
# RECONHECIMENTO E NORMALIZAÇÃO DE EXPRESSÕES TEMPORAIS EM PORTUGUÊS

Heraldo José Araújo Carneiro Filho

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Setembro de 2011

RECONHECIMENTO E NORMALIZAÇÃO
DE EXPRESSÕES TEMPORAIS EM PORTUGUÊS

Heraldo José Araújo Carneiro Filho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____

Prof. Geraldo Bonorino Xexéo, D.Sc.

_____

Prof. Jano Moreira de Souza, Ph.D.

_____

Prof. Sean Wolfgand Matsui Siqueira, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 2011

Carneiro Filho, Heraldo José Araújo

Reconhecimeto e Normalização de Expressões Temporais em Português/ Heraldo José Araújo Carneiro Filho. – Rio de Janeiro: UFRJ/COPPE, 2011.

XVI, 145 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2011.

Referências Bibliográficas: p. 128-145.

1. Extração de informação. 2. Reconhecimento de entidades mencionadas. 3. Reconhecimento de expressões temporais. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

DEDICATÓRIA

Aos meus pais, que me deram todo o suporte emocional (e financeiro) que precisei nesta jornada.

# Agradecimentos

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro, sem o qual seria difícil a conclusão deste mestrado em uma cidade tão distante de Fortaleza, minha terra natal.

Ao prof. Xexéo, meu orientador, por ter me aceitado como orientando e por ter compartilhado comigo seu conhecimento e experiência de maneira irrestrita, sempre com a combinação ideal de inteligência, criatividade e bom humor. Também por ter confiado na minha capacidade de executar este trabalho e por ter me permitido tratar de um tema que não está diretamente ligado às suas principais linhas de pesquisa. Obrigado pela confiança e pela paciência nos momentos mais delicados.

Ao prof. Jano e ao prof. Sean por fazerem parte da banca.

Aos meus pais, Heraldo e Zenilda, por terem apoiado as minhas escolhas, mesmo quando o bom senso dizia o contrário, e por nunca terem medido esforços para que eu e minha irmã tivéssemos e valorizássemos uma educação de qualidade. Espero sempre corresponder às suas expectativas e ser um orgulho para vocês.

À minha irmã, em breve Dra. Paula, por sempre encontrar um tempo, em meio à correria de hospitais e da vida acadêmica da medicina, para me ligar e me lembrar de que *tenho irmã*. A melhor delas.

À minha família como um todo, mas, em especial, ao núcleo paterno, aqui simbolizado na figura da minha querida tia Ana Beatriz, pelo amor, carinho e apoio incondicional em todos os momentos. Também à vovó Zenilda, por ainda fazer questão de, sempre que volto à Fortaleza, me receber com tanta alegria, mesmo estando com a saúde um pouco debilitada.

Ao meu padrasto, Walter, e, mais recentemente, à minha madrasta, Edésia, por, de *padrasto* e *madrasta*, não terem nada: são outro par de pais.

À minha namorada, Louise, por ter contribuído de forma significativa para que eu sempre buscasse meus sonhos e crescesse profissionalmente e como pessoa. Obrigado por ser minha inspiração na superação dos meus limites.

A todos os meus amigos cearenses e piauienses do Rio, Yuri, Samuel, Renan (é catarinense, mas entra nesta lista mesmo assim), Fabrício, Vinícius e, sobretudo, Hélio e Olivério, por estarem sempre presentes em quase todas as etapas da minha vida na

Cidade Maravilhosa. Obrigado por me permitirem contar com vocês. Saibam que sempre podem contar comigo, onde quer que estejamos.

Aos meus amigos da UFRJ, especialmente, Rodrigo, Carlos Eduardo e Moisés, com quem compartilhei boa parte dos altos e baixos desta jornada, especialmente nos primeiros anos do mestrado.

Aos meus amigos da UFC, notadamente, Carlos Eduardo, Bruno, Silveira, Ronan e Ivan, com quem compartilhei vitórias e derrotas presencialmente nos quatro anos da graduação e virtualmente, nos do mestrado. Ao amigo e prof. Flávio e ao prof. Javam, por terem me iniciado nos caminhos da academia.

Aos meus amigos do Colégio 7 de Setembro, em especial, Carol, Mila, Raina, Saw, Tati, Victor, Felipe e Hugo. Mais de uma dezena de anos de amizade que, apesar de distante, se mantém presente.

Aos meus colegas de trabalho e amigos da Ancine e do Banco Central, por tornarem o horário comercial uma parte ainda mais divertida e prazerosa da minha vida. Aos chefes que tive durante esses anos, Roberto, Helmut, Miriam e Mario, por entenderem e aceitarem a minha ausência sempre que precisei ir à UFRJ no meio do expediente. Ao chefe que virou grande amigo, Pedro Jorge.

E, finalmente, a todos aqueles que, direta ou indiretamente, me apoiaram durante esse período e contribuíram na elaboração deste trabalho. Obrigado a todos que, de uma forma ou de outra, deixaram algo em mim.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

# RECONHECIMENTO E NORMALIZAÇÃO DE EXPRESSÕES TEMPORAIS EM PORTUGUÊS

Heraldo José Araújo Carneiro Filho

Setembro/2011

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Este trabalho faz uma revisão das áreas de extração de informação, reconhecimento de entidades mencionadas e reconhecimento de expressões temporais para propor um padrão de anotação temporal, acompanhado de uma sugestão de arquitetura, para a língua portuguesa. A partir disso, implementamos o CoppeTER – Coppe Temporal Expression Recognizer, um sistema híbrido, baseado em uma gramática de regras desenvolvidas manualmente e em aprendizado de máquina (mais especificamente, etiquetadores gramaticais e classificadores de máxima entropia), para reconhecimento e normalização de expressões temporais em português. A abordagem é validada em dois momentos, sendo o primeiro através de uma série de experimentos que avaliam a precisão dos classificadores estatísticos empregados. A validação do desempenho do sistema como um todo é levada a cabo em cima do arcabouço de avaliação do Segundo HAREM. Diante do bom desempenho do CoppeTER na tarefa, comparado com as abordagens existentes, os resultados atestam que ainda há bastante espaço para avançar o atual estado da arte em processamento temporal no idioma.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

RECOGNITION AND NORMALIZATION
OF TEMPORAL EXPRESSIONS IN PORTUGUESE

Heraldo José Araújo Carneiro Filho

September/2011

Advisor: Geraldo Bonorino Xexéo

Department: Systems Engineering and Computer Science

This work surveys the fields of information extraction, named entity recognition and temporal expression recognition in order to develop a temporal annotation scheme, along with suggested system architecture, for the Portuguese language. Taking that into account, we implement CoppeTER – Coppe Temporal Expression Recognizer, a hybrid system, based on a grammar of manually developed rules and machine learning (specifically, part-of-speech taggers and maximum entropy classifiers), for Portuguese temporal expression recognition and normalization.. The approach is evaluated in two steps, the first being though a series of experiments that measure the accuracy of the statistical classifiers employed in the system. The end-to-end system evaluation is carried on top of the evaluation framework used in Segundo HAREM. With the good performance achieved by the CoppeTER system in mind, as compared with existing approaches, the results attest that there is significant room for improvement in the current state of the art for temporal processing in Portuguese.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Lista de Abreviaturas e Siglas

ACE – Automatic Content Extraction

BI – Business Intelligence

BNF – Backus-Naur Form

DARPA – Defense Advanced Research Projects Agency

DCT – Document Creation Time

EI – Extração de Informação

EM – Entidade Mencionada

ET – Expressão (ou Entidade) Temporal

HAREM – Avaliação de Reconhecimento de Entidades Mencionadas

HMM – Hidden Markov Model

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

ISO – International Organization for Standardization

MUC – Message Understanding Conference

NER – Named Entity Recognition

NIST – National Institute of Standards and Technology

NLTK – Natural Language Toolkit

PLN – Processamento de Linguagem Natural

POS – Part-of-Speech

REM – Reconhecimento de Entidades Mencionadas

RET – Reconhecimento de Expressões (ou Entidades) Temporais

RI – Recuperação de Informação

SGML – Standard Generalized Markup Language

SVM – Support Vector Machine

TDT – Topic Detection and Tracking

XML – Extensible Markup Language

XSLT – Extensible Stylesheet Language Transformations

# 1. Introdução

Esta dissertação descreve, juntamente com uma proposta de diretivas de anotação temporal, o CoppeTER, uma abordagem para o reconhecimento e a normalização de expressões temporais em documentos, em linguagem natural, escritos em português. Neste capítulo, serão apresentados, além da motivação para o desenvolvimento deste trabalho, os objetivos e as contribuições que almejamos alcançar. Ao final do capítulo, descrevemos a organização e a estrutura do restante desta dissertação.

## 1.1. Motivação

Apesar do conceito de sobrecarga de informação preceder o advento da internet, foi a popularização dela e o crescimento exponencial da quantidade de informações livremente disponíveis na World Wide Web (Web) que tornaram essa concepção uma realidade inegável para muitos de nós. Tendo sido a computação o veículo do problema, não surpreendentemente, é ela também que procura apresentar a maior parte das soluções, mesmo que paliativas. Nesse sentido, a área de recuperação de informação (RI) tem sido um dos ramos da computação mais focados nessa tarefa (BERGHEL, 1997). Uma área correlata, que também tem papel decisivo no tratamento dessa questão, é área de extração de informação (EI), que busca capturar informações relevantes dentro de documentos e transformá-las em conteúdo estruturado e, idealmente, processável automaticamente por computadores (SARAWAGI, 2008).

Entre as formas em que a EI se dá, uma das mais populares e mais importantes é o reconhecimento de entidades mencionadas (REM) (SANTOS *et al.*, 2007), cujo propósito é identificar e extrair, do texto, unidades de informação que representem pessoas, locais, organizações, valores numéricos, datas, entre outros. Essa subárea da EI é particularmente importante para sistemas de resposta automática a perguntas (do inglês, *question-aswering*), uma vez que, geralmente, a resposta esperada para uma pergunta (especialmente aquelas que começam por *quem*, *quando*, *quanto*, *onde* e *o quê*) é uma entidade desse tipo.

Dentre os diversos tipos de entidades mencionadas (EMs), um dos conjuntos mais importantes é o das chamadas entidades (ou expressões) temporais (ETs), que engloba as expressões que carregam consigo um componente temporal implícito ou

explícito e que transmitem ideia de tempo, duração ou frequência, por exemplo: “na quinta-feira”, “11 de setembro de 2001”, “na década de 80”, “duas semanas”, “três dias depois”, “diariamente”, entre uma infinidade de outras.

Para realmente compreender uma informação, é necessário entender sua estrutura temporal (AHN *et al.*, 2007), uma vez que ela é um componente essencial da narrativa (BITTAR, 2009). Tendo isso como objetivo final, o reconhecimento de expressões temporais (RET) dá um primeiro passo nesse sentido e busca identificar essas entidades em todo o conteúdo dos documentos para, então, classificá-las, interpretá-las e normalizá-las, podendo, ainda, anotá-las no próprio texto para posterior processamento por outros sistemas de *software*. Os sistemas que definem o estado da arte dessa área são capazes também de relacionar essas ETs entre si e com eventos no texto, através da linguagem de marcação temporal TimeML, por exemplo.

A tarefa de reconhecimento e normalização de expressões temporais, portanto, consiste em identificar, em textos, expressões que se refiram a pontos ou períodos no tempo e transformá-las em referências temporais em formato padrão, isto é, que possam ser processadas por computador. Em última análise, o objetivo dessa tarefa é possibilitar a ordenação dos eventos em um texto e a compreensão da dimensão temporal da linguagem, conforme dissemos, parte essencial do processo mais amplo e difícil de entendimento da linguagem natural.

A tarefa de normalização, em particular, é interessante e desafiante, pois, enquanto algumas referências temporais aparecem, no texto, com formatos bem definidos, outras são expressas através de um sem-número de construções de linguagem natural, além de, por vezes, serem ambíguas, demandando a análise das palavras ou sentenças adjacentes para permitir uma interpretação única e precisa.

Essas tarefas, apesar de iniciais no processo complexo de compreensão textual, não são triviais. Nós, como humanos, somos quase sempre capazes de nos localizar no tempo precisamente. Assim, quando queremos nos referir a um período ou dia específico, geralmente, falamos coisas como “no dia 16 de outubro”, “na terça”, “próxima semana” ou “daqui a dois dias”, expressões que são instantaneamente compreendidas por nós, mas que, para computadores, exigem uma cadeia de processamento temporal, por vezes, complexa. Até textos jornalísticos, que costumam ser razoavelmente formais, usam essas expressões dependentes de contexto no corpo

das notícias. Para todas elas, é preciso entender o contexto temporal para determinar que data ou período está sendo referenciado exatamente (FERRO *et al.*, 2005).

A complexidade no processamento de expressões temporais não se resume apenas à ambiguidade inerente aos tipos de ETs mais usadas e à necessidade de contexto para interpretá-las de maneira precisa. Não é incomum encontrar documentos em que a ordem dos eventos no texto não representa a ordem cronológica em que eles ocorrem. Notícias, por exemplo, costumam mencionar os eventos mais importantes primeiro para, somente depois, descrever os menos importantes que aconteceram antes deles (BITTAR, 2009).

Além disso, e como em muitos outros aspectos da linguagem natural, verifica-se um determinado grau de vagueza na interpretação de muitas expressões temporais (BAPTISTA *et al.*, 2008), por exemplo: a expressão "há dois anos" deve ser interpretada como se referindo ao intervalo de tempo entre 1º de Janeiro e 31 de Dezembro de 2009 ou a uma data exata nesse ano relativa ao momento da enunciação (hoje)? Questões como essa demandam uma análise do texto não apenas nos níveis morfológico e sintático, mas também semântico, o que implica que abordagens puramente algorítmicas tendem a não ser tão bem-sucedidas.

O conjunto de elementos lexicais envolvidos na composição de expressões temporais em linguagem natural é relativamente extenso, porém é, ainda assim, suficientemente limitado para que se conceba como possível a meta de se atingir uma cobertura próxima da exaustividade. Essa possibilidade de cobertura quase completa, apesar de ajudar, está longe de garantir que o sistema será capaz interpretar e normalizar corretamente as entidades temporais identificadas. A despeito dessas dificuldades, a pesquisa em torno do processamento de informação temporal em linguagem natural tem recebido atenção crescente da comunidade acadêmica, especialmente das áreas de RI e linguística computacional, nos últimos anos (MANI *et al.*, 2005). Desde as *Message Understanding Conferences* (MUC) e, posteriormente, da série de avaliações conjuntas anuais *Automatic Content Extraction* (ACE), a área de RET (majoritariamente em língua inglesa) tem percebido crescimento constante, com diversas abordagens desenvolvidas e excelentes resultados reportados.

Esse recente ressurgimento de interesse dos pesquisadores na área não é à toa. O tempo é uma dimensão importante de qualquer espaço informacional e pode ser muito útil, por exemplo, para RI. Atualmente, os sistemas de RI não tiram proveito de toda a

informação temporal disponível no conteúdo dos documentos para melhorar seus resultados, e, consequentemente, a experiência do usuário. Uma rápida verificação em qualquer dos motores de busca e sistemas de RI mais usados da Web atesta que o prisma temporal desses mecanismos se restringe à mera ordenação de resultados por data. Com a velocidade com que a quantidade de informação gerada aumenta no mundo digital, o conceito de tempo como uma a dimensão através da qual as informações podem ser organizadas e exploradas se estabelece como muito importante (ALONSO *et al.*, 2007).

Além da RI, há outras áreas da computação que podem se beneficiar diretamente da pesquisa em RET, entre elas as áreas de inteligência artificial, mineração de dados, visualização de dados e processamento de linguagem natural. Adicionalmente, além de ser essencial para sistemas de resposta automática a perguntas, a capacidade de reconhecer expressões temporais no texto pode ser um forte diferencial para sistemas de tradução automática, sumarização e detecção e acompanhamento de tópicos (MAKKONEN *et al.*, 2003a, SETZER, 2001).

Os primeiros trabalhos na área – que se limitavam à simples identificação das entidades temporais no texto, sem qualquer tentativa de classificação, interpretação ou normalização – alcançaram resultados promissores com abordagens baseadas em aprendizado de máquina. No entanto, depois que se passou a incluir a normalização (que, obviamente, pressupõe a classificação e a interpretação) entre as metas do RET, os novos sistemas gradualmente passaram a ser implementados em grande parte através de transdutores de estado finito ou etiquetadores baseados em regras, ainda que muitos deles adotassem uma abordagem híbrida, com uso de métodos estatísticos na composição da sua arquitetura. Essas abordagens parecem ser as mais efetivas quando o objetivo inclui as duas tarefas. De fato, o conjunto relativamente limitado de palavras que se combinam para formar expressões temporais sugere que sistemas baseados em regras podem ser implementados satisfatoriamente com esforço razoavelmente pequeno e, ainda assim, prover bons resultados (CASELLI *et al.*, 2009).

Embora a área de reconhecimento e normalização de expressões temporais em linguagem natural seja muito bem desenvolvida para a língua inglesa – e alguns outros idiomas como francês, alemão, espanhol e chinês –, este ainda não é o caso para o português, onde o RET ainda se encontra em estágio incipiente. As avaliações conjuntas das duas edições do HAREM (Avaliação de Reconhecimento de Entidades

Mencionadas), de fato, deram os primeiros passos nessa direção, criando uma pista específica para ETs no evento, no contexto da qual foi construída uma proposta de anotação padrão e desenvolvidas ferramentas de avaliação. Os resultados alcançados pelos sistemas considerados como o estado da arte para o processamento temporal em língua portuguesa, contudo, estão bem aquém dos observados em tarefas similares em outros idiomas. À exceção de um, os trabalhos apresentados na pista TEMPO do Segundo HAREM (a última edição do evento, em 2008) tratam apenas do problema da identificação de expressões temporais e, ainda assim, superficialmente. Além disso, a proposta de anotação preconizada pelos organizadores não se aproximou o suficiente dos padrões e das linhas de investigação estabelecidos internacionalmente (HAGÈGE *et al.*, 2008b).

Face ao exposto, apresentamos, nesta dissertação, o CoppeTER – Coppe Temporal Expression Recognizer, uma abordagem híbrida, que combina um sistema de regras e aprendizado de máquina, para identificar, extrair, classificar, normalizar e anotar expressões temporais em português, buscando aproximar os padrões adotados e resultados alcançados ao estado da arte da área.

## 1.2. Objetivos e contribuições

O processamento temporal textual tem como objetivo reconhecer e normalizar as expressões temporais presentes no texto para, depois, associá-las aos eventos e estados de coisas que modificam, de modo a possibilitar uma ordenação, ainda que parcial, segundo uma sequência cronológica. Naturalmente, essa meta é demasiada ambiciosa, dado o estágio embrionário em que se encontra o RET em português. Pretendemos, portanto, nesta dissertação, dar apenas alguns passos nessa direção, apresentando uma abordagem para a incontornável tarefa de reconhecimento e normalização de ETs.

Sendo assim, este trabalho, depois de uma extensa análise da área, propõe um esquema de anotação temporal para o português, mais próximo do padrão adotado internacionalmente (TIMEX2/TimeML), mas sem se distanciar tanto das diretivas desenvolvidas para o Segundo HAREM. Tal esquema é implementado no que batizamos de CoppeTER – Coppe Temporal Expression Recognizer, um sistema híbrido para o reconhecimento e a normalização de expressões temporais em português, que combina regras e aprendizado de máquina para alcançar precisão em par com o estado da arte para a língua inglesa.

Entre as contribuições, podemos citar, ainda, a composição de um corpus de mais três mil notícias, a partir do qual derivamos dois conjuntos de dados para treinamento e avaliação de classificadores de desambiguação de expressões temporais relativas.

## 1.3. Estrutura da dissertação

Esta dissertação está organizada em sete capítulos, sendo o primeiro deles esta introdução, que apresentou o contexto e a motivação do trabalho, bem como os objetivos almejados e as contribuições deixadas. No segundo capítulo, ampliamos essa contextualização, apresentando uma revisão da literatura das áreas extração de informação, reconhecimento de entidades mencionadas e reconhecimento de expressões temporais. O capítulo 3 finaliza essa revisão descrevendo os métodos e ferramentas utilizados: padrões de anotação temporal, etiquetadores *part-of-speech* (POS) e classificadores estatísticos.

O quarto capítulo traz o detalhamento da proposta de um esquema de anotação que aproxima os padrões TIMEX2 e do Segundo HAREM e define o que o sistema considera como expressão temporal e como ele procura anotá-las. Ao final desse capítulo, apresentamos uma visão geral da arquitetura do CoppeTER, descrevendo como os módulos interagem entre si. Com base nessa arquitetura, o capítulo 5 expõe a implementação de cada um dos módulos do sistema.

Por fim, o sexto capítulo reporta os procedimentos utilizados para avaliar este trabalho, apresenta os resultados alcançados, compara os valores encontrados com os de outros sistemas e faz uma análise crítica do CoppeTER e da própria avaliação. O capítulo 7, o último desta dissertação, apresenta as conclusões, as considerações finais e as sugestões e possibilidades de trabalhos futuros.

# 2. Expressões temporais

Este capítulo faz uma revisão das áreas relevantes e descreve o problema atacado nesta dissertação. A seção 2.1 apresenta a disciplina de extração de informação, a grande área na qual está inserida a tarefa de reconhecimento de entidades mencionadas. Essa tarefa é detalhada na seção 2.2. Finalmente, a seção 2.3 apresenta a área de reconhecimento de expressões temporais e contextualiza o problema tratado pela proposta deste trabalho.

## 2.1. Extração de informação

O objetivo da pesquisa em extração de informação (EI) é construir sistemas que encontrem e conectem informações relevantes ao passo em que ignoram informações irrelevantes ou não essenciais (COWIE *et al.*, 1996). Essa área de pesquisa é desafiante e tem ganhado cada vez mais importância com a popularização da internet e o crescimento, sem precedentes, da quantidade de informações livremente disponíveis na World Wide Web (Web). Com essa explosão de conteúdo, em sua maior parte textual e em linguagem natural, disponível na Web, a EI tem um papel expressivo na organização dessa enorme massa de dados para processamento computacional. Assim, tendo essa abundância de informação potencialmente útil ao seu dispor, um sistema de EI é capaz de transformar os dados brutos em informação tratada, coerente e computacionalmente processável.

Para atingir seu objetivo, a área de EI atua na identificação e extração automática de entidades, seus relacionamentos e seus atributos descritores a partir de fontes de dados não estruturadas (em particular, texto em linguagem natural) e na transformação desses dados em uma representação estruturada (SARAWAGI, 2008). Esse processo pode consistir na seleção de estruturas ou na combinação de dados encontrados, explicita ou implicitamente, em um ou mais documentos. Um exemplo da entrada e saída de um sistema de EI poderia ser o seguinte:

*"O show de Paul McCartney, confirmado na última quarta-feira (6), acontecerá no dia 22 de maio, no Estádio Olímpico João Havelange, o "Engenhão", localizado no bairro de Engenho de Dentro, Zona Norte carioca."*

| | |
|---|---|
| **Evento** | Show |
| **Artista** | Paul McCartney |
| **Data** | 22/05/2011 |
| **Local** | Estádio Olímpico João Havelange |
| **Bairro** | Engenho de Dentro |
| **Cidade** | Rio de Janeiro, RJ, Brasil |

**Figura 2.1. Exemplo de entrada e saída de um sistema de EI**

Essa representação estruturada de dados previamente não estruturados tem como objetivo permitir o processamento de informações computacionalmente, para, por exemplo, inseri-las em um banco de dados ou possibilitar inferências lógicas a partir do conteúdo de um texto.

Com a infinidade de assuntos sobre os quais há dados e textos disponíveis na Web atualmente, as possibilidades de aplicação de EI são evidentemente ilimitadas. Na literatura, já encontramos aplicações nas mais diversas áreas, entre elas: *business intelligence* (BI), análise de sentimento, busca de currículos, busca de patentes, busca de e-mails, acompanhamento de notícias, limpeza de dados, classificados, bancos de citações, bancos de opiniões, etc. Aplicações como comparadores de preços em lojas on-line e de criação automática de portais impulsionaram a pesquisa acadêmica e o uso comercial das tecnologias desenvolvidas na área. Atualmente, uma subárea de pesquisa particularmente importante tenta extrair dados estruturados de literatura científica disponível eletronicamente, especialmente nos domínios da biologia e da medicina (BIRD *et al.*, 2009), reconhecendo, por exemplo, proteínas, genes e interações entre eles em artigos dessas áreas de conhecimento.

### 2.1.1. Origem

A EI surgiu dentro da comunidade de processamento de linguagem natural (PLN), a partir de competições com o objetivo de reconhecer dados, como nomes de pessoas e organizações – as chamadas entidades mencionadas (em inglês, *named entities*), sobre as quais falaremos na próxima seção –, em artigos e notícias (SARAWAGI, 2008). Há, portanto, um forte relacionamento entre EI e PLN.

As *Message Understanding Conferences* (MUCs), a série de competições supracitadas, aconteceram entre 1987 e 1998, tendo sido iniciadas e financiadas pela DARPA (*Defense Advanced Research Projects Agency*) para encorajar o desenvolvimento de novos e melhores métodos para EI. A primeira e a segunda MUC, chamada de MUCK1 em 1987 e MUCK2 em 1989, usavam um pequeno número de

mensagens navais. Nas duas edições seguintes, chamadas de MUC-3 e MUC-4, o gênero dos textos foi alterado, passando a ser usadas notícias sobre incidentes terroristas em países latino-americanos (COWIE *et al.*, 1996).

Ao longo das suas sete edições, as MUCs foram responsáveis pela base da metodologia e tecnologia da área de EI (HOBBS *et al.*, 2010), sendo sua influência sentida até hoje. Essa significância é tão grande que alguns pesquisadores da área classificam as abordagens de EI em duas classes, as abordagens MUC e as pós-MUC (CHANG *et al.*, 2006).

Com o fim da era das MUCs, surgiram as avaliações *Automatic Content Extraction* (ACE), do *National Institute of Standards and Technology* (NIST), que começaram em 1999 com um estudo piloto e tiveram edições praticamente anuais até 2008. O objetivo do programa ACE era desenvolver a tecnologia de PLN para apoiar a compreensão automática e a extração de significado da linguagem natural, inferindo as entidades mencionadas, as relações entre essas entidades e os eventos dos quais essas entidades participam (DODDINGTON *et al.*, 2004).

No geral, o objetivo do programa ACE é motivado pelas mesmas questões das MUCs que o precederam. A ACE, entretanto, difere significativamente ao procurar identificar os objetos alvo (isto é, as entidades, as relações e os eventos) de fato em vez de simplesmente se preocupar com as palavras que os identificam no texto, como era feito nas MUCs. Isso é uma tarefa mais abstrata e substancialmente distinta, uma vez que necessita explicitamente de capacidade de inferência para produzir respostas corretas.

Por vezes, também é difícil separar a EI da, mais abrangente e mais madura, área de recuperação de informação (RI). É importante frisar, contudo, que as duas têm objetivos distintos: esta tem como meta a seleção de um subconjunto de documentos relevantes a partir de um conjunto maior, em geral, enxergando o texto como um simples saco de palavras (em inglês, *bag of words*), enquanto aquela procura extrair informação levando em conta a estrutura do texto dos documentos. Em poucas palavras, RI extrai os documentos relevantes a partir de coleções, enquanto EI extrai informações relevantes a partir documentos. Há, porém, uma grande sinergia entre as duas áreas. Em particular, ao extrair informações relevantes de maneira estruturada, a EI pode ser muito importante para tornar o problema da sobrecarga de informação, um dos grandes desafios da área de RI atualmente, mais tratável.

Além dessas, a pesquisa em EI tem ainda interseções com as comunidades de aprendizado de máquina, bancos de dados, linguística computacional, web, análise de documentos, entre outras.

### 2.1.2. Tipos de extração

Os tipos de objetos extraídos de textos através da EI variam em complexidade: há casos de conjuntos fechados (como, por exemplo, a identificação de unidades federativas -- "Rio de Janeiro", "Ceará", etc.), conjuntos regulares (ex. números de telefone – "8396 4608", "21 3637.7354", "+55 (21) 2466-2469", etc.), padrões complexos (como endereços) e padrões ambíguos (por exemplo, nomes de pessoas, que, na maioria das vezes, precisam de contexto e mais de uma evidência no texto para serem classificados como tais).

Esses objetos também variam em aridade: tendo como exemplo a frase abaixo, podemos capturar entidades singulares (nomes de pessoas, como "Murillo Ferreira" e "Roger Agnelli", e locais, como "Rio de Janeiro", por exemplo), relacionamentos binários (relação pessoa-cargo, pessoa: Roger Agnelli, cargo: CEO, relação empresa-local, empresa: Vale, local: Rio de Janeiro) ou registros n-ários (relação: sucessão, empresa: Vale, cargo: presidente, entra: Murillo Ferreira, sai: Roger Agnelli).

*"O executivo Murillo Ferreira foi indicado para a presidência da Vale. Roger Agnelli, atual presidente da companhia, em evento na sede da empresa no Rio de Janeiro, não quis comentar o comunicado divulgado ao mercado."*

Entre os tipos de estruturas extraídas, temos: entidades, relacionamentos entre entidades, adjetivos descrevendo entidades e estruturas complexas, como tabelas (PINTO *et al.*, 2003) e listas (COHEN *et al.*, 2002, EMBLEY *et al*,. 1999). Entidades são tipicamente formadas por sintagmas nominais, sendo as mais comuns as entidades mencionadas (que definiremos melhor mais adiante). Relacionamentos se dão entre duas ou mais entidades relacionadas de maneira pré-determinada. Um problema similar é o do *semantic role labeling*, em que se busca encontrar os diferentes argumentos semânticos de um predicado ("O menino chutou a bola ao gol."; quem chutou: o menino; o que foi chutado: a bola; para onde: ao gol).

Os tipos de extração ainda podem ser classificados a partir de outras dimensões, como a granularidade da extração (registro, sentença, parágrafo, documento, coleção de

documentos) e a heterogeneidade da fonte (documentos estruturados, páginas geradas automaticamente, documentos parcialmente estruturados, documentos livres).

### 2.1.3. Tarefas

A pesquisa em EI começou com avaliações conjuntas que tinham como objetivo o preenchimento de *slots* pré-definidos, tarefa que consistia, basicamente, em preencher, automaticamente, informações em campos pré-definidos de um formulário a partir de um texto: por exemplo, para cada texto descrevendo um atentado terrorista, identificar o local, o número de feridos, as armas usadas, etc. (os tais campos pré-determinados).

Com o passar dos anos, essas avaliações separaram os objetivos em cinco grandes tarefas principais: *template element* (TE), *template relation* (TR), *scenario template* (ST), *named entities* (NE) e *co-reference* (CO). As três primeiras foram as tarefas que permearam as primeiras competições de EI, em particular, no contexto das MUCs. Elas tinham como meta a extração de informações pré-determinadas, em domínios específicos, sobre eventos e entidades. A tarefa TE trata da extração de atributos relacionados às entidades a partir de informações de qualquer parte do texto. A tarefa TR registra os relacionamentos entre essas entidades, enquanto a tarefa ST define o cenário no contexto do qual as entidades e relacionamentos extraídos formarão uma representação estruturada, originada do texto livre (o supracitado preenchimento de *slots*).

A tarefa NE (as já citadas entidades mencionadas) tem como objetivo marcar no texto nomes de pessoas, organizações, lugares, expressões temporais, valores monetários, quantidades numéricas e afins. Como dito anteriormente, o foco deste trabalho está na anotação (incluindo a normalização) de expressões temporais (um tipo de entidade mencionada, portanto) em português e, por isso, essa tarefa será analisada mais detalhadamente na próxima seção.

Por fim, a tarefa CO determina que sejam identificadas as informações relativas a expressões co-referenciadas, isto é, todas as menções de cada entidade (independente de variações de forma ou do uso de pronomes). Essa tarefa tem interseção substancial com o problema de resolução de anáforas em PLN (SAQUETE *et al.*, 2002).

Essa lista de tarefas não é exaustiva, uma vez que a definição e o escopo exatos das subáreas de EI não são consensuais e muitas conferências, competições e avaliações

conjuntas combinam subtarefas de EI de formas distintas de acordo com seu objetivo, domínio e visão da área.

### 2.1.4. Abordagens

Os primeiros sistemas de EI eram baseados em regras codificadas manualmente, mas, rapidamente, foram aplicadas técnicas de aprendizado automático a partir de exemplos. Sistemas codificados manualmente demandam especialistas humanos para definir regras, expressões regulares ou programas para realizar a extração. Sistemas baseados em aprendizado requerem exemplos etiquetados manualmente para treinar modelos de extração. Apesar da recente predominância de métodos de aprendizado estatístico, não há uma melhor abordagem. Os dois métodos são bastante usados e dependem do tipo de tarefa que se esteja executando. Há ainda os métodos híbridos, que têm sido aplicados com sucesso em alguns domínios.

As abordagens para EI também podem ser divididas em dois grandes tipos. Na abordagem de engenharia do conhecimento, as gramáticas são construídas a mão, e os padrões do domínio descobertos por especialistas humanos através de introspecção e inspeção de corpora, em um processo bastante trabalhoso, meticuloso e demorado. Esses métodos manuais, em geral, são mais fáceis de construir, depurar e manter, porém, demandam muito tempo e esforço humano para serem suficientemente abrangentes, mesmo para domínios limitados. Porém, com habilidade e experiência, sistemas com bons resultados não são conceitualmente difíceis de desenvolver. De fato, em geral, os melhores sistemas têm sido os desenvolvidos dessa forma, em particular em tarefas ST. As maiores desvantagens dessa abordagem são o forte acoplamento ao domínio e a necessidade de especialistas com expertise não somente linguística, mas também no domínio. A maioria desses sistemas utiliza expressões regulares, gramáticas ou regras construídas manualmente. Alguns exemplos desses sistemas são FASTUS (APPELT *et al.*, 1993), DBLife (DOAN *et al.*, 2006) e Avatar (JAYRAM *et al.*, 2006).

Na abordagem de sistemas de treinamento automático, são usados métodos estatísticos, sempre que possível, para aprender regras a partir de corpora anotados ou a partir de interação com o usuário. Métodos baseados em aprendizado são fáceis de implementar quando há dados de treinamento disponíveis em quantidade suficiente ou quando esses dados são passíveis de geração automática. Tais métodos são muitas vezes capazes de reconhecer padrões sutis ou difíceis de identificar e codificar manualmente.

