtude do grande desequilíbrio numérico existente entre as diferentes categorias de EM[17]. Dado este fato, além da grande quantidade de instâncias existente no REPENTINO (450.000) e também a alta taxa de repetição de instâncias unigrama[18] de EM em categorias diferentes (9%), optou-se por uma modelagem estatística ajustada do *gazetteer* no presente trabalho, cujo fator de ajuste aplicado a determinada categoria é proporcional ao número de instâncias unigrama pertencentes à categoria. A Tabela 6.2 apresenta as categorias existentes no REPENTINO, a distribuição de instâncias n-grama e unigrama e o fator de ajuste por categoria[19]. Vale ressaltar que nos cálculos das probabilidades são ignoradas palavras identificadas como *stopwords* (artigos, preposições e conjunções). Observa-se que o conjunto de estados presentes neste *gazetteer*, denominado *S_Cl_GAZ* e apresentado na Tabela 6.2, difere do conjunto *S_Cl_EM* (equação 6.9), também utilizado no modelo de classificação. Entretanto, como o objetivo do ICC-HMM é reconhecer somente as EM das categorias *pessoa* e *organização*, e a única categoria, além das próprias, que interfere na predição dessas EM é a categoria *local* (por conter considerável quantidade de instâncias compartilhadas com os tipos *pessoa* e *organização*), conclui-se que a divergência entre os grupos *S_Cl_EM* e *S_Cl_GAZ* não causa prejuízo à solução proposta, pois as categorias *pessoa*, *organização* e *local* estão presentes nos dois conjuntos.

Tabela 6.2 - Fator de ajuste e distribuição das instâncias por categoria no *gazetteer* REPENTINO

Categoria	**Número de instâncias**		**Fator de ajuste**
	N-grama	**Unigrama**	
Pessoa	286.297	879.774	0,642
Organização	46.869	239.338	0,175
Local	49.451	96.978	0,071
Evento	25.357	60.453	0,044
Arte/Mídia/Comunicação (A/M/C)	15.232	37.200	0,027
Produto	9.199	20.978	0,015
Abstração	5.807	14.280	0,010
Papelada	4.427	11.067	0,008

[17] Os números da distribuição de instâncias por categorias de EM, assim como outros detalhes do REPENTINO, são apresentados na Seção 5.3.
[18] Consideram-se instâncias unigrama os exemplos de nomes contidos no REPENTINO compostos por somente uma palavra.
[19] As instâncias n-grama são as EM compostas por uma ou mais palavras, conforme registradas originalmente no REPENTINO, e o conjunto das instâncias unigrama contém cada palavra presente nas instâncias n-grama.

Tabela 6.2 (continuação)

Categoria	Número de instâncias		Fator de ajuste
	N-grama	Unigrama	
Outros	1.771	4.357	0,003
Substância	1.468	3.601	0,003
Natureza	867	2.133	0,002
Total	446.745	1.370.158	

Como exemplo de utilização do *gazetteer* no modelo, dados os números apresentados na Tabela 6.2, o cálculo de $P_{\text{gazetteer}}(Cl_t* \mid \text{“}Macedo\text{”})$ gera as seguintes probabilidades não nulas de geração de estados (onde *c*(*evento*) indica a contagem do *evento* nos exemplos de treinamento e a variável *ajuste* representa o fator de ajuste da categoria, obtido da Tabela 6.2):

- $P_{\text{gazetteer}}(PESSOA \mid \text{“}Macedo\text{”}) = \frac{c(PESSOA,\text{“}Macedo\text{”})}{c(\text{“}Macedo\text{”})} \cdot ajuste = \frac{836}{891} \cdot 0{,}642 = 0{,}602$;
- $P_{\text{gazetteer}}(ORG \mid \text{“}Macedo\text{”}) = \frac{c(ORG,\text{“}Macedo\text{”})}{c(\text{“}Macedo\text{”})} \cdot ajuste = \frac{40}{891} \cdot 0{,}175 = 0{,}008$;
- $P_{\text{gazetteer}}(LOCAL \mid \text{“}Macedo\text{”}) = \frac{c(LOCAL,\text{“}Macedo\text{”})}{c(\text{“}Macedo\text{”})} \cdot ajuste = \frac{15}{891} \cdot 0{,}071 = 0{,}001$.

6.3.3 – Modelos de *back-off*

O cenário ideal para um sistema de REM é que os exemplos de treinamento contenham todos os *tokens* e *n-gramas* possíveis, de modo a atenderem plenamente as expressões de probabilidade do modelo durante o processo de obtenção de sequência ótima de estados. No entanto, isso não ocorre na prática, e faz-se então necessária a criação dos modelos de *back-off*, que, apesar de mais genéricos e portanto com menor poder de acurácia em relação aos modelos de alto nível, cumprem a tarefa de garantir que a cada *token* observável seja associado um estado.

O ICC-HMM utiliza somente um nível de *back-off* abaixo dos modelos de alto nível. A principal alteração implementada no modelo de *back-off*, em relação ao modelo de alto nível, é a substituição do *token O* pelo seu *stemming f_stem* nos cálculos das probabilidades. Optou-se por essa abordagem por dois motivos, primeiro porque os modelos de *back-off* tendem a degradar a acurácia dos modelos de etiquetação de sequências (Bikel et al., 1999), então não é recomendada a criação de muitos níveis de *back-off*, e o segundo motivo é que, pelo fato do presente modelo ser baseado na soma de

probabilidades, e não no produto, e do modelo de *back-off* utilizar somente as *features* dos *tokens* (*F* e *f_stem*), ao invés dos próprios *tokens*, reduz-se de forma considerável o risco de não obtenção da sequência ótima de estados, pois a quantidade de exemplos das features no treinamento tende a ser suficiente para garantir a robustez dos cálculos de probabilidade. Além disso, nos casos de palavras ou bigramas desconhecidos nos modelos principal e de *back-off* no cálculo de uma das probabilidades, a probabilidade resultante (soma das parcelas de cálculos de probabilidades) não é anulada, mas sim passa a contar com uma parcela a menos no resultado da soma. Por fim, como forma de garantir a geração do estado dado qualquer *token*, assume-se que, nos raros casos de palavras e bigramas desconhecidos nos modelos principal e de *back-off* para os dois estados no modelo de identificação, atribui-se probabilidade final igual a 1 para o estado *não-EM*, ou seja, assume-se que essas palavras não são entidades mencionadas.

Os cálculos de probabilidade que compõem os modelos de identificação e classificação do ICC-HMM, apresentados nas equações 6.1 a 6.6, foram alterados para contemplar o modelo de *back-off* definido para cada um deles. A Tabela 6.3 apresenta o modelo de *back-off* definido para cada parcela, onde f_stem_t representa o stemming do token O_t.

Tabela 6.3 - Modelos de *back-off* do ICC-HMM

Modelo de alto nível	**Modelo de *back-off***	
$P(Id_t \mid Id_{t\text{-}1}, <O,F>_{t\text{-}1:\,t+1})$	$P(Id_t \mid Id_{t\text{-}1}, F_{t\text{-}1:\,t+1})$	Eq. (6.10)
$P(Id_t \mid O_t)$	$P(Id_t \mid f\_stem_t)$	Eq. (6.11)
$P(Cl_t \mid Cl_{t\text{-}1}, <O,F>_{t\text{-}1:\,t+1})$	$P(Cl_t \mid Cl_{t\text{-}1}, F_{t\text{-}1:\,t+1})$	Eq. (6.12)
$P(Cl_t \mid O_t)$	$P(Cl_t \mid f\_stem_t)$	Eq. (6.13)
$P(Cl_t \mid O_{t\text{-}1})$	$P(Cl_t \mid f\_stem_{t\text{-}1})$	Eq. (6.14)
$\sum_{u=t-2}^{t+2} P(Cl_t \mid O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2})/4$, $u \neq t$	$\sum_{u=t-2}^{t+2} P(Cl_t \mid f\_stem_u \subset f\_stem_{t\text{-}2,t\text{-}1,t+1,t+2})/4$, $u \neq t$	Eq. (6.15)
$P_{\text{gazetteer}}(Cl_t^* \mid O_t)$	$P_{\text{gazetteer}}(Cl_t^* \mid f\_stem_t)$	Eq. (6.16)

O cálculo do peso ϕ, usado no coeficiente aplicado à probabilidade de alto nível (coeficiente ϕ) e de *back-off* (coeficiente 1- ϕ), é similar ao apresentado em (Bikel et al., 1999), e é definido por:

$$\Phi = \left(1 - \frac{c(Y)}{c(Y')}\right) \cdot \frac{1}{1+\frac{ocorrências\ únicas\ de\ Y}{c(Y)}}, \qquad \text{Eq. (6.17)}$$

sendo que $P(X|Y)$ é a probabilidade do modelo de alto nível, $P(X|Y')$ a probabilidade do modelo de *back-off* e $c(Y)$ é a contagem de Y nos exemplos de treinamento.

O primeiro fator da equação 6.17, $\left(1 - \frac{c(Y)}{c(Y')}\right)$, distribui o peso de acordo com o nível de especialização de cada modelo, de forma que, quanto mais específico o modelo, maior o peso a ele atribuído, e o segundo fator, $\frac{1}{1+\frac{ocorrências\ únicas\ de\ Y}{c(Y)}}$, calcula o grau de suavização[20], inversamente proporcional ao número de ocorrências únicas do modelo, e pode ser interpretado da seguinte forma: quanto menor o índice de ocorrências únicas do modelo de alto nível, maior é o valor do segundo fator da equação 6.17 e, consequentemente, maior é o grau de certeza do modelo de alto nível (que usa o coeficiente Φ) e menor grau é atribuído ao modelo de *back-off* (que usa o coeficiente Φ-1).

O cálculo de Φ é similar ao apresentado em (Bikel et al., 1999), entretanto no ICC-HMM usa-se o índice de ocorrências únicas de Y, ao invés do índice de saídas únicas em t-1, uma vez que a solução proposta explora o contexto do *token* corrente no intervalo t-2 a t+2, ou seja, não é restrita aos bigramas t-1 $\rightarrow$ t.

6.4 – DECODIFICAÇÃO

Conforme apresentado previamente neste capítulo, o modelo de identificação do ICC-HMM transforma a sequência $O = O_1\ O_2\ ...\ O_T$ na sequência $O = (O, Id)_1\ (O, Id)_2\ ...\ (O, Id)_T$, de forma que $S\_Id = \{EM, não\text{-}EM\}$. Essa transformação é decodificada com o uso do algoritmo de Viterbi apresentado em (Rabiner, 1990), que, através de técnicas de programação dinâmica, garante eficiência no cálculo da sequência mais provável de estados com complexidade[21] $\theta(TN^2)$, ao invés de $\theta(2TN^T)$, que é a complexidade associada ao cálculo das probabilidades de todas as sequências candidatas (Rabiner, 1990).

[20] Suavização é o nome dado para a utilização de mais de um nível de modelo para calcular determinada probabilidade, de modo distribuir a massa probabilística de acordo com a adaptação de cada nível aos exemplos de treinamento (Feldman e Sanger, 2007).
[21] N é o número de estados do modelo e T o tamanho da sequência observável.

O modelo de classificação, por sua vez, recebe como entrada os *tokens* pré-etiquetados pelo modelo de identificação. Ao percorrer a sequência $(O, Id)= (O, Id)_1 (O, Id)_2 ... (O, Id)_T$, há uma alteração no comportamento do algoritmo de Viterbi nos instantes t em que $Id_t = não\text{-}EM$. O algoritmo representado pelo pseudocódigo na Figura 6.3 detalha este comportamento durante a decodificação do modelo de classificação do ICC-HMM.