O esforço nesse tipo de abordagem está concentrado na construção do corpus de treinamento. A portabilidade desses sistemas para outros domínios e línguas é mais fácil, uma vez que dados de treinamento estejam disponíveis em quantidade suficiente. A inexistência, alto custo ou necessidade de grande volume de dados de treinamento podem dificultar ou impossibilitar o uso dessa abordagem. Entre as técnicas utilizadas para implementação, temos classificadores (generativos, como Bayes ingênuo, e discriminativos, como modelos de máxima entropia) e modelos de sequência, como modelos ocultos de Markov (*hidden Markov models* ou HMMs) (LEEK, 1997) e campos aleatórios condicionais (*conditional random fields* ou CRFs) (LAFFERTY, 2000). Alguns exemplos clássicos de sistemas baseados em aprendizado de máquina para EI são SRV (FREITAG, 1998) e Rapier (CALIFF *et al.*, 1999).

Apesar de distintos, os dois métodos são complementares. O uso de uma ou de outra abordagem (ou de uma híbrida) vai depender da disponibilidade de recursos e especialistas, nível de resultados esperados e disponibilidade de dados de treinamento.

Não obstante não estar relacionado ao domínio do problema desta dissertação, cabe notar que as abordagens tradicionais de EI não atendem bem ao conteúdo da Web, uma vez que não fazem proveito da formatação, e, em alguns casos, da semiestruturação, provida pelas *tags* HTML. Em função disso, sistemas de EI para a web tendem a usar *wrappers* (KUSHMERICK, 1997), abordagens menos linguísticas que fazem uso da árvore de elementos das páginas HTML para extrair conteúdo através de regras. Esses *wrappers* podem ser gerados manualmente (o que requer bastante tempo e *expertise*) ou automaticamente (através de técnicas de aprendizado de máquina, supervisionado ou não).

### 2.1.5. Avaliação de sistemas de EI

A maior parte da metodologia de avaliação de sistemas de EI foi desenvolvida no âmbito das MUCs. Os participantes da MUC tomaram emprestados os conceitos de precisão (*precision*) e revocação (*recall*) da área de RI para pontuar os *templates* preenchidos (LAVELLI *et al.*, 2004).

Em RI, a precisão é definida como a fração dos documentos relevantes entre os recuperados, enquanto a revocação é a fração dos documentos relevantes que foram recuperados (BAEZA-YEATES *et al.*, 1999), conforme ilustrado na Figura 2.2.

$$\text{Precisão} = \frac{\#(\text{itens recuperados relevantes})}{\#(\text{itens recuperados})} \qquad \text{Revocação} = \frac{\#(\text{itens relevantes recuperados})}{\#(\text{itens relevantes})}$$

**Figura 2.2. Fórmulas das medidas de precisão e revocação**

No contexto da MUC, dada a resposta do sistema e a resposta correta, de forma análoga à definição original de RI, a precisão do sistema era determinada pela quantidade de *slots* preenchidos corretamente pelo sistema dividido pelo número total de preenchimentos feitos pelo sistema. A revocação, por sua vez, era definida como o número de *slots* preenchidos corretamente pelo sistema dividido pelo número de preenchimentos corretos possíveis.

A idéia de usar somente a precisão (*accuracy*) pode parece plausível à primeira vista, afinal, em geral, em RI e EI, tratamos apenas com duas classes (relevante e irrelevante ou correto e incorreto). Contudo, o uso apenas dessa medida não é apropriado para problemas dessas áreas, pois, em quase todas as circunstâncias, os dados são enviesados: normalmente, mais de 99,9% dos documentos são irrelevantes em RI (MANNING *et al.*, 2008), assim como a maioria das palavras em um texto não faz parte de uma entidade que se deseje extrair. Um sistema que objetiva maximizar a precisão pode, portanto, parecer ter um bom desempenho simplesmente considerando todos os documentos como não relevantes (sistemas de RI) ou todas as palavras como não parte de alguma entidade (no caso de EI), o que seria completamente indesejável.

Ter duas medidas é particularmente interessante em situações nas quais uma pode ser mais importante que a outra: por exemplo, pessoas que utilizam buscadores na web, em geral, desejam que todos os documentos na primeira página de resultado sejam relevantes (ou seja, precisão alta) e nem sempre estão interessados em conhecer todos os documentos relevantes; em uma busca nos arquivos do disco rígido, por outro lado, o usuário quer encontrar o máximo possível de resultados relevantes (revocação alta).

Com o objetivo de se comparar o desempenho de diferentes sistemas utilizando apenas um único valor numérico, foi criada a medida F, uma média harmônica ponderada da precisão e revocação, dada pela fórmula (MANNING *et al.*, 2008): $F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$, onde P é a precisão, R é a revocação, $\alpha$ é o peso da precisão (entre 0 e 1) e $\beta^2 = \frac{1-\alpha}{\alpha}$. Dependo da representação escolhida para a medida F, valores de $\alpha > 1$ ou $\beta < 1$ enfatizam a precisão, enquanto valores de $\alpha < 1$ ou $\beta > 1$

dão mais peso à revocação. A medida F balanceada padrão, comumente escrita como $F_1$ (abreviação de $F_{\beta=1}$), dá pesos iguais para precisão e revocação, fazendo $\alpha = 1/2$ ou $\beta = 1$. Nesse caso, a fórmula acima fica simplificada: $F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R}$. Vale ressaltar que, apesar das três medidas apresentadas nesta seção terem valores entre 0 e 1, é comum encontrá-las escritas como porcentagens (e, naturalmente, numa escala entre 0 e 100).

Cabe lembrar que, apesar das medidas de precisão e revocação terem servido a comunidade de EI muito bem como duas medidas separadas de desempenho dos sistemas, há discussões sobre a adequação da medida F nas avaliações da área em função de determinados comportamentos indesejáveis (MAKHOUL *et al.*, 1999).

### 2.1.6. Estado da arte em EI

Nos últimos anos, um novo paradigma, a Extração de Informação Aberta (EIA), do inglês *Open Information Extraction* (OIE), vem recebendo bastante atenção da comunidade acadêmica. Esse paradigma, apresentado em (BANKO *et al.*, 2008), é capaz de descobrir e extrair relações em textos independentemente do domínio e sem intervenção humana, sendo particularmente interessante para a Web.

Além da EIA, um tópico que tem ganhado bastante atenção da comunidade acadêmica é EI em documentos não-textuais (como áudio, imagens, vídeo, etc.). Esses documentos multimídia apresentam a próxima fronteira para a EI. Posteriormente, a organização da informação estruturada advinda desses diferentes tipos de documentos será um desafio interessante.

## 2.2. Entidades mencionadas

O termo “entidade mencionada”, tradução usada pela comunidade de língua portuguesa para o original em inglês *named entity*, bastante difundido atualmente, foi cunhado na MUC-6 para designar unidades de informação como nomes, incluindo pessoas, organizações, locais e expressões numéricas (horas, datas, valores monetários e percentuais). A identificação dessas referências no texto é reconhecidamente uma das mais importantes subtarefas de EI, tendo sida batizada originalmente como *Named Entity Recognition and Classification* (NERC), apesar de, atualmente, ser mais

popularmente conhecida como *Named Entity Recognition* (NER) ou Reconhecimento de Entidades Mencionadas (REM) em português.

A palavra *named*, no termo *named entity*, procura restringir o escopo da tarefa a entidades para as quais existam designadores rígidos. Em metafísica, lógica, filosofia da linguagem e semântica, um designador rígido é um termo, cunhado por KRIPKE (1980), que se refere à mesma coisa em todos os mundos possíveis. Por exemplo, a instituição de pós-graduação e pesquisa em engenharia, fundada em 1963, na então Universidade do Brasil, hoje Universidade Federal do Rio de Janeiro (UFRJ), é conhecida como Coordenação (ou Coordenadoria) dos Programas de Pós-Graduação em Engenharia, Instituto Luiz Alberto Coimbra de Pós-Graduação e Pesquisa em Engenharia, COPPE/UFRJ ou simplesmente Coppe. Todas essas expressões se referem à mesma entidade. Esses designadores rígidos incluem nomes próprios e termos biológicos como nomes de espécies e substâncias. Há também um consenso na comunidade de pesquisa da área em classificar expressões temporais e alguns tipos de expressões numéricas (valores monetários, unidades de medida, etc.) como entidades mencionadas (EMs). Tendo como objeto as duas frases citadas na seção anterior, podemos exemplificar a saída de um sistema de REM (as entidades mencionadas são marcadas com etiquetas XML, num processo conhecido na área como marcação, anotação ou etiquetagem):

*"O show de* <PESSOA>*Paul McCartney*</PESSOA>*, confirmado* <DATA>*na última quarta-feira (6)*</DATA>*, acontecerá* <DATA>*no dia 22 de maio*</DATA>*, no* <LOCAL>*Estádio Olímpico João Havelange*</LOCAL>*, o* <LOCAL>*"Engenhão"*</LOCAL>*, localizado no bairro de* <LOCAL>*Engenho de Dentro*</LOCAL>*,* <LOCAL>*Zona Norte carioca*</LOCAL>*."*

*"O executivo* <PESSOA>*Murillo Ferreira*</PESSOA> *foi indicado para a presidência da* <ORGANIZACAO>*Vale*</ORGANIZACAO>*.* <PESSOA>*Roger Agnelli*</PESSOA>*, atual presidente da companhia, em evento na sede da empresa no* <LOCAL>*Rio de Janeiro*</LOCAL>*, não quis comentar o comunicado divulgado ao mercado."*

REM é um dos usos mais comuns da tecnologia de EI (HOBBS *et al.*, 2010), sendo particularmente importante para sistemas de resposta automática a perguntas (do inglês, *question-answering*), pois, comumente, a resposta esperada para uma pergunta é

uma EM, principalmente para perguntas do tipo “quem?”, “quando?”, “quanto?”, “onde?” e similares.

### 2.2.1. Origem

Um dos primeiros artigos publicados sobre REM foi (RAU, 1991), que descrevia um sistema para extrair e reconhecer nomes de companhias usando heurísticas e regras codificadas manualmente. Até 1995, não houve muita atividade na área, mas, com a MUC-6, o interesse pelo REM cresceu e não diminuiu desde então. Nela foi proposto pela primeira vez que a tarefa de REM fosse medida de uma forma independente, após ter sido considerada durante vários anos como apenas uma parte da tarefa mais geral de extrair informação de um texto (GRISHMAN *et al.*, 1996).

Após a MUC, vários outros eventos de avaliação tendo como foco o REM se seguiram, como a MET (MERCHANT *et al.*, 1996), a MET-2 (CHINCHOR, 1998), a IREX (SEKINE *et al.*, 2000), a tarefa partilhada das CoNLLs (SANG *et al.*, 2003) e as ACEs. Enquanto a MET adotou diretamente a tarefa da MUC aplicando-a a japonês, espanhol e chinês, a tarefa partilhada da CoNLL procurou fomentar a investigação em sistemas de REM independentes do idioma, usando textos em holandês, espanhol, inglês e alemão, mas reduzindo significativamente a classificação, que passou a conter apenas quatro categorias semânticas: LOC (local), ORG (organização), PER (pessoa) e MISC (diversos).

O ACE 2004 propôs a pista *Entity Detection and Tracking* (EDT), em que o objetivo era fazer o reconhecimento de entidades, independentemente de serem mencionadas através de nomes próprios, o que aumentava consideravelmente a dificuldade da tarefa. Em outras palavras, o REM passou a compreender todo o reconhecimento semântico de entidades, sejam elas descritas por nomes comuns, próprios, pronomes ou sintagmas nominais.

Em português, tivemos o HAREM em 2006 e o Segundo HAREM em 2008, avaliações conjuntas organizadas pela Linguateca (um projeto que objetiva avançar o processamento computacional do português) e sobre as quais dedicaremos seções específicas mais adiante.

### 2.2.2. Classificações

A tarefa de REM é, em geral, dividida em duas partes: reconhecimento e classificação. O reconhecimento de EMs diz respeito à identificação e delimitação no texto das expressões que atendem aos critérios para serem classificadas como tais. A classificação corresponde à categorização da EM encontrada em um dos tipos possíveis para o domínio em questão (em geral, um dos citados a seguir).

Trabalhos seminais na área tratavam apenas do reconhecimento de nomes próprios (THIELEN, 1995). No geral, os tipos mais estudados de EMs são três categorias de nomes próprios: nomes de pessoas (*person*), locais (*location*) e organizações (*organization*). Desde a MUC-6, esses tipos são coletivamente conhecidos como *enamex*. O tipo "local" pode ser detalhado em múltiplos subtipos, como cidade, estado, país, etc. (FLEISCHMAN, 2001). O mesmo autor também examina a subcategorização (ex.: político, artista, empresário) do tipo "pessoa" (FLEISCHMAN *et al.*, 2002). Em um trabalho anterior, BODENREIDER *et al.* (2000) apresentam um uso inovador do tipo "pessoa": extrair expressões que contenham nomes de pessoas na terminologia biomédica, como mal de Parkinson (doença), tendão de Aquiles (parte do corpo), manobra de Heimlich (procedimento), respiração de Cheyne-Stokes (condição) e similares.

Nas conferências ACE, o tipo "instalação" (*facility*) engloba os tipos "local" e "organização", enquanto o tipo "GPE" é usado para representar uma entidade geopolítica (cidade, estado, país, etc.). O tipo "miscelânea" (*miscellaneous*) é utilizado nas conferências CoNLL e inclui nomes próprios fora do âmbito do *enamex*. Outra classe comum de EMs é "produto" (*product*) (BICK, 2004).

Além desses, também são bastante predominantes na literatura os *timexes* (de *time expressions*), outro termo cunhado nas conferências MUC, que incluem os tipos "data" (*date*) e "hora" (*time*) e são os objetos de estudo desta dissertação. Outros tipos comuns são a classe *numex* (de *numerical expressions*), que englobam os tipos "dinheiro" (*money*) e "percentual" (*percent*).

Finalmente, há exemplos na literatura de tipos usados com propósitos específicos: "filme", "cientista", "e-mail", "telefone", "área de pesquisa", "projeto", "autor", "livro", "cargo", "marca", entre outros (NADEAU *et al.*, 2007).

### 2.2.3. Desafios

Apesar de, a princípio, parecer que tirar proveito da capitalização das palavras é suficiente para reconhecer boa parte dos tipos de EMs (em particular, obviamente, as que envolvem nomes próprios), esse tipo de pista é quase inútil se a palavra estiver no início da frase. As dificuldades do REM advêm – além da diversidade de pontuação, grafia, espaçamento e formatação dos textos – da variação e ambigüidade comuns às EMs. Um homem chamado "Hélio Macedo" pode ser mencionado ao longo de um texto também pelo seu nome completo, por "Sr. Hélio" ou simplesmente "Hélio". Da mesma forma, o Departamento de Computação de uma determinada universidade pode ser citado como "Depto. de Computação" ou simplesmente "DC". Um bom sistema de REM deve ser capaz de reconhecer que todas essas referências fazem alusão à mesma entidade.

Além desses, há casos de ambigüidade especialmente difíceis: o nome "Ford" pode fazer referência à marca, à empresa de automóveis ou ao seu fundador, assim como a sigla "JFK" pode indicar o ex-presidente americano, seu filho ou o aeroporto de Nova Iorque batizado em sua homenagem. Outro tipo de ambigüidade é em relação a palavras comuns, sendo o emblemático exemplo na língua inglesa a palavra "*may*", que pode referir-se ao quinto mês do calendário gregoriano ou ao verbo que expressa possibilidade. Além disso, outro desafio reside no fato de que é comum um texto conter EMs de diferentes línguas (como nomes estrangeiros), o que pode tornar abordagens excessivamente atreladas a um determinado idioma impraticáveis para alguns domínios.

Um próximo desafio na área de REM é a extração de EMs a partir de outras mídias como áudio e vídeo (BASILI *et al.*, 2005).

### 2.2.4. Abordagens

Apesar de estudos e trabalhos iniciais terem se baseado majoritariamente em regras codificadas manualmente, os mais recentes têm usado aprendizado de máquina supervisionado, sendo esta a abordagem dominante atualmente para o problema de REM. Algumas dessas técnicas incluem HMMs (BIKEL *et al.*, 1997), árvores de decisão (SEKINE, 1998), modelos de máxima entropia (BORTHWICK *et al.*, 1998), máquinas de vetor de suporte (*support vector machines* ou SVMs) (ASAHARA *et al.*, 2003) e CRFs (McCALLUM *et al.*, 2003). Contudo, quando não há dados de treinamento suficientes, regras codificadas manualmente permanecem sendo a

abordagem preferida, como demonstrado por SEKINE *et al.* (2004), que desenvolveu um sistema de REM para 200 tipos de entidades.

Apesar de difíceis de adaptar para novos domínios, sistemas baseados em regras codificadas manualmente ainda oferecem melhor desempenho quando se é desejada uma classificação detalhada (*fine-grained*) das EMs. Já a abordagem de aprendizado de máquina, boa para uma classificação superficial (*coarse-grained*) é facilmente adaptável para novos domínios uma vez que dados de treinamento suficientes estejam disponíveis. Esse tipo de abordagem, em geral, utiliza uma adaptação do esquema de anotação IOB (de *Inside*, *Outside* e *Beginning*), também conhecido como BIO (HOBBS *et al.*, 2010), conforme descreveremos a seguir.

A notação IOB era empregada originalmente na tarefa de *chunking* (ou *shallow parsing*) em lingüística computacional para marcar palavras (*tokens*) como dentro (I), fora (O) ou iniciando (B) um *chunk*, conforme exemplificado na Tabela 2.1. No REM, que quase sempre requer a classificação das EMs em várias categorias, as etiquetas são expandidas para incluir também o tipo de EM, de forma análoga à demonstrada na Tabela 2.2.

**Tabela 2.1. Exemplo de uso da notação IOB/BIO**

| **Palavra** | O | ministro | Palocci | deixou | o | cargo | . |
|---|---|---|---|---|---|---|---|
| **Etiqueta** | B | I | I | O | B | I | O |

**Tabela 2.2. Exemplo de uso da notação IOB/BIO para REM**

| **Palavra** | José | Sérgio | Gabrielli | preside | a | Petrobras | . |
|---|---|---|---|---|---|---|---|
| **Etiqueta** | B-PERSON | I-PERSON | I-PERSON | O | O | B-ORGANIZATION | O |

Além das palavras e das etiquetas-alvo, os corpora de treinamento nas abordagens baseadas em aprendizado estatístico também utilizam uma grande variedade de traços. Traços são descritores ou atributos característicos das palavras, como valores booleanos (se a palavra é capitalizada ou não, por exemplo), numéricos (como o tamanho da palavra em caracteres) ou nominais (ex.: a palavra normalizada em letras minúsculas e sem acentos), e compõem uma abstração que as representa como um vetor (NADEAU *et al.*, 2007). Exemplos de vetores de traços seriam as colunas de cada palavra da frase da Tabela 2.3.

**Tabela 2.3. Exemplos de vetores de traços**

| **Traço \ Frase** | O | ministro | Palocci | deixou | o | cargo | . |
|---|---|---|---|---|---|---|---|

| **Capitalizada** | Sim | Não | Sim | Não | Não | Não | Não |
|---|---|---|---|---|---|---|---|
| **Tamanho** | 1 | 8 | 7 | 6 | 1 | 5 | 1 |
| **Normalizada** | o | ministro | palocci | deixou | o | cargo | . |

Boa parte dos traços utilizados é atrelada a características das palavras, como os relativos à capitalização (se começa com letra maiúscula, se todas as letras são maiúsculas), pontuação (se tem ponto, hífen ou apóstrofe no meio da palavra), morfologia (prefixo, sufixo, terminação, raiz), etiqueta POS (se é verbo, substantivo, etc.) e outros. Contudo, também há outros tipos de traços como os relativos à existência da palavra em almanaques e às características do documento e do corpus (freqüências, co-ocorrências, anáforas e afins).

Há ainda abordagens puramente baseadas em listas (*list lookup*), nas quais o sistema só reconhece entidades existentes em uma lista pré-definida conhecida como almanaque (em inglês, *gazetteer*) na comunidade de REM em língua portuguesa (NADEAU *et al.*, 2007). Apesar de ser demasiadamente simples (não lida, por exemplo, com variações e ambigüidades nas EMs), esse tipo de abordagem é rápida de ser implementada, tendo como único custo a criação e manutenção da lista. Independentemente disso, quase todos os sistemas de REM mais elaborados usam, em maior ou menor grau, almanaques como parte de sua abordagem, seja ela baseada em regras ou aprendizado de máquina. MIKHEEV *et al.* (1999) apresenta os resultados do seu sistema no MUC-7 para diferentes níveis de uso de almanaques, tendo a rodada sem almanaques alcançado resultados significativamente ruins apenas para entidades do tipo "local".

Outra abordagem bastante popular em REM é o relativamente recente aprendizado semisupervisionado, também conhecido como *bootstrapping* (NADEAU *et al.*, 2007). Essa técnica envolve um pequeno grau de supervisão: em geral, um pequeno conjunto de exemplos iniciais, a partir dos quais o sistema tentará identificar estruturas contextuais comuns que o permitam encontrar outras instâncias do tipo de entidade em questão em contextos similares. O sistema é então realimentado com esses novos exemplos encontrados, e o processo continua indefinidamente ou até que determinado critério de parada seja satisfeito.

### 2.2.5. Primeiro HAREM

O HAREM (Avaliação de Sistemas de Reconhecimento de Entidades Mencionadas) foi uma primeira tentativa de organizar o estado da arte para a tarefa de REM em língua portuguesa. A competição, no formato de avaliação conjunta, foi organizada pela Linguateca e contou com a participação de dez equipes de seis diferentes países (SANTOS *et al.*, 2006a). Uma avaliação conjunta é um modelo de avaliação em que vários grupos comparam, com base num conjunto de tarefas consensuais, o progresso dos seus sistemas numa dada área, usando para isso um conjunto de recursos comum e uma métrica consensual (SANTOS *et al.*, 2008).

As diretivas da competição elencaram três subtarefas: identificação, classificação semântica e classificação morfológica. A identificação avalia a habilidade dos sistemas de delimitar corretamente as EMs, independentemente de qualquer classificação. A classificação semântica consiste na atribuição de uma categoria semântica e tipo para cada EM identificada a partir e um conjunto de categorias e tipos pré-determinados (a Figura 2.3 na seção seguinte mostra essa hierarquia para o Segundo HAREM). A classificação morfológica diz respeito ao preenchimento de atributos referentes a gênero e número, quando aplicáveis.

O corpus do HAREM, chamado de coleção dourada ou CD, é uma coleção de textos de diferentes origens e gêneros que tiveram suas EMs identificadas e classificadas semântica e morfologicamente (a distribuição por categoria semântica é pouco diferente da utilizada no Segundo HAREM, ilustrada na Figura 2.3), além de ter sido revisada independentemente de acordo com um conjunto de diretivas aprovadas e discutidas por todos os participantes (CARDOSO *et al.*, 2007a, CARDOSO *et al.*, 2007b). Diferentemente do MUC, que utilizou uma abordagem *top-down* para escolher as categorias semânticas, o HAREM o fez de maneira *bottom-up*, analisando os textos, identificando entidades relevantes e construindo uma hierarquia de classificação (SECO, 2007). SANTOS *et al.* (2006b) apresenta mais detalhes sobre o processo de criação da CD.

Dos dez sistemas participantes, apenas dois, NERUA e MALINCHE (sobre os quais falaremos na seção dedicada a revisão dos sistemas que reconhecem ETs em português), utilizaram uma abordagem de aprendizado de máquina baseada em corpora previamente anotados (no caso, originalmente, em espanhol). O melhor sistema no primeiro HAREM obteve uma medida F de 0,63. Em comparação, o melhor sistema do

MUC alcançou 0,9642. Essa disparidade reflete tanto o desenvolvimento menos avançado do REM em português quanto um maior rigor da avaliação do HAREM (SECO, 2007).

### 2.2.6. Segundo HAREM

No Segundo HAREM, ocorrido entre 2007 e 2008, foi mantida a filosofia subjacente ao Primeiro HAREM, nomeadamente o modelo semântico e o modelo geral de avaliação (CARVALHO *et al.*, 2008). Entre as principais diferenças estão a inclusão de duas novas pistas: a ReRelEM, que avalia a detecção de relações entre as EMs (incluindo, mas não limitada à, resolução de co-referências) e a TEMPO, que trata do reconhecimento e normalização de entidades temporais.

O conjunto de etiquetas usado no Segundo HAREM não é significativamente distinto do usado no Primeiro HAREM (Figura 2.3). O número de categorias nas duas avaliações é idêntico: dez categorias, as quais permaneceram intactas em relação à sua designação (CARVALHO *et al.*, 2008). As categorias LOCAL e TEMPO foram as que sofreram alterações mais substanciais, sendo esta pormenorizada em seção posterior deste trabalho, uma vez que utilizamos substancialmente suas diretivas de anotação e avaliação.

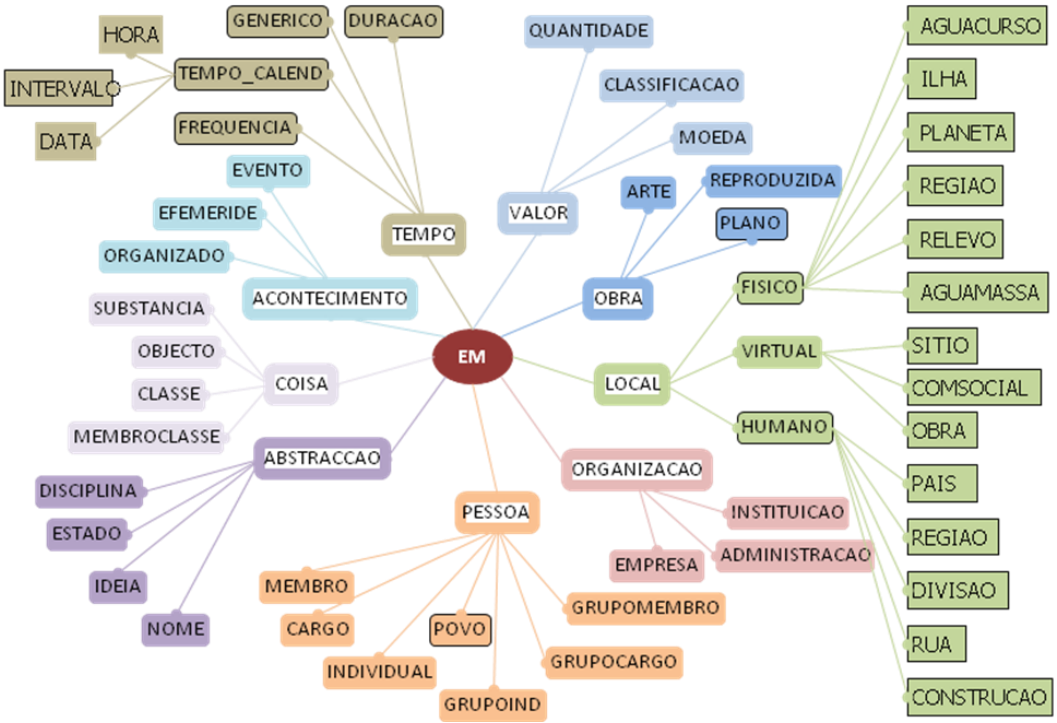



**Figura 2.3. Classificação semântica das EMs no Segundo HAREM (CARVALHO *et al.*, 2008)**

A nova CD também foi significativamente modificada, passando a incluir, além dos já tradicionais textos de notícias e páginas web, novos gêneros textuais, como blogs, wikis, enciclopédias (Wikipedia) e questões usadas em avaliações de sistemas de resposta automática a perguntas (FREITAS *et al.*, 2010), conforme ilustrado na Figura 2.4. Diferentemente do Primeiro HAREM, transcrições orais e textos literários foram bem menos utilizados. O corpus é constituído de 129 documentos, contabilizando um total de 147.991 palavras (CARVALHO *et al.*, 2008). A Figura 2.5 mostra a distribuição quantitativa das 7.847 EMs anotadas na CD por categoria semântica. Vale ressaltar que a categoria TEMPO ganhou mais importância nessa segunda edição do HAREM, sendo responsável por 15,21% das EMs anotadas (no Primeiro HAREM, as ETs totalizavam apenas 8,5%).

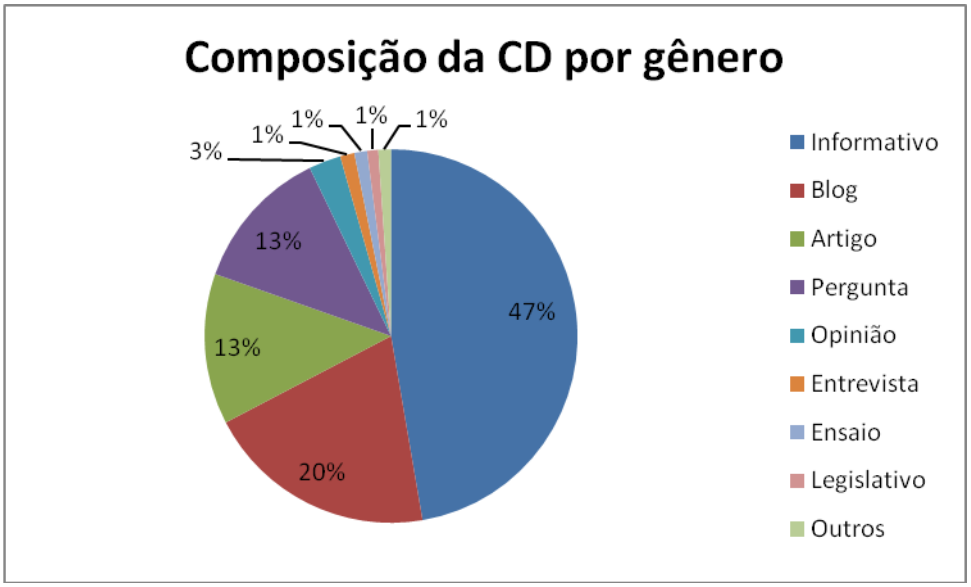



**Figura 2.4. Composição da coleção dourada do Segundo HAREM por gênero**

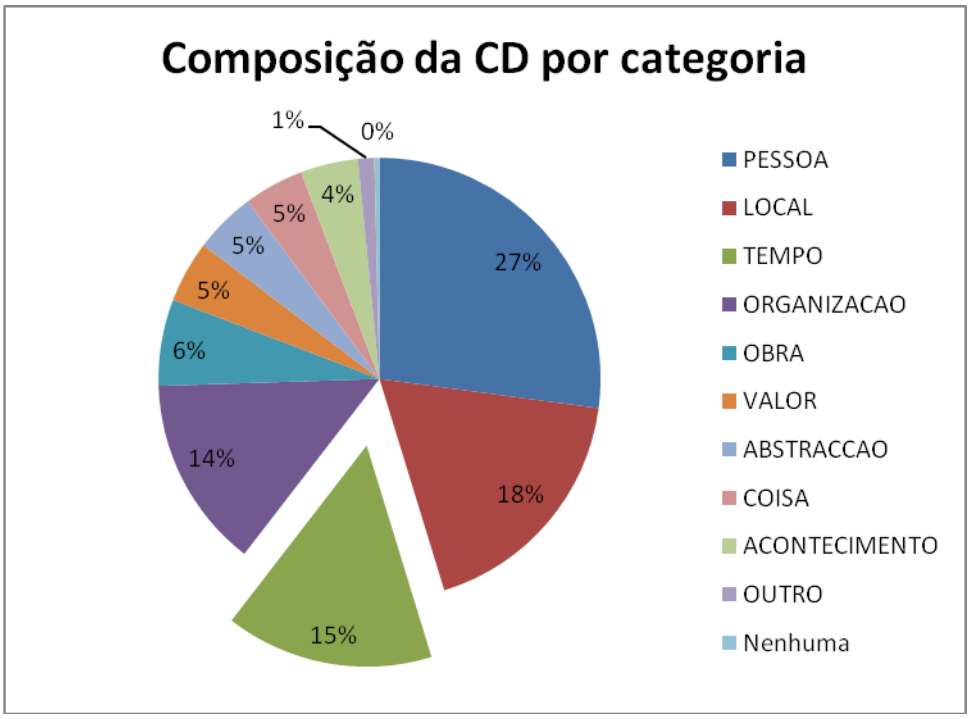



**Figura 2.5. Composição da coleção dourada do Segundo HAREM por categoria**

Dos dez participantes, apenas dois (Priberam e REMBRANDT, que detalharemos mais adiante) reconhecerem o conjunto completo de categorias, tipos e subtipos. Além disso, desses dez sistemas, apenas um adotou uma abordagem de aprendizado de máquina (especificamente, *co-training*); os outros se basearam em regras, dicionários, almanaques e ontologias (FREITAS *et al.*, 2010), o que mostra que a comunidade de REM em língua portuguesa, diferentemente da situação atual para o inglês, ainda não adotou técnicas de aprendizado de máquina (provavelmente, em virtude da falta de corpora anotados).

Atestando a superioridade da abordagem baseada em aprendizado estatístico (pelo menos no tocante a identificação e classificação semântica, que têm grande peso na avaliação), o Priberam foi o sistema com melhor medida F (0,5711), tendo ficado, no entanto, muito próximo do REMBRANDT, o segundo melhor sistema, cuja melhor corrida obteve 0,5674. Estes dois sistemas juntamente com o XIP-L2F/Xerox (que, diferentemente dos dois, é baseado em regras) foram os únicos a obter valores de medida F superiores a 0,5 (CARVALHO *et al.*, 2008).

# 2.3. Expressões temporais

A palavra “temporal” significa algo relacionado a tempo. O dicionário define o tempo como um período contínuo não espacial e indefinido através do qual os eventos ocorrem em uma sucessão aparentemente irreversível, criando no ser humano a idéia de passado, presente e futuro (HOUAISS *et al.*, 2001). Outra definição o classifica como um intervalo separando dois pontos em um contínuo, isto é, uma duração. Dado que a noção de evento aparece frequentemente em conjunto com o conceito de tempo, também é cabida a sua definição: algo que acontece, uma ocorrência.

O tempo já foi estudado em diversas disciplinas, em particular, religião, filosofia, física, lógica e arte. Desde que se há registro escrito, filósofos sempre discutiram o tempo e seu impacto na humanidade, notavelmente Gottfried Leibniz, Immanuel Kant, Henri Bergson e Martin Heidegger. Em física, além do seu uso prático como medida (a visão realista de Isaac Newton; por exemplo, o tempo que leva para um objeto se mover de um ponto A até um ponto B), o tempo também é parte central das mais avançadas teorias, incluindo a da relatividade geral de Albert Einstein. Finalmente, na arte, o tempo é objeto de pinturas (Salvador Dalí), de músicas e da literatura (“*An Experiment with Time*” de John William Dunne, “*Nueva refutación del tiempo*” de Jorge Luis Borges e “*About Time*” de Paul Davies, para citar alguns).