Considerando-se que:

- *S_Cl_EM* é o conjunto de estados usado no modelo de classificação contendo as 10 classes de EM apresentadas na equação 6.9;
- *S_Cl_naoEM* é o conjunto de estado usado no modelo de classificação contendo somente o estado *não-EM*;
- Cl_t é o estado do modelo de classificação mais provável no tempo t;
- $(O, Id)_1 (O, Id)_2 ... (O, Id)_T$ é a sequência de junções entre *tokens* e estados gerada pela decodificação do modelo de identificação do ICC-HMM;
- $\delta_t(i) = \max P[Cl_1 Cl_2 ... Cl_t = i , O_1 O_2 ... O_t \mid \lambda]$ é a máxima probabilidade obtida por uma sequência de estados no tempo t, associada às primeiras t observações e cujo estado final é $S\_Cl\_EM_i$ ou $S\_Cl\_naoEM_i$;
- Por recursividade, $\delta_{t+1}(j) = [\max \delta_t(i) \cdot a_{ij}] \cdot P(S\_Cl\_EM_j$ ou $S\_Cl\_naoEM_j \mid O_{t+1})$;
- $\psi_t(j)$ é o vetor usado para memorizar o estado que maximizou $\delta_{t+1}(j)$ para cada t e cada j.

A decodificação da sequência $O_1 O_2 ... O_T$ no modelo de classificação se dá em 4 etapas:

1. Inicialização ($t = 1$)
   Se ($Id_t = não\text{-}EM$) {
   $\delta_1(i) = \pi_i \cdot P(S\_Cl\_naoEM_i \mid O_1)$, $i = 1$ }
   Caso contrário {
   $\delta_1(i) = \pi_i \cdot P(S\_Cl\_EM_i \mid O_1)$, $1 \le i \le 10$ }
   $\psi_1(i) = 0$.
2. Recursividade[22] ($2 \le t \le T$)
   Se ($Id_t = não\text{-}EM$) {
   $\delta_t(j) = max_{(1 \le i \le N)} [\delta_{t-1}(i) \cdot a_{ij}] \cdot P(S\_Cl\_naoEM_j \mid O_t)$, $j = 1$ }
   Caso contrário {
   $\delta_t(j) = max_{(1 \le i \le N)} [\delta_{t-1}(i) \cdot a_{ij}] \cdot P(S\_Cl\_EM_j \mid O_t)$, $1 \le j \le 10$ }
   $\psi_t(j) = argmax_{(1 \le i \le N)} [\delta_{t-1}(i) \cdot a_{ij}]$, $1 \le j \le 10$
3. Finalização ($t = T$)
   $Cl_T = argmax_{(1 \le i \le N)} [\delta_T(i)]$.
4. Recuperação da sequência (*backtracking*)
   $Cl_t = \psi_{t+1}(Cl_{t+1})$, $t = T - 1, T - 2, ..., 1$.

Figura 6.3 - Pseudocódigo do algoritmo de Viterbi adaptado, utilizado na decodificação do modelo de classificação do ICC-HMM

Na Figura 6.3, as etapas 1 e 2 percorrem a sequência temporal t, de 1 a T, com o objetivo de memorizar as maiores probabilidades de junção entre observações e estados para cada t

[22] A função *max* retorna a maior probabilidade e a função *argmax* retorna o argumento (estado) que maximizou a probabilidade

e *j*, enquanto que a recuperação efetiva da sequência de estados mais provável se dá nas etapas 3 e 4.

A principal variação deste algoritmo em relação ao algoritmo de Viterbi apresentado na seção 4.3 está relacionada ao fato da sequência de *tokens* de entrada conter algumas junções pré-definidas pelo modelo de identificação que permanecem inalteradas durante a decodificação do modelo de classificação, que são as junções identificadas com o estado *não-EM*. Isso faz com que, nos instantes *t* associados a essas junções, o algoritmo utilize o conjunto de estados *S_Cl_naoEM*, que contém somente um estado denominado *não-EM*, e nos demais instantes seja utilizado o conjunto de estados *S_Cl_EM*, contendo as dez classes de EM existentes no modelo (*pessoa*, *organização*, *local*, etc.).

A Figura 6.4 ilustra este comportamento usando a sentença "*Paulo César frequentou o Senado Federal ativamente*" como exemplo.

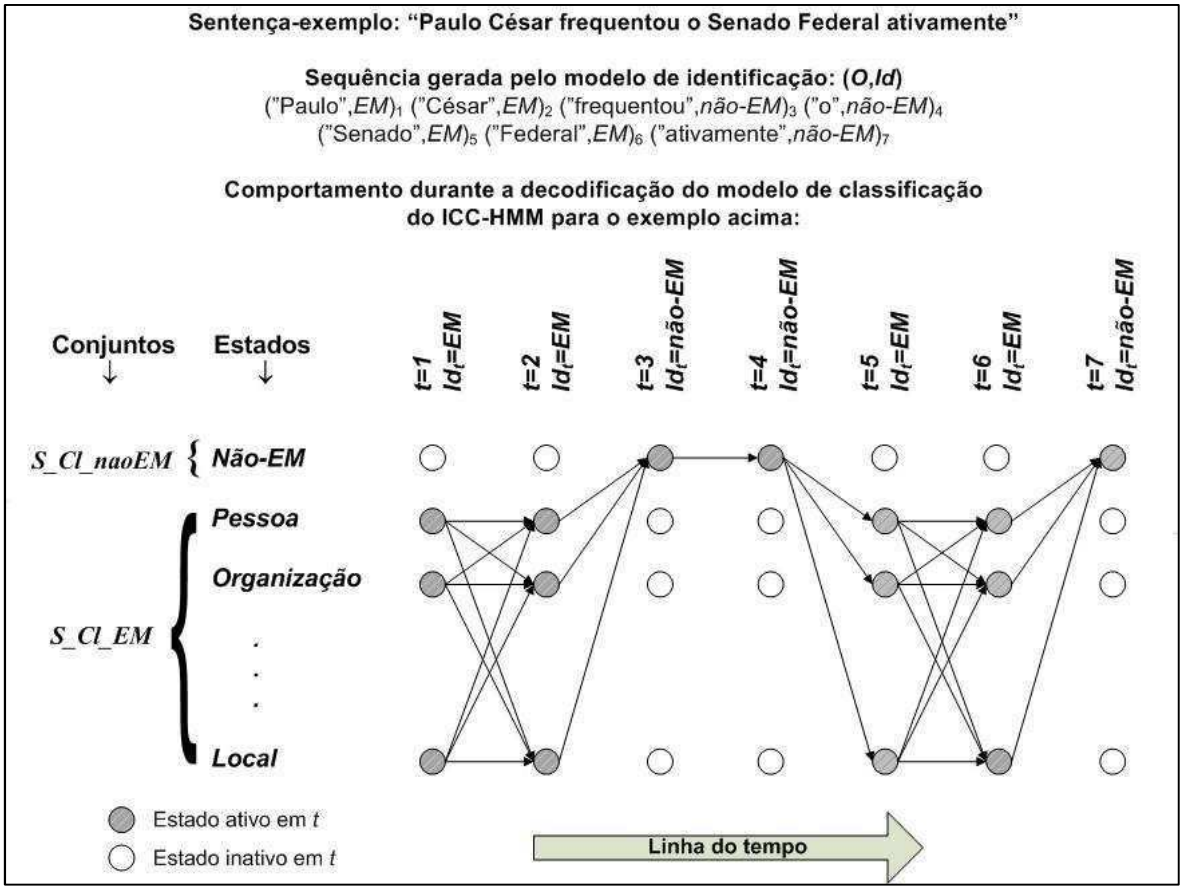


Figura 6.4 - Exemplo de comportamento do modelo de classificação do ICC-HMM durante a decodificação

Na Figura 6.4, pode-se perceber que nos instantes *t*=3, 4 e 7 a probabilidade dos *tokens* "*frequentou*", "*o*" e "*ativamente*" gerarem o estado *não-EM* é igual a 1, pois é o único estado disponível nesses instantes t. Por outro lado, percebe-se também que nos instantes t=1, 2, 5 e 6 a probabilidade do estado *não-EM* ser gerado, respectivamente, pelos *tokens* "*Paulo*", "*César*", "*Senado*" e "*Federal*" é nula, pois nesses instantes toda a massa probabilística é dividida entre os estados do conjunto *S_Cl_EM*.

Além disso, nos instantes *t*=3 e 7, o único cálculo realizado no modelo é $P(Cl_t \mid Cl_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1})$, apresentado na equação 6.2, pois este calcula a probabilidade de transição de estados, que varia de acordo com o estado anterior, contido em *S_Cl_EM*. Os demais cálculos (equações 6.3 a 6.6) são dispensáveis por estar associados à probabilidade de geração do estado, dado o contexto do *token* corrente, ou seja, como há somente um estado nesses instantes, essa probabilidade é 1, independente do *token* corrente. Já no instante *t*=4, nenhum cálculo é realizado, pois tanto a probabilidade de transição de estados quanto a da sua geração são invariáveis, uma vez que os estados em *t* e *t*-1 são únicos, iguais a *não-EM*.

Conforme apresentado no início desta seção, o algoritmo de Viterbi garante eficiência no cálculo da sequência mais provável de estados com complexidade $\theta(TN^2)$. Como, no ICC-HMM, esse cálculo implica na execução do algoritmo de Viterbi por duas vezes, a primeira associada ao modelo de identificação e a segunda ao modelo de classificação, conclui-se que a obtenção da sequência de estados mais provável pelo ICC-HMM é realizada com complexidade $(4T + TN^2)$ multiplicações, pois durante a primeira execução o número de estados *N* é fixo, igual a 2 (*EM* e *não-EM*).

6.5 – TREINAMENTO

O treinamento do ICC-HMM é realizado através da contagem dos exemplos do corpus de treinamento, com o objetivo de registrar as probabilidades apresentadas nas equações 6.1 a 6.6. Tais contagens são representadas nas equações 6.18 a 6.24, onde *c*(*evento*) representa o número de vezes que o *evento* ocorre no corpus de treinamento:

$$P(Id_t \mid Id_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1}) = \frac{c(Id_t, Id_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1})}{c(Id_{t\text{-}1}, <O,F>_{t\text{-}1:\, t+1})}, \qquad \text{Eq. (6.18)}$$

$$P(Id_t \mid O_t) = \frac{c(Id_t\ ,\ O_t)}{c(O_t)}\ , \qquad \text{Eq. (6.19)}$$

$$P(Cl_t \mid Cl_{t\text{-}1}\ ,\ <O,F>_{t\text{-}1:\ t+1}) = \frac{c(Cl_t\ ,\ Cl_{t\text{-}1}\ ,\ <O,F>_{t\text{-}1:\ t+1})}{c(Cl_{t\text{-}1}\ ,\ <O,F>_{t\text{-}1:\ t+1})}\ , \qquad \text{Eq. (6.20)}$$

$$P(Cl_t \mid O_t) = \frac{c(Cl_t\ ,\ O_t)}{c(O_t)}\ , \qquad \text{Eq. (6.21)}$$

$$P(Cl_t \mid O_{t\text{-}1}) = \frac{c(Cl_t\ ,\ O_{t\text{-}1})}{c(O_{t\text{-}1})}\ , \qquad \text{Eq. (6.22)}$$

$$P(Cl_t \mid O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2}) = \frac{c\left(Cl_t\ ,\ O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2}\right)}{c\left(O_u \subset O_{t\text{-}2,t\text{-}1,t+1,t+2}\right)}\ , \qquad \text{Eq. (6.23)}$$

u=t-2, t-1, t+1 e t+2, e

$$P_{\text{gazetteer}}(Cl_t{*} \mid O_t) = \frac{c(Clt{*},\ Ot)}{c(Ot)}\ . \qquad \text{Eq. (6.24)}$$

Na equação 6.24, vale ressaltar que o corpus de treinamento considerado é o *gazetteer* REPENTINO, e não o corpus utilizado nos treinamentos das demais probabilidades. Por fim, a mesma estratégia usada nos treinamentos representados nas equações 6.18 a 6.24 deve ser aplicada ao treinamento dos respectivos modelos de *back-off*.

Experimentos foram realizados utilizando este modelo proposto e são apresentados no próximo capitulo, com o intuito de avaliar a acurácia do algoritmo desenvolvido e compará-lo com outros modelos de REM.

7 - EXPERIMENTOS E RESULTADOS

Este capítulo descreve os experimentos realizados no presente trabalho, apresenta uma análise comparativa dos seus resultados e, por fim, descreve o resultado apresentado como um produto utilizado no cenário forense.

7.1 – CENÁRIO

Os experimentos do presente trabalho foram divididos em duas etapas, realizadas de acordo com os seguintes objetivos:

1. Comparar o desempenho do ICC-HMM com os desempenhos apresentados na avaliação conjunta do segundo HAREM (Carvalho et al., 2008);
2. Avaliar a acurácia do ICC-HMM aplicado a textos forenses reais.

O segundo HAREM foi um evento realizado pelo projeto Linguateca (Santos et al., 2004) no ano de 2008 em Portugal, que objetivou avaliar sistemas de REM em textos na língua portuguesa. O evento contou com a participação de dez sistemas, sendo que o Rembrandt (Cardoso, 2008) foi o que apresentou melhores resultados na tarefa de reconhecer nomes de pessoas e organizações e, por isso, é o foco das comparações realizadas no presente trabalho. O Rembrandt é um sistema determinístico baseado em regras manuais e o ICC-HMM, proposto neste trabalho, é um sistema probabilístico baseado em aprendizado de máquina.

Conforme apresentado em (Carvalho et al., 2008), um dos objetivos do HAREM é avaliar sistemas especializados em reconhecer categorias específicas de EM e que, portanto, não têm o propósito de obter bons resultados em todas as categorias avaliadas no evento. Essa característica permite que o ICC-HMM seja avaliado e comparado especificamente nas categorias *pessoa* e *organização*, quanto às tarefas de identificação e classificação das EM.

O projeto Linguateca dispõe de dois corpora etiquetados com EM, o *Coleção Dourada 1*, doravante denominado *CD1*, usado na avaliação do primeiro HAREM (Santos et al., 2006) no ano de 2005, e o *Coleção dourada 2*, ou *CD2*, usado na avaliação do segundo HAREM. Durante a primeira etapa dos experimentos, o ICC-HMM utilizou o *CD1* como corpus de treinamento e o *CD2* para a avaliação e comparação com o sistema Rembrandt.

A segunda etapa de experimentos manteve a utilização do corpus *CD1* para o treinamento, entretanto a avaliação foi aplicada a um corpus formado por 32 textos de arquivos contidos em mídias apreendidas em duas grandes operações da Polícia Federal do Brasil, cujas EM foram etiquetadas e revisadas manualmente. Esse corpus é referenciado no presente trabalho como *corpus forense*.

7.2 – MÉTRICAS DE AVALIAÇÃO

A avaliação dos sistemas de REM é comumente realizada através das métricas de acurácia, que objetivam expressar por números o desempenho de um sistema quanto aos seus acertos e erros. Desde o evento MUC-6 (Grishman e Sundheim, 1996), quando ocorreram as primeiras avaliações conjuntas desses sistemas, tem-se usado as medidas precisão (*P*), do inglês *precision*, revocação (*R*), do inglês *recall*, e a combinação entre elas, denominada medida-F (*F*), do inglês *F-measure*. Tais medidas foram apresentadas em (Rijsbergen, 1979) e são expressas pelas seguintes razões:

$$P = \frac{Nº\ de\ EM\ reconhecidas\ corretamente\ pelo\ sistema}{Nº\ de\ EM\ reconhecidas\ pelo\ sistema}\ ; \qquad \text{Eq. (7.1)}$$

$$R = \frac{Nº\ de\ EM\ reconhecidas\ corretamente\ pelo\ sistema}{Nº\ de\ EM\ existentes\ no\ corpus}\ ; \qquad \text{Eq. (7.2)}$$

$$F = \frac{(\rho^2+1)\cdot P\cdot R}{\rho^2\cdot P + R}\ . \qquad \text{Eq. (7.3)}$$

No cálculo de *F*, o parâmetro real não negativo ρ tem a função de distribuir os pesos entre *P* e *R*, de modo que, se $\rho > 1$ então o peso de *R* é superior ao de *R*, e se $\rho < 1$ então ocorre o contrário. No presente trabalho, os valores de *F* representam a combinação harmônica entre *P* e *R*, dada por $\rho = 1$, da seguinte forma:

$$F = \frac{2\cdot P\cdot R}{P + R}\ . \qquad \text{Eq. (7.4)}$$

O HAREM também utilizou este sistema de métricas para avaliar as tarefas de identificação e classificação de EM. Conforme apresentado em (Santos et al., 2007), para que uma EM etiquetada por um sistema seja considerada correta no HAREM, todas as palavras que a compõem devem estar corretas. Caso isso não aconteça, mas o sistema anote pelo menos uma palavra correta da EM, esta anotação é considerada parcialmente correta por defeito ou por excesso. A etiquetação é parcialmente correta *por defeito* quando

o número de palavras que compõem a EM do sistema é inferior ao do treinamento e, quando ocorre o inverso, a etiquetação é parcialmente correta *por excesso*. É atribuído o valor 1 às EM pontuadas corretamente e, para as EM parcialmente corretas, é atribuído o valor representado na Equação 7.5.

$$0{,}5 \cdot \frac{número\ de\ palavras\ anotadas\ corretamente\ pelo\ sistema}{número\ de\ palavras\ resultante\ da\ união\ das\ EM\ do\ sistema\ e\ do\ treinamento}. \quad \text{Eq. (7.5)}$$

A Tabela 7.1 apresenta alguns exemplos de pontuação atribuída à anotação de EM. A primeira linha da tabela mostra a etiquetação parcialmente correta por excesso, uma vez que o *token* "*tem*" foi anotado erroneamente, na segunda linha ocorre acerto parcial por defeito, pois o sistema não etiquetou os *tokens* "*Ricardo*", "*Ramos*" e "*Silva*", e a terceira linha apresenta um exemplo de etiquetação correta.

Tabela 7.1 - Exemplos de pontuação para EM anotadas do tipo PESSOA

Treinamento	**Saída do sistema**	**Pontuação**
Paulo	Paulo tem	$0{,}5 \cdot \frac{1}{2} = 0{,}25$
Paulo Ricardo Ramos Silva	Paulo	$0{,}5 \cdot \frac{1}{4} = 0{,}125$
Paulo Roberto	Paulo Roberto	1

A exemplo do HAREM, no presente trabalho as avaliações foram realizadas considerando as tarefas de identificação e classificação de EM. A primeira tarefa testa a capacidade de identificação do sistema, ou seja, a informação sobre um *token* ser ou não ser EM, independente de ter sido classificado como pessoa, organização ou qualquer outra categoria. A avaliação da segunda tarefa, classificação, verifica se o *token* foi classificado corretamente pelo sistema de REM, de acordo com a categoria de EM à qual ele pertence.

7.3 – CORPORA UTILIZADOS

Conforme mencionado, neste trabalho foram utilizados três corpora, o *CD1*, o *CD2* e o corpus forense. O ICC-HMM usou o *CD1* como corpus de treinamento e os demais foram usados para a avaliação do sistema.

7.3.1 – *CD1* - o corpus de treinamento

Para treinar o ICC-HMM foi utilizado o corpus *Coleção Dourada do primeiro HAREM* (Rocha e Santos, 2007), ou *CD1*, composto por 133.569 palavras, sendo 16.821 delas agrupadas em 8.976 entidades mencionadas etiquetadas.

A escolha da *CD1* como corpus de treinamento se deu pelo fato de ser um dos dois únicos corpora etiquetados com entidades mencionadas disponíveis gratuitamente[23].

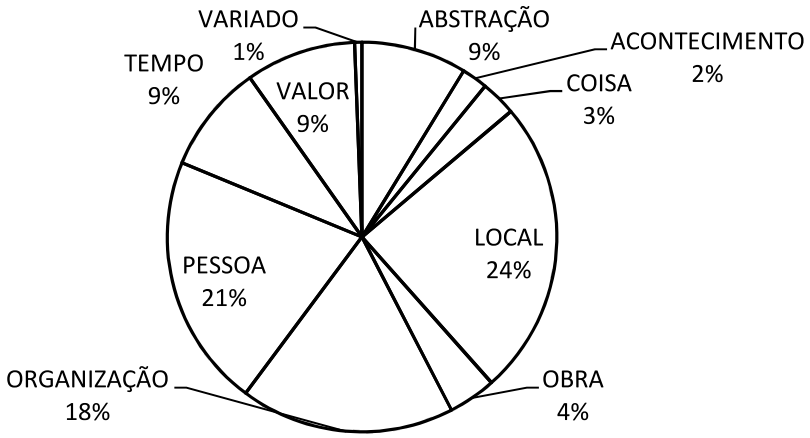


Figura 7.1 - Distribuição das categorias de EM presentes na *CD1*

A Figura 7.1 apresenta a distribuição das categorias de EM contidas nesse corpus. As entidades mencionadas *pessoa* e *organização* juntas representam 39% das EM presentes na CD1 e que, além dessas, existem outras oito categorias de EM anotadas. Apesar do ICC-HMM objetivar o reconhecimento de nomes de pessoas e organizações, é de fundamental importância que o corpus de treinamento contenha outras categorias de EM etiquetadas, pois isso garante o equilíbrio do modelo probabilístico. Se fossem consideradas no treinamento do ICC-HMM somente as categorias *pessoa* e *organização*, o modelo de classificação seria tendencioso ao distribuir a massa de probabilidades de classificação de um *token* entre as categorias candidatas, pois, uma vez que o *token* fosse identificado como EM pelo modelo de identificação do ICC-HMM, a classificação deste *token* ficaria limitada às categorias *pessoa* e *organização*. Por exemplo, se o *token* "Recife" for identificado como EM pelo modelo de identificação, o modelo de classificação só terá as opções *pessoa* e *organização* para classificá-lo.

[23] O outro corpus disponível gratuitamente é a *CD2*, apresentado na Seção 7.3.2.

Os documentos presentes na *CD1* originam de fontes distintas, contendo diferentes tipos textuais e variações no idioma. As Figuras 7.2 e 7.3 apresentam essas distribuições.

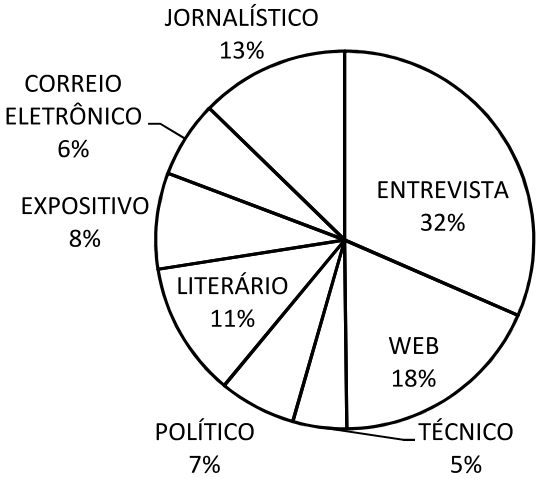


Figura 7.2 - Distribuição das palavras pelo tipo textual do seu documento de origem no corpus *CD1*

O fato da *CD1* conter os 8 tipos textuais apresentados na Figura 7.2 é um fator relevante para o objetivo da etiquetação de textos forenses do ICC-HMM, pois essa característica aplicada ao treinamento contribui para uma melhor preparação do modelo quanto ao reconhecimento de nomes em textos de arquivos de domínio desconhecido ou escritos em linguagem informal. Quanto maior for o número de tipos textuais presentes no treinamento, maiores são as chances de serem identificadas semelhanças entre textos de domínios desconhecidos e os textos treinados. Além disso, a existência de tipos textuais contendo escrita informal na *CD1*, como os tipos *entrevista* e *correio eletrônico*, contribui para que o treinamento tenha melhor adaptação aos textos forenses.