Para este trabalho, adotamos o conceito de que o tempo envolve o processo de medir eventos nas unidades escolhidas (milissegundo, minuto, século, etc.). Uma linha do tempo, também conhecida como cronologia, é uma interpretação linear de eventos na ordem em que eles aconteceram. Um calendário é um sistema projetado para representar o tempo físico e define os valores, chamados granularidades, que interessam para o usuário, em geral, segmentos específicos da linha do tempo. O mais conhecido é o calendário gregoriano (também chamado de calendário ocidental ou calendário cristão), que é baseado na rotação da Terra em seu próprio eixo e sua revolução ao redor do Sol. Ele foi criado em 1582 pelo Papa Gregório XIII, e suas granularidades são ano, mês, dia (sua unidade básica de medida de tempo), hora, minuto e segundo (Figura 2.6).

| **Número** | **Mês** | **Dias** |
|---|---|---|
| 1 | Janeiro | 31 |
| 2 | Fevereiro | 28 ou 29 |
| 3 | Março | 31 |
| 4 | Abril | 30 |
| 5 | Maio | 31 |

| **Granularidade** | **Fração** |
|---|---|
| Hora | $\frac{1}{24}$ do dia |

| | | |
|---|---|---|
| 6 | Junho | 30 |
| 7 | Julho | 31 |
| 8 | Agosto | 31 |
| 9 | Setembro | 30 |
| 10 | Outubro | 31 |
| 11 | Novembro | 30 |
| 12 | Dezembro | 31 |
| | **Total (ano)** | 365 ou 366 |

| | |
|---|---|
| Minuto | $\frac{1}{60}$ da hora |
| Segundo | $\frac{1}{60}$ do minuto |

**Figura 2.6. Granularidades do calendário gregoriano**

### 2.3.1. O tempo na linguagem natural

Há inúmeras maneiras de se expressar o tempo em idiomas como o português, sendo, não surpreendentemente, o tempo verbal uma das principais. O tempo verbal é a expressão gramatical de localização no tempo e envolve a mudança da forma de elementos sintáticos como verbos e auxiliares, por exemplo, "Pedro comeu um sanduíche" ou “Pedro está comendo um sanduíche”. Outro mecanismo linguístico capaz de indicar se um evento ocorreu ou não é o modo verbal (“Quando Pedro comer o sanduíche...”).

Uma terceira maneira de indicar tempo, durações e frequências é através de expressões referentes a tempo: as expressões de tempo ou expressões temporais (ETs). Alguns exemplos incluem "na quinta-fei1ra", "11 de setembro de 2001", "na década de 80", "duas semanas" e "diariamente". Apesar de utilizarmos os dois mecanismos do parágrafo anterior para desambiguar e normalizar expressões temporais, é neste tipo de indicativo temporal que nos concentraremos nesta dissertação. Um trabalho posterior que tenha como intuito a ancoragem e ordenação de eventos numa linha do tempo deverá explorar exaustivamente os três mecanismos.

Como a linguagem natural é ambígua e está em constante evolução, há mais de um modo de se representar uma mesma informação temporal: "Natal" e "25 de dezembro" podem significar a mesma coisa, mas são expressas distintamente, uma da perspectiva da data comemorativa e a outra como um simples valor do calendário. Quando levamos em conta outras línguas, além da óbvia diferença das palavras no idioma ("Natal" em português, "Christmas" em inglês, "Navidad" em espanhol, etc.), ainda podemos apontar que um mesmo evento temporal pode remeter a diferentes valores no calendário dependendo da cultura e do país (Dia do Trabalho, Dia dos Namorados, Dia da Independência, Dia dos Pais, entre outros).

Para se ancorar eventos em uma linha do tempo, se faz necessário normalizar as ETs para uma representação comum que possa ser mapeada. Esse é um passo essencial

para unificar as diferentes maneiras que um mesmo ponto do tempo pode ser expresso em linguagem natural. Uma simples data absoluta, por exemplo, pode ser representada por um sem-número de variações textuais: "11/09/1985", "11-09-1985", "11/9/85", "11.09.1985", "11 de setembro de 1985", "11 de set. de 85", entre outros. Esses exemplos mostram que um bom desempenho de qualquer processamento temporal posterior dependerá de uma identificação e normalização bem feita das ETs.

### 2.3.2. Classificações

A informação temporal de um documento pode ser originária de duas fontes: metadados e conteúdo textual. Metadados temporais dizem respeito, em geral, à data de criação, publicação ou última modificação do documento aos quais estão associados. A data de criação do documento (*document creation time* ou DCT) é especialmente importante para ancorar ETs em uma linha do tempo.

O segundo tipo de informação temporal envolve a análise linguística do conteúdo textual dos documentos a partir de técnicas de extração de informação e extração de entidades mencionadas, sendo as entidades temporais conceitos relacionados ao tempo e representados nos documentos como sequências (não necessariamente contíguas) de *tokens* ou palavras. Em particular, entidades temporais são explicitadas na forma de expressões temporais que, na maioria das vezes, correspondem a um ponto numa linha do tempo. Uma ET é, portanto, uma porção de conteúdo textual que expressa uma informação temporal direta ou inferida, incluindo principalmente datas ("11/09/1985") ou sintagmas preposicionais contendo expressões de tempo ("no sábado").

Não há um consenso na literatura quanto à classificação de ETs. SCHILDER *et al.* (2001) distinguem entre dois tipos de expressões: as que denotam tempo e as que denotam eventos. As primeiras incluem trechos que expressam informação temporal referenciáveis em um calendário. Sintaticamente falando, geralmente, essas expressões são sintagmas preposicionais, adverbiais ou substantivos ("na sexta-feira", "hoje", "o quarto trimestre"). O segundo grupo, as expressões que denotam eventos, tem uma dimensão temporal implícita, uma vez que todas as situações possuem um componente temporal. Nessas expressões, no entanto, não há qualquer ligação direta ou indireta com um calendário. Elas são, quase sempre, verbos ou sintagmas nominais ("aumentou" ou "a eleição"). Esta dissertação está focada no primeiro grupo de expressões (as que

denotam tempo), para o qual listamos os critérios mais comuns de classificação na Tabela 2.4, sendo a classificação semântica e a classificação quanto à necessidade de contexto as mais encontradas nos trabalhos da área.

**Tabela 2.4. Critérios de classificação de expressões temporais**

| Critério de classificação | Categorias |
|---|---|
| Semântica | Data, hora, duração, frequência, intervalo e genérico |
| Contexto | Absoluta e relativa (ou indexada ou referencial) |
| Referencial | Dêitica e anafórica |
| Precisão | Precisa e imprecisa (ou indeterminada ou vaga) |
| Clareza | Explícita e implícita |
| Composição | Simples e complexa |

### 2.3.2.1. Classificação semântica

Apesar de ubíqua, a classificação semântica varia bastante entre as diferentes conferências e avaliações, tanto nas definições quanto na quantidade de categorias utilizadas. Em geral, a categoria "data" (a mais comum) inclui ETs que representam um momento no calendário, isto é, expressões temporais que correspondem a uma ancoragem única na linha do tempo, como as exemplificadas na Tabela 2.5 abaixo. A segunda categoria mais comum ("hora") engloba os horários de relógio (granularidade inferior ao dia do calendário). Não é incomum encontrar trabalhos que agrupam as categorias "data" e "hora" em uma só (normalmente com o nome daquela) ou que consideram "hora" como uma subcategoria de "data".

**Tabela 2.5. Exemplos de categorias de classificação semântica**

| Categoria | Exemplos |
|---|---|
| Data | 11/09/1985, 11 de setembro, sábado, próximo dia 11, na semana passada |
| Hora | 19:00, 18h, às 14 horas |
| Duração | dois meses, durante algum tempo, por um ano, [levou] cinco horas |
| Frequência | semanalmente, todos os dias, dia sim dia não, duas vezes por mês |
| Intervalo (data) | entre 2004 e 2007, de março a junho de 2008 |
| Intervalo (duração) | de 3 a 6 meses, entre duas e três semanas, 10-15 dias |
| Genérica | [odeio] segundas-feiras, Fevereiro [é o mês mais curto] |

Outra categoria semântica bastante comum é a "duração": ETs que se referem a uma duração de tempo contínuo. Ao contrário das datas, trata-se de expressões que não exprimem propriamente a localização de um evento (*calendarização* do evento), mas sim quantificação de ordem temporal, sendo constituídas por nomes de unidades de medida de tempo e determinantes com função de quantificadores (e.g.. numerais).

Podem, por vezes, ser introduzidas, facultativamente, pelas preposições “durante”, “por” ou “em” e respondem adequadamente à interrogativa “durante/por/em quanto tempo?”.

A categoria “frequência” corresponde às expressões temporais que permitem várias instâncias de ancoragem com a linha do tempo, ou seja, exprimem uma repetição no tempo. O tipo “intervalo”, apesar de menos presente que os restantes na literatura, não é tão incomum em textos em linguagem natural, podendo tratar-se de intervalos de data ou intervalos de duração, conforme os exemplos da Tabela 2.5. Correspondem a expressões complexas, compostas por duas expressões temporais simples, mas que, semanticamente, formam uma única entidade mencionada e que tem explicitamente dois limites temporais. Nos trabalhos que os mencionam diretamente, os intervalos são mais comumente tratados como subcategorias de “data” e “duração”.

Finalmente, a categoria "genérica" trata de ETs que não se referem uma data específica, embora a expressão linguística seja composta por unidades lexicais que denotam elementos temporais. Essas expressões não ancoram qualquer processo na linha do tempo e não são realmente expressões temporais no sentido de que nenhuma informação temporal está associada a qualquer processo, mas mantêm um significado temporal que pode ser importante para a resolução de referências temporais.

#### 2.3.2.2. Classificação quanto à necessidade de contexto e referencial

A distinção entre ETs absolutas e relativas se deu desde a MUC-6, a primeira a tratar de entidades temporais. Absolutas seriam ETs que indicassem um minuto, hora, dia, mês, ano, etc. específico (por exemplo, “11 de setembro”) enquanto as restantes seriam tidas como relativas. Apesar dessa distinção conceitual, as ETs não eram de fato classificadas na anotação, sendo a única categorização presente nas etiquetas referente à classificação semântica: data (*DATE)* ou hora (*TIME*), dependendo da amplitude da expressão (maior ou menor que um dia, respectivamente).

SETZER (2001) propôs que a separação entre ETs absolutas e relativas fosse redefinida, passando as absolutas a serem aquelas que podem ser posicionadas de forma não ambígua numa linha do tempo sem qualquer informação adicional de outras partes do texto no qual ocorrem, sendo esta a definição que adotamos neste trabalho. Na literatura, também encontramos trabalhos que fazem essa mesma separação, mas com

nomenclatura distinta, chamando as ETs relativas de expressões indexadas (*indexical*) ou referenciais.

A maioria das durações e frequências é classificada como ETs absolutas, enquanto praticamente todas as da categoria "hora" são relativas. Expressões de datas como "setembro de 1985" ou "3 de março de 1987" são expressões absolutas, mas a simples omissão do ano, tornaria ambas relativas, pois seria preciso determinar a que ano se referem, o que não é possível sem estudar o contexto em que elas ocorrem. Não surpreendentemente, a maior parte das ETs encontradas em textos é classificada como relativa (VICENTE-DÍEZ *et al.*, 2008).

As expressões temporais relativas podem ainda ser classificadas quanto ao referencial. Por exemplo, a ET da frase "*A festa aconteceu* <DATA>*ontem*</DATA>" é relativa, pois sua resolução implica conhecer o momento da enunciação (que, em geral, é a DCT). Consequentemente, se essa sentença tiver sido enunciada em 11 de setembro de 2010, pode-se inferir que a festa aconteceu no dia 10 de setembro de 2010. Por esse critério, essa ET relativa é classificada como dêitica (WIEBE *et al.*, 1998), isto é, relativa ao momento da enunciação.

Uma ET relativa como a da frase "*A festa aconteceu* <DATA>*no dia anterior*</DATA>" é, por sua vez, classificada como anafórica, pois, para ancorá-la na linha do tempo, é preciso conhecer outra data de um evento que funciona como referência, isto é, a localização temporal da ET é independente do momento em que a sentença foi produzida, sendo a referência outra data ou evento que aparece no contexto textual ou discursivo. Esse ponto no tempo ao qual outras ETs fazem referência (e que muda ao passo em que o texto progride) é conhecido como foco temporal (*temporal focus*) ou tempo de referência (*reference time*). Desse modo, se o texto completo fosse "*Ele chegou à cidade* <DATA>*em 11 de setembro de 2010*</DATA>. *A festa aconteceu* <DATA>*no dia anterior*</DATA>.", poderíamos inferir que a festa ocorreu em 10 de setembro de 2010. Vale notar que o referencial, nesse caso, é a ET da frase anterior e não o momento da enunciação. Algumas ETs anafóricas (em particular, aquelas sem um indicador de direção, como "seguinte" ou "anterior") podem depender também do modo e tempo do verbo que modificam para determinar se o ponto no tempo ao qual se referem é antes ou depois do referencial (AHN *et al.*, 2005).

#### 2.3.2.3. Classificação quanto à precisão, clareza e composição

Expressões temporais precisas são aquelas para as quais podemos determinar um valor de data no calendário, hora, duração ou frequência. Expressões que indicam informação temporal, mas de forma vaga, isto é, de modo que é difícil colocá-la precisamente em uma linha do tempo são classificadas como imprecisas: por exemplo, "em algumas semanas" e "há algum tempo" não podem ser representadas por pontos ou intervalos exatos no tempo.

O critério de clareza separa as ETs em explícitas e implícitas (ALONSO, 2008), incluindo, nestas, nomes de feriados, datas comemorativas ou eventos que carreguem consigo um significado temporal não explícito diretamente no texto, mas que podem ser mapeadas e ancoradas na linha do tempo: “Natal” (25 de dezembro), “Dia da Independência” (7 de setembro) , “Dia dos Namorados” (12 de junho), “Réveillon”, “Ano Novo” (31 de dezembro), etc.

Por fim, enquanto a maioria das expressões temporais é classificada como simples, SETZER (2001) identificou a existência de ETs compostas, isto é, expressões temporais formadas pela composição de outras ETs (estas simples), como “dois dias depois do dia 14 de maio” ou “de hoje a quinze dias”.

### 2.3.3. Histórico da área

As primeiras referências à extração de expressões temporais surgiram na sexta das MUCs, em novembro de 1995. A MUC-6, como as anteriores, envolvia a avaliação de sistemas de EI. Contudo, somente a partir desta, foi criada uma tarefa separada para REM. Ela era composta por três subtarefas, dentre as quais uma tratava de expressões temporais. Essa avaliação, a primeira do tipo, desafiava os candidatos a identificar e anotar as diversas EMs, entre elas, as expressões temporais, conhecidas como *timex*, a tag SGML utilizada nas marcações. Nessa subtarefa, apenas ETs absolutas deviam ser etiquetadas. Além disso, só eram reconhecidos os tipos “data” e “hora”, apesar de incluírem nomes de feriados e datas comemorativos (como “Dia da Independência”) como datas opcionais.

Exemplos de frases anotadas com o padrão TIMEX (usando suas *tags* SGML) poderiam ser: ‘O *Rio de Janeiro sediará as Olimpíadas em* <TIMEX TYPE=“DATE”>*julho de 2016*</TIMEX>’ e ‘*Amanhã, almoçaremos às* <TIMEX TYPE=“TIME”>*13h*</TIMEX>’. Na MUC-6, também não havia qualquer requisito de

interpretação (normalização) das expressões. Na tarefa NE da conferência (não há avaliação do reconhecimento de ETs isoladamente), o sistema mais bem avaliado alcançou 97% de precisão e 96% de revocação (GRISHMAN *et al.*, 1996). As *timexes* representavam 10% do corpus usado na avaliação (para efeito de comparação, a classe *enamex* representava 82% e a *numex*, os 8% restantes).

A MUC-7, em 1998, se diferenciou da MUC-6 por tratar do REM multilingual, passando a contar com os idiomas chinês e japonês. Além disso, pediu também a identificação de ETs relativas e de feriados e datas especiais referenciadas por nome (agora, não mais opcionais). No corpus dessa avaliação, as *timexes* correspondiam a 25% do total de EMs (sendo 15% da categoria "hora" e 85% de "data"), enquanto a classe *numex* respondia por 6% e a *enamex* por 69%. O melhor sistema na tarefa NE teve uma medida F de 93,39% (95% de precisão e 92% de revocação).

Em 2000, no contexto do programa *Translingual Information Detection, Extraction and Summarization* (TIDES) da DARPA, começou a se desenvolver o padrão TIMEX2, o padrão mais utilizado até hoje para anotação de ETs. O TIDES é um ambicioso e amplo programa de desenvolvimento tecnológico da DARPA que foca no processamento e entendimento automático de linguagem. O avanço fundamental do TIMEX2, comparado ao TIMEX original concebido nas MUCs, foi a inclusão da tarefa de normalização, além do reconhecimento, de ETs (usando o padrão ISO 8601 para representar as datas e horas normalizadas).

O esquema de anotação TIMEX2 tem como base os esforços iniciados na MUC-7, estendendo o leque de ETs identificadas e anotadas e incluindo diretivas e atributos para normalização desses dados, e tem esse nome por utilizar uma única *tag* XML, a <TIMEX2>, para marcar as ETs nos textos. Durante o período em que o padrão estava sendo desenvolvido, MANI *et al.* (2000), num dos seminais e mais referenciados trabalhos da área, criou o TEMPEX, um etiquetador temporal automático que já anotava documentos de texto no padrão TIMEX2. Na mesma época, SETZER *et al.* (2000a) desenvolveu, independentemente, um outro padrão de anotação que representava tanto ETs como eventos e relações temporais. Tal padrão, mais tarde, daria origem à TimeML, o atual estado da arte para anotação de informação temporal em textos.

Além de fomentar o desenvolvimento do TIMEX2, o TIDES disponibilizou dois corpora com anotações temporais (8.243 ETs marcadas): um paralelo com 95 diálogos (44.081 palavras) em espanhol e traduções para inglês e outro com 193 documentos

(171.535 palavras) do corpus em inglês do TDT-2 (*Topic Detection and Tracking*) da *Linguistic Data Consortium* (LDC). Esses corpora foram anotados por seis anotadores humanos, tendo alcançado consenso médio, respectivamente de 0,73 e 0,83 de medida F para identificação e normalização de ETs (FERRO *et al.*, 2002), o que mostra que tal tarefa é difícil até para pessoas.

No estudo piloto sobre anotação de relações temporais de SETZER *et al.* (2001), os autores identificam que de 30-40% dos erros dos anotadores pode ser atribuído à ambiguidade das diretivas de anotação ou à erros em aplicá-las. O trabalho, que tem como base uma combinação do esquema proposto em (SETZER *et al.*, 2000a, SETZER *et al.*, 2000b) para anotação de eventos e relações temporais e das diretivas do TIDES para anotação de ETs (FERRO *et al.*, 2001), lista ainda outros problemas encontrados na construção de corpora temporalmente anotados: imprecisão das diretivas e entendimento, cansaço e descuido dos anotadores.

A especificação TIMEX2 foi, em seguida, usada na ACE de 2002, como o padrão de anotação para atributos temporais de relações. A ACE começou tratando do problema de detecção e acompanhamento de entidades (*Entity Detection and Tracking* ou EDT), similar ao REM. Em 2001, passou a ter também a tarefa de detecção e caracterização de relações (*Relation Detection and Characterization* ou RDC). Em sua edição de 2004, o programa passou a tratar especificamente de ETs, introduzindo, além das três tarefas de entidades, eventos e relações, uma tarefa específica, dedicada exclusivamente, para o reconhecimento e normalização de expressões de tempo (ACE, 2004), a *Timex2 Detection and Recognition* (TDR), depois mais conhecida como *Time Expression Recognition and Normalization* (TERN). Essa tarefa pedia que os sistemas participantes detectassem e normalizassem as ETs mencionadas nos corpora de acordo com a versão do padrão de anotação TIMEX2 definida em (FERRO *et al.*, 2004). A ACE 2004 trabalhou com três idiomas (inglês, chinês e árabe), mas somente os dois primeiros estavam presentes na TERN (TERN, 2004), conforme mostram as estatísticas da Tabela 2.6.

**Tabela 2.6. Estatísticas da ACE 2004**

| Corpus | Documentos | Palavras | ETs | ETs / documento |
|---|---|---|---|---|
| Inglês (treinamento) | 767 | 265.809 | 8.047 | 10,5 |
| Chinês (treinamento) | 466 | 147.502 | 4.415 | 9,5 |
| Inglês (avaliação) | 192 | 54.614 | 1.828 | 9,5 |
| Chinês (avaliação) | 256 | 66.921 | 2.429 | 9,5 |

A TERN determinava a marcação tanto de datas e horas absolutas como relativas, além das durações, expressões ancoradas em eventos (“dois dias antes da viagem”) e frequências. A avaliação de 2004 oferecia dois cenários de participação: reconhecimento e normalização (processamento completo no qual os sistemas tinham seu desempenho avaliado na detecção, delimitação e normalização das ETs) ou apenas reconhecimento (no qual se avaliavam apenas detecção e delimitação de ETs). Isso evidenciou uma divisão entre as abordagens de EI temporal: os sistemas que participaram apenas do cenário de reconhecimento eram todos baseados em aprendizado de máquina, enquanto os sistemas que atacaram a tarefa completa (reconhecimento e normalização) eram puramente baseados em regras (AHN *et al.*, 2005). A razão para tal é clara: o simples reconhecimento de ETs pode ser facilmente visto como uma tarefa de etiquetagem de sequências (*sequence labeling*), o que permite tirar proveito dos inúmeros sistemas de aprendizado de máquina já existentes para isso.

Apesar de ter incluído uma nova tarefa, relativa à detecção e reconhecimento de valores (monetários, percentuais, e-mail, telefones, entre outros), no tocante às ETs, muito pouco mudou (FERRO *et al.*, 2005), tendo a ACE 2005 continuado basicamente com os mesmo requisitos da edição anterior (ACE, 2005). Seis grupos participaram da avaliação de ETs, dos quais quatro para inglês e dois para chinês, conforme os resultados da Tabela 2.7 e da Tabela 2.8. Os resultados para o chinês foram significativamente superiores, pois a avaliação no idioma não envolvia a normalização, tornando a tarefa bem mais simples que a conduzida para o inglês.

**Tabela 2.7. Resultados da ACE 2005 para a língua inglesa**

| **Participante** | Language Computer Corp. | Lockheed Martin | Janya Inc. | University of Amsterdam |
|---|---|---|---|---|
| **Escore** | 63,7 | 56,2 | 54,8 | 33,2 |

**Tabela 2.8. Resultados da ACE 2005 para a língua chinesa**

| **Participante** | Polytechnic University of Hong Kong (chinês) | Peking University (chinês) |
|---|---|---|
| **Escore** | 83,7 | 79,0 |

À exceção da inclusão do espanhol, a ACE 2007 também repetiu sem maiores modificações a avaliação da edição anterior, de 2005. Conforme listado na Tabela 2.9 e na Tabela 2.10, das oito equipes participantes, quatro atacaram o problema em inglês, uma em chinês e outra em espanhol. A última edição da ACE, em 2008, mudou de foco

e passou a tratar apenas da detecção e reconhecimento de entidades e relações intra e interdocumentos (ACE, 2008).

**Tabela 2.9. Resultados da ACE 2007 para a língua inglesa**

| **Participante** | Lockheed Martin | IBM | University of Amsterdam | Macquarie University |
|---|---|---|---|---|
| **Escore** | 61,6 | 59,3 | 58,2 | 48,3 |

**Tabela 2.10. Resultados da ACE 2007 para chinês e espanhol**

| **Participante** | Lockeed Martin (chinês) | Universidad Carlos III de Madrid (espanhol) |
|---|---|---|
| **Escore** | 14,8 | 46,5 |

Paralelamente, em 2002, foi dado início ao desenvolvimento do maior corpus anotado com informações temporais e de eventos durante *workshop Time and Event Recognition for Question Answering Systems* (TERQAS). Essa coleção, chamada de TimeBank, continha 300 notícias com anotações detalhadas, em TimeML, de termos denotando eventos, ETs, sinais temporais e ligações representativas de relações temporais (PUSTEJOVSKY *et al.*, 2003). A última versão do corpus (TimeBank 1.2), de 2006, contém 183 notícias anotadas com base na especificação 1.2.1 da TimeML. A coleção contabiliza mais de 61 mil tokens, 7.935 eventos, 1.414 ETs, 688 sinais temporais e 6.418 ligações temporais.

Em 2006, aconteceu o *International Symposium on Temporal Representation and Reasoning* (TIME 2006), com cinco artigos sobre o tempo em linguagem natural.

Em 2007, no contexto do 4th *International Workshop on Semantic Evaluations* (SemEval-2007), ocorreu a TempEval-1, um exercício de avaliação com o intuito de avançar a pesquisa na identificação automática de ETs, eventos e relações temporais em textos. O SemEval surgiu com a ampliação do escopo do SenseEval, um *workshop* de desambiguação do sentido de palavras (*Word Sense Desambiguation* ou WSD), a partir de sua quarta edição (as três primeiras, com o nome original, acontecerem em 1998, 2001 e 2004).

Apesar de recente, a TempEval foi a primeira avaliação conjunta a atacar o problema das relações temporais (VERHAGEN *et al.*, 2009). Ela foi dividida em três tarefas (A, B e C), para as quais foram disponibilizados dados de treinamento e teste (baseados no TimeBank 1.2) contendo anotações que identificavam limites de sentenças, ETs (no formato TIMEX3), eventos (em TimeML) e um subconjunto de relações temporais consideradas relevantes. O formato TIMEX3 é o esquema de

anotação de ETs adotado dentro da linguagem TimeML e, à exceção de algumas diretivas de anotação de frequências, é exatamente igual ao TIMEX2.

A tarefa A pedia que fossem identificadas as relações temporais entre expressões de tempo e eventos na mesma sentença, enquanto a tarefa B tinha como requisito a identificação das relações entre a DCT e expressões de eventos. A tarefa C, finalmente, consistia em atribuir uma relação temporal entre os eventos principais (geralmente, o verbo sintaticamente dominante) de sentenças adjacentes. As tarefas A e B envolvem casos restritos de ancoragem temporal, enquanto a tarefa C cobre um caso específico de ordenação temporal. Em todas elas, os dados estavam identificados com as anotações supracitadas e cabia aos participantes fornecerem apenas as etiquetas corretas (o tipo de relação temporal) para os dados de teste (VERHAGEN *et al.*, 2007). As três tarefas podem, portanto, ser vistas como tarefas de classificação nas quais, dados links temporais, deve-se atribuir um tipo de relação a partir de um conjunto pré-definido (BEFORE, AFTER, OVERLAP, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER ou VAGUE).

Dos seis participantes, três usaram exclusivamente métodos estatísticos, enquanto um utilizou um sistema baseado em regras e dois, uma abordagem híbrida (VERHAGEN *et al.*, 2009). O desempenho dos sistemas (precisão, revocação e medida F) foi bastante similar e alguns deles nem sequer diferiram muito da *baseline* (Tabela 2.11).

**Tabela 2.11. Resultados do TempEval-1 (medida F nas formas estrita / relaxada)**

| **Sistema** | **Tarefa A** | **Tarefa B** | **Tarefa C** |
|---|---|---|---|
| *Baseline* | 57/60 | 56/57 | 47/53 |
| CU-TMP | 61/63 | 75/76 | 54/58 |
| LCC-TE | 58/60 | 73/74 | 55/58 |
| NAIST | 61/63 | 75/76 | 49/53 |
| USFD | 59/60 | 73/74 | 54/57 |
| WVALI | 62/64 | 80/81 | 54/64 |
| XRCE-T | 34/41 | 66/71 | 42/58 |

Os resultados do primeiro TempEval indicaram que a tarefa A era muito geral, contribuindo, em parte, para o baixo consenso entre os anotadores. Levando isso em consideração, a segunda edição do TempEval, realizada no SemEval 2010, foi mais elaborada e, além de ser multilingual, propôs seis tarefas (ETs, eventos e quatro de relações temporais), mais específicas e detalhadas, em vez das três originais, objetivando tornar tanto a preparação dos dados como a extração de relações temporais

mais fácil, simples e consensual (PUSTEJOVSKY *et al.*, 2009). Os dados (contendo entre 10 e 60 mil tokens) para os seis idiomas da avaliação (inglês, espanhol, chinês, francês, coreano e italiano) foram manualmente anotados (com base na TimeML 1.2.1) e preparados independentemente uns dos outros, não havendo, portanto, um corpus paralelo.

A tarefa A no TempEval-2 consistia em anotar as ETs em um texto usando a etiqueta TIMEX3 da TimeML, identificá-las (como data, hora, duração ou frequência) e normalizá-las. A tarefa B pedia que fossem anotados os eventos no texto usando a etiqueta EVENT da TimeML e determinados os valores de atributos como tempo (*tense*), modo (*aspect*), polaridade (*polarity*) e modalidade (*modality*). As tarefas C, D, E e F tratam da identificação das relações temporais, respectivamente, entre expressões de tempo e eventos na mesma sentença (tarefa A da TempEval-1), entre eventos e a DCT (tarefa B da TempEval-1), entre os principais eventos de duas sentenças consecutivas (tarefa C da TempEval-1) e entre dois eventos onde um evento domine sintaticamente o outro, como, por exemplo, "ela ouviu uma explosão" ou "ela falou que eles adiaram a reunião" (VERHAGEN *et al.*, 2010).

**Tabela 2.12. Resultados do TempEval-2 para a língua inglesa**

| | A | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Equipe** | **Medida F** | **Tipo** | **Valor** | **B** | **C** | **D** | **E** | **F** |
| Edinburgh | 0,84 | 0,84 | 0,63 | 0,80 | | | | |
| HeidelTime | 0,86 | 0,96 | 0,85 | | | | | |
| JU_CSE | 0,26 | 0,00 | 0,00 | 0,52 | 0,63 | 0,80 | 0,56 | 0,56 |
| KUL | 0,84 | 0,91 | 0,55 | | | | | |
| NCSU | | | | | 0,63 | 0,68 | 0,51 | 0,66 |
| TERSEO | 0,71 | 0,98 | 0,65 | | | | | |
| TIPSem | 0,85 | 0,92 | 0,65 | 0,83 | 0,55 | 0,82 | 0,55 | 0,60 |
| TRIOS | 0,85 | 0,94 | 0,76 | 0,77 | 0,65 | 0,79 | 0,56 | 0,60 |
| TRIPS | 0,85 | 0,94 | 0,76 | 0,68 | 0,63 | 0,76 | 0,58 | 0,59 |
| USFD2 | 0,82 | 0,90 | 0,17 | | 0,63 | | 0,45 | |

Apesar da disponibilidade de dados, apenas dois idiomas (inglês e espanhol) foram abordados pelos sistemas participantes. A Tabela 2.12 apresenta os resultados para o inglês de cada sistema nas tarefas do TempEval-2. É interessante apontar que a classificação de ETs na tarefa A (coluna "Tipo") é notavelmente mais simples que a normalização (coluna "Valor").

### 2.3.4. Expressões temporais em outras línguas

Uma das primeiras referências na literatura tratando de expressões temporais em outros idiomas apresenta um método para tradução de ETs em japonês para o inglês (BOND *et al.*, 1997). Nesse trabalho, os autores desenvolvem um módulo separado do sistema completo de tradução, baseado em regras, para lidar com as idiossincrasias das ETs nas duas línguas.

Apesar de GAGNON *et al.* (1996) descrever um método de geração de textos em francês capaz de transmitir informação temporal de maneira coerente, VAZOV (2001a) é o primeiro a introduzir um sistema de extração de ETs para essa língua. Esse sistema busca, no texto, marcadores de ETs (expressões regulares) contidos em três listas pré-definidas. Em seguida, checa se esses marcadores fazem parte de uma ET maior, verificando seus contextos à esquerda e à direita através de regras (de um *chart-parser*). Para evitar a sobregeração, são aplicadas restrições que podem diminuir os limites da ET identificada. Em (VAZOV, 2001b), esse sistema, além de ser aperfeiçoado com regras e restrições mais complexas, é avaliado em um corpus de 50 milhões de palavras, tendo alcançado 85% de precisão e 95% de revocação.

Ainda para o francês, BALDWIN (2002) desenvolveu um sistema de anotação temporal automática, que aprende regras a partir de um corpus anotado no padrão TIMEX2. O sistema foi avaliado em um corpus de 45 notícias (30 para o conjunto de treinamento e 15 para o conjunto de teste), contendo 584 tags TIMEX2, obtendo 69% de precisão e 54% de revocação. Recentemente, BITTAR (2009) detalhou uma cadeia de módulos de anotação de informação temporal em textos em Francês (depois de passarem por etiquetagem *part-of-speech* e análise morfológica e sintática). Esse sistema de anotação utiliza a TimeML e, além de etiquetar e normalizar ETs, identifica e anota expressões de eventos. O etiquetador do sistema utiliza uma gramática bastante abrangente para identificação de ETs e um script Perl para a normalização. A avaliação, no mesmo corpus utilizado por BALDWIN (2002), reportou precisão de 73,6% e revocação de 60,1% (medida F de 66,85%).

Outra língua para a qual há pesquisa em ETs é o espanhol, tendo SAQUETE *et al.* (2000) implementado um dos primeiros sistemas para esse idioma. Sua abordagem era baseada em uma gramática para identificação de ETs e um conjunto de regras de resolução, inclusive para ETs anafóricas. O trabalho foi avançado em (SAQUETE *et al.*, 2002) e propôs um esquema próprio de anotação baseado em *tags* XML. A avaliação foi

feita em cima de um corpus de 16 artigos de dois jornais (contendo um total de 237 ETs) e alcançou precisão e revocação de 95,59% e 82,28% respectivamente.

VICENTE-DÍEZ *et al.* (2008) demonstra uma abordagem empírica baseada em regras de resolução que detectou e normalizou corretamente 81,19% das ETs do corpus de avaliação da ACE 2007 para espanhol, além de apresentar uma proposta de tipologia de ETs. Recentemente, SAURÍ *et al.* (2010) definiu as diretivas para o estado da arte na anotação de ETs em espanhol, direcionadas para a avaliação TempEval 2010 e baseadas na TimeML (PUSTEJOVSKY *et al.*, 2005).

O chinês, em particular, é uma língua para a qual uma abordagem específica é importante, uma vez que difere significativamente dos idiomas ocidentais. É comum que sistemas para esses idiomas utilizem informações de modo e tempo dos verbos para resolução de ETs. Em chinês, isso não é possível, pois os verbos têm apenas uma única forma, independente de descreverem um evento no passado ou no futuro. LI *et al.* (2001) leva isso em consideração e, em um trabalho pioneiro, propõe estratégias para extração de informação temporal em chinês, incluindo um arcabouço de conceitos e modelos de arquitetura e implementação.