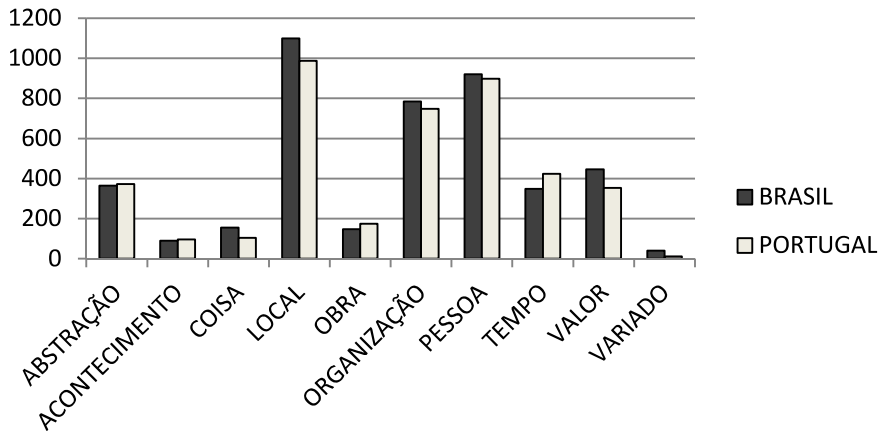


Figura 7.3 - Distribuição das categorias de EM por variação do idioma no corpus *CD1*

Conforme apresentado na Figura 7.3, a *CD1* contém textos escritos no idioma português de Portugal e do Brasil, sendo que a distribuição das EM quanto ao idioma é praticamente equânime. Apesar desta não ser a situação ideal para a segunda etapa dos experimentos, uma vez que a todos os textos forenses são escritos no idioma português do Brasil, entendeu-se que o fato dessas variações linguísticas serem bastante tênues não justifica a eliminação de cerca de 50% dos exemplos de treinamento para a obtenção de um conjunto de treinamento 100% escrito em português do Brasil.

7.3.2 – *CD2* - o corpus dos experimentos

O corpus avaliado na primeira etapa dos experimentos é a *CD2* - *Coleção Dourada do segundo HAREM* ((Carvalho et al., 2008) e (Mota et al., 2008a)). Este corpus é composto por 74.350 palavras divididas em 129 documentos e que contém mais de 7.000 entidades mencionadas etiquetadas.

A Figura 7.4 apresenta a distribuição das EM no *CD2* e mostra que, assim como ocorre com a *CD1*, as categorias de EM *pessoa* e *organização* representam 41% das categorias presentes no corpus. As demais oito categorias etiquetadas no corpus não são avaliadas no presente trabalho.

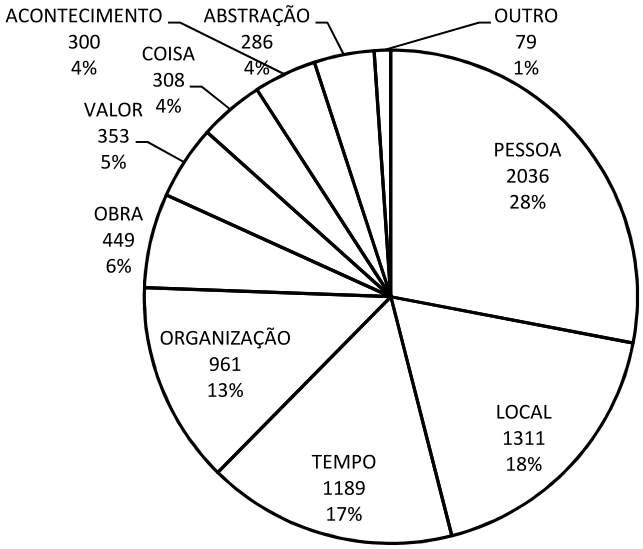


Figura 7.4 - Distribuição das categorias de EM presentes na *CD2*

A Figura 7.5 apresenta as variações de tipos textuais da *CD2*, que também ocorrem na *CD1* e são características que contêm alguns pontos em comum com as encontradas em textos forenses. Nesse sentido, destacam-se na CD2 a presença de vários tipos associados à

escrita informal, como *opinião*, *blog pessoal*, *entrevista* e *texto privado*, e também a grande variedade de tipos, que são 13 ao total.

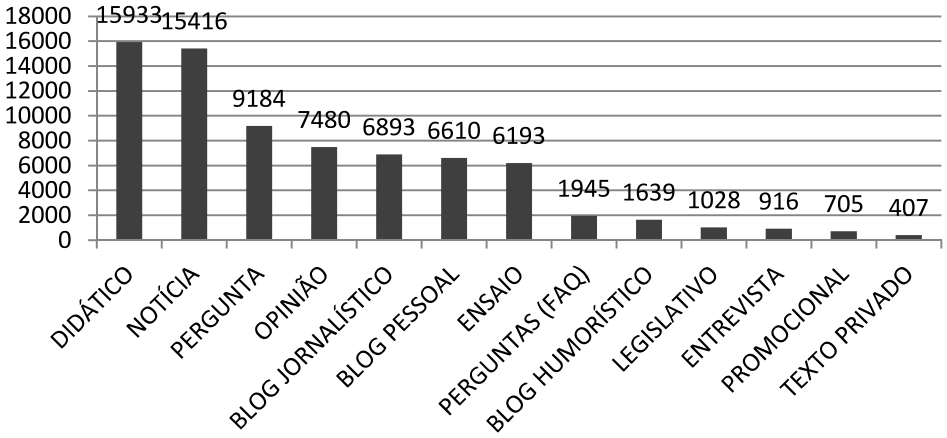


Figura 7.5 - Distribuição das palavras pelo tipo textual do seu documento de origem na *CD2*

Por fim, a Figura 7.6 apresenta a distribuição das palavras da *CD2* por variação no idioma, onde se observa que cerca de 60% delas estão contidas em textos escritos em português de Portugal. Entretanto, como os exemplos de treinamento também são divididos entre essas duas variações da língua e a construção frasal delas é bastante similar, entende-se que a avaliação dos textos escritos no idioma português de Portugal não implica em prejuízo para a interpretação dos resultados. Além disso, há a necessidade de avaliação do corpus completo para possibilitar as comparações desejadas com os resultados do sistema Rembrandt.

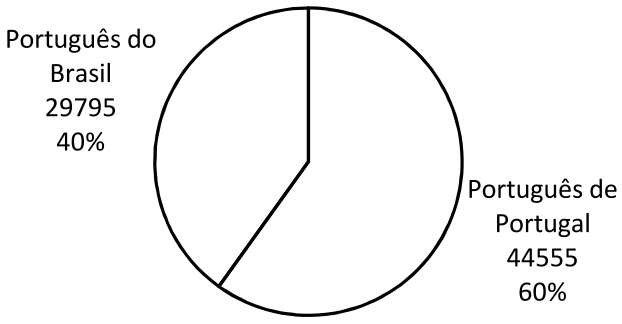


Figura 7.6 - Distribuição das palavras por variação do idioma

Os corpus *CD1* e *CD2* são disponibilizados publicamente para *download* no projeto Linguateca[24].

[24] *http://www.linguateca.pt.*

7.3.3 – Corpus forense

O corpus forense foi utilizado na segunda etapa dos experimentos. Este corpus contém 6.421 palavras divididas em 32 documentos textuais extraídos de arquivos de mídias apreendidas em duas operações da Polícia Federal do Brasil.

A seleção desses documentos foi realizada da seguinte forma: primeiramente, buscou-se agrupar os documentos em tipos textuais distintos contidos no material apreendido e, em seguida, de forma aleatória foram escolhidos os documentos de cada grupo em quantidades proporcionais aos tamanhos desses grupos. As EM das categorias *pessoa* e *organização* contidas nos documentos do corpus foram etiquetadas manualmente de modo a possibilitar a avaliação do corpus. Os grupos identificados, a quantidade de documentos por grupo e o número de EM etiquetadas em cada grupo são apresentados na Tabela 7.2.

Tabela 7.2 - Distribuição de documentos por tipo textual

Tipo textual	**Número de documentos**	**Número de EM *pessoa***	**Número de EM *organização***
Contrato	6	26	24
Licitação – Edital	6	21	27
Licitação – Proposta	4	11	14
Procuração	3	11	8
E-mail	5	13	17
Mensagem instantânea	6	11	9
Certificado de registro cadastral	2	2	6

De acordo com a Tabela 7.2, os contratos, licitações e mensagens são os tipos de documentos mais frequentes no corpus, que, no total, contém 95 nomes de pessoas e 105 nomes de organizações etiquetados. Essa distribuição é apresentada na Figura 7.7.

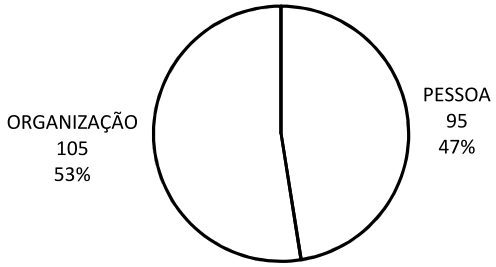


Figura 7.7 - Distribuição das categorias de EM no corpus forense

A Figura 7.7 mostra que, ao contrário do que ocorre nos corpora *CD1* e *CD2*, no corpus forense a quantidade de ocorrências da categoria *organização* (105) é superior à da categoria pessoa (95). Isso pode estar relacionado ao fato das investigações da Polícia Federal serem frequentemente associadas a delitos cometidos contra empresas e órgãos públicos.

Vale ressaltar que o texto contido no corpus forense não pode ser disponibilizado publicamente, por motivo de sigilo policial.

7.4 – AMBIENTE DE DESENVOLVIMENTO E EXECUÇÃO

O protótipo do ICC-HMM foi desenvolvido na linguagem de programação *Java* através do ambiente de desenvolvimento integrado *Eclipse Helios SR1* e, para armazenar os dados modelados, foi utilizado o SGBD[25] *MySQL Server 5.5.10*.

Os experimentos foram executados em um computador com processador *Intel Core2* com 2,33 GHz de *clock* e 3 GB de memória *RAM*.

7.5 – TAREFAS DE APOIO

Algumas tarefas de mineração de texto foram necessárias em fase de pré-processamento ou durante a modelagem do sistema, essas tarefas são a seguir apresentadas:

- Processamento preparatório: como os corpora *CD1* e *CD2* são disponibilizados no formato XML, a conversão de documentos para o formato em texto claro foi necessária somente para a geração do corpus forense, quando foram convertidos arquivos de extensão *doc*, *pdf* ou *html* em arquivos de extensão *txt*, que armazenavam somente o conteúdo textual do arquivo original, livre da presença de metadados; tal conversão foi realizada através do software comercial *Guidance Encase Forensics* (Guidance Software, 2011);
- Lista de abreviações: utilizada como recurso auxiliar na tarefa de segmentação de sentenças, a fim de possibilitar a correta interpretação do caractere *ponto*; a lista das abreviações utilizadas no ICC-HMM é apresentada no Anexo A;

[25] Sistema gerenciador de banco de dados.

- Segmentação em sentenças: tarefa que delimita as sentenças de um texto, é necessária porque modelos de REM dependem da interpretação da construção frasal para conseguir boa acurácia (Weiss et al., 2005); a segmentação foi implementada através de adaptações da API[26] *SentenceDetector* do software livre *CoGrOO* (Kinoshita et al., 2007);
- Tratamento de contrações: a separação de preposições contraídas com artigos, pronomes ou advérbios, como "*do*" ("*de*"+"*o*"), "*neste*" ("*em*"+"*este*") e daqui ("*de*"+"*aqui*"), permite que a tarefa da tokenização considere a menor unidade gramatical possível, o que aumenta a padronização das sequências textuais tratadas pelo modelo; a lista de contrações resolvidas no ICC-HMM é apresentada no Anexo B;
- Tokenização: esta tarefa é resolvida em duas etapas, primeiro, dentro de uma sentença, consideram-se *tokens* as palavras separadas por um ou mais espaços em branco, posteriormente cada *token* é tratado quanto à presença de abreviação, pontuação ou caracteres especiais e, se for necessário, subdivide-se o *token*;
- Lista de *Stopwords*: conjunto de palavras ignoradas em determinadas tarefas, por ocorrer em grande quantidade no treinamento e serem consideradas irrelevantes ou prejudiciais quanto ao objetivo que a tarefa deve atingir; são usadas no cálculo das probabilidades de contexto e de *gazetteer* do modelo de classificação do ICC-HMM; a lista das *stopwords* utilizadas na solução proposta é apresentada no Anexo C;
- *Stemming*: redução do *token* ao seu morfema, é utilizado como *feature* no modelo de *back-off* do ICC-HMM. Esta tarefa foi implementada através da API *ARBTFBrazilianAnalyzer* do software livre *Lucene* (Gospodnetic e Hatcher, 2004).

7.6 – EXPERIMENTOS

Os experimentos realizados envolveram a etapa de treinamento do modelo e em seguida a etapa de avaliação do mesmo.

[26] API é acrônimo para *Application Programming Interface*, uma interface de acesso para um componente de *software* reutilizável.

7.6.1 - Treinamento

O processo de treinamento do ICC-HMM se deu através da obtenção dos valores de probabilidade utilizados pelo modelo, apresentados na Seção 5.5, com base nos exemplos de treinamento do corpus *CD1*. Esses valores foram armazenados em tabelas de uma base de dados do SGBD *MySQL*.

7.6.2 – Primeira etapa

Na primeira etapa dos experimentos, foram avaliadas três variações do ICC-HMM, com o objetivo de comparar os resultados entre essas variações e o modelo proposto, quanto à acurácia e o tempo de execução. Esses testes foram aplicados ao corpus *CD2* e as variações propostas foram as seguintes:

1. **ICC-HMM com modelo único de identificação e classificação**: esta variação, doravante denominada "*modelo único*", resultou em um sistema que não contém o modelo de identificação do ICC-HMM apresentado na Seção 5.3.1. Além disso, o modelo de classificação, que passou a ser modelo único do sistema, foi adaptado de modo a conter somente um conjunto de estados, formado pelas dez categorias de EM presentes na *CD2* (vide Figura 7.4) mais o estado *não-EM*. Todas as outras características do ICC-HMM foram mantidas. Dessa forma, o modelo se tornou similar ao apresentado em (Todorovic et al., 2008), tendo como principais diferenças as *features* selecionadas e a forma de se explorar o contexto do *token* corrente;
2. **ICC-HMM sem uso de *gazetteer***: doravante identificada por "*sem gazetteer*", esta variação do ICC-HMM propôs a retirada do cálculo de geração do estado pelo *gazetteer* REPENTINO;
3. **ICC-HMM sem contexto**: doravante denominada "*sem contexto*", esta variação propôs a retirada do cálculo da probabilidade de contexto, apresentada na Seção 6.3.2.

Observa-se que essas variações provocam a retirada de três recursos do ICC-HMM, que são o modelo de identificação, o uso do *gazetteer* e a exploração do contexto do *token* corrente. Os experimentos da primeira etapa foram realizados com o intuito de identificar e

analisar o impacto provocado pela ausência desses três recursos no desempenho do modelo proposto.

7.6.3 – Segunda etapa

A segunda etapa dos experimentos objetivou comparar o desempenho do ICC-HMM com o do sistema Rembrandt, que foi o primeiro colocado na avaliação conjunta do segundo HAREM para a tarefa de reconhecer nomes de pessoas e organizações no corpus *CD2*.

A fim de possibilitar a comparação, o ICC-HMM foi aplicado ao corpus *CD2* e foram registradas as métricas de avaliação associadas às EM dos tipos *pessoa* e *organização* anotadas.

7.6.4 – Terceira etapa

A terceira etapa dos experimentos se caracterizou pela utilização do ICC-HMM para o reconhecimento das entidades dos tipos *pessoa* e *organização* no corpus forense.

Para fins de comparação, o mesmo experimento foi realizado com a utilização do sistema Rembrandt[27] na sua versão 1.3 beta. O Rembrandt possui um arquivo de configuração que define os parâmetros sob os quais o sistema é executado. Para o presente experimento, foram utilizados os parâmetros apresentados no Anexo D.

7.7 – RESULTADOS E DISCUSSÃO

Os resultados dos experimentos realizados são apresentados e discutidos baseados nestas três etapas de experimentos.

7.7.1 – Primeira etapa

O primeiro resultado obtido dos experimentos é referente ao comparativo das variações do ICC-HMM, conforme pode ser observado nas Tabelas 7.3 (tarefa de identificação de EM) e 7.4 (tarefa de classificação de EM). A Tabela 7.3 mostra que o ICC-HMM apresentou

[27] O código fonte do Rembrandt foi obtido no sítio web de endereço *http://xldb.di.fc.ul.pt/Rembrandt/*.

melhor capacidade de identificação de EM no corpus *CD2* em relação às variações do seu modelo.

Tabela 7.3 - Resultados da avaliação do ICC-HMM e variações - tarefa de identificação

Variação do Modelo	Classe da EM	#EM Corpus	#EM Identificadas Corretas dentre EM do Corpus	#EM Sistema	#EM Identificadas Corretas dentre EM do Sistema	*P*	*R*	*F*
ICC-HMM	PES	2036	1339,25	2008	1590,09	0,792	0,658	0,719
	ORG	961	657,07	2064	1126,07	0,545	0,684	0,607
Modelo Único	PES	2036	1313,63	1648	1249,37	0,758	0,645	0,697
	ORG	961	617,28	1344	741,21	0,551	0,642	0,593
Sem Gazetteer	PES	2036	1339,25	1274	988,06	0,775	0,658	0,712
	ORG	961	657,07	3061	1442,57	0,471	0,684	0,558
Sem Contexto	PES	2036	1339,25	1967	1411,93	0,718	0,658	0,687
	ORG	961	657,07	2171	1021,79	0,471	0,684	0,558

A medida da precisão (*P*), quando aplicada à tarefa de identificação de uma classe de EM, representa o percentual de EM que o sistema identificou corretamente dentre todas as EM classificadas como *pessoa* ou *organização* pelo próprio sistema, enquanto que a medida da revocação (*R*) revela o percentual de EM identificadas corretamente dentre as EM do tipo *pessoa* ou *organização* existentes no corpus. Esta avaliação, em especial a revocação, está diretamente relacionada ao resultado gerado pelo modelo de identificação do ICC-HMM, cuja única função é identificar se determinado *token* é ou não EM. A superação em cerca de 2% do ICC-HMM em relação à variação *Modelo Único*, apresentada na Tabela 7.3, sugere que a utilização do modelo de identificação em um sistema de REM pode contribuir para a melhora da acurácia na tarefa de identificar EM.

Além disso, a Tabela 7.3 mostra que as variações *Sem Gazetteer* e *Sem Contexto* implicam na piora da precisão do modelo proposto em 7,4% para a EM *organização* e entre 1,7% e 7,4% para a EM *pessoa*. Observa-se ainda que as medidas de revocação dessas variações são idênticas à medida do ICC-HMM. Isso se justifica pelo fato dessas variações utilizarem o mesmo modelo de identificação do ICC-HMM, assim a tarefa de identificar se um *token* é EM ou não gera os mesmos resultados para todos eles.

A Tabela 7.4 apresenta os resultados associados à tarefa de classificação de EM.

Tabela 7.4 - Resultados da avaliação do ICC-HMM e variações - tarefa de classificação

Variação do Modelo	Velocidade (tokens/s.)	Classe da EM	#EM Corpus	#EM Sistema	#EM Corretas	*P*	*R*	*F*
ICC-HMM	10	PES	2036	2008	1394,31	0,694	0,685	0,690
		ORG	961	2064	676,25	0,328	0,704	0,447
Modelo Único	7	PES	2036	1648	1105,61	0,671	0,543	0,600
		ORG	961	1344	521,75	0,388	0,543	0,453
Sem Gazetteer	17	PES	2036	1274	921,04	0,723	0,452	0,556
		ORG	961	3061	664,15	0,217	0,691	0,330
Sem Contexto	17	PES	2036	1967	1333,09	0,678	0,655	0,666
		ORG	961	2171	631,76	0,291	0,657	0,403

Esses resultados mostram que, em geral, o ICC-HMM também supera o desempenho dos demais modelos quando a tarefa é classificar as EM. A única exceção é a medida-F obtida pelo *modelo único* na tarefa de classificação da EM *organização*, que superou o ICC-HMM em 0,6%. Apesar dessa diferença entre os valores da medida-F ser insignificante do ponto de vista prático, ela chama a atenção para o valor da medida de precisão que compõe a medida-F, que é superior à do ICC-HMM em 6%. Essa diferença pode ter ocorrido devido a erros de identificação do ICC-HMM, entretanto tais erros representam o risco que o modelo assume por priorizar a revocação em relação à precisão. Nesse caso, por exemplo, o ICC-HMM reconheceu mais de 70% dos nomes de organizações no texto, enquanto que o modelo único reconheceu 54,3% desses nomes.

A Tabela 7.4 mostra também que a utilização do *gazetteer* contribui consideravelmente para melhorar a acurácia na tarefa de classificação, uma vez que a diferença da medida-F com e sem esse recurso é superior a 13% para a EM *pessoa* e 11% para a EM *organização*. A ausência do *gazetteer* revelou um comportamento inesperado do modelo, uma vez que os resultados foram caracterizados por alta precisão e baixa revocação na classificação da EM *pessoa* e o oposto ocorreu com a EM *organização*. Isso pode representar insuficiência de exemplos de treinamento que, como se pode observar nos resultados apresentados, foi compensada com a utilização do *gazetteer* no modelo principal, o ICC-HMM.

A informação sobre o contexto do *token* corrente também se mostrou um importante recurso para obter melhora da acurácia do sistema. A Tabela 7.4 mostra que houve piora em relação ao ICC-HMM de cerca de 2% e 4% na medida-F para as EM *pessoa* e *organização*, respectivamente, o que justifica a utilização deste recurso no ICC-HMM.

A velocidade de etiquetação do modelo, cujos valores são exibidos na segunda coluna da Tabela 7.4, representa o número médio de *tokens* etiquetados por segundo durante a execução do algoritmo de REM, considerando as tarefas de identificação e classificação. Os seus resultados mostram que a utilização do *gazetteer* e do contexto têm impacto relevante no fator tempo, representando diminuição da velocidade em mais de 40%.

Os resultados apresentados na Tabela 7.4 e ilustrados no gráfico de dispersão da medida-F da Figura 7.8 mostram que os modelos tiveram maior dificuldade para reconhecer nomes de organizações do que de pessoas, enquanto a visualização gráfica dos resultados apresentada na Figura 7.9 evidencia que esta dificuldade está diretamente relacionada à capacidade de precisão dos modelos, que em 100% dos casos foram inferiores às medidas de revocação, fato esse que não ocorreu com a EM *pessoa*.