Em (LI *et al.*, 2004), os autores propõem e implementam um modelo para resolução de relações temporais em chinês, combinando conhecimento linguístico e duas abordagens aprendizado de máquina (árvore de decisão probabilística e classificador bayesiano ingênuo). Uma implementação completamente estatística de etiquetagem automática de ETs para chinês é exposta em (HACIOGLU *et al.*, 2005). Essa abordagem usa oito classificadores SVM (*one-versus-all*) e, em experimentos no corpus do TERN 2004 (466 documentos no conjunto de desenvolvimento e 256 no de teste, contendo 147 mil e 67 mil palavras respectivamente), contabilizou 83,8% de precisão e 74% de revocação (medida F de 78,6%). Os dois trabalhos também trazem análises detalhadas das contribuições de cada traço no desempenho dos classificadores utilizados.

Para o finlandês, MAKKONEN *et al.* (2003b) desenvolveu uma abordagem para extração e formalização de ETs, com o intuito de examinar a similaridade temporal entre documentos de notícias no contexto de *Topic Detection and Tracking* (TDT), uma área de pesquisa que procura detectar novos eventos numa *stream* de notícias e acompanhar todos os documentos posteriores que discutam o mesmo evento. Apesar de não dar muitos detalhes, o trabalho utiliza *parsing* sintático e transdutores de estados

finitos (um tipo de autômato de estados finitos) para o reconhecimento de ETs e uma álgebra de calendário (NING *et al.*, 2002) baseada na granularidade temporal definida por GORALWALLA *et al.* (2001) para formalização. Os autores ainda definem uma abordagem para comparação de similaridade temporal entre dois documentos. A avaliação em um corpus de 134 documentos (com 322 ETs) mostrou precisão de 98% e revocação de 91% para ETs simples e, respectivamente, 95% e 81 para ETs compostas.

BERGLUND (2004) é o pioneiro na extração de informação temporal de textos em sueco, com uma abordagem baseada em regras (para extração e normalização de ETs) e árvores de decisão (para ordenação de eventos no texto). O sistema foi avaliado para um corpus de 100 documentos de notícias de acidentes em estradas e obteve precisão de 82,4% e revocação de 95,4% (medida F de 88,4%) na detecção e 100% e 78,1% (87,7%), respectivamente, na interpretação (normalização) de ETs. O trabalho tem como objetivo detectar, normalizar e anotar, utilizando TimeML, apenas variações de ETs que são relevantes para um domínio específico (BERGLUNG *et al.*, 2006a): o de narrativa de acidentes em estradas, para uso em um programa, chamado Carsim, que as converte automaticamente em cenas animadas em 3D (BERGLUND *et al.*, 2006b).

Para o italiano, um primeiro passo foi dado em (MAGNINI *et al.*, 2006) com a criação do Italian Content Annotation Bank (I-CAB), um corpus de notícias anotado com informação semântica em diferentes níveis, sendo ETs (LAVELLI *et al.*, 2005), em particular, o primeiro nível (usando o esquema TIMEX2). Além disso, temos o recente trabalho de CASELLI *et al.* (2009), que apresenta o TETI, um etiquetador no padrão TIMEX3, para o idioma, baseado em regras e composto por quatro componentes (detector, gramática, dicionário de gatilhos e dicionários de modificadores). Esse sistema, avaliado com base em um corpus manualmente anotado de 42 artigos (contendo um total de 367 ETs), alcançou precisão de 82,95% e revocação de 90,17%, perfazendo uma medida F de 86,41%.

Além desses idiomas, há outros para os quais a pesquisa em reconhecimento e normalização ETs ainda conta com poucos trabalhos. TREUMUTH (2008) apresenta uma primeira abordagem simples baseada em regras para a normalização de ETs em estoniano e reporta resultados com precisão entre 75% e 85% e revocação entre 45% e 55%. JANG *et al.* (2004) estendeu as diretivas de anotação do TIMEX2 para coreano e implementou um etiquetador temporal, chamado KTX, baseado em uma estratégia simples, similar a BALDWIN (2002), de aprendizado a partir de um dicionário induzido

e padrões criados manualmente, tendo obtido precisão e revocação de 87% na avaliação com um corpus de 200 artigos anotados manualmente.

### 2.3.5. Expressões temporais em português

Os primeiros trabalhos que trataram de ETs em português surgiram no contexto do primeiro HAREM, que tinha a categoria TEMPO como uma das categorias de entidades mencionadas marcáveis na avaliação conjunta. As diretivas de marcação, contudo, eram bastante incipientes, limitando-se apenas a identificação e classificação (sem qualquer esforço no caminho da normalização) dos tipos DATA, HORA, PERIODO (entidade mencionada que se refere a um intervalo de tempo contínuo e não repetido, com apenas um início e um fim; por exemplo, "Inverno", "anos 80", "século XXI", etc.) e CICLICO (períodos recorrentes, como "véspera de Natal", "1º de janeiro", "Páscoa", entre outros). Na competição, dos dez sistemas participantes, apenas dois não eram capazes de reconhecer nenhum tipo de entidade da categoria TEMPO.

Em (SARMENTO, 2007), os autores descrevem a participação do sistema SIEMÊS no HAREM. Esse sistema combina regras de análise de contexto com a consulta de almanaques (com 450 mil exemplos de nomes de entidades distribuídos em 11 classes e 103 subclasses) e obteve, para a categoria TEMPO, precisão de 85,1% e revocação de 61% (medida F de 0,71) como resultados da avaliação global da classificação semântica.

O sistema MALINCHE (SOLORIO, 2007) utilizou um classificador SVM treinado em cima de um corpus anotado com o esquema BIO, usando nove traços (a própria palavra, as duas palavras anteriores e as duas seguintes, informação ortográfica, posição na frase, etiqueta POS e etiqueta BIO), e obteve precisão e revocação de 87,7% na avaliação da identificação e classificação de entidades da categoria TEMPO.

O bom resultado desses sistemas (SIEMÊS e MALINCHE) é, contudo, aparente, pois leva em consideração apenas as entidades avaliadas como corretas ou parcialmente corretas na etapa de identificação, no que foi chamado de avaliação relativa pelos organizadores da competição. Este cenário permite avaliar o desempenho dos sistemas apenas na tarefa de classificação semântica, independentemente do desempenho na tarefa de identificação (SANTOS *et al.*, 2007b). Os três trabalhos descritos a seguir fizeram uso da avaliação absoluta, levando em conta todas as entidades presentes na

coleção dourada, no cálculo de seus resultados e, por isso, não podem ser diretamente comparados aos dois supracitados.

FERRÁNDEZ *et al.* (2007) apresenta o sistema NERUA que, apesar de usar uma estratégia de votação a partir dos resultados de três classificadores (HMMs, máxima entropia e *memory-based learning*) para a maior partes das entidades detectadas e classificadas, trabalha com uma abordagem baseada em conhecimento para os tipos DATA, HORA e CICLICO, obtendo precisão de 53,58% e revocação de 66,57% (medida F de 59,37%) para a categoria TEMPO na competição.

BICK (2006) apresenta o PALAVRAS-NER, um sistema baseado em uma gramática constritiva, que trata o reconhecimento de EMs como uma tarefa integrada da etiquetagem gramatical feita pelo PALAVRAS (BICK, 2000). Apesar de, diferentemente da maioria (que utiliza abordagens estatísticas e aprendizado de máquina), esse sistema, desenvolvido inteiramente com regras escritas manualmente, teve uma das melhores performances do HAREM, tendo alcançado o primeiro lugar em diversas tarefas. Na categoria TEMPO, em particular, obteve precisão de 76,1% e revocação de 68,7% (medida F de 72,2%).

Por fim, o sistema Stencil/NooJ (MOTA *et al.*, 2007), que faz uso de uma série de recursos lingüísticos para REM (dicionários e gramáticas), construídos manualmente e organizados de modo a serem aplicados numa cadeia de processamento, obteve as melhores pontuações na identificação e classificação da categoria TEMPO: 83,24% de precisão de 74,61% de revocação (medida F de 0,7869).

Ainda com avaliação em cima do corpus do HAREM, mas fora da competição, LOUREIRO (2007) disserta sobre o reconhecimento de obras, valores, relações de parentesco e tempo, tratando, inclusive, da normalização de ETs. Sua abordagem é baseada no Xerox Incremental Parser (XIP), um analisador sintático bastante flexível e extensível por regras. Para reconhecimento das entidades supracitadas, são definidas regras no XIP, enquanto que para a normalização de ETs são usadas funções desenvolvidas na linguagem de programação Python invocadas a partir do XIP. A avaliação reportou precisão de 82,36% e revocação de 72,53% na tarefa de identificação e precisão de 90,56% e revocação de 79,74% na classificação. Os resultados da normalização de ETs foram avaliados manualmente e contabilizaram 84% de acertos, em virtude do sistema não conseguir normalizar horas e ETs coordenadas (como “dias 3 e 4 de março”). Além disso, o sistema consegue normalizar poucas datas relativas. Um

exemplo de tal deficiência é que ele não usa o tempo do verbo na normalização e, consequentemente, não ancora corretamente expressões como "Ele partiu segunda-feira" e "Ele parte segunda-feira" (aquela refere-se à segunda-feira anterior e esta, à próxima segunda-feira). Outro problema reside na ancoragem de expressões como "há três meses", pois o sistema simplesmente subtrai três meses da data do documento, resultando em um dia específico quando, muitas vezes, se quer dizer o mês (por exemplo, supondo a data do documento como 03/05/2011, "há três meses" pode significar fevereiro de 2011 e não a data de 2 de fevereiro de 2011).

No Segundo HAREM, as ETs ganharam tratamento especial, passando a ter uma pista específica dentro da competição, a pista TEMPO. Foram definidas, separadamente dos outros tipos de EMs, diretivas de anotação e normalização de ETs levando em conta os recentes avanços da literatura e direções gerais de trabalhos recentes no âmbito do processamento de ETs em textos, em particular, no contexto da TimeML e da campanha TempEval (HAGÈGE *et al.*, 2008b). Os tipos de ETs anotados foram divididos em TEMPO_CALEND (tempo calendário), DURACAO (duração), FREQUENCIA (frequência) e GENERICO (genérico). Além do atributo TIPO, a etiqueta EM deve ser preenchida, em alguns casos, também com o atributo SUBTIPO, como exemplificaremos a seguir.

As entidades do tipo TEMPO_CALEND são expressões que permitem inserir o predicado que elas modificam numa linha temporal (como um ponto ou um intervalo) e incluem as datas (absolutas ou referenciais), os intervalos e as horas, que correspondem aos subtipos DATA, HORA e INTERVALO. O subtipo DATA ainda deve ser classificado de acordo com o referencial, tendo o atributo TEMPO_REF preenchido como ABSOLUTO nas datas absolutas e como ENUNCIACAO (se o referencial for o momento de enunciação) ou TEXTUAL (se o referencial for um evento ou uma ET no próprio texto) nas datas relativas.

Para as datas relativas, ainda devem ser computados os atributos de normalização SENTIDO e VAL_DELTA. Aquele permite dar uma informação complementar que tem por finalidade a normalização de ETs referenciais, indicando se o valor temporal se situa cronologicamente antes ou depois do tempo de referência, enquanto este deve ser valorado com uma expressão que indique a distância temporal entre o tempo do evento denotado pela ET e o momento de referência.

Por fim, uma das maiores diferenças da pista TEMPO do Segundo HAREM em relação às diretivas do primeiro HAREM é a inclusão do atributo VAL_NORM, correspondente ao valor normalizado, como um primeiro passo para a normalização de ETs, estando presente, exclusivamente, para as datas absolutas, as horas e as durações.

Na pista do TEMPO, tomaram parte sete dos dez participantes do Segundo HAREM, embora sejam verificadas diferenças relativas à forma como cada um participou: seis sistemas trabalharam com o atributo TIPO; cinco sistemas tentaram preencher TIPO e SUBTIPO; dois sistemas, além de TIPO e SUBTIPO, valoraram o atributo TEMPO_REF e somente um sistema atacou a tarefa completa, incluindo a normalização (BAPTISTA *et al.*, 2008). Independentemente da disparidade no nível de granularidade com o qual essas equipes trataram as ETs, ainda assim, esta forte participação evidencia o interesse da comunidade de processamento computacional do português pelo tema.

O mais simples, do ponto de vista de ETs, desses trabalhos é o CaGe, um sistema híbrido apoiado por dicionários e regras de desambiguação (MARTINS, 2008). Apesar de seu foco ser o reconhecimento de entidades geográficas e de não ter participado da pista TEMPO oficialmente, o sistema reconhecia dois tipos de ETs: datas absolutas e nomes de períodos temporais (por exemplo, "idade média"). Sua abordagem, demasiadamente simples, consistia no uso de dicionários, almanaques e expressões regulares. Seus resultados não foram publicados discriminando a categoria TEMPO.

Outro sistema que não reportou separadamente os resultados da pista TEMPO, apesar de detectá-las, foi o da Priberam (AMARAL *et al.*, 2008). Esse sistema de REM é construído com o uso de regras contextuais e tem por base um léxico, com classificação morfossintática e semântica, ligado a uma ontologia multilíngüe, estruturada através de relações de proximidade conceitual.

CRAVEIRO *et al.* (2008) participou da pista TEMPO com PORtuguese Temporal EXpressions Tool (PorTexTO), um sistema que identifica ETs na língua portuguesa por meio de padrões de expressões, criados a partir de co-ocorrências existentes em referências temporais. O PorTexTO é dividido em dois módulos (processador de co-ocorrências e anotador) e processa os documentos frase a frase, trabalhando apenas com aquelas que contenham palavras incluídas em uma lista de palavras-chave temporais. O processador de co-ocorrências computa uma lista de

expressões onde as co-ocorrências detectadas têm uma distância máxima de n palavras antes ou depois das palavras temporais de referência. Essas expressões são agregadas e analisadas manualmente para darem origem a expressões regulares utilizadas pelo anotador para reconhecer e marcar as ETs no texto. Na competição, a melhor corrida do sistema, alcançou precisão de 68,71% de precisão e 54,7% de revocação (medida F de 0,6091) para identificação e precisão de 66,94% e revocação de 54,19% (medida F de 0,599) para classificação.

O REMBRANDT (Reconhecimento de Entidades Mencionadas Baseado em Relações e ANálise Detalhada do Texto) é um sistema de REM e de detecção de relações entre entidades (DRE) que explora intensamente a Wikipédia como fonte de conhecimento e aplica um conjunto de regras gramaticais que aproveitam os vários indícios internos e externos das EM para extrair o seu significado (CARDOSO, 2008). O tratamento de ETs, entretanto, não utiliza qualquer dado da Wikipédia, sendo completamente baseado em regras. Sua melhor corrida na CD do Segundo HAREM, para a categoria TEMPO, contabilizou 60,98% de precisão e 40,93% de revocação (medida F de 0,4899) para identificação e precisão de 59,04% e revocação de 40,3% (medida F de 0,479) para classificação.

Outro sistema participante do Segundo HAREM que faz uso da Wikipédia como fonte de conhecimento externo é o REMMA (Reconhecimento de Entidades Mencionadas do MedAlert), mas, assim como o REMBRANDT, essa informação não é utilizada pelo anotador de ETs (FERREIRA *et al.*, 2008). O anotador TEMPO do sistema utiliza uma lista de palavras pré-definidas combinadas com um conjunto de expressões regulares para identificar, classificar e anotar as ETs. Os resultados da avaliação do REMMA reportaram precisão de 47,44% e revocação de 25,38% (medida F de 0,3307) para classificação de ETs na CD do Segundo HAREM.

Por fim, HAGÈGE *et al.* (2008a) apresentaram o XIP-PT (Xerox Incremental Parser), o único trabalho que teve como objetivo realizar todas as dimensões da tarefa da pista TEMPO (incluindo a de normalização): um sistema de REM, desenvolvido numa colaboração entre o L2F (INESC-ID Lisboa) e o XRCE (Xerox Research Centre Europe, Grenoble, França), a partir do trabalho de LOUREIRO (2007), que tem como uma de suas principais características a completa integração da abordagem numa cadeia geral do processamento do português que vai da segmentação à análise e anotação morfossintática (*part-of-speech tagging* ou *POS tagging*). Trata-se de um sistema

baseado em regras de gramáticas locais (regras de reescrita ordenadas) que usam o contexto imediato à esquerda e à direita para delimitar e classificar EMs (HAGÈGE *et al.*, 2009). Essas regras trabalham em cima de unidades lexicais representadas por conjuntos de traços (atributos e valores) que explicitam a informação linguística associada a essas entradas (não só informação morfossintática, mas também de natureza semântica).

O módulo de anotação de ETs começa com o pré-processamento, que tem como saída uma lista de unidades lexicais, na forma supracitada, possivelmente ambígua. Nessa etapa, são incluídos novos traços que correspondem à informação temporal inicial (por exemplo, meses recebem traços indicando seu número de 1 a 12 e a palavra "semana" ganha um traço que indica se tratar de uma medida de tempo). Em seguida, o sistema utiliza dois módulos de desambiguação – um apenas com regras e outro misto, combinando HMMs e regras construídas manualmente – para decidir sobre as categorias das palavras (e resolver casos de ambigüidades como o da palavra "Natal", que pode significar a capital do Rio Grande do Norte ou o período festivo do ano). Depois disso, as gramáticas locais agrupam elementos lexicais, geralmente enriquecidos por nova informação relevante relativa ao tempo, para assim formar expressões temporais. A etapa seguinte verifica as dependências sintáticas para caracterizar de forma mais pormenorizada certos tipos de ETs que não podem ser classificadas com um simples contexto local. Finalmente, são efetuados cálculos numéricos externos, através de chamadas a funções na linguagem de programação Python, para normalização das ETs.

Os resultados obtidos pelo sistema na campanha de avaliação do Segundo HAREM, considerando identificação e classificação de ETs, foram de 85% de precisão e 76% de revocação (HAGÈGÈ *et al.*, 2010), comprovando ser este o sistema mais avançado para o processamento de ETs em português. Mesmo sem qualquer outro participante com o qual comparar, para a tarefa de normalização (tendo o sistema tratado somente de datas absolutas e referenciais, estas apenas parcialmente), os autores reportaram medida F de 0,74.

A partir desses números, e de acordo com (BAPTISTA *et al.*, 2008), do ponto de vista dos resultados do Segundo HAREM, é possível considerar que, de um modo geral, o patamar do estado da arte, para a classificação de ETs, se situa em valores na ordem dos 0,75 para a precisão, revocação e medida F.

# 3. Métodos e ferramentas

Este capítulo descreve os métodos e ferramentas utilizados na implementação da proposta desta dissertação, com uma breve revisão de seu uso na área. Na seção 3.1, introduzimos os conceitos de aprendizado de máquina e classificação estatística para, então, descrever os classificadores empregados neste trabalho. Na seção 3.2, apresentamos o processo de etiquetagem gramatical e descrevemos alguns tipos de etiquetadores testados na implementação do sistema CoppeTER. Por fim, na seção 3.3, detalhamos alguns dos padrões temporais já citados que estão diretamente relacionados à presente proposta.

## 3.1. Aprendizado de máquina

Aprendizado de máquina é o ramo da inteligência artificial que estuda algoritmos que permitam que computadores desenvolvam comportamentos e os evoluam com base em dados empíricos. Um dos maiores focos dessa disciplina é permitir que a máquina aprenda a reconhecer padrões complexos e, com base nos dados, tome decisões inteligentes. A definição mais famosa da área diz que, dada uma tarefa T e uma medida de desempenho P, um programa aprende a partir de uma experiência E se seu desempenho na tarefa T melhora com a experiência E (MITCHELL, 1997).

A chave do processo de aprendizado é a capacidade de generalização a partir da experiência (BISHOP, 2006). Os dados que caracterizam essa experiência, geralmente, têm origem em uma distribuição de probabilidade desconhecida. O aprendizado de máquina está alicerçado nessa capacidade de extrair algo mais geral sobre tal distribuição que permita com que um programa produza boas respostas em novos casos da tarefa em questão.

### 3.1.1. Aprendizado supervisionado

Duas grandes subáreas de aprendizado de máquina são o aprendizado supervisionado e o aprendizado não supervisionado (há, ainda, o mais recente aprendizado semi-supervisionado, mas não entraremos em detalhes aqui) que se diferenciam pela

utilização ou não de dados etiquetados (exemplos de treinamento). Neste trabalho, fazemos uso apenas de técnicas de aprendizado supervisionado.

No aprendizado supervisionado, cada exemplo de treinamento é um par composto por um objeto de entrada (quase sempre um vetor) e um valor de saída desejada. Um algoritmo de aprendizado supervisionado analisa os dados de treinamento e infere função que tenta prever o valor de saída correto para qualquer objeto de entrada válido (generalização). Dependendo do tipo de saída, chamamos essa função de classificador, se os valores de saída forem discretos, ou de função de regressão, se forem contínuos (RUSSEL *et al.*, 2002). Nesta dissertação, usamos apenas classificadores, descritos a seguir.

### 3.1.2. Classificação estatística

Classificação é a tarefa de escolher uma classe ou rótulo (valores de saída discretos, como vimos) correto para uma dada entrada. Na maioria das vezes, cada entrada é considerada isoladamente das outras entradas, e o conjunto de classes ou rótulos é conhecido a priori. Alguns exemplos de uso de problemas de classificação poderiam ser: decidir se um e-mail é spam ou não, decidir de qual categoria uma notícia trata (a partir de um conjunto de categorias pré-definido, como, por exemplo, “esportes”, “ciência”, “tecnologia”, “política” e “economia”) e decidir se uma determinada ocorrência da palavra “banco” quer dizer uma instituição financeira ou um móvel para sentar. A motivação para o uso de classificadores no sistema implementado no presente trabalho é tomar decisões como esta última.

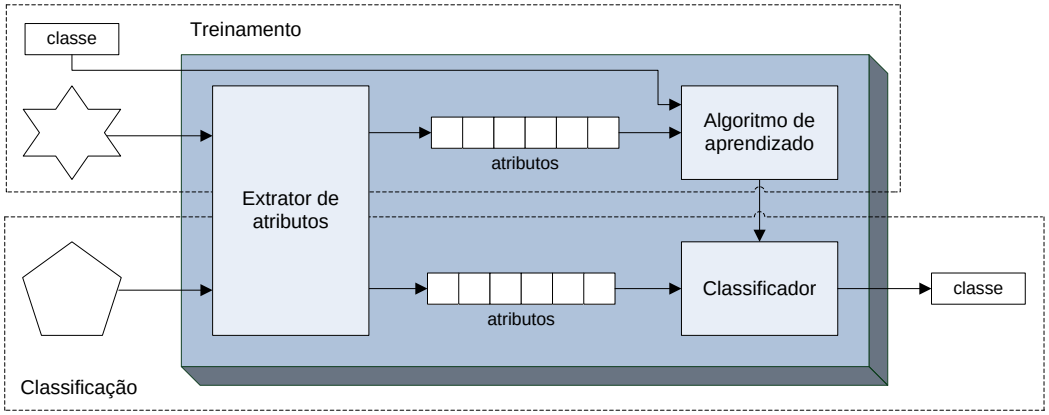



**Figura 3.1. Exemplo do processo de classificação estatística**

O processo de classificação estatística (Figura 3.1) acontece em dois momentos distintos. No primeiro deles, chamado de treinamento, um extrator de atributos é utilizado para converter cada objeto de entrada em um conjunto de atributos (*feature set*). Esses conjuntos de atributos capturam as informações básicas sobre cada entrada, que serão usadas na classificação. Os pares (conjunto de atributos, classe) são então alimentados no algoritmo de aprendizado de máquina para gerar um modelo ou uma função (o classificador). Na classificação, o segundo momento do processo, o mesmo extrator de atributos é utilizado para converter objetos de entrada desconhecidos (isto é, objetos novos, nunca vistos pelo sistema) em conjuntos de atributos, que serão alimentados no classificador para determinar a que classe o objeto pertence.

A seleção dos atributos (escolha do subconjunto de atributos mais relevantes para a tarefa de classificação em questão) e a decisão de como codificá-los para o método de aprendizado podem ter um enorme impacto no desempenho do classificador. Tipicamente, esse é um processo de tentativa e erro, guiado pela intuição do que parece ser mais relevante para o problema. À primeira vista, pode parecer uma boa ideia usar o máximo de atributos que se consiga extrair dos objetos de entrada, contudo, esse tipo de abordagem frequentemente resulta no problema da superadaptação (*overfitting*), em que o classificador perde sua essencial capacidade de generalização.

Nas próximas seções, detalharemos os dois tipos de classificadores empregados nesta dissertação: o classificador bayesiano ingênuo (*naive Bayes classifier*) e o classificador de máxima entropia (*maximum entropy classifier*).

#### 3.1.2.1. Classificador bayesiano ingênuo

O classificador bayesiano ingênuo é um classificador probabilístico simples, baseado na aplicação do Teorema de Bayes, que adota um modelo de probabilidade com atributos independentes. Em outras palavras, este classificador supõe que a presença (ou a ausência) de um atributo de uma classe não é relacionada à presença (ou à ausência) de qualquer outro atributo. O termo "ingênuo" vem dessa suposição.

Para escolher uma classe para um objeto de entrada, o classificador de Bayes ingênuo, primeiramente, calcula a probabilidade a priori de cada classe (que é determinada pelo simples cálculo da frequência de cada classe no conjunto de treinamento). Em seguida, cada atributo contribui com a estimativa de probabilidade de cada classe ao ser multiplicado pela probabilidade de que os valores de entrada daquela

classe tenham aquele atributo. A classe cuja estimativa de probabilidade for a maior (máxima probabilidade) é atribuída ao objeto de entrada como resultado da classificação.

De modo mais formal, dada uma classe C e um conjunto de atributos A, o classificador escolhe o maior valor de $P(C \mid A)$, que é a probabilidade de um objeto de entrada pertencer à classe C, dado que é representado pelo conjunto de atributos A. Sendo, por definição, $P(C \mid A) = P(A,C)/P(A)$, note que o denominador $P(A)$ será o mesmo para todas as classes e, portanto, se estamos interessados apenas em escolher a classe de maior probabilidade, podemos desprezá-lo, ou seja, é suficiente calcularmos apenas $P(A,C)$. A estimativa de probabilidade de cada classe pode, então, ser expandida como a probabilidade da classe multiplicada pela probabilidade do conjunto de atributos dada a classe: $P(A,C) = P(C) \times P(A \mid C)$. Finalmente, como os atributos são, pela suposição do modelo, independentes uns dos outros (dada a classe), podemos separar a probabilidade de cada atributo individual: $P(A,C) = P(C) \times \prod_{a \in A} P(a \mid C)$. Essa é exatamente a equação que descrevemos informalmente no parágrafo anterior: $P(C)$ é a probabilidade a priori de uma classe C, enquanto $P(a \mid C)$ é a contribuição individual de cada atributo para a estimativa de probabilidade da classe.

A suposição ingênua de Bayes, contudo, não é realista, uma vez que atributos comumente apresentam, quando não forte, pelo menos, alguma dependência (BIRD *et al.*, 2009). O efeito nocivo disso é exemplificado pelo problema da dupla contagem (*double-couting*): o modelo pode acabar contando duas vezes atributos com forte dependência entre si, o que é, efetivamente, ainda que de maneira indesejada, o mesmo que dar peso 2 para a característica do objeto de entrada representada por esses atributos. Por não ter que se preocupar com como os atributos interagem entre si, este classificador pode ser treinado de forma muito eficiente.

### 3.1.2.2. Classificador de máxima entropia

A máxima entropia é uma técnica de estimativa de distribuição de probabilidade amplamente usada para uma variedade de tarefas de processamento de linguagem natural, entre elas, modelagem de linguagem, etiquetagem *part-of-speech*, segmentação de textos e desambiguação de sentido de palavras (*word sense desambiguation*).

O modelo de classificador de máxima entropia generaliza o modelo usado pelo classificador bayesiano ingênuo, sendo uma alternativa comum a este, pois não assume independência estatística entre os atributos. Contudo, o processo de treinamento desse tipo de modelo pode ser significativamente mais lento e, por isso, ele nem sempre é apropriado quando há um grande número de classes a aprender.

Sendo entropia uma medida de incerteza, pode soar estranha a ideia de se buscar maximizá-la, uma vez o que um classificador estatístico deveria supostamente fazer, em última análise, é minimizá-la. O princípio por trás da máxima entropia, no entanto, é que, sem conhecimento externo (i.e. quando nada mais é sabido), a distribuição deve ser tão uniforme quanto possível, ou seja, deve ter entropia máxima. Restrições na distribuição, derivadas dos dados de treinamento, informam ao método onde ser minimamente não-uniforme. Essa formulação da máxima entropia tem uma solução única que pode ser encontrada pelo algoritmo de dimensionamento iterativo melhorado (*improved iterative scaling* ou IIS), isto é, esse algoritmo encontra a distribuição de entropia máxima que é consistente com as restrições dadas (NIGAM *et al.*,1999).

Conforme dito acima, neste modelo, os dados de treinamento são usados para definir um conjunto de restrições sobre a distribuição condicional. Cada restrição expressa uma característica dos dados de treinamento que também deve estar presente na distribuição aprendida. Mais formalmente, supondo que um atributo é dado por uma função $f_i(A,C)$, onde A é um objeto de entrada e C é uma classe, a máxima entropia nos permite restringir o modelo de distribuição de tal forma que ele tenha o mesmo valor esperado para um atributo que este teria no conjunto de dados, que significa dizer que ele deve ter a propriedade: $\frac{1}{|D|}\sum_{A\in D} f_i(A,C(A)) = \sum_A P(A)\sum_C P(C \mid A) f_i(A,C)$. Na prática, como a distribuição de objetos de entrada $P(A)$ é desconhecida, não há interesse em modelá-la, então, usam-se os dados de treinamento, sem as classes, como uma aproximação dessa distribuição: $\frac{1}{|D|}\sum_{A\in D} f_i(A,C(A)) = \frac{1}{|D|}\sum_{A\in D}\sum_C P(C \mid A) f_i(A,C)$.

Ao usar a máxima entropia, o primeiro passo é identificar um conjunto de funções de atributo que sejam relevantes na classificação dos objetos de entrada. Daí, para cada atributo, deve-se medir o valor esperado em cima dos dados de treinamento e usar isso como uma restrição na distribuição do modelo. Quando as restrições são construídas dessa forma, é garantida a existência de uma distribuição única que tenha

entropia máxima e que seja da seguinte forma exponencial (DELLA PIETRA *et al.*, 1997): $P(C \mid A) = \frac{1}{Z(A)} \exp\left( \sum_i \lambda_i f_i(A,C) \right)$, onde cada $f_i(A,C)$ é um atributo, $\lambda_i$ é um parâmetro a ser estimado e $Z(A)$ é simplesmente um fator de normalização para garantir uma probabilidade apropriada: $Z(A) = \sum_C \exp\left( \sum_i \lambda_i f_i(A,C) \right)$. Além disso, também é garantido que haja apenas um máximo global e nenhum máximo local, o que nos permite, por exemplo, usar um algoritmo de subida de encosta (*hillclimbing*) para encontrar a solução de entropia máxima que atende às restrições supracitadas, que é exatamente o papel do algoritmo de dimensionamento iterativo melhorado (IIS), citado no início desta seção. O IIS é usado para estimar os parâmetros indicados na forma exponencial acima, encontrado conjuntos com estimativa cada vez maior a cada iteração, até chegar à solução final. Uma descrição pormenorizada do algoritmo pode ser consultada em (DELLA PIETRA *et al.*, 1997).

### 3.1.3. Implementação utilizada

Na implementação do sistema proposto no próximo capítulo, optamos por usar um classificador de máxima entropia, em vez de um classificador de Bayes ingênuo, em virtude, obviamente, de não podermos assumir que os atributos utilizados são independentes entre si.

A escolha do modelo de máxima entropia, em detrimento de outros modelos de classificação estatística, está embasada na tese de RATNAPARKHI (1998), que demonstra que uma série de importantes tipos de ambiguidades encontradas em linguagem natural pode ser resolvida com precisão próxima, igual ou superior ao estado da arte, usando apenas esta técnica de modelagem estatística (não é incomum encontrar trabalhos que usam mais de um classificador e um processo de votação para alcançar alta precisão). Segundo o autor, os poucos métodos que apresentam desempenho ligeiramente superior aos apresentados nesse trabalho, em geral, demandam muito mais supervisão, na forma de intervenção humana ou recursos adicionais.

Reforçando esta escolha, também citamos o trabalho de AHN *et al.* (2005), que empregou, com sucesso, o modelo de máxima entropia para desambiguar expressões temporais relativas, exatamente o objetivo do uso da classificação estatística neste trabalho.

Aproveitando a escolha do arcabouço de processamento temporal NLTK (que apresentaremos no Capítulo 5), utilizamos, em nossa implementação, o modelo de máxima entropia disponibilizado pelo *framework*. Apesar das várias opções de algoritmo de estimativa de parâmetros, optamos por utilizar o tradicional IIS.

Um aspecto que precisamos levar em conta na implementação e sobre o qual ainda não falamos é a obtenção das funções de atributos $f_i(A,C)$. Cabe neste momento, contudo, fazer um esclarecimento de terminologia. Em modelos de máxima entropia, o termo *atributo* pode se referir a dois conceitos, que podem causar confusão se não forem previamente distinguidos: chamamos de atributo de entrada (*input-feature)* ou contexto (*context*) o atributo de um objeto de entrada e de atributo associado (*joint-feature*), a combinação de classes e atributos que recebe um parâmetro próprio e que é, de fato, utilizada no modelo. Esses atributos associados devem ter valores numéricos.

$$f_i(A,C) = \begin{cases} 1, \text{ se } v = A_i \text{ e } c = C \\ 0, \text{ caso contrário} \end{cases}$$

**Figura 3.2. Função de mapeamento entre atributos de entrada e atributos associados**

Para fazer o mapeamento entre quaisquer tipos de atributos para valores numéricos, usamos a seguinte função da Figura 3.2, onde $A$ é o objeto de entrada e $C$ é a classe. Note que esse método cria um atributo de entrada para cada combinação de atributo de entrada ($A_i$), valor do atributo ($v$) e classe ($c$) que ocorra pelo menos uma vez no conjunto de treinamento. Vale observar que os vetores de atributos associados são, consequentemente, bastante esparsos, e, por isso, a implementação do NLTK usa listas para poupar espaço de memória.

# 3.2. Etiquetagem gramatical

Em linguística computacional, a etiquetagem gramatical (*part-of-speech tagging*), mais conhecida como etiquetagem *part-of-speech* (ou ainda, simplesmente, etiquetagem POS), é o processo de marcar as palavras de um texto (corpus) com uma determinada etiqueta gramatical, baseado tanto na sua definição quanto no seu contexto (o relacionamento com palavras adjacentes na frase). Tais etiquetas também são conhecidas como classes de palavras, classes morfológicas ou categorias lexicais (MANNING *et al.*, 1999). Esse processo de atribuição de classes de palavras é

fundamental para muitas das aplicações de PLN, tais como análise sintática e semântica (*parsing*), análise do discurso e processamento de fala (JURAFSKY *et al.*, 2000).