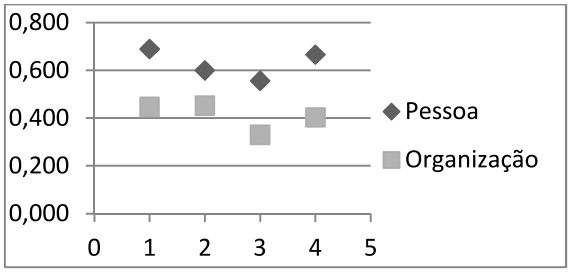


Figura 7.8 - Dispersão dos valores da medida-F por EM, obtidas nas avaliações apresentadas na Tabela 7.4 (os números do eixo horizontal representam os modelos: 1=ICC-HMM; 2=Modelo Único; 3=Sem Gazetteer e 4=Sem Contexto)

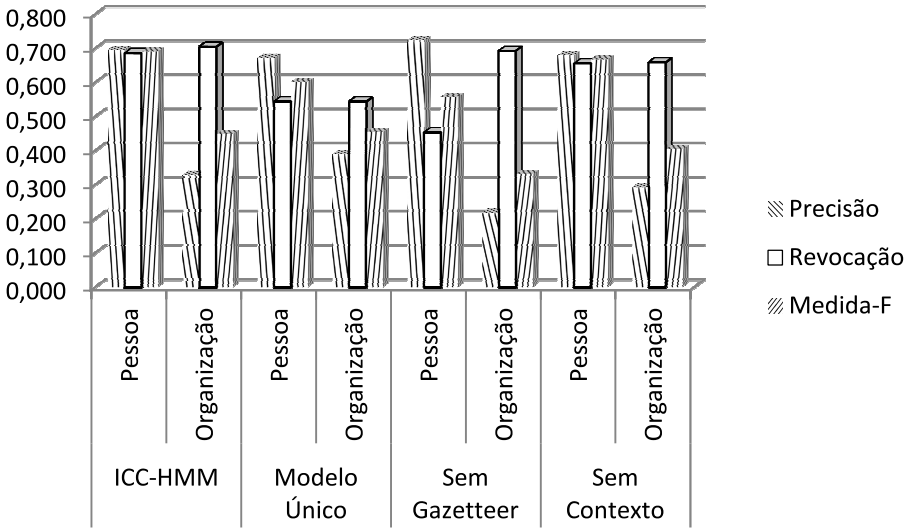


Figura 7.9 - Visualização gráfica dos resultados associados à tarefa de classificação de EM apresentados na Tabela 7.4

Para melhorar a precisão de um modelo de REM, têm-se duas opções: reduzir a quantidade de EM etiquetadas ou aumentar a quantidade de acertos na etiquetação. Entretanto, em um modelo de REM considerado estável[28], essas grandezas são diretamente proporcionais, o que significa que a redução da quantidade de EM etiquetadas provoca a redução de acertos, que, por sua vez, implica em uma piora da medida de revocação.

Em se tratando do REM no domínio forense, quando não é possível obter um equilíbrio entre as medidas de precisão e revocação, opta-se pela revocação maior que a precisão. Isso se justifica pelo fato de, durante uma investigação policial, qualquer informação ser potencialmente relevante até que se prove o contrário, ou seja, para a equipe de investigação é preferível o reconhecimento de uma lista de nomes errados à omissão de uma lista de nomes corretos. Conforme apresentado na Tabela 7.4 e na Figura 7.9, o ICC-HMM apresentou medidas de precisão e revocação equilibradas para a EM *pessoa* e medida de revocação superior à de precisão em mais de 100% para a EM *organização*.

7.7.2 – Segunda etapa

A segunda etapa dos experimentos objetivou comparar o ICC-HMM e o sistema Rembrandt quanto à tarefa de REM aplicada ao corpus *CD2*.

A Tabela 7.5 apresenta os resultados obtidos pelo Rembrandt na avaliação conjunta do segundo HAREM para a tarefa de reconhecer nomes de pessoas e organizações. Esses resultados estão registrados em (Mota et al., 2008) e (Cardoso, 2008). A Tabela 7.7 reapresenta, para fins de comparação, os resultados obtidos pelo ICC-HMM aplicados ao mesmo corpus avaliado pelo Rembrandt, a *CD2*.

Tabela 7.5 - Resultados obtidos pelo sistema Rembrandt no segundo HAREM (corpus avaliado: *CD2*)

Classe de EM	*P*	*R*	*F*
PES	0,768	0,537	0,632
ORG	0,535	0,323	0,403

[28] No presente trabalho, considera-se "modelo estável" o modelo de aprendizado que possui os parâmetros de treinamento e todos os cálculos de probabilidade do modelo formal definidos.

Tabela 7.6 - Resultados obtidos pelo ICC-HMM (corpus avaliado: *CD2*)

Classe de EM	*P*	*R*	*F*
PES	0,694	0,685	0,690
ORG	0,328	0,704	0,447

Os valores obtidos para a medida-F no ICC-HMM superam os obtidos pelo Rembrandt em cerca de 10%. Conforme apresentado no gráfico da Figura 7.10, observa-se que essa superação é reflexo da grande diferença existente nas medidas de revocação entre os dois sistemas, na qual o ICC-HMM é superior em 14,8% para a EM *pessoa* e em 38,1% para a EM *organização*. Em contrapartida, as medidas de precisão do Rembrandt são superiores às do ICC-HMM em 7,4% para a EM *pessoa* e 20,7% para a EM *organização*.

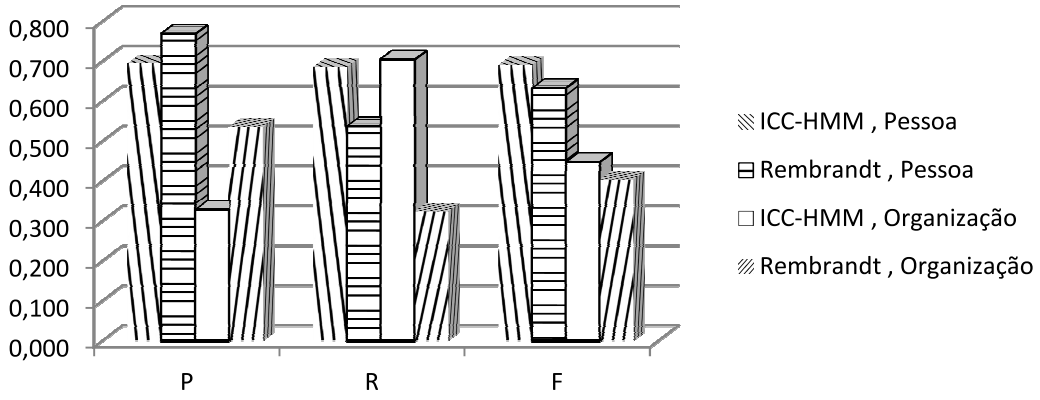


Figura 7.10 - Medidas *P*, *R* e *F* obtidas pelo ICC-HMM e pelo Rembrandt

Conforme já mencionado no presente trabalho, para um sistema de REM aplicado ao domínio forense, no caso de não se conseguir obter o equilíbrio entre *P* e *R*, é preferível que se tenha *R* superior a *P*, requisito este que foi atendido pelo ICC-HMM de acordo com os resultados dos experimentos aplicados ao corpus *CD2*.

7.7.3 – Terceira etapa

A terceira etapa dos experimentos objetivou avaliar a aplicação do algoritmo do ICC-HMM ao corpus forense (vide Seção 7.3.3) e comparar os seus resultados com aqueles obtidos pelo sistema Rembrandt aplicado ao mesmo corpus. Esses resultados são apresentados na Tabela 7.7.

Tabela 7.7 - Resultados do ICC-HMM e do Rembrandt aplicados ao corpus forense

Sistema	Velocidade (tokens/seg.)	Classe da EM	#EM Corpus	#EM Sistema	#EM Corretas	*P*	*R*	*F*
ICC-HMM	5	PES	95	104	58,72	0,565	0,618	0,590
		ORG	105	158	51,10	0,323	0,487	0,389
Rembrandt	59	PES	95	76	43,10	0,567	0,454	0,504
		ORG	105	34	21,90	0,644	0,209	0,315

Em linhas gerais, esta comparação se manteve similar à realizada com o corpus *CD2*, ou seja, o ICC-HMM obteve as melhores medidas–F (diferença entre 7,4% e 8,6%) e de revocação (diferença entre 16,4% e 27,8%), e o Rembrandt as melhores medidas de precisão (diferença entre 0,2% e 32,1%). Além disso, para os dois sistemas os melhores desempenhos mantiveram associados à EM *pessoa*, quando comparados com a EM *organização*.

A Tabela 7.7 mostra que, enquanto o Rembrandt reconheceu somente 21% dos nomes de organizações presentes no corpus, o ICC-HMM chegou a quase 50% de reconhecimento, o que é mais eficiente do ponto de vista prático policial, mesmo tendo esse resultado ocasionado a alta média de dois erros de classificação (falso-positivos) a cada três EM classificadas, uma vez que *P*=0,323. O mesmo comportamento foi observado em relação à EM *pessoa*, para a qual o ICC-HMM reconheceu 62% dos nomes presentes do corpus, enquanto que o Rembrandt reconheceu 49%.

A análise das EM contidas no corpus forense mostra que a baixa acurácia associada à EM *organização* está em parte relacionada à ocorrência de nomes desconhecidos e não iniciados em letra maiúscula nos textos forenses, uma situação que pôde ser resolvida com a utilização do contexto do ICC-HMM em alguns casos, entretanto, na maior parte das vezes, as informações do contexto presentes no corpus não sugeriram qualquer relação com a EM organização. Por exemplo, a sentença "A *sincraft mandou 10 cópias do material.*"[29] contém a EM "*sincraft*" da categoria organização, entretanto este nome é desconhecido dos exemplos de treinamento e do *gazetteer* utilizados no ICC-HMM, além de ser iniciado em letra minúscula e não haver informações de contexto dentro da sentença que associem o nome a uma organização. A solução para esse tipo de problema pode ser o aumento da quantidade de exemplos de treinamento, a expansão da janela de contexto ou a utilização de informações sobre EM previamente etiquetadas com alta probabilidade de

[29] O nome da EM presente nesta sentença foi alterado, por motivo de sigilo policial.

acerto no mesmo texto. Entretanto, problemas como a carência de textos etiquetados para REM na língua portuguesa e impactos causados no desempenho computacional do modelo podem tornar essas soluções inviáveis do ponto de vista prático.

Dentre os 32 documentos de texto contidos no corpus forense, todos tiveram ao menos uma EM do tipo *pessoa* ou *organização* reconhecida corretamente pelo ICC-HMM, enquanto que o Rembrandt não reconheceu nenhum nome em dois desses documentos. Esse resultado favorece a aplicabilidade do ICC-HMM à tarefa de utilização do REM como filtro de arquivos analisados manualmente em investigações policiais, conforme proposto em (Dalben e Claro, 2011). O objetivo dessa tarefa é, durante a etapa de análise do material apreendido pela polícia, priorizar a análise manual dos arquivos que contêm nomes de pessoas ou organizações, a fim de se evitar esforço desnecessário com a análise de arquivos irrelevantes. Em uma situação hipotética, se essa tarefa fosse aplicada ao corpus forense avaliado no presente experimento, o ICC-HMM selecionaria todos os arquivos corretamente.

Chama a atenção o fato da velocidade do sistema Rembrandt ser mais de dez vezes superior à do ICC-HMM. O presente trabalho não objetivou analisar o algoritmo desenvolvido quanto ao seu desempenho computacional, por isso as causas desta discrepância entre as velocidades não foram avaliadas, devendo esta tarefa ser realizada em trabalhos futuros.