Cada corpus etiquetado (para treinamento ou análise, por exemplo) ou etiquetador adota um ou mais conjuntos de etiquetas, que são coleções finitas que definem quais são as etiquetas usadas. A Tabela 3.1 abaixo apresenta um exemplo de um possível conjunto de etiquetas. Esses conjuntos podem ser mais ou menos granulares: enquanto o conjunto de etiquetas usado pelo PALAVRAS (BICK, 2000), tido como provavelmente o melhor etiquetador para a língua portuguesa, é composto por apenas 18 etiquetas, o corpus do Penn Treebank e o corpus Brown utilizam, respectivamente, 45 e 87 etiquetas.

**Tabela 3.1. Exemplo de conjunto de etiquetas**

| **Etiqueta** | **Significado** | **Exemplos** |
|---|---|---|
| ADJ | Adjetivo | novo, velho, bom, ... |
| ADV | Advérbio | bastante, lentamente, .. |
| ART | Artigo | um, uma, a, o, ... |
| CNJ | Conjunção | se, ou, que, e, mas, ... |
| S | Substantivo | casa, carro, árvore, ... |
| P | Preposição | após, até, desde, ... |
| V | Verbo | comer, beber, dançar, ... |

A princípio, pode parecer que é suficiente ter um enorme dicionário com a etiqueta gramatical correspondente a cada palavra. Contudo, conforme aprendemos na escola e apontamos acima, a etiqueta gramatical não é baseada apenas na definição do termo, mas também no seu contexto, como ilustrado nos dois exemplos da Tabela 3.2. Observe que o termo "casa" é um verbo na primeira sentença, enquanto que, no segundo exemplo, é um substantivo. Apesar de não encontrarmos essas ambiguidades em linguagens artificiais, em linguagens naturais, elas não são nada raras, e, sem o conhecimento do contexto (isto é, as palavras adjacentes), não seria possível fazer a devida distinção.

**Tabela 3.2. Exemplos de frases etiquetadas gramaticalmente**

| **Palavra** | *Em* | *novembro* | , | *Maurício* | *se* | *casa* | *com* | *Lygia* | . |
|---|---|---|---|---|---|---|---|---|---|
| **Etiqueta** | P | S | , | S | CNJ | V | P | S | . |
| **Palavra** | *Em* | *dezembro* | , | *Maurício* | *comprará* | *uma* | *casa* | *nova* | . |
| **Etiqueta** | P | S | , | S | V | ART | S | ADJ | . |

Os primeiros algoritmos de etiquetagem gramatical eram baseados em uma arquitetura em dois estágios: no primeiro, se atribuía uma lista das etiquetas possíveis

para cada palavra usando um dicionário para, então, no segundo estágio, a partir de uma grande lista de regras de desambiguação escritas manualmente, reduzir cada uma dessas listas a uma única etiqueta.

A maioria dos algoritmos de etiquetagem gramatical automática se encaixa em um de dois grandes grupos: os baseados em regras e os estocásticos (também chamados de probabilísticos). Os etiquetadores gramaticais baseados em regras são geralmente compostos por uma grande base de regras de desambiguação especificando, por exemplo, que uma palavra ambígua é um substantivo em vez de um verbo se ela é precedida por um artigo. Etiquetadores estocásticos resolvem ambiguidades usando um corpus de treinamento para calcular a probabilidade de uma determinada palavra ter determinada etiqueta num determinado contexto. Um terceiro tipo de etiquetador gramatical amplamente usado são os etiquetadores baseados em transformação, que compartilham características de ambos os anteriores e têm como representante mais famoso o etiquetador Brill (BRILL, 1995).

### 3.2.1. Etiquetadores estocásticos

O etiquetador probabilístico mais trivial que se pode construir é um que simplesmente atribua a etiqueta mais provável do corpus (isto é, a de maior frequência) a todas as palavras. Evoluindo essa ideia, um etiquetador um pouco menos trivial pode, para cada termo, atribuir a etiqueta mais provável para ele. Este etiquetador é a definição do mais simples da classe de etiquetadores de n-gramas, o etiquetador de unigramas, que utiliza apenas um item de contexto (que, na realidade, significa não levar em conta contexto algum, uma vez que só utiliza o termo em análise).

Um etiquetador de n-gramas utiliza um contexto composto pela palavra em análise e pelas etiquetas dos n-1 termos anteriores. A Figura 3.3 ilustra o contexto (as células destacadas de cinza) usado por um etiquetador de trigramas (ou 3-gramas, isto é, um etiquetador de n-gramas com n = 3), onde $p_n$ é a palavra atual para a qual estamos tentando atribuir a etiqueta $e_n$. Neste caso, o etiquetador n-grama escolhe a etiqueta mais provável, dado o contexto.

| | | | | |
|---|---|---|---|---|
| Palavras: | $p_{n-2}$ | $p_{n-1}$ | $p_n$ | $p_{n+1}$ |
| Etiquetas: | $e_{n-2}$ | $e_{n-1}$ | $e_n$ ? | $e_{n+1}$ ? |

**Figura 3.3. Contexto de um etiquetador de 3-gramas**

As abordagens mais poderosas de etiquetadores estocásticos usam métodos de classificação estatística (como os listados na seção 3.1.2) utilizando como atributos, além do contexto descrito acima, outras características das palavras como capitalização, sufixos de diferentes tamanhos, formas normalizadas, etc. Entre eles, os baseados em modelos ocultos de Markov (*hidden Markov models* ou HMMs) são especialmente populares e bem-sucedidos nesta tarefa de etiquetagem gramatical.

### 3.2.2. Etiquetadores baseados em transformação

A ideia da etiquetagem gramatical baseada em transformação é, depois de atribuída uma etiqueta para cada palavra da sentença, voltar e consertar os erros. Deste modo, o etiquetador sucessivamente melhora a etiquetagem. Isso é feito compilando uma lista de regras de correção transformacionais.

As regras geradas a partir do corpus de treinamento, geralmente, são da forma "troque a etiqueta $e_1$ pela etiqueta $e_2$ no contexto $C$". Contextos, neste caso, podem ser, por exemplo, a palavra anterior (ou seguinte), a sua etiqueta ou, até, a simples existência de uma determina etiqueta a duas ou três palavras da palavra atual.

Na fase de treinamento, o etiquetador gera milhares de regras candidatas, e cada uma delas é pontuada de acordo com a melhoria líquida que proporciona: o número de etiquetas incorretas que a regra corrige menos o número de etiquetas corretas que ela incorretamente modifica. Uma propriedade interessante de etiquetadores desse tipo é que as regras geradas são passíveis de interpretação linguística.

Como dissemos anteriormente, o etiquetador Brill (BRILL, 1995) foi o primeiro a utilizar esta abordagem, sendo praticamente sinônimo de etiquetador baseado em transformação até hoje.

## 3.3. Padrões temporais

Chamamos de padrões temporais quaisquer esquemas de representação, marcação ou anotação de dados com conteúdo temporal. Nesta seção, detalharemos os padrões temporais utilizados direta ou indiretamente neste trabalho, quais sejam: o ISO 8601 (o padrão mais utilizado para representação de datas e horas), o TIMEX2 (o esquema mais usado para anotação de expressões temporais) e o padrão proposto e avaliado no Segundo HAREM para ETs em português.

### 3.3.1. ISO 8601

A ISO 8601 (“ISO 8601: Data elements and interchange formats — Information interchange — Representation of dates and times”) é uma norma internacional da Organização Internacional para Padronização (International Organization for Standardization ou ISO) para representação e troca de dados relacionados a datas e horas publicada inicialmente em 1988 e revisado pela última vez em 2004. Esse padrão unificou e substituiu uma série de outras normas mais antigas da ISO, em particular, a ISO 2014, famosa por ter introduzido originalmente a representação numérica *big-endian* (explicada abaixo) de datas (“AAAA-MM-DD”). O objetivo da ISO 8601 é fornecer um método bem-definido e sem ambiguidade para representação de datas e horas, de forma a evitar interpretações erradas quando esses tipos de dados são transferidos entre países com diferentes convenções numéricas.

O princípio da norma é que o componente temporal mais significativo (o ano) apareça primeiro na cadeira de dados (*big-endian*), progredindo de forma decrescente até o menor deles (a fração de segundo), passando por dia, hora, minuto e segundo. Esse tipo de ordenação numérica para representação de dados é chamado de *big-endian*. O método também trata da comunicação de informação temporal em diferentes fusos horários através da distância do Tempo Universal Coordenado (*Coordinated Universal Time* ou UTC), o padrão de tempo que regula relógios e horários em todo o mundo. Entre os benefícios de se usar essa representação estão: facilidade de escrita e leitura por sistemas computacionais, facilidade de comparação e ordenação e independência de país, idioma e linguagem de programação.

A ISO 8601 utiliza o calendário gregoriano, o padrão internacional de calendário civil. Os anos devem ser representados com quatro dígitos (“AAAA”), para evitar problemas como o do *bug* do milênio no ano 2000. As datas devem ser representadas no formato “AAAA-MM-DD”, “AAAA-MM” ou “AAAAMMDD”, onde “MM” corresponde ao número do mês de 01 a 12 e “DD”, ao dia de 01 a 31. Semanas são representadas nos formatos “AAAA-WSS”, “AAAAWSS”, “AAAA-WSS-D” ou “AAAAWSSD”, onde “SS” refere-se ao número da semana no ano entre W01 e W53 e “D” é o número do dia da semana entre 1 (segunda-feira) e 7 (domingo).

Para horas, o padrão usa um sistema de relógio de 24 horas nos formatos “hh:mm:ss”, “hhmmss”, “hh:mm”, “hhmm ou “hh”, onde “hh” refere-se a hora entre 00 e 24, “mm” corresponde aos minutos entre 00 e 59 e “ss”, aos segundos entre 00 e 59. O

fuso horário pode ser especificado nos formatos “<hora>Z”, “<hora>±hh:mm”, “<hora>±hhmm” ou “<hora>±hh”, onde “hh” e “mm” referem-se às horas e minutos de deslocamento a partir do UTC. O primeiro formato indica que o horário está no UTC (zero de deslocamento). Na ausência dessas informações, é assumido que está sendo representada a hora local. A combinação de datas e horas é feita no formato “<data>T<hora>”, onde “<data>” e “<hora>” podem ser, respectivamente, quaisquer datas ou horas construídas seguindo as diretivas supracitadas.

A norma também prevê a representação de durações nos formatos “PnYnMnDTnHnMnS”, “PnW” ou “P<data>T<hora>”, onde “n” é substituído pelo valor dos elementos de data e hora – representados pelas letras Y (anos), M (meses), W (semanas), D (dias), H (horas), M (minutos) e S (segundos) – que o precedem, e “<data>” e “<hora>”, por quaisquer datas ou horas dentro das respectivas regras, conforme indicado acima.

**Tabela 3.3. Exemplos de aplicação da norma ISO 8601**

| **Exemplo** | **Representação de acordo com a ISO 8601** |
|---|---|
| 3 de janeiro de 2010<br>(ou o domingo da última semana de 2009) | 2010-01-03 ou 20100103<br>(ou 2009-W53-7 ou 2009W537) |
| Março de 2008 | 2008-04 |
| Terceira semana de 2001 | 2001-W03 ou 2001W03 |
| Seis da manhã | 06:00:00 ou 060000 ou 06:00 ou 0600 ou 06 |
| Seis da tarde, horário de Brasília | 18:00-03 (ou 1800-03:00 ou 18-0300 ou ... ) |
| Sete e meia da noite do dia 3 de março de 1988 | 1988-03-03T1930 ou 19880303T19:30:00 ou ... |
| Quatro anos | P4Y (ou P0004-00-00T00 ou ...) |
| Dois meses, cinco dias e trinta horas | P2M5DT30H (ou P0000-02-06T06:00 ou ...) |
| Sete semanas | P7W |

A Tabela 3.3 demonstra o uso do padrão através de diversos exemplos. A ISO 8601 ainda descreve métodos para representação de datas ordinais e outros tipos de entidades temporais. Esses, entretanto, estão fora do escopo desta dissertação.

### 3.3.2. TIMEX2

Como mencionado anteriormente, o padrão TIMEX2 foi desenvolvido no âmbito do TIDES e evoluiu ao longo de uma série de versões (2001, 2003 e duas em 2005), sendo a última a versão 1.1 de setembro de 2005 (FERRO *et al.*, 2005). O TIMEX2 é construído em cima da norma ISO 8601 descrita na seção anterior.

Nesse esquema, as ETs são anotadas usando uma etiqueta XML (*eXtensible Markup Language*) para englobar o texto que a determina. No início da expressão,

portanto, é inserida a etiqueta <TIMEX2> ao passo que, no final, é colocada a *tag* </TIMEX2> (a mesma etiqueta só que de fechamento, com uma barra). Por exemplo, a ET da frase "Viajo em dezembro" seria anotada (por ora, sem atributos) como "Viajo em <TIMEX2>setembro</TIMEX2>". As *tags* TIMEX2 podem contar com um ou mais atributos, quais sejam:

**Tabela 3.4. Atributos possíveis das etiquetas TIMEX2**

| **Atributo** | **Função** | **Exemplo** |
|---|---|---|
| VAL | Contém a forma normalizada da data/hora da ET anotada. | VAL="1985-09-13" |
| MOD | Captura os modificadores temporais, com valores padronizados como BEFORE (antes), MORE_THAN (mais que), START (início) e APPROX (aproximadamente). | MOD="APPROX" |
| ANCHOR_VAL | Contém a forma normalizada da data/hora que ancora uma ET relativa. | ANCHOR_VAL="1985-09-11" |
| ANCHOR_DIR | Captura a direção/orientação relativa entre os atributos VAL e ANCHOR_VAL, com valores padronizados como WITHIN (dentro de), STARTING (a partir de) e BEFORE (antes de). | ANCHOR_DIR="BEFORE" |
| SET | Identifica expressões que denotam conjuntos de tempos. | SET="YES" |
| COMMENT | Contém quaisquer comentários que o anotador queira adicionar. | COMMENT="ET ambígua" |

### 3.3.3. Segundo HAREM

A proposta de anotação de expressões temporais do Segundo HAREM foi desenvolvida em 2008 para passar a considerar a normalização de ETs no padrão de anotação do primeiro HAREM. Esta proposta considera como ETs, expressões que, semanticamente, denotam: um momento no calendário (ponto ou intervalo), uma quantificação temporal (duração), uma repetição de eventos no tempo ou, ainda, o emprego genérico de algumas dessas expressões, geralmente associadas à noção de tempo (HAGÈGE *et al.*, 2008b).

No padrão, considera-se que a totalidade da expressão temporal deve ser delimitada entre as etiquetas <EM ID=... CATEG="TEMPO"> e </EM>, incluindo a preposição que a introduzir, no caso da expressão temporal ser um sintagma preposicional (e.g. "no ano passado"), ou o determinante no caso de ser um sintagma nominal (e.g. "dois dias depois"). O atributo TIPO é o único atributo obrigatório do elemento EM de categoria TEMPO. Os diferentes valores do atributo TIPO são:

TEMPO_CALEND (tempo calendário), DURACAO (duração), FREQUENCIA (frequência) e GENERICO (genérico). Para o tipo TEMPO_CALEND, ainda são admitidos os subtipos DATA, HORA e INTERVALO.

| **Atributo** | **Função** | **Exemplo** |
|---|---|---|
| CATEG | Indica a categoria da entidade mencionada. Para ETs, deve ser sempre preenchido com TEMPO. | CATEG=“TEMPO” |
| TIPO | Indica o tipo da entidade mencionada dentro da categoria. Para a categoria TEMPO, são aceitos os valores TEMPO_CALEND, DURACAO, FREQUENCIA e GENERICO. | TIPO=“TEMPO_CALEND” |
| SUBTIPO | Indica o subtipo da entidade mencionada dentro do tipo. Usado apenas para o tipo TEMPO_CALEND. Pode ter os valores DATA, HORA e INTERVALO. | SUBTIPO=“DATA” |
| TEMPO_REF | Indica o referencial temporal da ET anotada. No caso de datas absolutas, o valor deve ser ABSOLUTO. No caso de datas relativas, o valor deve ser ENUNCIACAO (quando a ET for relativa ao momento da enunciação) ou TEXTUAL (no caso de referência textual). | TEMPO_REF=“TEXTUAL” |
| SENTIDO | Indica se a ET se situa cronologicamente antes ou depois do tempo de referência. Os possíveis valores do atributo SENTIDO são ANTERIOR, POSTERIOR e SIMULT. | SENTIDO=“ANTERIOR” |
| VAL_DELTA | Indica a distância temporal entre o tempo do evento denotado pela ET e o momento de referência. . | VAL_DELTA=“A0M0S2D0H0M0S0” |
| VAL_NORM | Indica o valor normalizado de datas, horas e durações absolutas. | VAL_NORM=“+19850911T----E—LM-“ |

Os valores possíveis de VAL_DELTA são representados da maneira seguinte: “AnMnSnDnHnMnSn”, onde “n” é substituído pelo valor dos elementos temporais – representados pelas letras A (anos), M (meses), S (semanas), D (dias), H (horas), M (minutos) e S (segundos) – que o precedem.

O atributo VAL_NORM é atribuído a algumas entidades TEMPO, sendo apenas um primeiro passo para a normalização de expressões temporais. Este atributo deve estar presente exclusivamente para as datas, horas e durações (apenas absolutas). Seu formato é “<era><ano><mes><dia>T<hora><min>E<est>LM<lim>”, onde: <era> corresponde a um caractere (“+” ou “-“) conforme a data seja depois ou antes da nossa era (considerando o calendário gregoriano); <ano> corresponde a quatro caracteres dígitos que representam o valor do ano; <mes> corresponde a dois dígitos que

representam o valor do mês; <dia> corresponde a dois dígitos que representam o valor do dia; <hora> corresponde a dois dígitos que representam o valor da hora; <min> corresponde a dois dígitos que representam o valor dos minutos; e <est> corresponde a duas letras capitalizadas correspondente às estações do ano (“IN” para inverno, “PR” para primavera, “VE” para verão e “OU” para outono).

O campo <lim> indica se a expressão normalizada introduz um intervalo de tempo com limite anterior ou limite posterior não determinado (em aberto), sendo seus respectivos valores “A”, “P” e “-”, este quando a data não corresponde a um intervalo com um dos limites aberto. Finalmente, para expressões de tipo DURACAO, o valor do atributo VAL_NORM corresponde ao valor utilizado para VAL_DELTA e exprime uma distância temporal.

# 4. Proposta

A proposta desta dissertação está dividida em três partes. Na primeira, apresentamos as extensões ao padrão ISO 8601 para captura de diversas unidades temporais não previstas na especificação original. A segunda parte apresenta um esquema de anotação que combina os padrões TIMEX2 e HAREM, levando em conta os avanços recentes na área e aproveitando o que há de melhor nos dois, aproximando mais a captura de ETs em português das linhas de investigação já estabelecidas internacionalmente. A terceira descreve a arquitetura do etiquetador, baseado no padrão proposto, implementado nesta dissertação.

## 4.1. Extensões ao ISO 8601

Antes de detalharmos o formato proposto para anotação de expressões temporais, apresentaremos as extensões introduzidas aos padrões do ISO 8601. A maioria delas tem como base as especificações do TIMEX2.

### 4.1.1. Unidades não especificadas

Apesar de o ISO 8601 permitir a omissão de valores nas representações de data e hora a partir da esquerda (truncando) ou da direita (reduzindo a precisão), não há uma maneira padronizada para representar lacunas nos componentes do valor representado. Assim como introduzido no padrão TIMEX2, utilizamos o caractere “X” para preencher as posições das unidades não especificadas, desconhecidas ou não aplicáveis à determinada expressão temporal. A Tabela 4.1 traz alguns exemplos de uso dessa extensão, que é especialmente útil na representação de durações, frequências, ETs imprecisas e ETs genéricas, conforme descrito mais adiante.

**Tabela 4.1. Exemplo de unidades não especificadas**

| **Exemplo** | **Representação** |
|---|---|
| Setembro | XXXX-09 |
| Domingo | XXXX-WXX-7 |
| Dia 1º | XXXX-XX-01 |
| 25 de dezembro | XXXX-12-25 |

### 4.1.2. Décadas, séculos e milênios

As especificações do ISO 8601 não fornecem meios para a captura de períodos como décadas, séculos e milênios. Embora, a exemplo do proposto no Segundo HAREM, sejamos capazes de representar tais durações em seus equivalentes em anos (“P10Y” para uma década, “P200Y” para dois séculos, etc.), fazê-lo implica em perda de semântica que pode ser importante para um processamento posterior. Como exemplo, quando falamos que algo aconteceu *há vinte anos*, a probabilidade dessa ET significar vinte anos exatos é significativamente superior do que quando falamos que algo aconteceu *há duas décadas*. Por isso, defendemos que preservar a semântica dessa imprecisão, introduzida pelo uso de ETs de granularidade maior, é importante para sistemas de processamento temporal que venham a tirar proveito das informações anotadas. Essa extensão consiste na inclusão de três unidades na ISO 8601 para representar durações em décadas (DE), séculos (CE, de *century* em inglês, seguindo o padrão do TIMEX2) e milênios (ML).

| **Exemplo** | **Representação** |
|---|---|
| Duas décadas | P2DE |
| Um século | P1CE |
| Meio século | P0,5CE |
| Um milênio | P1ML |

| **Exemplo** | **Representação** |
|---|---|
| Década de 80 | 198 |
| Século XX | 19 |
| Século IX | 09 |
| Terceiro milênio | 2 |

**Figura 4.1. Exemplos de durações e datas com décadas, séculos e milênios**

Para a representação de datas expressas nessas granularidades, seguimos a diretiva do TIMEX2 e usamos apenas os algarismos das casas correspondentes: apenas o algarismo da casa dos milhares para milênios (1985), até o algarismo das centenas para séculos (1985) e até o algarismo das dezenas para décadas (1985). A Figura 4.1 elenca alguns exemplos de durações e datas envolvendo os períodos em questão.

### 4.1.3. Semestres, quadrimestres, trimestres e bimestres

Essas frações do ano são muito comuns em textos na língua portuguesa e, tal qual justificamos acima, pensamos ser essencial capturar a semântica expressa em seu uso, em vez de simplesmente traduzi-las no intervalo de meses correspondente no calendário. Para isso, aproveitamos os *tokens* introduzidos no TIMEX2 para semestres (H1, H2 e HX) e trimestres (Q1, Q2, Q3, Q4 e QX) e acrescentamos outros para quadrimestres (U1, U2, U3 e UX) e bimestres (B1, B2, B3, B4, B5, B6 e BX). Os

*tokens* que têm um algarismo como segunda letra indicam a ordem no ano: H1 representa o primeiro semestre, assim como Q4 aponta para o quarto trimestre. Seu uso deve se dar na posição do mês no formato padrão de datas do ISO 8601. Essas frações do ano também podem ser usadas em expressões de duração e, para representá-las, utilizamos os *tokens* terminados com a letra "X" dentro do formato ISO 8601 para durações: "três semestres" é, portanto, valorado como "P3QX". Outros exemplos são mostrados abaixo na Tabela 4.2.

**Tabela 4.2. Exemplos de fraçoes do ano**

| Exemplo | Representação |
|---|---|
| Segundo semestre de 1985 | 1985-H2 |
| Primeiro trimestre de 2008 | 2008-Q1 |
| Terceiro quadrimestre de 1994 | 1994-U3 |
| Quinto bimestre de 1998 | 1998-B5 |
| Um semestre | P1HX |
| Dois trimestres | P2QX |
| Três bimestres | P3BX |

### 4.1.4. Estações do ano

Ainda que não sejam tão comuns no português do Brasil, em virtude da pouca variabilidade climática durante as diferentes estações do ano na maior parte do país, expressões temporais que fazem referência às estações do ano são perfeitamente ancoráveis na linha do tempo e, por isso, adotamos extensões para registrá-las, uma vez que não há qualquer previsão nesse sentido no padrão ISO 8601. Seguindo a linha do TIMEX2, fazemos uso de quatro *tokens* (SP, SU, FA e WI) na posição do mês dentro do formato de representação de datas para indicar, respectivamente, a primavera (*spring*), o verão (*summer*), o outono (*fall*) e o inverno (*winter*), conforme os exemplos da Tabela 4.3.

**Tabela 4.3. Exemplos de estações do ano**

| Exemplo | Representação |
|---|---|
| Verão de 1969 | 1969-SU |
| Sempre viajo *no outono* | XXXX-FA |
| Primavera de 1985 | 1985-SP |
| Adoro o *inverno* | XXXX-WI |

### 4.1.5. Fins de semana

Da mesma forma que escolhemos capturar a semântica por trás de expressões como décadas, semestres e estações do ano, também o faremos para ETs que especifiquem explicitamente o fim de semana, que, dependendo da percepção, pode começar na noite de sexta-feira ou no sábado, se estendendo até o fim do domingo. Compreendemos que o entendimento do que caracteriza o fim de semana deve ficar a cargo do usuário ou da aplicação final da informação temporal. Esta extensão ao padrão ISO 8601 se dá no formato utilizado para representação de dias da semana (AAAA-WSS-D), substituindo o dia da semana pelo *token* WE (de *weekend*), como feito no TIMEX2. Seguindo essa diretiva, portanto, o fim de semana dos dias 7 e 8 de janeiro de 2011 seria representado como "2011-W01-WE".

### 4.1.6. Manhã, tarde e noite

Outro ponto que, seguindo a proposta do TIMEX2, consideramos interessante capturar é o uso de expressões referentes a períodos do dia – manhã, tarde e noite –, que apesar de subjetivas (onde começam e terminam cada um desses períodos é sujeito à interpretação individual e, provavelmente, não é tão discreto como gostaríamos, sendo talvez melhor modelado por instrumentos de lógica *fuzzy*). Os *tokens* utilizados para os períodos supracitadas são, respectivamente, MO (de *morning*), AF (de *afternoon*) e NI (de *night*) e substituem a hora no formato do ISO 8601, conforme exposto abaixo (Tabela 4.4). Vale ressaltar que expressões como "nove horas da manhã" continuam sendo representadas simplesmente como "09:00". A extensão em questão diz respeito apenas a expressões para as quais não seja explicitada uma hora precisa.

**Tabela 4.4. Exemplos de períodos do dia**

| Exemplo | Representação |
|---|---|
| Manhã de 11 de setembro de 1985 | 1985-09-11TMO |
| *Tardes* são tristes | TAF |
| Noite de 3 de março de 1987 | 1987-03-03TNI |

### 4.1.7. Eras geológicas

Apesar de possivelmente pouco relevantes para o processamento temporal, a exemplo do TIMEX2, incluímos extensões para representação de expressões temporais em

referência a um passado distante, em particular, as que seguem a convenção científica usada para eventos geológicos (milhares, milhões e bilhões de anos atrás, usando, respectivamente, as abreviações KA, MA e GA). A forma de uso está listada na tabela a seguir.

**Tabela 4.5. Exemplos de eras geológicas**

| Exemplo | Representação |
|---|---|
| Dez mil anos atrás | KA10 |
| Quatro milhões de anos atrás | MA4 |
| Alguns milhões de anos atrás | MA |
| Dois bilhões de anos atrás | GA2 |

# 4.2. Padrão de anotação

Para ser anotada, a expressão deve ter como núcleo sintático uma palavra ou expressão numérica cujo sentido transmita uma unidade ou conceito temporal (como "dia" ou "mensalmente", por exemplo) o que chamamos de gatilho lexical ou gatilho temporal. A Tabela 4.6 exemplifica alguns desses gatilhos, separados por classe gramatical.

**Tabela 4.6. Exemplos de gatilhos temporais**

| Classe | Gatilhos |
|---|---|
| Substantivos | minuto, tarde, meia-noite, dia, noite, semana, mês, trimestre, ano, década, século, milênio, semestre, sábado, janeiro |
| Nomes próprios | Páscoa, Natal, Corpus Christi |
| Padrões regulares | 19:00, 19h, 1998, 11/09/1985 |
| Adjetivos | diário, mensal, bianual |
| Advérbio de modo | atualmente, ultimamente, diariamente, mensalmente |
| Advérbio de tempo | hoje, amanhã, agora, ontem, futuramente |

Vale lembrar que, no caso de os gatilhos lexicais aparecerem em expressões que designam algo que não seja uma entidade temporal (por exemplo, "Adorei [o filme] *Doce Novembro*" ou "Li *1984* [o livro de George Orwell] na escola"), tais expressões não devem ser anotadas.

Assim como na especificação do TIMEX2, e diferentemente do padrão HAREM, adotamos, sempre que possível, a especificação ISO 8601 na normalização das expressões temporais. Além das datas, horas, durações e frequências precisas, também normalizamos algumas ETs imprecisas.

### 4.2.1. Formato da etiqueta

Seguindo os padrões em voga, as ETs devem ser anotadas inserindo uma *tag* SGML (*Standard Generalized Markup Language*) em volta da expressão: "Voltarei <ET>amanhã</ET>". Adicionalmente, as etiquetas podem contar um ou mais dos atributos listados na Tabela 4.7.

**Tabela 4.7. Atributos usados na etiqueta do padrão de anotação**

| Atributo | Função | Exemplo |
|---|---|---|
| TIPO | Identifica o tipo semântico da ET. | TIPO="CALENDARIO" |
| REF | Indica o tempo de referência de ETs relativas. | REF="E" |
| DELTA | Indica o valor de deslocamento a partir do tempo de referência em ETs relativas. | DELTA="P2D" |
| DIR | Indica a direção do deslocamento a partir do tempo de referência em ETs relativas. | DIR="+" |
| VAL | Valor normalizado da ET. | VAL="2009-03-10" |
| LIM | Quando aplicável, indica o limite do intervalo aberto definido pela ET. | LIM="<=" |
| MOD | Indica a presença de um modificador de imprecisão na ET. | MOD="APROX" |
| QUANT | A quantidade de vezes que um evento ocorre por unidade de tempo (em ETs de frequência). | QUANT="2" |
| FREQ | A unidade de tempo (ou módulo) de uma freqüência. | FREQ="P1W" |

Um exemplo de anotação completa seria (considerando que a DCT do documento que contém a frase é 10 de março de 2009): '*Voltarei* <ET REF="E" DELTA="P1D" DIR="+" VAL="2009-03-11">*amanhã*</ET>'. Nas seções seguintes, elencaremos as diferentes variações de ETs que nos propusemos a anotar neste trabalho. A menos que indicado o contrário, consideraremos, para expressões relativas, sempre o tempo de referência como 10 de março de 2009.

### 4.2.2. Delimitação das expressões temporais

Nesta proposta, diferentemente da diretiva dada pelo TIMEX2 (FERRO *et al.*, 2005) e em consonância com (BAPTISTA *et al.*, 2008), consideramos que deve ser delimitada a totalidade da expressão temporal, isto é, incluindo a preposição que a introduzir, no caso da expressão temporal ser um sintagma preposicional (por exemplo, "*no ano passado"*), ou o determinante no caso de ser um sintagma nominal (ex.: "*todos os dias"*).

A motivação para a escolha deste caminho reside na noção de que, na maioria das ETs, os elementos ditos gramaticais (sobretudo preposições e determinantes) são não apenas partes integrantes dessas expressões, como contribuem de modo decisivo para a classificação das ETs nos diferentes tipos de categoria semântica. Naturalmente,

essa decisão acarreta – por uma questão não só de coerência, mas também de simplicidade – a inclusão, em algumas expressões temporais, de certas preposições que não fazem parte da ET propriamente dita. Isso acontece, sobretudo, no caso das ETs genéricas, como em "*Eu gosto* <ET>*da primavera*</ET>". Nesse caso, a preposição "de", enquanto elemento que introduz o complemento do verbo "gostar", em nada contribui para a interpretação da expressão temporal. Se relevante para a aplicação final, o tratamento das regências verbais deve, portanto, constituir um problema distinto, a ser resolvido independentemente do reconhecimento das ETs.

Por outro lado, em consonância com o TIMEX2, também incluímos dentro da etiqueta de anotação todos os pré- e pós-modificadores de expressões temporais (e.g. "até", "a partir de", "mais de", "aproximadamente", etc.).

## 4.2.3. Tipos semânticos

Nesta proposta, distinguimos entre seis tipos semânticos de expressões temporais (veja mais detalhes sobre classificação semântica de ETs na seção 2.3.2.1), a saber: datas de calendário, durações, frequências, idades, genéricas e indefinidas. Cada um deles tem um valor correspondente que deve ser preenchido no atributo TIPO das etiquetas de anotação. A Tabela 4.8 resume o panorama da classificação semântica adotada. As seções seguintes descrevem as diretivas de anotação para cada um dos tipos elencados.

**Tabela 4.8. Classificação semântica adotada**

| **Tipo** | **Valor do atributo TIPO** |
|---|---|
| Datas de calendário | CALENDARIO |
| Durações | DURACAO |
| Frequências | FREQUENCIA |
| Idades | IDADE |
| Genéricas | GENERICA |
| Indefinidas | INDEFINIDA |

### 4.2.3.1. Datas de calendário

As entidades do tipo CALENDARIO são expressões que permitem ancorar o predicado que elas modificam numa linha temporal. Para serem classificadas como tal, o ano precisa estar presente nessas expressões ou deve existir a possibilidade de inferi-lo pelo contexto. A normalização dessas ETs deve, no mínimo, preencher o atributo VAL.

Se a expressão for relativa, também é esperado que sejam preenchidos os atributos REF, DELTA e DIR. Para calcular o valor normalizado (e ser capaz de preencher o atributo VAL), é preciso conhecer o momento de enunciação ou inferir a que data ou evento no texto a ET faz referência. Nesse caso, REF pode receber o valor "E", quando a expressão for relativa ao momento da enunciação, ou "T", quando o referencial for outra data ou evento que apareça no contexto textual ou discursivo.

Além disso, para propriamente ancorar a ET na linha temporal, é antes necessário normalizar o valor de deslocamento (delta) a partir do tempo de referência (seja ele o momento da enunciação ou textual) e em que direção ele se dará (passado ou futuro). Esses dados devem ser preenchidos nos atributos DELTA (usando a notação do ISO 8601 para durações) e DIR ("-" para referências cronologicamente anteriores ao tempo de referência e "+" para as posteriores). Vale notar, finalmente, que, nesses casos, o atributo VAL corresponderá ao valor do tempo de referência somado ou subtraído da distância temporal explicitada no atributo DELTA, dependendo do sinal da DIR.

Prevemos ainda uma situação de uso diferente para o atributo DELTA: para expressões do tipo "no dia 3" ou "no domingo" não há muito valor em interpretar o delta entre a data de referência e a data expressa na ET. A intenção do atributo é guardar o significado e não o valor de deslocamento de fato. Para essas expressões, portanto, defendemos que sejam guardados no DELTA exatamente os componentes que permitem chegar ao ponto desejado no calendário a partir da data de referência. Para tal, em lugar de representar a diferença como uma duração, deve ser usado o formato do ISO 8601 para datas e horas juntamente com a extensão para unidades não especificadas apresentado anteriormente. Nos exemplos citados, gravaríamos, respectivamente, "XXXX-XX-03" e "XXXX-WXX-7" no atributo DELTA das ETs. A Tabela 4.9 logo adiante enumera diversos exemplos (o atributo TIPO="CALENDARIO" foi omitido das etiquetas para poupar espaço).