7.8 – APLICAÇÃO FORENSE

Ao final dos experimentos, o ICC-HMM gerou uma lista contendo todos os nomes de pessoas e organizações reconhecidos no corpus, com as seguintes informações:

- Nome reconhecido;
- Categoria de EM;
- Arquivo que contém o nome reconhecido;
- Identificação da mídia que contém o arquivo;
- Identificação da operação policial que resultou na apreensão da mídia;
- Outros arquivos/mídias onde o mesmo nome foi reconhecido.

A geração dessa lista tem o objetivo de prover informações à equipe de investigação acerca dos nomes contidos nas mídias apreendidas, em momento prévio à análise manual dessas mídias, como forma de revelar nomes desconhecidos e possibilitar a priorização da análise manual dos arquivos que contêm EM.

8 - CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta as principais conclusões do presente trabalho, bem como os direcionamentos futuros.

Um dos problemas hoje enfrentados pelas forças policiais brasileiras é a carência de métodos e ferramentas que as auxiliem no processo de análise de mídias computacionais apreendidas em grande quantidade. Como consequência, acaba-se por realizar a análise manual dessas mídias sem que se tenha qualquer informação prévia sobre o conteúdo das mesmas. O conhecimento acerca dos nomes das pessoas e organizações contidos nas mídias apreendidas, em momento prévio à sua análise manual, pode contribuir para a revelação de nomes latentes desconhecidos, bem como pode servir como um filtro para a redução de mais de 90% da quantidade de arquivos analisados manualmente (Dalben e Claro, 2011).

Assim, o presente trabalho propôs o desenvolvimento de um modelo de Reconhecimento de Entidades Mencionadas (REM) baseado no Modelo Oculto de Markov (HMM) para reconhecer nomes de pessoas e organizações contidos em textos forenses. O algoritmo desenvolvido baseado no modelo de REM proposto, denominado ICC-HMM (*Identification-Classification Context HMM*), é uma variação do HMM, cuja principal característica é a divisão do modelo principal em dois submodelos, um para identificar e outro para classificar as EM encontradas no texto, sendo que o submodelo de classificação usa informações extraídas do contexto da palavra corrente como forma de melhorar a acurácia. Além dessa característica, o ICC-HMM usa um conjunto de *features* de palavras e um *gazetteer* como recursos adicionais do modelo que objetivam facilitar o reconhecimento de nomes de pessoas e organizações.

Experimentos foram realizados utilizando dois corpora mantidos pelo projeto Linguateca (Santos et al., 2004), um para treinar e outro para avaliar o modelo, e um corpus forense extraído de arquivos contidos em mídias apreendidas pela Polícia Federal do Brasil, e os resultados foram avaliados em três etapas, a seguir descritas:

1. **REM no corpus do Linguateca e comparação dos resultados com três variações do modelo ICC-HMM**: os testes mostraram que as três variações

resultam na redução da acurácia do ICC-HMM; as variações foram: (i) ICC-HMM sem modelo de identificação, (ii) ICC-HMM sem *gazetteer* e (iii) ICC-HMM sem informações de contexto;

2. **REM no corpus do Linguateca e comparação dos resultados com o sistema Rembrandt** (Cardoso, 2008)[30]: O ICC-HMM foi superior nas medidas de revocação e medida-F, enquanto que o Rembrandt foi superior nas medidas de precisão;
3. **REM no corpus forense e comparação dos resultados com o sistema Rembrandt**: o ICC-HMM foi superior nas medidas de revocação e medida-F; o Rembrandt foi superior nas medidas de precisão.

A partir da análise dos resultados obtidos, conclui-se que um sistema de REM de aprendizado de máquina usando modelo bipartido (identificação-classificação) pode alcançar acurácia superior à dos melhores sistemas de REM para a língua portuguesa, em especial quando não se tem informação prévia acerca do domínio textual e quando as medidas de revocação são mais importantes que as de precisão.

Além disso, pode-se concluir que *gazetteers* e informações do contexto das palavras são importantes recursos usados para melhorar a acurácia de modelos de REM.

Por fim, o presente trabalho mostrou que o modelo do ICC-HMM é aplicável ao cenário forense, para o qual pode oferecer as seguintes contribuições:

- Geração de lista de arquivos que contêm pelo menos um nome de pessoa ou organização, com o objetivo de funcionar como um filtro dos arquivos a serem analisados manualmente e assim evitar esforços desnecessários com a análise manual de arquivos irrelevantes, conforme apresentado em (Dalben e Claro, 2011);
- Geração de listas contendo todos os nomes de pessoas e organizações reconhecidos em mídias apreendidas, em momento prévio à análise manual dos arquivos contidos nessas mídias, com o objetivo de enriquecer a base de conhecimento da equipe de investigação e também direcionar as atividades de análise das mídias apreendidas.

Portanto, o objetivo traçado pelo presente trabalho foi alcançado, uma vez que foi criado um modelo probabilístico baseado no HMM capaz de superar os resultados obtidos por

[30] O sistema Rembrandt obteve os melhores resultados na avaliação conjunta do segundo HAREM (Carvalho et al., 2008) na tarefa de reconhecer nomes de pessoas e organizações.

sistemas de REM na tarefa de reconhecer nomes de pessoas e organizações na língua portuguesa, foram registradas as suficiências e deficiências observadas durante os experimentos e, por fim, foi criado um protótipo representativo do modelo que serve como linha de base para a implementação de uma aplicação de REM no cenário forense.

8.1 - LIMITAÇÕES

As principais limitações associadas à solução proposta foram:

- A aplicação do ICC-HMM é restrita à etiquetação de textos não estruturados, assim os resultados obtidos com a etiquetação de textos estruturados tendem a ser piores que os apresentados no presente trabalho;
- Somente foi avaliada a acurácia do modelo proposto para a tarefa de reconhecer nomes de pessoas e organizações, portanto nada se pode afirmar sobre o desempenho do modelo aplicado ao reconhecimento de outras categorias de EM.

8.2 – TRABALHOS FUTUROS

Dado o fato do REM aplicado ao cenário forense ser um tópico ainda pouco explorado academicamente, alguns trabalhos são sugeridos como complemento às pesquisas que originaram o ICC-HMM. As propostas são as seguintes:

- Realização de um estudo comparativo com outros modelos de REM probabilísticos, como o CRF (Lafferty et al., 2001), MEMM (McCallum et al., 2000) e SVM (Vapnik, 1998), aplicado ao cenário forense;
- Criação de um modelo de relacionamento entre EM, com o objetivo de identificar a existência de vínculos entre diferentes documentos textuais;
- Análise do desempenho computacional do ICC-HMM, tratando de temas como a otimização da estrutura de dados utilizada no modelo e a viabilidade de utilização do processamento distribuído para a execução da tarefa de REM;
- Estudo de viabilidade de modificação do ICC-HMM de modo a:
    - Considerar todas as palavras da sentença corrente como informações de contexto;
    - Fazer uso de informações associadas a sentenças etiquetadas previamente no texto como forma agregar contexto às sentenças ainda não etiquetadas;
    - Incluir o uso de sinônimos no modelo como forma melhorar a sua acurácia.

REFERÊNCIAS BIBLIOGRÁFICAS

ACE Group, 2000. *Entity Detection and Tracking - Phase 1 (ACE Pilot Study task Definition),* s.l.: s.n.

Amaral, C., Figueira, H., Mendes, A., Mendes, P., Pinto, C. e Veiga, T., 2008. Adaptação do sistema de reconhecimento de entidades mencionadas da Priberam ao HAREM. In:: *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* s.l.:Cristina Mota e Diana Santos, pp. 171-179.

Barreto, F., Branco, A., Ferreira, E., Mendes, A., Nascimento, M.F., Nunes, F. e Silva, J., 2006. Open resources and tools for the shallow processing of Portuguese: The TagShare project. *Proceedings of LREC2006*.

Baum, L. e Petrie, T., 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, Volume 37.

Bikel, D. M., Schwartz, R. e Weischedel, R. M., 1999. An Algorithm that Learns What‘s in a Name. *Machine Learning - Special issue on natural language learning,* 34(1-3), pp. 211-231.

Cardoso, N., 2008. REMBRANDT - Reconhecimento de Entidades Mencionadas Baseado em Relações e ANálise Detalhada do Texto. In:: *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* s.l.:Linguateca, pp. 195-211.

Carvalho, P., Oliveira, H.G., Mota, C., Santos, D. e Freitas, C., 2008. Segundo HAREM: Modelo geral, novidades e avaliação. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca, 2008,* pp. 11-31.

Chang, C.H., Kayed, M., Girgis, M. R. e Shaalan, K., 2006. A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering,* 18(10), pp. 1411-1428.

Chinchor, N. A., 1998. *Overview of MUC-7/MET-2.* San Diego, s.n.

Chiticariu, L., Krishnamurthy, R., Li, Y., Reiss, F. e Vaithyanathan, S., 2010. Domain Adaptation of Rule-based Annotators for Named-Entity Recognition Tasks. *EMNLP*.

Cohen, W. W. e Sarawagi, S., 2004. Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods. *KDD '04*

Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

Computational Linguistics & Psycholinguistics Research Center, 2009. *Thirteenth Conference on Computational Natural Language Learning,* Boulder, USA: s.n.

Dalben, O. J. e Claro, D. B., 2011. Uma Análise do Reconhecimento Textual de Nomes de Pessoas e Organizações na Computação Forense. *Proceeding of the Sixth International Conference on Forensic Computer Science – ICoFCS 2011*, pp. 7 - 15.

Edward, K., Baryamureeba, V. e Pauw, G., 2008. Towards Domain Independent Named Entity Recognition. *International Journal of Computing and ICT Research*, Volume 2, pp. 84-95.

Eleutério, P. e Machado, M., 2011. *Desvendando a Computação Forense.* 1 ed. São Paulo, BRA: Novatec.

Feldman, R. e Sanger, J., 2007. *The text mining handbook: advanced approaches analyzing advanced unstructured data.* New York: CAMBRIDGE UNIVERSITY PRESS.

Florian, R., Ittycheriah, A., Jing, H. e Zhang, T., 2003. Named entity recognition through classifier combination. *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, Volume 4.

Gospodnetic, O. e Hatcher, E., 2004. *Lucene in Action (In Action series).* s.l.:s.n.

Grishman, R. e Sundheim, B., 1996. Message understanding conference - 6: A brief history. *Proc. International Conference on Computational Linguistics*.

Guidance Software, 2011. *Guidance Encase Forensic.* [Online] Available at: http://www.guidancesoftware.com/forensic.htm [Acesso em 10 11 2011].

Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X. e Su, Z., 2009. Domain adaptation with latent semantic association for named entity recognition. *NAACL '09 Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Kazama, J. e Torisawa, K., 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. *EMNLP-CoNLL*.

Kinoshita, J., Salvador, L. N. e Menezes, C. E. D., 2007. CoGrOO - An OpenOffice Grammar Checker. *ISDA '07 Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*.

Lafferty, J., McCallum, A. e Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*.

Levinson, S. E., Rabiner, L. e Sondhi, M., 1983. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal,* 62(4), pp. 1035-1074.

Louis, A. e Engelbrecht, A., 2011. Unsupervised discovery of relations for analysis of textual data. *Digital Investigation*, Volume 7, pp. 154-171.

Mayfield, J., McNamee, P. e Piatko, C., 2003. Named entity recognition using hundreds of thousands of features. *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, Volume 4.

McCallum, A., Freitag, D. e Pereira, F., 2000. Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the 7th International Conference on Machine Learning (ICML 2000)*, pp. 591-598.