Por fim, um último ponto em que este trabalho difere dos existentes está na anotação de intervalos de datas. Enquanto a maioria gera duas anotações (uma para cada limite do intervalo), entendemos que o intervalo como um todo compõe uma única expressão temporal. Nesse sentido, utilizamos a notação do ISO 8601 para representação de intervalos de datas separando os limites temporais dos intervalos com uma barra ("/").

**Tabela 4.9. Exemplos de datas de calendário anotadas**

| Tipo | Exemplo |
|---|---|
| Dia absoluto | Nasci <ET VAL="1985-09-11">no dia 11 de setembro de 1985</ET>. |
| Mês absoluto | Minha irmã nasceu <ET VAL="1987-03">em março de 1987</ET>. |
| Dia impreciso | Minha irmã nasceu <ET VAL="1987-03-XX">em algum dia de março de 1987</ET>. |
| Ano absoluto | Iremos ao Rio <ET VAL="2016">em 2016</ET>. |
| Hora absoluta | Minha irmã nasceu <ET VAL="1987-03-03T12:00">no dia 3 de março de 1987 ao meio-dia</ET>. |
| Dia relativo | Cheguei <ET REF="E" DELTA="P1D" DIR="-" VAL="2009-03-09">ontem</ET>. |
| Dia relativo | Viajarei <ET REF="E" DELTA="XXXX-XX-15" DIR="+" VAL="2009-03-15">no dia 15</ET>. |
| Dia relativo | Ele chegou <ET REF="E" DELTA="XXXX-WXX-6" DIR="-" VAL="2009-03-07">no sábado</ET>. |
| Dia relativo | Estive viajando <ET REF="E" DELTA="P1W" DIR="-" VAL="2009-W10">na semana passada</ET>. |
| Dia relativo | Ele chegou <ET REF="T" DELTA="P3D" DIR="+" VAL="2009-03-13">três dias depois</ET>. |
| Mês relativo | Ele viajou <ET REF="T" DELTA="P1M" DIR="+" VAL="2009-04">no mês seguinte</ET>. |
| Ano relativo | Estou no Rio <ET REF="E" DELTA="P1Y" DIR="-" VAL="2008">desde o ano passado</ET>. |
| Hora relativa | Minha irmã comemorou seu aniversário <ET REF="E" DELTA="XXXX-03-03T19:00" DIR="-" VAL="2009-03-03T19:00">no dia 3 às 19h</ET>. |
| Intervalo | Os jogos aconteceram <ET VAL="2010-03/2010-04">entre março e abril de 2010</ET>. |

### 4.2.3.2. Durações

A anotação das durações requer, além da identificação do tipo, apenas o atributo VAL preenchido com o valor normalizado da quantificação temporal expressa pela ET. Diferentemente da proposta do Segundo HAREM, com o intuito de nos aproximarmos dos padrões usados internacionalmente (a exemplo do TIMEX2 e da TimeML), optamos por utilizar a notação de representação de durações do ISO 8601.

Além disso, sanamos duas deficiências identificadas no Segundo HAREM (BAPTISTA *et al.*, 2008). A primeira delas diz respeito à inclusão de uma unidade menor que o segundo (milissegundos), a fim de permitir o tratamento adequado de, por exemplo, resultados desportivos. Novamente, utilizamos para tal a notação do ISO 8601, que também permite uma representação mais elegante que a proposta no Segundo HAREM para frações de outras granularidades: enquanto a duração presente na frase "O processo durou seis meses e meio" seria anotada (o valor normalizado) como "A0M6S0D15H0M0S0" no padrão do HAREM, pelo ISO 8601, teríamos o mesmo valor representado por "P6,5M", que, além de mais concisa, remove a ambigüidade

existente na primeira normalização (isto é, se o valor normalizado representa seis meses e meio ou seis meses e quinze dias, que, apesar de similares, são durações de precisão e clareza distintas).

A segunda lacuna preenchida pela presente proposta está na anotação de intervalos de durações. Para isto, estendemos a especificação do ISO 8601, para possibilitar a normalização desse tipo de ET, seguindo o padrão adotado de separar os limites temporais de intervalos com uma barra ("/"), conforme exemplificado na Tabela 4.10.

Por fim, tendo em mente que o objetivo de um padrão de anotação temporal é representar as relações temporais expressas em um documento, sem âncoras, a normalização de ETs de duração só transmite a informação de por quanto tempo algo aconteceu, sem qualquer dado sobre quando isso ocorreu. Assim, para maximizar a informação temporal capturada, seguimos a proposta do TIMEX2 e incluímos atributos para ancoragem de durações relativas (REF, DIR e REF_VAL). Os atributos REF e DIR funcionam tal e qual exposto anteriormente para datas, enquanto o novo atributo REF_VAL captura o valor do tempo de referência. A granularidade desse valor deve variar com a intenção do texto: em "Há quatro anos hoje", o atributo REF_VAL seria preenchido com "2009-03-10", a data atual do tempo de referência; por outro lado, em "Há quatro anos", provavelmente, a intenção do autor ou do interlocutor é ancorar a expressão no ano corrente (ou seja, REF_VAL seria "2009"). Quando não estiver clara a intenção da expressão, deve-se utilizar a mesma granularidade da duração (nos exemplos acima, portanto, o ano). Pode-se argumentar que, como nas especificações do TIMEX2, os atributos supracitados deveriam ser preenchidos para todas as durações, uma vez que, em última análise, à exceção de ETs do tipo "uma gestação leva nove meses", quase todas as durações são relativas. Contudo, optamos por classificar como tais apenas as durações claramente ancoráveis, isto é, durações para as quais a ancoragem contribui no entendimento da expressão, havendo um componente referencial explícito. Outros exemplos de uso dos atributos para normalização de durações relativas estão listados na Tabela 4.10 (todos com o atributo TIPO="DURACAO" omitido por economia de espaço).

**Tabela 4.10. Exemplos de durações anotadas**

| **Tipo** | **Exemplo** |
|---|---|
| Duração absoluta | A construção levou <ET VAL="P6M">seis meses</ET>. |

| | |
|---|---|
| Intervalo absoluto | A reunião durará <ET VAL="PT2H/PT3H">entre duas e três horas</ET>. |
| Duração absoluta | O processo levou <ET VAL="PXDE">décadas</ET>. |
| Duração relativa | <ET VAL="P4Y" REF="E" DIR="-" REF_VAL="2009">Nos últimos quatro anos</ET>, o presidente Lula governou o Brasil. |
| Duração relativa | <ET VAL="P4Y" REF="T" DIR="+" REF_VAL="2009">Nos quatro anos seguintes</ET>, o país será governado por uma mulher. |
| Duração relativa | Ele responderá <ET VAL="P3D" REF="E" DIR="+" REF_VAL="2009-03-10">em três dias</ET>. |

#### 4.2.3.3. Frequências

A proposta do Segundo HAREM não trata da normalização de frequências, a despeito de sugerir caminhos para o futuro em (BAPTISTA *et al.*, 2008). Embora o padrão TIMEX2 utilize uma abordagem distinta, optamos por segui-la apenas em parte, acatando também algumas das sugestões dos autores supracitados e incluindo dois atributos suplementares (QUANT e FREQ) para a normalização dessas ETs. Essa direção é similar à seguida na TimeML e estudada como possibilidade futura no apêndice das especificações TIMEX2 (FERRO *et al.*, 2005).

Antes de detalharmos a maneira como capturamos o significado dessas ETs, cabe definir o que é uma frequência ancorável, uma vez que o tratamento dado pode ser diferente dependendo da possibilidade de classificação da expressão como tal. Uma frequência ancorável é aquela que pode ser facilmente instanciada como um conjunto de datas ou outras unidades precisas no calendário. A expressão "todos os dias", por exemplo, é ancorável, uma vez que podemos facilmente apontar quais são os dias em que o evento associado a ela ocorre em um calendário (nesse caso, todos). Por outro lado, as ETs "dia sim dia não" e "a cada dois dias", apesar de pouco diferente quantitativamente, não são ancoráveis, uma vez que não conseguimos dizer com exatidão, tendo apenas seu conteúdo semântico como base, em quais dias do calendário os eventos aos quais são associadas ocorrem.

Na normalização de frequências não ancoráveis, o atributo QUANT indica o número de vezes em que o evento ou processo se repete, enquanto o atributo FREQ representa a granularidade ou módulo dessa frequência. Aquele é preenchido por valores numéricos, ao passo em que este segue a notação usada para durações. Para frequências ancoráveis, utilizamos a mesma estratégia do TIMEX2, a saber: mantendo o formato ISO 8601 para datas, preenchemos as posições faltantes com "X" para indicar que elas podem ser ancoradas a quaisquer instâncias de tais componentes. Os exemplos da Tabela 4.11 abaixo deixam mais claro o uso desse artifício. Novamente, omitimos o

atributo TIPO=“FREQUENCIA” das etiquetas para economizar espaço e facilitar a ilustração dos atributos relevantes para o tipo em questão.

O atributo QUANT ainda pode receber alguns valores especiais para quantificar frequências imprecisas (que, por definição, também não são ancoráveis), isto é, aquelas para as quais é explicitado ou não se pode inferir um valor numérico quantificador. Um exemplo desse caso seria a ET “algumas vezes por mês”: apesar de facilmente identificarmos seu módulo (a cada um mês), não sabemos quantas vezes por mês o evento ocorre. Para expressões assim, o atributo QUANT deve ser preenchido com o caractere “?”. Algumas expressões, apesar de imprecisas, trazem sinais de intensidade, como em “muitas vezes por dia”. Para não desperdiçar essa valiosa informação, usamos outros dois caracteres para valorar o atributo QUANT, quais sejam: “+” para eventos muito frequentes e “-“ para os pouco frequentes. Vale ressaltar que mesmo ETs que têm tanto o quantificador quanto o módulo imprecisos devem ser anotadas, como se vê na anotação dos exemplos supracitados mostrados na Tabela 4.11.

Por fim, apesar de sermos capazes de anotar e capturar com alguma precisão a semântica de expressões temporais de frequência razoavelmente complexas como “no dia 5 de cada mês”, ainda não há na literatura nenhuma proposta para dar conta de expressões cujo significado global parece combinar o valor de frequência com o de localização temporal, como acontece em “todas as primeiras segundas-feiras de cada mês”.

**Tabela 4.11. Exemplos de frequências anotadas**

| **Tipo** | **Exemplo** |
|---|---|
| Frequência não ancorável | Irei <ET QUANT=“2” FREQ=“P1W”>duas vezes por semana</ET>. |
| Frequência não ancorável | A Copa do Mundo acontece <ET QUANT=“1” FREQ=“P4Y”>a cada quatro anos</ET>. |
| Frequência não ancorável | Malho <ET QUANT=“PT2H” FREQ=“P1D”>duas horas por dia</ET>. |
| Frequência ancorável | Faço isso <ET VAL=“XXXX-XX-XX”>diariamente</ET>. |
| Frequência não ancorável | Vou à academia <ET QUANT=“1” FREQ=“P2D”>dia sim dia não</ET>. |
| Frequência imprecisa | Jogamos futebol <ET QUANT=“?” FREQ=“P1M”>algumas vezes por mês</ET>. |
| Frequência ancorável | Jogo tênis <ET VAL=“XXXX-WXX-3TMO”>todas as quartas-feiras pela manhã</ET>. |
| Frequência ancorável | Recebo meu salário <ET VAL=“XXXX-XX-05”>no dia 5 de cada mês</ET>. |

### 4.2.3.4. Idades

Não obstante nenhum dos padrões para anotação de expressões temporais prever o reconhecimento de idades, em função de sua similaridade com expressões de duração (o que implicaria na necessidade do sistema deliberadamente distinguir entre os dois de qualquer forma), optamos por tratá-las, anotá-las e normalizá-las. Caberá ao usuário ou à aplicação final da informação temporal determinar sua eventual utilidade e uso. A normalização desse tipo de ET utiliza apenas o atributo VAL para guardar a representação da idade, usando o formato ISO 8601 para durações. A Tabela 4.12 traz alguns exemplos, como nos outros casos, omitindo o atributo TIPO="IDADE".

**Tabela 4.12. Exemplos de idades anotadas**

| Tipo | Exemplo |
|---|---|
| Idade | João tem <ET VAL="P19Y">19 anos de idade</ET>. |
| Idade | Pedro, <ET VAL="P16Y">16</ET>, cresceu no bairro. |
| Idade | O bebê <ET VAL="P6M">de apenas seis meses</ET> foi encontrado na rua. |

### 4.2.3.5. Genéricas e indefinidas

As expressões que sejam compostas por unidades lexicais que denotam elementos temporais, mas que não se refiram a datas específicas são classificadas como genéricas. Diferentemente do HAREM, optamos por normalizar essas ETs, ainda que de forma limitada, seguindo o padrão adotado no TIMEX2, usando a extensão proposta para unidades não especificadas. O atributo TIPO dessas expressões temporais deve ser preenchido com o valor "GENERICA".

Também marcamos expressões que, apesar de utilizadas de modo referencial, são indefinidas. Geralmente, essas ETs são acompanhadas de um artigo indefinido e, apesar de incomuns em textos jornalísticos, não são difíceis de serem encontradas em obras literárias. Elas devem ser marcadas com o valor "INDEFINIDA" no atributo TIPO e normalizadas quando e até onde possível. As expressões genéricas e indefinidas podem ter um papel relevante no cálculo de referências temporais, pelo que importa identificá-las adequadamente. Exemplos das duas, com os respectivos atributos TIPO="GENERICA" e TIPO="INDEFINIDA" omitidos das etiquetas, estão listados na Tabela 4.13 a seguir.

**Tabela 4.13. Exemplos de ETs genéricas e indefinidas anotadas**

| Tipo | Exemplo |
|---|---|
| Genérica | Adoro o <ET VAL="XXXX-WI">inverno</ET>. |
| Genérica | <ET VAL="XXXX-02">Fevereiro</ET> é o mês mais curto do <ET VAL="XXXX">ano</ET>. |
| Genérica | Odeio <ET VAL="XXXX-WXX-7>domingos</ET>. |
| Indefinida | Era <ET VAL="XXXX-04-XXTAF">uma tarde fria de abril</ET>. |

## 4.2.4. Outros atributos

Além dos atributos citados nos detalhamentos dos tipos semânticos reconhecidos pela atual proposta, ainda identificamos a necessidade de capturar a semântica relativa a limites e modificadores comumente encontrados associados às expressões temporais. Para isso, introduzimos dois atributos (LIM e MOD) e mostramos como utilizá-los nas subseções a seguir. Vale lembrar que tais modificadores, quando presentes, devem estar inclusos na etiqueta de anotação, uma vez que são parte essencial do significado da expressão temporal.

### 4.2.4.1. Limites

É comum encontrar, em linguagem natural, modificadores que dão uma noção de limite às expressões temporais, tais como "até", "a partir de", "depois de", entre outros. Esses modificadores delimitam um intervalo temporal, à semelhança da noção de intervalo da matemática, sendo um dos extremos aberto e o outro, fechado. Intervalos fechados são tratados de forma distinta, como um subtipo de datas ou durações, conforme explicitado anteriormente. O atributo LIM é usado para representar a ideia de intervalo temporal transmitida pelo modificador presente na expressão, sendo preenchido com o operador relacional (maior que, menor que, maior ou igual a e menor ou igual a) que melhor represente o extremo fechado do limite. Os exemplos da Tabela 4.14 ilustram sua utilização.

**Tabela 4.14. Exemplos de ETs anotadas com limites**

| Tipo | Exemplo |
|---|---|
| Data (semana) | Estive viajando <ET REF="E" DELTA="P1W" DIR="-" LIM="<=" VAL="2009-W10">até a semana passada</ET>. |
| Duração | Ele responderá <ET VAL="P3D" LIM=">">depois de três dias</ET>. |

#### 4.2.4.2. Modificadores

Outro ponto em que diferimos do padrão do Segundo HAREM, a exemplo do utilizado no TIMEX2 e na TimeML, levando em conta a sugestão feita por (BAPTISTA *et al.*, 2008) e implementada para todos os tipos neste trabalho, é na consideração e tratamento da imprecisão. Em ETs como "por volta de três horas" ou "aproximadamente dois dias", a indefinição (também chamada de vagueza, em particular, no contexto do Segundo HAREM) introduzida pelas expressões "por volta de" e "aproximadamente" não era capturada. No presente trabalho, o fazemos através do atributo MOD, utilizando, para esse caso, por exemplo, o valor "APROX" (vide exemplo na Tabela 4.15). Outros valores desse atributo representam modificadores de datas, como "no início de" (INICIO), "no final de" (FIM) e "meados de" (MEIO), e de durações, como menos que (MENOS), mais que (MAIS), igual ou menos que (IGUAL_OU_MENOS) e igual ou mais que (IGUAL_OU_MAIS).

**Tabela 4.15. Exemplos de ETs anotadas com modificadores**

| **Tipo** | **Exemplo** |
|---|---|
| Data (mês) | Viajaremos <ET REF="E" DELTA="XXXX-04" DIR="+" VAL="2009-04" MOD="INICIO">no início de abril</ET>. |
| Data (ano) | Volto <ET VAL="2011" MOD="MEIO">em meados de 2011</ET>. |
| Data (dia) | Falarei com ela <ET REF="E" DELTA="XXXX-XX-20" DIR="+" VAL="2009-03-20" MOD="APROX">por volta do dia 20</ET>. |
| Duração | A construção levou <ET VAL="P6M" MOD="APROX">aproximadamente seis meses</ET>. |
| Duração | A reunião durou <ET VAL="PT3H" MOD="MAIS">mais de três horas</ET>. |

## 4.3. Arquitetura

Para executar as diretivas de anotação e normalização supracitadas, foi implementado o CoppeTER (Coppe Temporal Expression Recognizer), um reconhecedor de expressões temporais para o português. O sistema segue a arquitetura ilustrada na Figura 4.2, sendo dividido em dois componentes principais e, se assim desejado e respeitadas as interfaces de entrada e saída, independentes.

Diferentemente do XIP (LOUREIRO, 2007; HAGÈGE *et al.*, 2008a; HAGÈGE *et al.*, 2009; HAGÈGE *et al.*, 2010), avaliado como o atual estado da arte para o reconhecimento e normalização de expressões temporais em português (BAPTISTA *et al.*, 2008), optamos por construir o componente de processamento temporal fora da

parte de análise morfossintática do sistema, pois, além da independência entre os componentes, consideramos tal abordagem mais elegante e mais fácil de manter, evoluir e reutilizar.

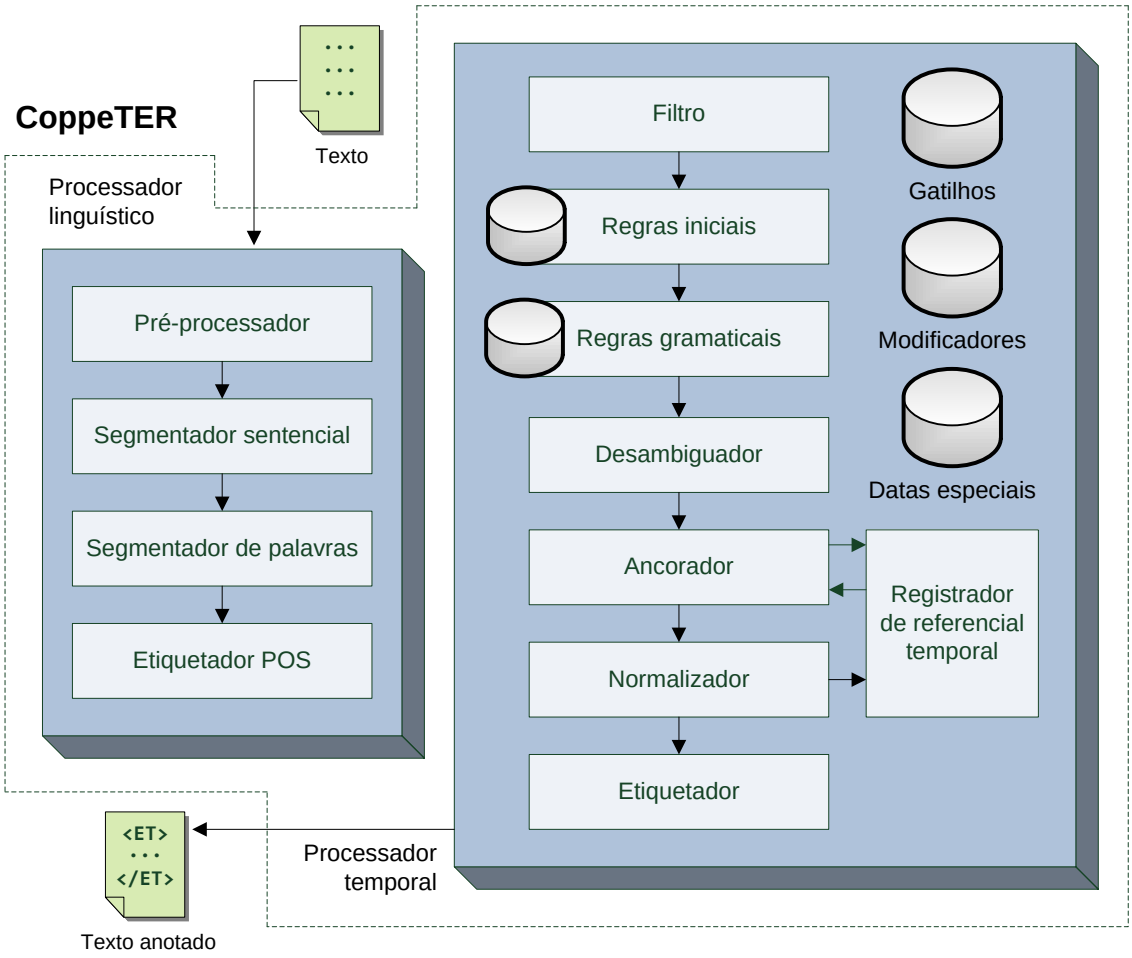



**Figura 4.2. Arquitetura do CoppeTER**

O primeiro componente é o processador linguístico, que é dividido em quatro módulos: pré-processador, segmentador sentencial, segmentador de palavras e etiquetador POS. O pré-processador é responsável por sanitizar o texto e executar quaisquer procedimentos necessários para preparar a entrada aos módulos seguintes. O segmentador sentencial recebe como entrada o texto puro e fornece como saída uma lista de sentenças, que são, uma a uma, passadas para os módulos seguintes. O segmentador de palavras recebe uma sentença e retorna uma lista de palavras (ou melhor, *tokens*, uma vez que os caracteres de pontuação também formam palavras). Cada uma das sentenças, agora quebradas em listas de palavras, é processada, então, pelo etiquetador POS que retorna, por fim, uma lista de pares, contendo cada palavra da

sentença e sua etiqueta POS associada. O conjunto dessas sentenças processadas e etiquetadas é a saída do componente de processamento lingüístico e a entrada do processador temporal.

O componente de processamento temporal utiliza três dicionários e um conjunto de regras como fontes de dados de seus módulos. O primeiro módulo é um filtro que usa o dicionário de gatilhos para eliminar frases que não contenham alguma palavra com conteúdo temporal, evitando, assim, a tentativa desnecessária de aplicação das diversas regras do sistema e, consequentemente, acelerando o processamento. Pode-se questionar se, nesse sentido, não seria interessante posicionar o filtro antes do etiquetador POS na cadeia de módulos do processador linguístico, contudo, optamos por deixá-lo na entrada do processador temporal, pois os gatilhos são essencialmente conhecimento temporal e colocá-los no processador linguístico criaria um acoplamento, que enxergamos como indesejável por ora, entre os componentes.

Sistemas de reconhecimento e normalização de expressões temporais têm sido implementados em grande parte através de transdutores de estado finito ou etiquetadores baseados em regras. Essas abordagens parecem ser as mais efetivas quando o objetivo inclui as duas tarefas. De fato, o conjunto relativamente limitado de palavras que se combinam para formar expressões temporais sugere que sistemas baseados em regras podem ser implementados satisfatoriamente com esforço razoavelmente pequeno e, ainda assim, prover bons resultados (CASELLI *et al.*, 2009). Por isso, seguimos esta direção no presente trabalho.

As sentenças que passam pelo filtro são processadas por um conjunto de regras separadas em uma cadeia de dois estágios. No primeiro deles, são reconhecidos números, expressões com padrões simples (em particular, aquelas passíveis de casamento por expressão regular), gatilhos temporais e modificadores (palavras como “aproximadamente”, “até”, “a partir de”, “próximo”, “seguinte”, etc., que ajudam na resolução de alguns tipos de ETs, como as relativas e as imprecisas). É nesse estágio também que são reconhecidas, com auxílio de um dicionário, as expressões referentes a feriados, datas especiais e comemorativas. O segundo módulo desta cadeia é responsável pelo efetivo reconhecimento das expressões temporais, através de um conjunto de regras gramaticais executadas iterativamente.

Antes de proceder à normalização, alguns tipos de ETs passam por um módulo desambiguador para resolver ambiguidades, em particular, os casos de palavras como

“hoje” e algumas expressões temporais relativas (por exemplo, “no sábado”). Em seguida, para ETs relativas, o ancorador identifica o foco temporal e fornece os últimos parâmetros necessários à normalização e posterior anotação das expressões encontradas. Caso o referencial seja uma data ou evento do texto, o ponto de ancoragem é provido pelo registrador de referencial temporal.

O processo de normalização propriamente dito acontece no módulo normalizador, levando em conta todos os traços disponíveis para resolução das expressões temporais. No CoppeTER, a estrutura interna resultante da normalização já é bastante similar ao padrão de anotação detalhado anteriormente, cabendo ao último módulo da cadeia, o etiquetador, apenas a formatação das *tags* e atributos em SGML. No caso de o usuário solicitar que as ETs sejam anotadas nos padrões TIMEX2, TimeML ou HAREM, o etiquetador fará um mapeamento do esquema de anotação CoppeTER para um dos referidos padrões.

Tendo como base esta arquitetura, detalharemos a implementação do sistema CoppeTER no próximo capítulo.

# 5. Implementação

Este capítulo descreve a implementação do sistema CoppeTER, seguindo as diretivas de anotação propostas e a arquitetura desenhada no capítulo anterior. Detalhamos os aspectos relevantes de cada um dos módulos dos dois componentes, além de ilustrar exemplos práticos de entradas e saídas quando aplicável.

## 5.1. Visão geral

O sistema é todo desenvolvido em Python (PYTHON, 2011), uma linguagem de programação dinâmica bastante popular e com uma extensa gama de bibliotecas disponíveis. Em geral, a versão da linguagem não é uma questão importante, sendo a maioria delas razoavelmente compatíveis entre si. No caso de Python, em particular, esse é um ponto que requer uma ponderação. A versão 2.7 segue a linha de compatibilidade da linguagem desde o seu início e é atualmente a mais difundida. A nova versão, 3.2, representa o futuro de Python, apesar de ter tomado a difícil decisão de alterar aspectos básicos da linguagem, tornando boa parte das aplicações escritas em versões anteriores à 3.0 completamente incompatíveis. Utilizamos a versão 2.7 da linguagem, pois ela ainda é a versão utilizada pela maioria das bibliotecas de terceiros.

O critério de escolha da linguagem se baseou em três aspectos: biblioteca de processamento lingüístico, portabilidade e produtividade. Tendo o primeiro aspecto como primordial, estudamos as três bibliotecas mais maduras de processamento lingüístico publicamente disponíveis na web e as avaliamos segundo os critérios elencados na Tabela 5.1.

**Tabela 5.1. Comparação de bibliotecas de processamento linguístico**

| Biblioteca (Linguagem) Critério | **NLTK (Python)** | **LingPipe (Java)** | **OpenNLP (Java)** |
|---|---|---|---|
| **Portabilidade** | Sim | Sim | Sim |
| **Código aberto** | Sim | Sim | Sim |
| **Facilidade de uso** | Excelente | Bom | Bom |
| **Documentação** | Livro (publicado), tutoriais e classes | Livro (em desenvolvimento), tutorias e classes | Manual e classes |
| **Corpora em português** | Floresta Sintática e MacMorpho | Nenhum | Nenhum |
| **Tokenizadores** | Sentenças, palavras | Sentenças, palavras | Palavras |
| **Etiquetadores POS** | N-grama, Brill, HMM, | HMM, CRF | MaxEnt |

| | | | |
|---|---|---|---|
| | TnT, MaxEnt, Naive Bayes | | |
| **Classificadores** | Árvores de decisão, MaxEnt, Naive Bayes | KNN, naive Bayes, perceptron, regressão logística, Bernoulli | MaxEnt |
| **Métricas de avaliação** | Sim | Sim | Sim |
| **Produtividade** | Excelente | Bom | Bom |

O NLTK (LOPER *et al.*, 2002, BIRD *et al.*, 2006, BIRD *et al.*, 2009, NLTK, 2011) e o LingPipe (LINGPIPE, 2011) se mostraram bem mais maduros e completos, tanto em termos de ferramentas e classes disponibilizadas quanto de documentação de uso, do que o OpenNLP (OPENNLP, 2011). A eventual opção pelo NLTK (Natural Language Toolkit) teve por base os critérios de facilidade de uso e produtividade, não só da biblioteca em si, mas também da linguagem de programação Python. Além disso, o NLTK já traz adaptados às suas interfaces e padrões – e, portanto, prontos para utilização – dois corpora em língua portuguesa, nos permitindo experimentar diferentes configurações de etiquetadores POS com menor dispêndio de esforço e tempo.

## 5.2. Processador lingüístico

O processador lingüístico é a porta de entrada do CoppeTER, sendo composto por quatro módulos, conforme ilustrado na arquitetura do sistema: pré-processador, segmentador sentencial, segmentador de palavras e etiquetador gramatical. Nas subseções a seguir, destacaremos os principais aspectos de implementação desses elementos.

### 5.2.1. Pré-processador

O pré-processador do componente lingüístico do CoppeTER é responsável por garantir que o texto fornecido como entrada do sistema esteja dentro dos padrões esperados. Para tal, ele verifica se o *encoding* do texto é compatível com o utilizado internamente e, em caso negativo, converte para o padrão utilizado pelo sistema (Unicode). Além disso, o pré-processador remove excesso de *whitespace* (caracteres de espaço e controle que são representados visualmente como espaços em branco) no texto e, opcionalmente, remove *tags* HTML, se assim o usuário desejar.

### 5.2.2. Segmentador sentencial

O segmentador sentencial utiliza a implementação do Punkt (KISS *et al.*, 2006) disponibilizada no NLTK, mas treinada em cima de um corpus em língua portuguesa. Quantitativamente, abreviações são a maior fonte de ambiguidades nos problemas de detecção de limites de sentenças, pois elas constituem até 30% dos possíveis candidatos a limites de sentenças em textos corridos. Este segmentador é baseado na ideia de que a maioria das ambiguidades existentes na determinação desses limites pode ser eliminada uma vez que as abreviações tenham sido identificadas. Em vez de ter como base pistas ortográficas, o Punkt é capaz de detectar abreviações com alta precisão usando três critérios: abreviações podem ser definidas como uma colocação entre uma palavra truncada e um ponto final, são geralmente curtas e podem conter pontos internos.

A abordagem é separada em dois classificadores (ambos usam *likelihood ratio*): o primeiro determina se um *token* é uma abreviação, enquanto o segundo detecta de que tipo é a combinação entre um *token* e o ponto que o sucede (palavra no fim da sentença, abreviação no fim da sentença, reticências no fim da sentença, iniciais de nome ou número ordinal). O sistema não faz uso de anotações adicionais, etiquetagem POS ou listas pré-compiladas. Além disso, informação ortográfica não é a evidência principal dos classificadores, sendo, portanto, ideal para o processamento de texto exclusivamente em minúsculas ou maiúsculas. Esse método foi testado em onze idiomas, tendo obtido precisão média de 98,74%. Para o português, o sistema alcançou 99,14% de precisão e 99,72% de revocação (99,43% de medida F).

### 5.2.3. Segmentador de palavras

A segmentação de palavras em idiomas como o português, diferentemente de línguas como o chinês, é relativamente simples. O módulo do CoppeTER utiliza uma adaptação de uma classe do NLTK, baseada em expressões regulares, para quebrar cada sentença do texto em uma lista das palavras e *tokens* que a constituem. A adaptação foi feita para corrigir incoerências percebidas nos resultados dados pela classe, em particular, no tratamento de reticências e nos limites de citações (com aspas simples ou duplas), além de implementar um procedimento para desfazer contrações (isto é, transformar partículas como “do” em “de o”).

### 5.2.4. Etiquetador gramatical

O motor de regras do CoppeTER e o módulo desambiguador fazem uso de informações de categoria gramatical para melhor identificar o contexto, a direção e o sentido das expressões temporais identificadas. A maioria dos etiquetadores gramaticais é baseada em técnicas de aprendizado estatístico. Para a implementação do módulo do sistema, testamos algumas abordagens clássicas, conforme descritas a seguir. Todos os métodos foram treinados em cima do corpus Bosque do projeto Floresta Sintá(c)tica (AFONSO *et al.*, 2001; FREITAS *et al.*, 2008). Esse corpus, disponibilizado pela Linguateca (LINGUATECA, 2011), foi analisado automaticamente pelo analisador sintático automático PALAVRAS (BICK, 2000) e revisado integralmente por linguistas, sendo composto por 9.368 frases (186 mil palavras) provenientes de textos dos jornais Folha de São Paulo (do Brasil) e Público (de Portugal). A Tabela 5.2 descreve as categorias gramaticais utilizadas no corpus e, consequentemente, nas regras do sistema.

**Tabela 5.2. Categorias gramaticais e símbolos usados no corpus Floresta Sintá(c)tica**

| **Categoria gramatical** | **Símbolo** |
|---|---|
| Nome, substantivo | N |
| Nome próprio | PROP |
| Adjetivo | ADJ |
| Verbo (finito, infinitivo, particípio, gerúndio) | V (V-FIN, V-INF, V-PCP, V-GER) |
| Artigo | ART (ART-DET, ART-IND) |
| Pronome (pessoal, determinado, independente) | PRON (PRON-PERS, PRON-DET, PRON-INDP) |
| Advérbio | ADV |
| Numeral | NUM |
| Preposição | PRP |
| Interjeição | INTJ |
| Conjunção (subordinativa, coordenativa) | CONJ (CONJ-S, CONJ-C) |

A primeira abordagem usa um encadeamento seqüencial de etiquetadores mais simples. Esse encadeamento, chamado de *sequential backoff tagger*, começa com um etiquetador padrão que apenas atribui a etiqueta mais comum encontrada no corpus a qualquer palavra (no caso, o substantivo, representado pelo símbolo "N"). Em cima dele, construímos um etiquetador baseado em afixos, isto é, um etiquetador probabilístico que determina a etiqueta de acordo com o prefixo ou sufixo da palavra – vale ressaltar que esses afixos não são morfológicos, mas sim de tamanho fixo (no caso, três caracteres). O encadeamento funciona de tal forma que, se o afixo da palavra sendo analisada não for conhecido, o etiquetador padrão é acionado e, consequentemente, atribuída a etiqueta padrão (a mais comum) à palavra em questão. Essa lógica de

encadeamento (o *backoff*) se repete para cada par de etiquetadores, conforme ilustrado na Figura 5.1.