McDonald, D. D., 1996. Internal and external evidence in the identification and semantic categorization of proper names. *Corpus processing for lexical acquisition ,* pp. 32-43.

Merchant, R., Okurowski, M. e Chinchor, N., 1996. The Multilingual Entity Task Overview. *Proceedings of the Tipster Text Program Phase II*, pp. 445-447.

Milidiú, R. L., Duarte, J. C. e Cavalcante, R., 2007. Machine Learning Algorithms for Portuguese Named Entity Recognition. *Revista Iberoamericana de Inteligencia Artificial,* 11(36), pp. 67-75.

Miller, S., Guinness, J. e Zamanian, A., 2004. Name Tagging with Word Clusters and Discriminative Training. *Proceedings of HLT-NAACL*, pp. 337-342.

Mota, C., Oliveira, H.G., Santos, D., Carvalho, P. e Freitas, C., 2008. Resumo de resultados do Segundo HAREM. In:: *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* s.l.:s.n., pp. 379-403 (Apêndice I).

Mota, C. e Santos, D., 2008. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* s.l.:Linguateca.

Mota, C., Santos, D., Carvalho, P., Freitas, C. e Oliveira, H.G., 2008a. Apresentação detalhada das colecções do Segundo HAREM. In:: *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.* s.l.:s.n., pp. 355-377.

Nadeau, d. e Sekine, s., 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, Volume 30, pp. 3-26.

Ng, A. Y. e Jordan, M. I., 2002. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. *Neural Information Processing Systems - NIPS*, Volume 2, pp. 841-848.

Pasca, M., Lin, D., Bigham, J., Lifchits, A e Jain, A., 2006. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, Volume 2.

Rabiner, L. R., 1990. A tutorial on hidden Markov models and selected applications in speech recognition. In:: *Readings in speech recognition* . San Francisco: Morgan Kaufmann Publishers Inc..

Ratinov, L. e Roth, D., 2009. Design challenges and misconceptions in named entity recognition. *CoNLL '09 Proceedings of the Thirteenth Conference on Computational Natural Language Learning*.

Rau, L. F., 1991. Extracting Company Names from Text.. *Conference on Artificial Intelligence Applications of IEEE*.

Rijsbergen, C. v., 1979. Information Retrieval. *Journal of the American Society for Information Science,* 30(2), pp. 374-375.

Riloff, e. e Jones, r., 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pp. 474-479.

Rocha, P. e Santos, D., 2007. Disponibilizando a <OBRA>Colecção Dourada</OBRA> do <ACONTECIMENTO>HAREM</ACONTECIMENTO> através do projecto <LOCAL|ORGANIZACAO|ABSTRACCAO>AC/DC</LOCAL|ORGANIZACAO|ABSTRACCAO>. *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área, 2007*, pp. 307-326.

Russell, S. J. e Norvig, P., 2010. *Artificial intelligence: a modern aproach.* 3 ed. s.l.:Prentice Hall.

Sang, E. F. T. K. e Meulder, F. D., 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, Volume 4.

Santos, D., 2000. O projecto Processamento Computacional do Português: Balanço e perspectivas. *Actas do V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR'2000)*, pp. 105-113.

Santos, D., Cardoso, N. e Seco, N., 2007. Avaliação no HAREM: métodos e medidas. In:: *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área.* s.l.:s.n., pp. 245-282.

Santos, D., Seco, N., Cardoso, N. e Vilela, R., 2006. HAREM: An Advanced NER Evaluation Contest for Portuguese. *Proceedings of LREC 2006 (LREC'2006)*, pp. 1986-1991.

Santos, D., Simões, A., Frankenberg-Garcia, A., Pinto, A, Barreiro, A., Maia, B., Mota, C., Oliveira, D., Bick, E., Ranchhod, E., Almeida, J.J.D., Cabral, L., Costa, L., Sarmento, L., Chaves, M., Cardoso, N., Rocha, P., Aires, R., Ilva, R., Vilela, R. e Afonso, S., 2004. Linguateca: um centro de recursos distribuído para o processamento computacional da língua portuguesa. *Proceedings of the international workshop "Taller de Herramientas y Recursos Linguísticos para el Espanõl y el Portugués", IX Iberoamerican Conference on Artificial Intelligence (IBERAMIA 2004)*, pp. 147-154.

Sarawagi, S., 2008. Information Extraction. *Foundations and Trends in Databases,* March, 1(3), pp. 261-377.

Sarmento, L., 2005. *Relatório Técnico sobre o REPENTINO,* Porto: s.n.

Sarmento, L., 2006. SIEMÊS – a Named-Entity Recognizer for Portuguese Relying on Similarity Rules. *PROPOR 2006 - Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada*.

Sarmento, L., Pinto, A. S. e Cabral, L., 2006. REPENTINO – A Wide-Scope Gazetteer for Entity Recognition in Portuguese. *PROPOR 2006*, pp. 31-40.

Shinyama, Y. e Sekine, S., 2004. Named Entity Discovery Using Comparable News Articles. *Proceedings of the 20th international conference on Computational (COLING '04)*.

Todorovic, B. T., Rancic, S.R., Markovic, Ivica M., Mulalic, E.H. e Ilic, V.M., 2008. Named Entity Recognition and Classification using Context Hidden Markov Model. *9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2008*.

Toral, A. e Muñoz, R., 2006. A Proposal To Automatically Build And Maintain Gazetteers For Named Entity Recognition By Using Wikipedia. *Workshop On New Text Wikis And Blogs And Other Dynamic Text Sources*.

Vapnik, V., 1998. *Statistical Learning Theory.* New York, NY: s.n.

Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, Volume 13, pp. 260-269.

Weiss, S. M., Indurkhya, N., Zhang, T. e Damerau, F. J., 2005. *Text Mining: predictive methods for analyzing unstructured information.* New York: Springer Science+Business Media Inc..

Yakhenko, O., Lita, L. V., Rosales, R. e Niculescu, S., 2007. Principled Generative-Discriminative Hybrid Hidden Markov Model. *NIPS - Workshop on Representations and Inference on Probability Distributions* .

Zhang, T. e Johnson, D., 2003. A robust risk minimization based named entity recognition system. *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, Volume 4.

Zhou, G. e Su, J., 2002. Named entity recognition using an HMM-based chunk tagger. *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

ANEXOS

A – LISTA DAS ABREVIAÇÕES USADAS NO ICC-HMM

A Tabela A.1 apresenta a lista das abreviações usadas na implementação do ICC-HMM

Tabela A.1 - Lista das abreviações usadas no ICC-HMM

"s.a.", "ltda.", "ltd.", "sr.", "sra.", "sr.ª" , "srª.", "dr.", "dra.", "mr.","mrs.", "s.m.j.", "cia.", "atm.", "qui.", "adm.", "fls.", "q.e.d.", "cap.", "ten.", "maj.", "cel.", "max.", "máx.", "mm.", "ton.", "srs.", "dpto.", "depto.", "pgto.", "tab.", "fig.", "rev.", "sto.", "sta.", "med.", "seg.", "ter.", "qua.", "qui.", "sex.", "sab.", "sáb.", "dom.", "jan.", "fev.", "mar.", "abr.", "mai.", "jun.", "jul.", "ago.", "set.", "out.", "nov.", "dez.", "att.", "doc.", "inc.", "talmai.", "freq.", "art.", "ref.", "arts.", "pág.", "pag.", "cont.", "adv.", "adj.", "min.", "mín.", "prof.", "profa.", "obs.", "profª.", "vol.", "org.", "eng.", "pg.", "jr.", "bel.", "v.ex.a.", "v.ex.a", "sr.a", "sr.ª", "v.ex.ª", "v.em.a", "v.em.ª", "v.rev.ma", "v.rev.mª", "vv.pp.", "v.mag.", "v.mag.a", "v.mag.ª", "vv.mm."

B – LISTA DAS CONTRAÇÕES TRATADAS NO ICC-HMM

A Tabela B.1 apresenta a lista das contrações tratadas na implementação do ICC-HMM

Tabela B.1 - Lista das contrações tratadas no ICC-HMM

De + o(s) – do(s)	Em + a(s) – na(s)
De + a(s) – da(s)	Em + um – num
De + um – dum	Em + uma – numa
De + uns – duns	Em + uns – nuns
De + uma – duma	Em + umas – numas
De + umas – dumas	A + à(s) – à(s)
De + ele(s) – dele(s)	Por + o – pelo(s)
De + ela(s) – dela(s)	Por + a – pela(s)
De + este(s) – deste(s)	De + outro – doutro(s)
De + esta(s) – desta(s)	De + outra – doutra(s)
De + esse(s) – desse(s)	Em + este(s) – neste(s)
De + essa(s) – dessa(s)	Em + esta(s) – nesta(s)
De + aquele(s) – daquele(s)	Em + esse(s) – nesse(s)
De + aquela(s) – daquela(s)	Em + aquele(s) – naquele(s)
De + isto – disto	Em + aquela(s) – naquela(s)
De + isso – disso	Em + isto – nisto
De + aquilo – daquilo	Em + isso – nisso
De + aqui – daqui	Em + aquilo – naquilo
De + aí – daí	A + aquele(s) – àquele(s)
De + ali – dali	A + aquela(s) – àquela(s)
Em + o(s) – no(s)	A + aquilo – àquilo

C – LISTA DAS *STOPWORDS* CONSIDERADAS NO ICC-HMM

A Tabela C.1 apresenta a lista das *stopwords* consideradas na implementação do ICC-HMM

Tabela C.1 - Lista das *stopwords* consideradas no ICC-HMM

"a","ainda","alem","ambas","ambos","antes",
"ao","aonde","aos","apos","aquele","aqueles",
"as","assim","com","como","contra","contudo",
"cuja","cujas","cujo","cujos","da","das","de",
"dela","dele","deles","demais","depois","desde",
"desta","deste","dispoe","dispoem","diversa",
"diversas","diversos","do","dos","durante","e",
"ela","elas","ele","eles","em","entao","entre",
"essa","essas","esse","esses","esta","estas",
"este","estes","ha","isso","isto","logo","mais",
"mas","mediante","menos","mesma","mesmas","mesmo",
"mesmos","na","nas","nao","nas","nem","nesse","neste",
"nos","o","os","ou","outra","outras","outro","outros",
"pelas","pelas","pelo","pelos","perante","pois","por",
"porque","portanto","proprio","proprios","quais","qual",
"qualquer","quando","quanto","que","quem","quer","se",
"seja","sem","sendo","seu","seus","sob","sobre","sua",
"suas","tal","tambem","teu","teus","toda","todas","todo",
"todos","tua","tuas","tudo","um","uma","umas","uns"

D – VALORES DE PARÂMETROS DO SISTEMA REMBRANDT

A Tabela D.1 apresenta os valores atribuídos aos parâmetros do sistema Rembrandt quando da sua utilização no experimentos do presente trabalho.

Tabela D.1 - Valores atribuídos aos parâmetros do sistema Rembrandt

Parâmetro	**Valor**
global.lang	pt
rembrandt.core.rules	HAREM
rembrandt.core.doEntityRelation	false
rembrandt.core.removeRemainingUnknownNE	true
rembrandt.core.removeTextualNumbers	true
rembrandt.input.encoding	ISO-8859-1
rembrandt.input.file	[*caminho_e_nome_do_corpus_avaliado*]
rembrandt.output.encoding	ISO-8859-1
rembrandt.output.file	[*caminho_e_nome_do_arquivo_ etiquetado_gerado*]
rembrandt.cache.usedocumentindex	true
rembrandt.cache.term_clause.enable	true
saskia.dbpedia.enabled	true
saskia.dbpedia.version	3.5.1
saskia.dbpedia.mode	local
saskia.dbpedia.local.fileformat	N-TRIPLE
saskia.wikipedia.enabled	true