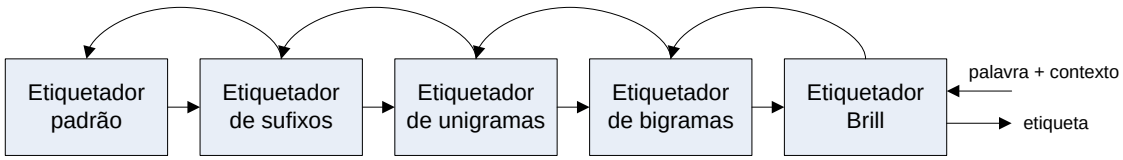



**Figura 5.1. Lógica de encadeamento do etiquetador Brill com *backoff***

Em seguida, construímos um etiquetador baseado em unigramas, tendo como *backoff* deste o etiquetador de afixos. A partir daí, implementamos um etiquetador de bigramas em cima do anterior. Conforme explicado na seção sobre etiquetadores baseados em modelos de n-grama do Capítulo 2, enquanto os unigramas levam em conta apenas a palavra sendo analisada, os bigramas também consideram a etiqueta da palavra anterior. Por fim, concluímos a implementação do etiquetador seqüencial *backoff* com um etiquetador Brill (BRILL, 1995). Todos os etiquetadores foram adaptados de classes existentes na biblioteca NLTK.

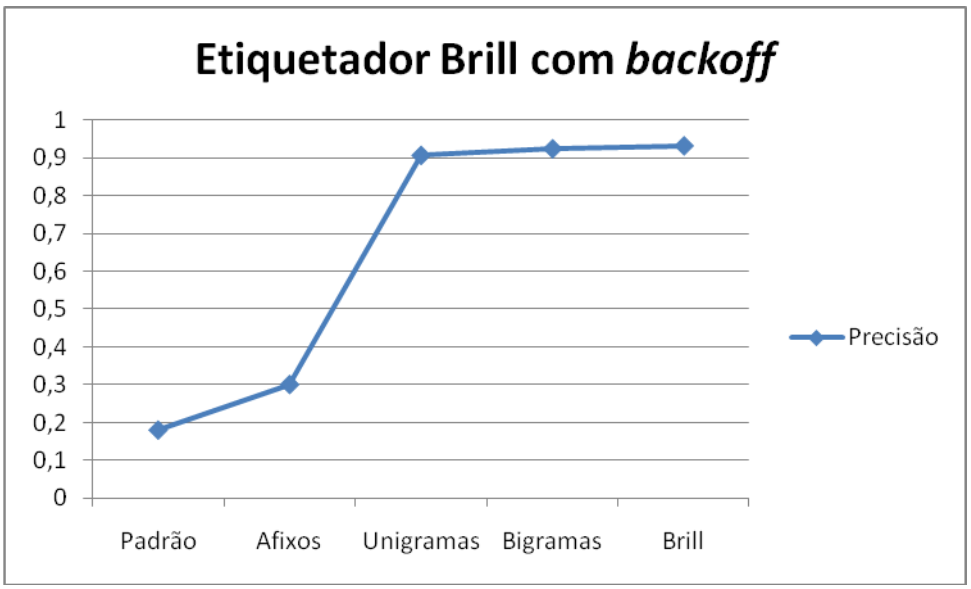



**Figura 5.2. Gráfico da precisão do etiquetador Brill com *backoff***

O gráfico da Figura 5.2 ilustra a precisão alcançada pelo encadeamento de etiquetadores. Cada ponto no gráfico representa a precisão alcançada pelo por um etiquetador tendo como componente final aquele que rotula o ponto. A simples

atribuição da etiqueta mais comum (substantivo) pelo etiquetador padrão alcança uma precisão de 17,99%. Um etiquetador de afixos acoplado ao etiquetador padrão atinge uma precisão de 30,04%. O uso de um etiquetador de unigramas em cima dele eleva a precisão para 90,61%. Com a adição de um etiquetador de bigramas contabilizamos precisão de 92,35% e, por fim, a utilização de um etiquetador Brill como componente final da abordagem percebe uma precisão de 93,12%. Apesar de termos testado a inclusão de um etiquetador de trigramas, o seu uso na abordagem descrita não trouxe qualquer ganho significativo em precisão.

As outras duas abordagens que incluímos entre as possíveis instanciações do etiquetador POS do CoppeTER são um etiquetador baseado em um classificador bayesiano ingênuo e uma implementação externa do TreeTagger (SCHMID, 1994), um etiquetador probabílistico baseado em árvores de decisão, adaptada para o padrão NLTK. Entre os atributos utilizados pelo classificador Bayes ingênuo estão: as duas palavras e etiquetas anteriores à palavra sendo analisada, a própria palavra, uma versão normalizada da palavra, os sufixos de uma, duas e três letras e a forma da palavra (número, pontuação, maiúsculas, minúsculas, capitalização, etc.). A Figura 5.3 compara o resultado alcançado pelas três abordagens no corpus Bosque.

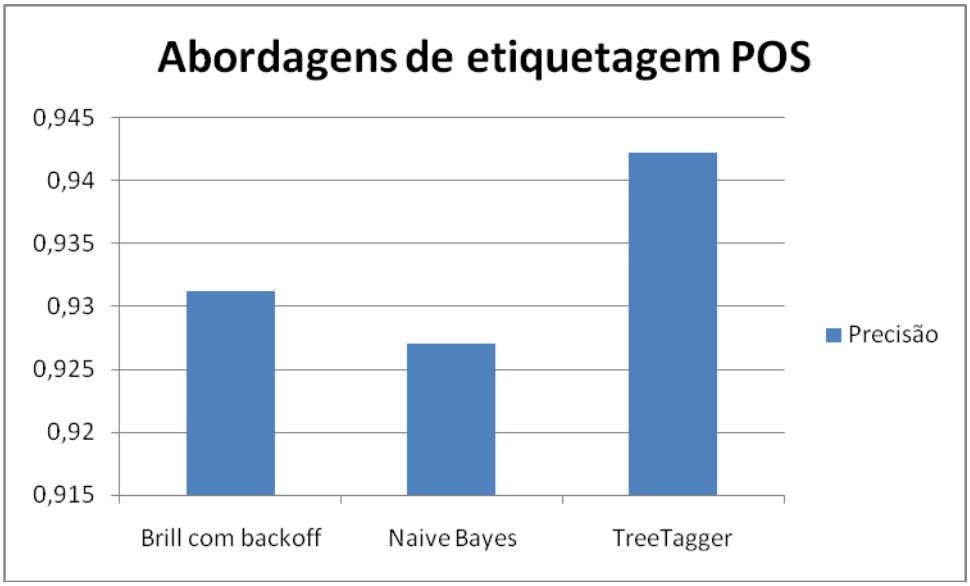



**Figura 5.3. Gráfico de desempenho das abordagens de etiquetagem POS**

Como etiquetador principal do sistema, utilizamos o Brill com *backoff*, conforme abordagem descrita acima. O TreeTagger, embora tenha apresentado um resultado melhor, está implementado com um programa externo, sendo necessário,

portanto, o disparo de um novo processo para cada sentença analisada. Em virtude disso e da integração mais fácil e rápida com o resto do sistema pelo uso de classes do próprio NLTK, optamos pelo etiquetador com a segunda melhor precisão. Há alguns casos de expressões temporais, contudo, para os quais invocamos o TreeTagger com o intuito de obter a forma do verbo principal da sentença no modo infinitivo, pois, entre os etiquetadores apresentados, ele é o único que fornece automaticamente essa informação. Essa chamada é feita no contexto do processo de desambiguação, detalhado mais adiante.

## 5.3. Processador temporal

Uma vez processado e etiquetado gramaticalmente, o texto segue para o processador temporal, o núcleo do sistema. Conforme ilustrado na arquitetura do sistema, compõem o processador de expressões temporais os seguintes módulos: filtro, motor de regras, desambiguador, ancorador, normalizador, registrador de referencial temporal e etiquetador. Além desses módulos, estão associadas a este componente três dicionários: um dicionário de gatilhos, um dicionário de modificadores e um dicionário de datas especiais. Nas subseções abaixo, destacaremos os principais aspectos de implementação desses elementos, começando, a seguir, pelos dicionários, que são referenciados por mais de um módulo do processador temporal e essenciais para o entendimento do funcionamento da abordagem.

### 5.3.1. Dicionário de gatilhos

O dicionário de gatilhos é uma base de dados englobando todas as palavras com conteúdo temporal que, quase sempre, são o núcleo das expressões temporais. Sua representação em memória é feita com a estrutura de dados de dicionário da linguagem Python. Uma entrada nesse dicionário tem como chave a própria palavra e como valor um conjunto de pares de atributo e valor que chamamos coletivamente de traços. Esse conjunto de traços também é implementado através de uma estrutura de dados de dicionário em Python. Os traços são muito importantes, pois são eles que carregam toda a semântica a partir da qual o sistema será capaz de interpretar e normalizar as expressões temporais de que os gatilhos fazem parte. A Figura 5.4 mostra exemplos de entradas do dicionário de gatilhos.

| **Palavra** | março |
|---|---|
| **Traço** | **Valor** |
| tipo | mes |
| unidade | M |
| mes | 3 |

| **Palavra** | dia |
|---|---|
| **Traço** | **Valor** |
| tipo | unidade |
| unidade | D |

| **Palavra** | sábado |
|---|---|
| **Traço** | **Valor** |
| tipo | dia |
| unidade | D |
| dia-semana | 6 |

| **Palavra** | meio-dia |
|---|---|
| **Traço** | **Valor** |
| tipo | hora |
| unidade | TH |
| hora | 12 |

| **Palavra** | manhã |
|---|---|
| **Traço** | **Valor** |
| tipo | turno |
| unidade | T |
| turno | MO |

| **Palavra** | ontem |
|---|---|
| **Traço** | **Valor** |
| tipo | dia |
| unidade | D |
| ref | E |
| direcao | - |
| delta | 1 |

| **Palavra** | diário |
|---|---|
| **Traço** | **Valor** |
| tipo | freq |
| unidade | D |

| **Palavra** | segundos |
|---|---|
| **Traço** | **Valor** |
| tipo | unidade |
| unidade | TS |
| plural | 1 |

**Figura 5.4. Exemplos de entradas do dicionário de gatilhos**

Note que os traços, as unidades e os valores tendem a se assemelhar aos conceitos introduzidos nas diretivas do padrão de anotação proposto no capítulo anterior. Isso é intencional e tem como intuito facilitar a posterior normalização das expressões temporais. As entradas no dicionário incluem o máximo de informação semântica possível, dados estes que serão complementados e agregados ao passo em que as regras de processamento temporal forem sendo executadas.

O exemplo da entrada da palavra “ontem” ilustra a quantidade de informações que uma única palavra gatilho pode fornecer: a simples ocorrência dessa palavra indica que a expressão temporal da qual ela faz parte tem granularidade de dia, é ancorável, é relativa (ao momento de enunciação) e refere-se ao passado, mais precisamente, ao dia anterior à hoje (sendo hoje, na teoria, o momento da enunciação ou, na prática, da criação ou publicação do documento que contém a ET).

Por fim, vale ressaltar que, apesar de termos demonstrado acima que a chave das entradas no dicionário de gatilhos é a palavra, algumas expressões (no sentido de serem compostas por mais de uma palavra) também podem se qualificar como tal. Um bom exemplo seria a expressão “depois de amanhã”, que seria o simétrico de “anteontem” no futuro. Nesse caso, tendo como base a entrada da palavra “ontem”, conforme ilustrada na Figura 5.4, a entrada que tem como chave a palavra (expressão) “depois de amanhã” difere apenas nos traços “direcao” e “delta” que têm como valores, respectivamente, “+” (indicando o futuro) e “2” (indicando que devem ser somados dois dias ao referencial).

## 5.3.2. Dicionário de modificadores

O dicionário de modificadores segue a mesma ideia e estrutura do dicionário de gatilhos, sendo, entretanto, composto de entradas que têm como chave palavras acessórias à construção do significado das expressões temporais. Além de melhor qualificá-las, esses modificadores são importantes, em particular, na resolução das ambigüidades comuns a essas expressões, sendo essenciais para possibilitar uma normalização precisa. A Figura 5.5 exemplifica algumas entradas correspondentes a modificadores comumente encontrados associados a expressões temporais. Vale ressaltar que essas palavras podem ocorrer em outros contextos. O processador temporal, contudo, só faz uso desses dados quando os modificadores são, de fato, parte integrante de uma expressão temporal.

| **Palavra** | passado |
|---|---|
| **Traço** | **Valor** |
| tipo | mod |
| mod | rel |
| ref | E |
| direcao | - |
| delta | 1 |

| **Palavra** | seguinte |
|---|---|
| **Traço** | **Valor** |
| tipo | mod |
| mod | rel |
| ref | T |
| direcao | + |
| delta | 1 |

| **Palavra** | até |
|---|---|
| **Traço** | **Valor** |
| tipo | mod |
| mod | lim |
| lim | <= |

| **Palavra** | de idade |
|---|---|
| **Traço** | **Valor** |
| tipo | mod |
| mod | idade |

**Figura 5.5. Exemplos de entradas do dicionário de modificadores**

O traço “mod”, exclusivo para entradas deste dicionário, indica que tipo de modificador a palavra é: o valor “rel” identifica modificadores relativos, “lim” é usado para modificadores que introduzem um limite à expressão temporal e “idade” distingue modificadores que servem como pistas de que uma ET que poderia ser classificada como duração é, na realidade, uma idade.

## 5.3.3. Datas especiais

O dicionário de datas especiais é responsável por reconhecer e traduzir expressões temporais implícitas referentes a nomes de feriados, datas comemorativas ou eventos que carreguem consigo um significado temporal não explícito diretamente no texto, mas que podem ser mapeadas e ancoradas na linha do tempo. O dicionário contém dois tipos de entradas: uma correspondente a expressões que se traduzem em datas do calendário e outra dedicada a expressões que, na maioria das vezes, dizem respeito a períodos do calendário (como “Carnaval” ou “Semana Santa”). Exemplos dos dois tipos de entradas do dicionário de datas especiais são apresentados na Figura 5.6. As expressões presentes

neste dicionário também são consideradas como gatilhos para efeitos de filtragem e reconhecimento de expressões temporais.

| Expressão | Ano | Mês | Dia |
|---|---|---|---|
| Quarta-feira de cinzas | 2011 | 3 | 9 |
| Quarta-feira de cinzas | 2010 | 2 | 17 |
| Dia dos Namorados | XXXX | 6 | 12 |
| Dia da Independência | XXXX | 9 | 7 |
| Natal | XXXX | 12 | 25 |
| Ano Novo, Réveillon | XXXX | 12 | 31 |

| Expressão | Ano | Período |
|---|---|---|
| Carnaval | 2011 | 03-05 / 03-09 |
| Carnaval | 2010 | 02-13 / 02-17 |
| Semana Santa | 2011 | 04-22 / 04-24 |

**Figura 5.6. Exemplos de entradas do dicionário de datas especiais**

Cabe lembrar que algumas vezes é difícil capturar a intenção original expressa no texto quando encontramos referências a eventos como "Natal" ou "Carnaval". Quando o texto se refere ao Natal, pode compreender apenas o dia 25 ou o período que começa na noite do dia 24 e vai até o final do dia 25. De modo similar, o Carnaval pode significar apenas a terça-feira de Carnaval, como todo o período que, para alguns, começa na sexta-feira e, para outros, no sábado, se estendendo até a Quarta-Feira de Cinzas.

Essas expressões podem ainda ser usadas de forma genérica em frases como "Adoro o Carnaval". Na abordagem executada neste trabalho, tentamos distinguir apenas entre o uso ancorável e o uso genérico dessas expressões, adotando, naquele caso, quando da normalização da ET em uma data ou período preciso do calendário, sempre a intenção que consideramos ser a mais comum (nos exemplos citados, dia 25 de dezembro para as ocorrências ancoráveis de "Natal" e o período de sábado até a Quarta-Feira de Cinzas, para as de "Carnaval").

### 5.3.4. Filtro

Conforme mencionado no capítulo anterior, o filtro usa o dicionário de gatilhos para eliminar frases que não contenham alguma palavra com conteúdo temporal, evitando, assim, a tentativa desnecessária de aplicação das diversas regras do sistema e, consequentemente, acelerando o processamento. Além dos gatilhos, o filtro também verifica a existência de padrões expressos por expressões regulares usadas pelo módulo

de regras iniciais (usadas para detectar, por exemplo, datas em formatos regulares como "11/09/1985"). Em outras palavras, uma sentença só passa pela cadeia de processamento temporal se contiver pelo menos um dos gatilhos presentes no dicionário de gatilhos ou contiver *tokens* que casem com uma das expressões regulares utilizadas pelas regras iniciais.

### 5.3.5. Regras iniciais

Antes de ser processada pela *pipeline* de regras, cada *token* da sentença é transformado de um par (composto pela palavra e sua etiqueta POS) – a saída do processador lingüístico – para a estrutura de dicionário de Python (um tipo de vetor associativo, também conhecido como *hash* ou tabela *hash*). Usaremos, sempre que necessária, a própria notação da linguagem Python para pares (tuplas), listas e dicionários (Tabela 5.3). Cabe apontar que os valores indicados na sintaxe podem ser não apenas tipos primitivos da linguagem, mas também outras estruturas e instâncias de objetos. Apesar de incluirmos as aspas simples nos exemplos abaixo, deste ponto em diante, sempre que irrelevante para a compreensão da estrutura, as omitiremos.

**Tabela 5.3. Notação da linguagem Python para estruturas de tuplas, listas e dicionários**

| **Estrutura** | **Sintaxe** | **Exemplo** |
|---|---|---|
| Tupla | (valor1, valor2, ...) | ('Nasci', 'V') |
| Lista | [valor1, valor2, valor3, ...] | ['Nasci', 'em', '1985', '.'] |
| Dicionário | {chave1: valor1, chave2: valor2, chave3: valor3, ...} | {token: 'Nasci', POS: 'V'} |

Cada *token* é inicialmente representado, portanto, por um dicionário contendo as chaves "token" e "POS", conforme indicado na Tabela 5.4 abaixo. A estrutura de dicionário é útil para agregar traços aos *tokens*, ao passo em que a sentença é processada pelas diversas regras do sistema. Esses traços podem ser provenientes tanto de regras implementadas em código como dos dicionários de gatilhos e modificadores.

**Tabela 5.4. Exemplos de saída e entrada do processador linguístico e das regras, respectivamente**

| **Saída do processador linguístico** | **Entrada das regras** |
|---|---|
| Lista de pares (palavra, POS) | Lista de dicionários {palavra, POS} |
| [(Nasci, V), (em, PRP), (1987, NUM), ('.', '.')] | [{token: Nasci, POS: V}, {token: em, POS: PRP}, {token: 1987, POS: NUM}, {token: '.', POS: '.'}] |

O processamento feito pelas regras iniciais tem como objetivo reconhecer partes de expressões temporais que seguem padrões regulares, números (inclusive, por

extenso) e gatilhos. Tendo como entrada a sentença na forma de uma lista de *tokens* (representados por estruturas de dicionário, como anteriormente explicado), conforme exemplificado na Tabela 5.4 acima, cada frase é percorrida *token* a *token* na tentativa de casá-los com uma das regras de expressões regulares ou com um dos *tokens* (ou expressões compostas por múltiplos *tokens*, quando este for o caso) presentes no dicionário de gatilhos. Exemplos de expressões regulares utilizadas nessa etapa do processamento são listados na Tabela 5.5. Além da própria expressão regular correspondente ao padrão procurado, esse tipo de regra traz ainda uma função de casamento que verifica se os valores estão dentro do domínio esperado e, em caso positivo, aplicam os traços relevantes no dicionário do *token*.

**Tabela 5.5. Exemplos de expressões regulares que compõem as regras iniciais**

| **Tipo** | **Expressão regular** | **Exemplos** | **Traços** |
|---|---|---|---|
| Datas | ^(\d{1,2})[\/\.\-](\d{1,2})[\/\.\-](\d{2}\|\d{4})$ | 11/9, 11/09, 11/09/85, 11/9/1985, ... | dia: 11<br>mes: 9<br>ano: 1985 |
| Horas | ^(\d{1,2})[h\:](\d{2})?$ | 9h, 9h15, 9:15, 11:30, ... | hora: 9<br>min: 15 |
| Intervalo de números | ^(\d+)\-(\d+)?$ | 15-18, 97-99, 2003-2006, ... | num_ini: 15<br>num_fim: 19 |
| Números | ^(\d+)$ | 15, 97, 2003, ... | num: 15 |

Tomando como exemplo a regra de expressão regular para datas, a função de casamento (código em Python) associada garante que, além de obedecer ao padrão regular esperado, os componentes do *token* estejam dentro dos limites do domínio correspondente (isto é, dia entre 1 e 31 e mês entre 1 e 12, por exemplo). Estando tudo de acordo, o dicionário do *token* na sentença é estendido com a inclusão dos traços "dia", "mês" e "ano" (se presente). Se o ano não estiver presente, este só será resolvido e inserido no dicionário uma vez que a frase seja processada pelo desambiguador, pelo ancorador e pelo normalizador. Lógica semelhante é executada para as outras regras.

O casamento de *tokens* com gatilhos é feito através de comparações simples entre *strings*. Uma vez casado, os traços do gatilho correspondente são inseridos no dicionário do *token*. No caso de gatilhos compostos por mais de um *token* (por exemplo, "a partir de"), a comparação é feita levando em conta não apenas o *token* atual na sequência, mas também os N - 1 seguintes (sendo N o número de *tokens* que compõem o gatilho). Na ocasião de um casamento de um gatilho composto, os *tokens* correspondentes na sentença são substituídos por um só, englobando todos os outros,

em um processo que chamamos de redução. A Tabela 5.6 ilustra uma redução dessa sorte.

**Tabela 5.6. Exemplo de redução**

| **Antes da redução** | **Depois da redução** |
|---|---|
| [{token: a, ...}, {token: partir, ...}, {token: de, ...}, {token: março, ...}, ...] | [{token: a partir de, lim: >=, …}, {token: março, ...}, …] |
| [{token: Às, ...}, {token: vinte, ...}, {token: e, ...}, {token: duas, ...}, {token: horas, ...}, ...] | [{token: Às, ...}, {token: vinte, num: 20, ...}, {token: e, ...}, {token: duas, num: 2, ...}, {token: horas, ...}, ...] |
| | [{token: Às, ...}, {token: vinte e duas, num: 22, ...}, {token: horas, ...}, ...] |

Por fim, as regras iniciais também incluem o reconhecimento e a redução de números por extenso. Cada palavra representativa de um número tem um traço "num" associado que indica a quantidade inteira, tal qual na regra de expressão regular para números. A diferença neste caso é que um número por extenso pode ser composto por vários *tokens*, que devem ser reduzidos e somados para se chegar ao resultado desejado.

Esse processo é feito em iterações, reduzindo um par de *tokens* por vez. Na primeira iteração, são reconhecidos os *tokens* singulares (no exemplo acima, "vinte" e "duas" separadamente). A partir daí, a cada iteração (se assim for necessário), a função de casamento verifica se o *token* da esquerda é numericamente maior que o *token* da direita, realiza a soma dos traços "num" e guarda o resultado no traço "num" do elemento criado pela redução (novamente, a combinação dos dois dicionários correspondentes aos *tokens* agregados). As duas iterações necessárias para reduzir a expressão "vinte e duas" estão representadas nas duas células da coluna direita da segunda linha da Tabela 5.6 acima.

## 5.3.6. Regras gramaticais

Nesta etapa, são efetivamente reconhecidas as expressões temporais, incluindo, além do gatilho temporal, as preposições e os modificadores cabíveis. Apesar dos gatilhos e modificadores já estarem reconhecidos isoladamente desde o módulo anterior, é neste que eles são reduzidos para formar um único objeto. Este módulo é composto por 127 regras codificadas manualmente, a partir da observação e análise de uma coleção de mais de três mil notícias em português coletadas a partir de sítios da web. Depois de experimentar com alguns formatos, adotamos o padrão descrito a seguir. Toda regra é composta por um número de prioridade, um conjunto de elementos e, opcionalmente,

uma função de casamento. As regras seguem o formato ilustrado na Figura 5.7 (em BNF).

```
<regra>    ::= <prioridade> <elemento> <função>
<traço>    ::= “<” <nome> “:” <valor> “>” | “<” <nome> “>”
<op>       ::= “|” | “&”
<elemento> ::= <traço> | “?” <traço> | “!” <elemento>
               | <elemento> <op> <elemento> | “(“ <elemento> “)”
```

**Figura 5.7. Formato das regras em BNF**

O número de prioridade (um número natural) serve para que, dadas duas ou mais regras passíveis de casamento com os termos em análise, o algoritmo saiba qual deve ser executada (quanto menor o número, mais prioritária a regra). Os elementos que compõem a regra de fato têm duas formas: uma especificando um traço, um operador e um valor e outra, apenas um traço. Esta pede apenas que o termo considerado contenha o traço especificado presente em seu dicionário, enquanto aquela exige, para o casamento do termo da frase com o elemento da regra, que o dicionário do *token* contenha o traço pedido preenchido com o valor indicado.

Para indicar que um elemento é opcional (ou seja, que a sua ausência não invalida o casamento da regra), deve ser inserido um caractere de interrogação (“?”) antes da especificação do traço. Os elementos podem ainda tomar parte em operações de lógica booleana: conjunções são separadas por “&” e disjunções, separadas por “|”, além da negação que é indicada pelo uso de um caractere de exclamação (“!”) antes do elemento. Além disso, essas expressões podem ser agrupadas com parênteses para alterar a precedência dos operadores lógicos.

Por fim, a terceira parte da regra, quando presente, indica o nome de uma função Python a ser invocada sempre que a regra for casada com os termos em análise. Essa função de casamento recebe como parâmetros a frase em análise e a posição inicial a partir da qual a regra foi casada com os termos e pode retornar nulo quando considerar que o casamento for inválido (por exemplo, para dar efeito a uma restrição de domínio que não pode ser representada diretamente pelas regras) ou um novo elemento que agregará e substituirá os termos presentes na frase (mais uma vez, o processo que chamamos de redução). Na ausência de tal função, o simples casamento dos elementos é suficiente, e implica na combinação dos termos casados em um só objeto, formado por

todos os traços dos termos componentes (e adicionado de um novo traço, chamado "tokens", com a lista dos objetos originais reduzidos na operação). Em geral, as funções de casamento são utilizadas para garantir a conformidade de domínios, incluir novos traços ou melhor mesclar os existentes. A Tabela 5.7 lista alguns exemplos das regras implementadas na gramática deste módulo do CoppeTER.

**Tabela 5.7. Exemplos de regras implementadas na gramática do sistema**

| # | P | Regra | Descrição |
|---|---|---|---|
| 1 | 1 | <tipo:mes> <token:de> <num> | Reconhece expressões como "setembro de 1985". A função de casamento inclui o traço "ano" com o valor do traço "num". |
| 2 | 1 | <mod:ind> <tipo:unidade> | Reconhece expressões indeterminadas do tipo "algum dia". A função de casamento inclui o traço "tipo" e um traço com o nome da unidade indeterminada (no caso, respectivamente, "tipo: dia" e "dia: ?"). |
| 3 | 1 | <token:dia> <num> | Reconhece expressões como "dia 11". A função de casamento inclui o traço "tipo: dia" e o traço "dia" com o valor de "num" ("dia: 11"). |
| 4 | 2 | <num> \| <tipo:dia> <token:de> <tipo:mes> | Reconhece expressões como "3 de março", "dia 3 de março", "algum dia de março de 87", etc. Por ser prioridade 2, essa regra sempre será executada depois das regras 1 e 3 (nos casos em que as duas também seriam aplicáveis). Além de alterar o traço "tipo" para "tipo: dia", a função de casamento inclui o traço "dia" quando o primeiro elemento for <num> (ex.: "dia: 11"). |
| 5 | 2 | (<tipo:dia> <tipo:turno> \| <tipo:hora>) \| (<tipo:turno> \| <tipo:hora> <tipo:dia>) | Reconhece expressões que combinem quaisquer tipos de dia reconhecidos anteriormente com turnos e horas (ex.: "dia 3 às 14h", "na manhã de ontem", "às 10 horas de 11/09/85", etc.). |
| 6 | 3 | ?<pos:prep> ?<pos:art> <tipo:ano> \| <tipo:mes> \| <tipo:dia> \| <tipo:turno> \| (<tipo:unidade> <mod:rel>) \| (<mod:rel> <tipo:unidade>) | Reconhece expressões acompanhadas de preposições (ex.: "desde o dia 3") e ETs relativas, como "três dias depois". Vale notar que modificadores como "desde" (que está presente no dicionário de modificadores como indicador de limite) não precisam ser reconhecidos explicitamente (por <mod:lim>), uma vez que <pos:prep> já casa com quaisquer preposições, sejam elas modificadores ou não. |
| 7 | 3 | <pos:prep> <num> | Reconhece expressões do tipo "em 87" ou "em 1985". A função de casamento inclui o traço "tipo: ano" e um traço "ano" com o valor de "num". |
| 8 | 3 | <token:entre> <calendario> <token:e> <calendario> | Reconhece expressões de intervalos de datas como "entre março e setembro". O traço "calendario" está presente em toda ET ancorável. |

As regras são ordenadas internamente de acordo com a prioridade (ascendente) e, depois, o número de elementos (descendente). Em outras palavras, a tentativa de casamento é feita primeiro com as regras mais prioritárias e com o maior número de elementos (no sentido de casar sempre a maior expressão possível). A aplicação das regras é feita de forma sequencial, uma regra de cada vez, a todas as sentenças do texto

(também sequencialmente, da primeira até a última sentença). Para cada sentença, a regra ativa começa pelo primeiro termo da sentença e invoca sucessivamente cada uma das suas cláusulas. Após esse passo, a regra muda o seu posicionamento para um termo à direita, até serem esgotadas todas as combinações possíveis de alinhamento da regra com a sentença. Nesse ponto, passa-se para a regra seguinte, sempre de acordo com a ordem supracitada. Esse processo é repetido até que se esgotem as regras e as sentenças do texto.

### 5.3.7. Desambiguador

As regras descritas na seção anterior trabalham apenas com o contexto imediato das expressões temporais. Há, contudo, expressões que precisam de mais informações para serem corretamente interpretadas. Esse trabalho é realizado pelo desambiguador, que tem por objetivo buscar os insumos necessários para remover as ambiguidades comumente encontradas nas expressões temporais relativas, sejam elas dêiticas ou anafóricas. Esse processo é de grande importância, uma vez que a maioria das ETs encontradas em textos, principalmente jornalísticos, é relativa.

Para precisamente desambiguar esse tipo de expressão, tendo por inspiração o trabalho de AHN *et al.* (2005), identificamos três questões que devem ser tratadas. A primeira delas diz respeito ao tempo de referência da ET relativa, isto é, o foco ou referencial temporal. A normalização correta da expressão temporal parte do pressuposto de que o sistema é capaz de identificar com precisão se a ET é relativa ao momento da enunciação ou ao contexto textual. De fato, o dicionário de modificadores inclui uma gama de expressões que, quando presentes, classificam, sem ambiguidade, uma ET como sendo relativa a um ou ao outro: por exemplo, enquanto a expressão "na próxima semana" é dêitica (isto é, o tempo de referência é o momento da enunciação e, portanto, devemos normalizá-la buscando a data de criação ou publicação do documento), a ET relativa "na semana seguinte" é anafórica (ou seja, faz referência a uma data ou evento anterior presente no texto e, para normalizá-la, precisamos encontrar essa data anterior nas sentenças anteriormente processadas). Nesses exemplos, a presença dos modificadores relativos "próxima" e "seguinte" trariam consigo, além de outros, o traço "ref", conforme exemplificado na seção sobre o dicionário de modificadores, que indicariam para o sistema, sem ambiguidade, que referencial temporal utilizar para ancorar e normalizar essas expressões temporais.

Todavia, há expressões temporais relativas que não trazem consigo quaisquer modificadores indicativos de referencial temporal como os vistos acima. No primeiro exemplo da Tabela 5.8, a ET "Na quinta-feira" da segunda frase do trecho exposto tem como tempo de referência (anafórico) a data mencionada na sentença anterior ("na quarta-feira, dia 16 do mês passado", esta, por sua vez, é dêitica). Essa mesma expressão temporal é encontrada também, em outro contexto, no segundo exemplo, mas, desta vez, seu foco temporal é dêitico. Essa distinção nem sempre é clara, podendo ser, inclusive, em alguns raros casos, de difícil classificação até mesmo para um indivíduo. As duas últimas linhas da tabela exemplificam dois casos que são simples de distinguir para pessoas, mas não triviais de fazê-lo algoritmicamente: no primeiro deles, a ET é uma frequência (equivalente a "toda quinta-feira"), ao passo em que, no segundo, trata-se de uma ET genérica.

**Tabela 5.8. Exemplos de diferentes focos temporais para uma mesma ET**

| **Exemplo** | **Foco temporal** |
|---|---|
| Pedro sumiu na quarta-feira, dia 16 do mês passado. *Na quinta-feira*, seu corpo foi encontrado. | Anafórico |
| Pedro sumiu no início do mês. *Na quinta-feira*, seu corpo foi encontrado. | Dêitico |
| *Na quinta-feira*, muitos fortalezenses saem para comer caranguejo. | Nenhum |
| *Quinta-feira* é o dia que mais gosto na semana. | Nenhum |

O segundo desafio para a desambiguação, e a consequente normalização, de expressões temporais relativas, é o chamado problema da direção (AHN *et al.*, 2005), que consiste em determinar se uma expressão se refere a um ponto no tempo antes ou depois do referencial temporal. Essa informação também é essencial para a interpretação precisa das ETs relativas. A Tabela 5.8 traz dois exemplos simples de como uma mesma ET, sem uso de qualquer informação contextual, pode ser ambígua quanto à direção a partir do tempo de referência. Em ambos os casos, o modo e o tempo do verbo principal da sentença são decisivos na resolução do que seria uma expressão ambígua sem o conhecimento dessa informação contextual. Novamente, a presença de modificadores relativos como "próxima" ou "seguinte", além de trazerem consigo a informação que identificaria sem ambiguidade o referencial temporal, também trariam um traço referente à direção e o deslocamento da expressão temporal a partir do seu referencial. Não sendo esse o caso, caberá ao desambiguador essa resolução.

**Tabela 5.9. Exemplos de ambiguidade quanto à direção**

| **Exemplo** | **Direção** |
|---|---|
| *Na quinta-feira*, fui ao cinema. | Antes do momento da enunciação |
| *Na quinta-feira*, vou ao cinema. | Depois do momento da enunciação |

Por fim, uma última questão que deve ser atacada para conseguirmos implementar uma abordagem de desambiguação de boa precisão – em particular, em textos do gênero jornalístico – é quanto à ambiguidade semântica decorrente do uso da palavra "hoje". Essa tarefa consiste em determinar se uma ocorrência desse termo é específica ou genérica, isto é, se ela se refere ao momento de enunciação ou ao presente (i.e. se ela tem o mesmo sentido de "atualmente", por exemplo). Apesar de se restringir a uma única palavra, esse caso é comum o suficiente que para ser tratado de modo específico (no corpus de três mil notícias com o qual trabalhamos, foram encontradas mais de 500 instâncias desse tipo de ambiguidade). Vale ressaltar que textos de outros gêneros, como o literário, talvez, não se beneficiassem tanto com a resolução deste problema. Para o gênero jornalístico, contudo, percebe-se ser uma tarefa de grande importância.

Problemas como esses são mais bem tratados por métodos estatísticos de aprendizado de máquina, capazes de aprender a partir de diversos exemplos de ocorrências similares. MANI *et al.* (2000) abordaram questões análogas com o emprego de regras, não tendo obtido resultados suficientemente bons. Optamos por, novamente, seguir a linha de AHN *et al.* (2005) e abordar esses casos com o uso de classificadores de máxima entropia, introduzidos no Capítulo 3 desta dissertação.

Antes de enumerarmos os atributos de entrada utilizados na composição dos classificadores implementados neste módulo desambiguador, cabe detalharmos o processo através do qual o sistema obtém um dos mais importantes desses atributos: o tempo verbal. Esse atributo é utilizado pelos três classificadores, sendo, inclusive, imprescindível para uma correta interpretação da direção da expressão temporal. Como nosso etiquetador POS não é capaz de fornecer essa informação, fazemos uso do etiquetador externo TreeTagger e do *software* gconjugue (versão 0.7.2) para obtê-la. O gconjugue (GCONJUGUE, 2011) é um *software* de conjugação verbal para a língua portuguesa baseada no conjugue (CONJUGUE, 2011), um conjugador desenvolvido desde 1995 que faz uso de um banco de verbos e paradigmas para cobrir a quase totalidade dos verbos de uso corrente, e no br.ispell (BR-ISPELL, 2011), um dicionário (vocabulário) de palavras em português.

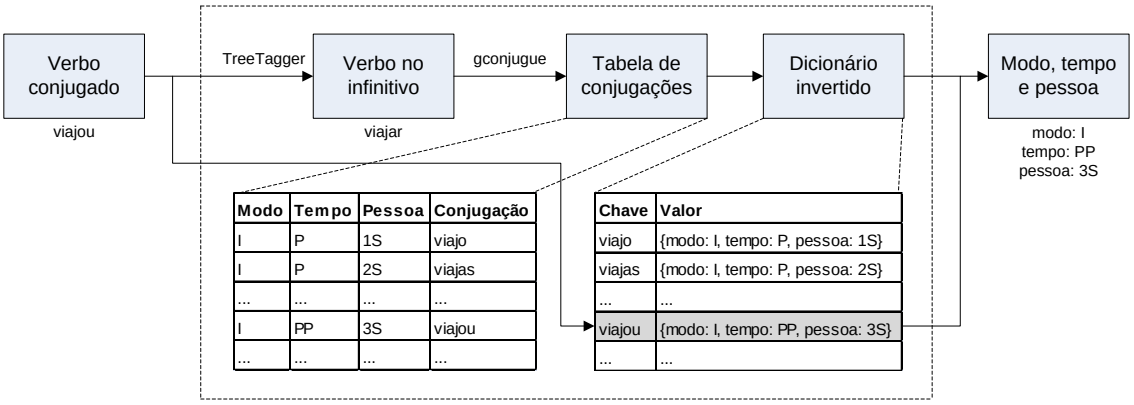



**Figura 5.8. Exemplo do processo de identificação da conjugação verbal**

O procedimento pormenorizado a seguir, em função do *overhead* em se executar dois programas externos, só é executado quando necessário, a saber: quando é preciso tratar de um dos problemas citados acima e o verbo da sentença sendo analisada não está em *cache*. O etiquetador TreeTagger é invocado para classificar morfologicamente a frase, pois, além da etiqueta POS, ele fornece a forma nominal infinitiva dos verbos encontrados. De posse disso, o CoppeTER executa o *software* gconjugue passando como parâmetro o verbo no infinitivo e obtendo como resposta todas as formas nominais do verbo, bem como a tabela completa de suas conjugações em todos os modos, tempos e pessoas verbais. Esses dados são armazenados em um dicionário invertido (que também funciona como o *cache* citado acima) tendo como chave cada conjugação do verbo e como valor um dicionário indicando o modo, o tempo e a pessoa verbal da conjugação. Desse modo, a partir do verbo conjugado da frase, conseguimos identificar precisamente a que modo, tempo e pessoa verbal ele está associado. Esse procedimento está ilustrado na Figura 5.8.

Os três classificadores de máxima entropia utilizados pelo CoppeTER para desambiguar expressões temporais relativas utilizam um conjunto praticamente idêntico de atributos de entrada, em sua maioria, léxicos, a saber: a posição da ET dentro da sentença (uma fração entre 0 e 1, proporcional à posição da ET na sentença, sendo 0 no início e 1 no final), a janela de contexto (de duas palavras), o verbo principal no infinitivo, o verbo principal na conjugação original da frase, seu modo verbal, seu tempo verbal, a pessoa e um atributo derivado do tempo verbal que aceita apenas três valores (passado, presente e futuro). A janela de contexto é formada por oito atributos, quais sejam: as duas palavras anteriores à expressão temporal, as duas palavras seguintes e as quatro etiquetas POS de cada uma delas.

**Tabela 5.10. Exemplos dos atributos de entrada utilizados pelos classificadores**

| ET | Tipo | Pos. | P-2 POS | P-1 POS | P+1 POS | P+2 POS | VI | VC | M | T | PE | T' | Classif. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| no domingo | dia | 1,00 | chover V | fraco ADJ | | | voltar | volta | I | P | 3S | 0 | DIR: + |
| no sábado | dia | 0,33 | tentei V | vir V | , , | mas CONJ | tentar | tentei | I | PP | 1S | -1 | DIR: - |
| hoje | | 0,00 | | ainda ADV | , , | cerca ADV | ser | são | I | P | 3P | 0 | DIA: 0 |
| hoje | | 0,30 | que CONJ | participou V | de PRP | reunião N | participar | participou | I | PP | 3S | -1 | DIA: 1 |

O classificador de foco temporal ainda leva em conta alguns atributos de entrada adicionais, entre eles o referencial temporal da última ET identificada no texto antes da expressão temporal em análise. Além deste, também é incluído, tanto no classificador de foco temporal quanto no classificador de direção, um atributo de entrada com o tipo da expressão temporal (obtido através do traço "tipo" presente no dicionário). A Tabela 5.10 elenca alguns exemplos dos atributos utilizados, em uma representação similar à utilizada pelo sistema internamente. A última coluna da tabela indica a classificação correta de cada instância, sendo os dois primeiros exemplos do classificador de direção e os dois últimos do classificador de significado do termo "hoje", indicando se seu uso é específico (1) – isto é, referente ao dia de hoje – ou genérico (0).

## 5.3.8. Registrador de referencial temporal

O registrador de referencial temporal mantém uma lista das expressões temporais ancoráveis reconhecidas e normalizadas até o momento dentro do documento. Sempre que o processamento temporal de uma sentença é encerrado (no módulo normalizador), suas expressões temporais são acrescentadas ao final desta lista, na ordem em que aparecem no texto, para serem recuperadas depois, caso uma ET relativa anafórica faça referência a uma delas posteriormente em outra sentença, pelo módulo ancorador, conforme explicaremos na próxima seção.

Nos casos em que uma sentença contenha mais de uma expressão temporal, o registrador a quebra em vários segmentos de acordo com a pontuação presente (vírgulas, parênteses, dois pontos, etc.) e, em vez de incluir na lista a sentença inteira, registra cada segmento (que contiver uma expressão temporal reconhecida e normalizada) separadamente (sempre na ordem em que aparecem no texto). Manter a ordem das expressões encontradas no registrador é importante para corretamente

ancorar expressões como “dois dias depois” que, em geral, se referem à ET reconhecida imediatamente antes. Quebrar sentenças com múltiplas expressões temporais, por outro lado, auxilia na ancoragem de expressões como “dois dias depois da entrevista”, que têm como referencial uma data ou evento anterior do texto.

### 5.3.9. Ancorador

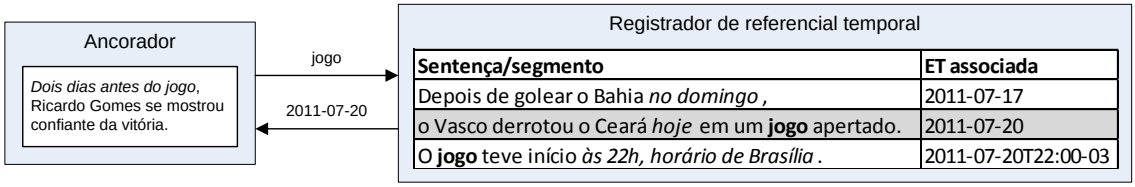
O ancorador é o módulo responsável por, a partir dos traços presentes na estrutura de dicionário representativa da expressão temporal, identificar, calcular e registrar o ponto na linha do tempo a partir do qual a ET em análise está deslocada. Para expressões temporais dêiticas, a ancoragem deve usar como referencial o momento da enunciação. A abordagem utilizada pelo sistema para esse tipo de ET relativa é simples e direta: atribuímos a data de publicação do documento (DCT) como o ponto referencial na linha temporal. Essa data, em geral, está presente de alguma forma nos metadados do documento, sejam eles, dependendo da origem e da forma de obtenção, atributos no sistema de arquivos, cabeçalhos no protocolo HTTP, *tags* no documento HTML ou até mesmo diretamente no início ou no fim do texto.

É comum, quando a origem do documento é a web, termos acesso a mais de uma data passível de ser considerada a data de publicação (e, portanto, o momento de enunciação), quais sejam: a data da última modificação do documento no cabeçalho HTTP (Last-Modified), as datas de criação e última alteração nas *tags* <meta> do HTML e, possivelmente, uma data de publicação e uma data de última atualização em texto no HTML. Nesses casos, a escolha da DCT pode não ser tão trivial. Apesar de termos implementado um procedimento para tal na construção do corpus de três mil notícias utilizado na análise e em alguns dos experimentos (no nosso caso, usamos a heurística de seguir exatamente a ordem acima invertida, escolhendo, portanto, a data mais ao fim da lista que estivesse presente), o CoppeTER, por padrão espera que a DCT seja fornecida como entrada juntamente com o documento.

A ancoragem de expressões temporais anafóricas, por sua vez, faz uso do registrador de referencial temporal para tentar identificar os pontos na linha do tempo aos quais essas ETs são relativas. Para as instâncias de expressões temporais como “dois dias depois” ou “na quinta-feira seguinte”, utilizamos o modelo simples de registro de foco temporal mais recente (*recency-based temporal focus tracking*), empregado pela maioria das abordagens da literatura (AHN *et al.*, 2005) e que consiste,

basicamente, em utilizar como âncora dessas ETs a última expressão temporal de granularidade compatível reconhecida e normalizada pelo sistema.

Finalmente, para expressões temporais anafóricas que fazem referência a eventos anteriores do texto, como "dois dias depois da entrevista" ou "uma semana antes do grande jogo", desenvolvemos uma abordagem para ancorar os casos mais simples. A ancoragem dessas ETs consiste em procurar o substantivo descritor do evento nas sentenças (e segmentos de sentenças) gravadas no registrador de referencial temporal, dando prioridade à mais recente e com granularidade mais compatível no caso de encontrar mais de uma, e utilizar a expressão temporal associada à sentença encontrada como âncora da ET em análise.




| Sentença/segmento | ET associada |
|---|---|
| Depois de golear o Bahia *no domingo* , | 2011-07-17 |
| o Vasco derrotou o Ceará *hoje* em um **jogo** apertado. | 2011-07-20 |
| O **jogo** teve início *às 22h, horário de Brasília* . | 2011-07-20T22:00-03 |

**Figura 5.9. Exemplo do processo de ancoragem**

A Figura 5.9 acima ilustra o processo descrito. Note que, no exemplo da figura, duas entradas do registrador contêm o substantivo caracterizador do evento ao qual a ET da sentença em processamento pelo ancorador faz referência ("jogo"). Pelo critério de dar prioridade à mais recente, a ET escolhida seria aquela associada à última entrada da lista ("2011-07-20T22:00-03"). Contudo, pelo critério adicional da granularidade mais compatível, a segunda entrada do registrador é selecionada – pois é a mais recente que especifica apenas a data, sem a hora – e, consequentemente, a ET correta ("2011-07-20") é retornada ao módulo ancorador. Em contraste, caso a expressão temporal em análise pelo ancorador fosse "duas horas antes do jogo", a última entrada seria a escolhida, devolvendo a ET "2011-07-20T22:00-03" como resposta.

O reconhecimento desses eventos é feito por um conjunto de regras simples que identificam substantivos, substantivos precedidos de adjetivos, entre algumas construções. Em virtude de o sistema não possuir módulos de processamento sintático e semântico, tampouco um módulo reconhecedor de eventos robusto, expressões temporais dessa sorte que fujam do conjunto dessas regras implementadas no CoppeTER (em particular, aquelas em que o evento é descrito por uma oração subordinada), infelizmente, quase nunca são corretamente ancoradas.

## 5.3.10. Normalizador

Como a estrutura interna utilizada pelo CoppeTER muito se assemelha com o padrão de anotação adotado e em virtude de boa parte do trabalho que poderia ser realizado por este módulo ser executado no âmbito das funções de casamento, o trabalho do normalizador é bastante facilitado. Com base nos traços acrescentados ao dicionário da expressão temporal em análise até este ponto, o módulo normalizador, em um processo que combina a verificação do tipo juntamente com a da presença de traços específicos, normaliza as informações temporais que compõem a definição da ET.

No tratamento de expressões temporais relativas, o normalizador é o responsável por normalizar o valor de deslocamento temporal de acordo com a unidade e calcular o ponto final na linha do tempo, levando em conta o referencial temporal, a direção e o delta de deslocamento normalizado. Somente após esse cálculo (para ETs relativas) é que o módulo é capaz de preencher o traço "val" (ou "ref_val", para durações relativas), que corresponde ao valor final normalizado da expressão temporal.

Uma vez encerrado o procedimento de normalização, conforme mencionado anteriormente, a sentença e as expressões temporais encontradas são submetidas ao registrador de referencial temporal, onde serão segmentadas (quando necessário) e inseridas na lista de entradas reconhecidas e normalizadas, para o caso de uma eventual necessidade de recuperação em resposta ao processo de ancoragem de uma ET relativa anafórica posterior.

## 5.3.11. Etiquetador

O módulo derradeiro da cadeia do processador temporal do sistema é incumbido de transformar a sentença de uma representação interna baseada em estruturas de dicionário de volta para o formato original em texto corrido, exceto que, desta vez, com as expressões temporais presentes devidamente reconhecidas, normalizadas e anotadas. Conforme já dito previamente, a estrutura interna resultante da normalização e, em última instância, de todo o processamento temporal realizado pelo CoppeTER, foi propositalmente implementada em sintonia com os atributos esperados pelo padrão de anotação especificado na proposta. Assim, o esforço despendido no processo de etiquetagem é significativamente reduzido.

Em face dessa similaridade, no caso de o esquema de anotação escolhido pelo usuário do sistema ser o padrão (isto é, o especificado na proposta), o trabalho do

etiquetador se resume a converter a lista dos *tokens* (novamente, que são instâncias de estruturas de dicionário da linguagem Python) que compõem a sentença em texto plano (seguindo as regras de formatação relativas à pontuação e afins), construir as etiquetas SGML de acordo com os traços requeridos para anotação de cada tipo de ET e, por fim, inserir essas *tags* em volta das expressões temporais identificadas.

Como o esforço para o desenvolvimento de um esquema de avaliação, a implementação de um programa avaliador e a construção de uma coleção dourada para o padrão de anotação especificado neste trabalho seria significativo, para sermos capazes de avaliar este trabalho de forma repetida com alguma facilidade e agilidade, implementamos duas subclasses do etiquetador padrão: uma para anotar o texto nos padrões TIMEX2 e TimeML (não criamos uma terceira subclasse para a TimeML, pois sua especificação TIMEX3 é praticamente idêntica ao padrão adotado no TIMEX2) e outra para fazê-lo no padrão do Segundo HAREM.

Apesar de termos considerado utilizar a linguagem XSLT (XSLT, 2011) para realizar tal transformação diretamente do documento anotado no esquema do CoppeTER para os padrões supracitados, optamos pela implementação em código Python por considerarmos ser este o caminho mais rápido e flexível. Em ambos os casos, em virtude de o esquema de anotação adotado neste trabalho ser mais granular que o desses padrões, o esforço para a adaptação da etiquetagem na implementação dessas subclasses foi relativamente pequeno. No caso do Segundo HAREM, em particular, esse mapeamento foi mais fácil, pois, além de ser menos complexo, o padrão prevê uma quantidade significativamente inferior de atributos a serem anotados.

# 6. Avaliação

Para medir o desempenho de um sistema, é imprescindível conduzir experimentos e avaliar os resultados alcançados. Nesse sentido, para termos uma avaliação abrangente do trabalho, submetemos o CoppeTER a dois experimentos distintos, sendo um deles para medir a precisão dos três classificadores utilizados pelo módulo desambiguador e outro para avaliar o desempenho do sistema como um todo na tarefa de reconhecimento e normalização de expressões temporais. Este seguiu o padrão de avaliação da comunidade Linguateca/HAREM, uma vez que, conforme explicitado anteriormente, além de a especificação de um padrão de avaliação próprio, a implementação de um programa avaliador seguindo essas diretivas e a construção de uma coleção dourada demandarem esforço considerável, uma abordagem própria de avaliação dificultaria a comparação dos resultados obtidos pelo sistema com o estado da arte da literatura.

## 6.1. Corpora

No desenvolvimento desta dissertação, trabalhamos com dois corpora: um coletado manualmente em sites de notícias em português, composto por 3.138 documentos sem qualquer anotação temporal, e o corpus utilizado na avaliação conjunta do Segundo HAREM, constituído de 129 documentos. Escolhemos compor o corpus construído manualmente por textos do gênero jornalístico pela qualidade gramatical e ortográfica (não raro, textos de *blogs* e outros sítios menos formais, além de não seguirem a norma culta da língua portuguesa, apresentam um texto menos limpo, podendo, portanto, introduzir ruído nos experimentos) e pela facilidade de obtenção desse material programaticamente, uma vez que praticamente qualquer sítio de notícias provê uma ou mais *feeds* de RSS (um formato de publicação para conteúdo frequentemente atualizado, originalmente chamado de *RDF Site Summary*, mas mais conhecido atualmente como *Really Simple Syndication*).

O corpus do Segundo HAREM tem 7.847 EMs anotadas em sua coleção dourada, sendo as expressões temporais responsáveis por 15,21% dessas entidades mencionadas (mais precisamente, por 1195 delas). Todas essas ETs foram, no mínimo, anotadas com os atributos do HAREM clássico: CATEG, TIPO e SUBTIPO. Em

virtude do esforço necessário para tal, no entanto, apenas 19% dessas expressões temporais foram completamente normalizadas, tendo, portanto, todos os atributos possíveis devidamente preenchidos. Essa subcoleção é constituída por 304 parágrafos e 12.992 palavras. A distribuição de tipos mostra uma prevalência (mais de 80%) de expressões temporais de data e hora (MOTA *et al.*, 2008).

## 6.2. Avaliação do desambiguador

Conforme detalhado na seção específica da implementação (5.3.7), o desambiguador é composto por três classificadores de máxima entropia que empregam um conjunto similar de atributos de entrada em sua composição. Pormenorizaremos os procedimentos de treinamento e teste, além de analisarmos o desempenho e a origem dos erros, de cada um desses classificadores a seguir.

### 6.2.1. Classificadores de foco temporal e de direção

Estes classificadores usam o mesmo conjunto de atributos de entrada, conforme descrito no capítulo anterior, à exceção do atributo referente ao referencial temporal da última ET reconhecida e normalizada no texto antes da expressão temporal em análise, que está presente apenas no classificador de foco temporal. Como não temos acesso a um conjunto de dados de treinamento para o experimento (isto é, uma lista de expressões temporais, já classificadas com foco temporal e direção corretos, a partir da qual o classificador pode, de fato, aprender), tivemos que construir tal base manualmente.

Há potencialmente uma infinidade de expressões temporais relativas, nos mais variados formatos e combinações léxicas. O esforço para avaliar completamente um corpus significativamente grande de notícias e classificar as ET relativas presentes em cada uma delas seria elevado, por isso, optamos por preparar o conjunto de dados a partir de algumas poucas variações dessas ETs, facilmente encontradas através de procedimentos simples de programação. Dessa forma, buscamos em nossa base de notícias por ocorrências de sentenças com expressões temporais relativas em formatos simples de se procurar programaticamente, quais sejam: ETs que se referem a um dia da semana ou a um mês isoladamente (sem quaisquer dos modificadores previstos no dicionário) e que são, na ausência de quaisquer outras informações contextuais, por definição, ambíguas quanto ao foco referencial e à direção. A Tabela 6.1 abaixo ilustra a

quantidade de expressões encontradas entre as mais de três mil notícias para cada variação buscada.

**Tabela 6.1. Quantidade de ETs encontradas para cada variação buscada**

| Expressão | Qtd. | Expressão | Qtd. | Expressão | Qtd. |
|---|---|---|---|---|---|
| sábado | 66 | janeiro | 20 | agosto | 17 |
| domingo | 43 | fevereiro | 11 | setembro | 16 |
| segunda | 61 | março | 19 | outubro | 13 |
| terça | 43 | abril | 18 | novembro | 17 |
| quarta | 38 | maio | 15 | dezembro | 19 |
| quinta | 56 | junho | 14 | **Total** | **551** |
| sexta | 52 | julho | 13 | | |

Os 514 documentos contendo essas 551 ocorrências foram, então, inseridos em uma base de dados MySQL (MYSQL, 2011) e processadas pelo CoppeTER para identificar as outras ETs do texto e derivar os traços das selecionadas através do desambiguador (que, até o momento, estava incompleto e gerando apenas o vetor de traços para classificação). Em seguida, de posse dos vetores de traços, realizamos a classificação manual de cada uma das 551 ETs quanto ao foco referencial – momento da enunciação (“E”), textual (“T”), frequência (“F”), genérica (“G”) ou indefinida (“I”), estes três últimos, que, na realidade, não são classificações válidas de foco temporal, são usados para distinguir os casos de expressões temporais de frequência, genéricas e indefinidas, que podem, à primeira vista, parecer ETs relativas normais – e à direção – anterior (“-“) ou posterior (“+”) em relação ao foco temporal – através de uma interface web em PHP (uma linguagem de programação dinâmica conhecida pela facilidade de ser usada para construção de sistemas web) (PHP, 2011) construída para tornar essa tarefa tão fácil e ágil quanto possível.

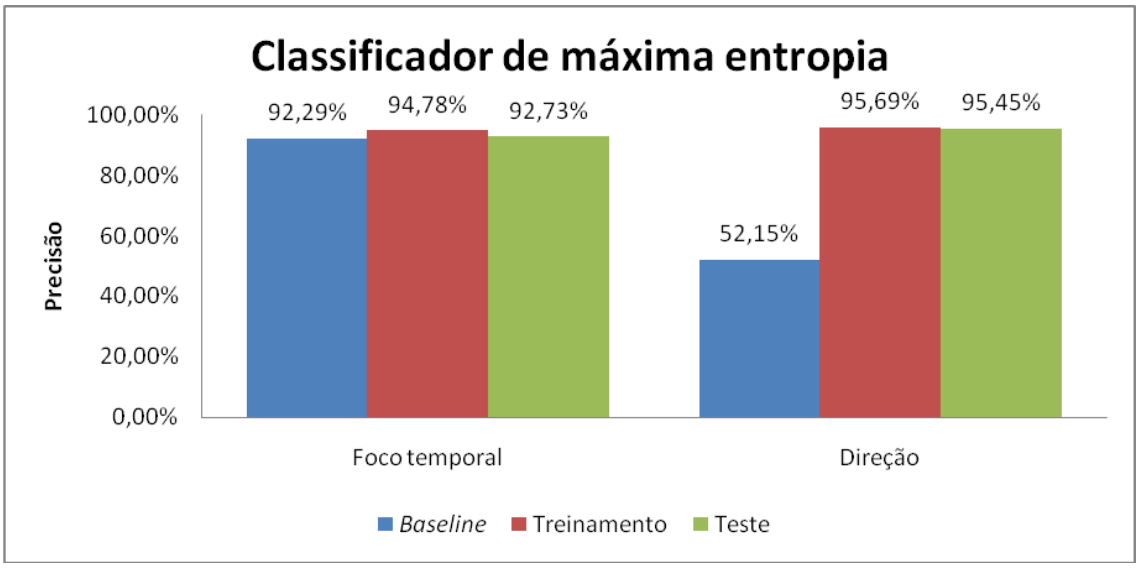



**Figura 6.1. Resultados dos classificadores de foco temporal e de direção**

Assim, com o conjunto de dados completamente classificado, separamos aleatoriamente 80% dos dados (441 expressões temporais) para treinamento e 20% (110 ETs), para teste. Treinamos um classificador de máxima entropia (usando a implementação padrão disponibilizada no pacote NLTK) para cada uma dos atributos-alvo, tendo alcançado, para a classificação do foco temporal, 94,78% de precisão no conjunto de treinamento e 92,73%, no conjunto de teste. Para a classificação da direção das expressões temporais, o classificador atingiu 95,69% de precisão no conjunto de treinamento e ligeiramente menos (95,45%), no de teste. Em ambos os experimentos, a *baseline* adotada é simplesmente atribuição da classe de maior frequência ("E" para o classificador de foco temporal e "-" para o classificador de direção). A Figura 6.1 ilustra comparativamente esses resultados.

No caso do foco temporal, em virtude dos dados serem altamente enviesados (mais de 92% das instâncias são dêiticas), o classificador de máxima entropia melhorou pouco o resultado que seria obtido pelo simples uso da *baseline*, sendo, inclusive, estatisticamente insignificante no conjunto de teste. O enviesamento dos dados tem como explicação o fato de a quase totalidade dos documentos utilizados serem do gênero jornalístico, gênero no qual, pela nossa observação, a maioria das ETs relativas é mesmo dêitica. A condução de um experimento similar tendo como base uma coleção mais heterogênea de documentos (e, portanto, mais representativa dos diversos gêneros textuais existentes) se faz necessária para uma melhor análise do desempenho de classificadores estatísticos para esta tarefa.

A classificação da direção, por outro lado, demonstrou resultados mais condizentes com a faixa de valores esperada para o experimento. Diferentemente da classificação do foco temporal e dos resultados reportados em empreitada semelhante por AHN *et al.* (2005), o classificador de máxima entropia alcançou, nesta tarefa, melhoria expressiva em cima do *baseline* proposto. Como imaginamos intuitivamente que textos jornalísticos costumem tratar mais do passado (apesar de quase sempre recente, obviamente) do que do futuro, suspeitávamos encontrar significativamente mais expressões relativas com direção anterior ao foco temporal ("-"). Contudo, este não foi o caso, uma vez que não notamos tanto enviesamento nos dados em função da homogeneidade de gênero textual da coleção (pouco mais de 52% das instâncias).

### 6.2.2. Classificador de significado do termo "hoje"

Apesar de este classificador usar virtualmente o mesmo conjunto de atributos de entrada dos dois anteriores, não pudemos, obviamente, aproveitar os conjuntos de dados de treinamento e teste preparados manualmente para a classificação do foco temporal e da direção. Da mesma forma, portanto, como não tínhamos acesso a conjuntos de dados prontos para treinar o classificador, tivemos que construir tal base manualmente. Desta vez, contudo, em virtude de tratarmos de apenas um termo neste classificador, a tarefa de coleta e classificação do conjunto de dados foi um tanto mais fácil.

Novamente, através de procedimentos simples de programação, buscamos em nossa base de notícias por ocorrências de sentenças com a palavra "hoje", que, tais quais os casos previamente tratados, são, na ausência de quaisquer outras informações contextuais, por definição, ambíguas semanticamente. Nesta busca, encontramos 508 ocorrências (em exatamente 508 documentos), que também foram inseridas na base de dados MySQL e processadas pelo CoppeTER para gerar o vetor de atributos de entrada para classificação. Realizamos a classificação manual de cada uma das 508 ETs – indicando, para cada uma delas, se o significado do termo era específico (o dia de hoje) ou genérico (no sentido de "atualmente", "nos dias de hoje" e afins) – através da interface web em PHP construída para os experimentos da seção anterior.

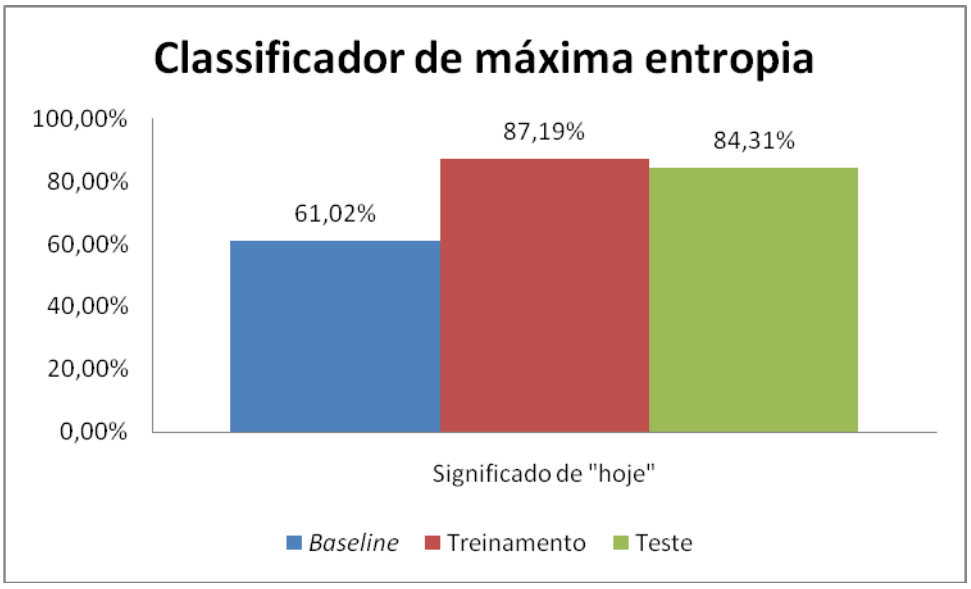



**Figura 6.2. Resultados do classificador de significado do termo "hoje"**

Outra vez, com o conjunto de dados completamente classificado, separamos aleatoriamente 80% dos dados (406 expressões temporais) para treinamento e 20% (102 ETs), para teste. Treinamos um classificador de máxima entropia e verificamos 87,19% de precisão para o conjunto de treinamento e 84,31%, para o de teste. Neste experimento, a *baseline* adotada também foi a simples atribuição da classe de maior frequência (no caso, a que classifica o significado do temo como sendo referente ao dia de hoje). Um gráfico comparativo desses resultados é apresentado na Figura 6.2.

Enquanto, na classificação da direção, percebemos uma *baseline* bem próxima da encontrada por AHN *et al.* (2005) – apesar de, diferentemente deles, termos verificado bons resultados na tarefa –, nesta classificação, encontramos exatamente o oposto: resultados semelhantes no desempenho do classificador, mas *baseline* significativamente distinta. Os autores supracitados reportaram uma *baseline* extremamente enviesada, com 90% das instâncias do problema sendo específicas, ao passo em que, em nosso conjunto de dados, contabilizamos apenas pouco mais de 61%. A precisão do classificador, por outro lado, tanto no conjunto de dados de treinamento quanto no de teste, demonstrou valores bem similares aos alcançados pelos autores (85%).

Acreditamos que tamanha diferença de *baseline* esteja alicerçada em uma combinação da homogeneidade do gênero textual (novamente, frisamos que só tínhamos textos jornalísticos na base) e de particularidades do próprio uso da língua portuguesa (isto é, talvez, seja mais comum e natural utilizarmos "hoje" com o sentido

de “atualmente” em português do que se usa “*today*” com o mesmo intuito em inglês). Entre as classificações incorretas feitas pelo sistema, notamos que a quase totalidade dos erros é, não surpreendentemente, proveniente de sentenças nas quais o verbo principal está no presente (tempo verbal).

## 6.3. Avaliação do sistema com base no Segundo HAREM

Para avaliar o desempenho do sistema CoppeTER na tarefa completa de reconhecimento e normalização de expressões temporais em português, fizemos uso de todo o arcabouço da pista TEMPO da avaliação conjunta do Segundo HAREM, que incluí, além de uma coleção dourada (com todas as ETs identificadas, classificadas, normalizadas e anotadas; já descrita qualitativa e quantitativamente na seção 2.2.6), diretivas de avaliação e plataforma de *software* para conduzir os experimentos de maneira padronizada, automática e repetível, com mínimia intervenção humana. Além do evidente benefício de conseguirmos executar uma avaliação quase completa do trabalho de forma mais fácil e ágil, reutilizar esta estrutura nos possibilita comparar diretamente nossos resultados com os dos sistemas participantes do Segundo HAREM, entre os quais, inclusive, se encontra o sistema XIP, que é considerado o atual estado da arte para a tarefa na língua portuguesa.

### 6.3.1. Esquema de avaliação

Conforme descrevemos na seção 2.3.5, no Segundo HAREM, as expressões temporais ganharam uma pista própria dentro do evento e, com isso, diretivas próprias de avaliação. Ainda assim, embora a proposta desta tarefa tenha sido feita pelos organizadores de forma independente da proposta de reconhecimento das entidades pertencentes a outras categorias (HAREM clássico), a avaliação dos sistemas no que diz respeito às entidades da categoria TEMPO foi levada a cabo de forma integrada com o HAREM clássico. Em outras palavras, não foi desenhado um novo processo de avaliação exclusivo para as ETs, mas sim integrado na sequência de avaliação do HAREM clássico um novo módulo para atribuir uma pontuação adicional às entidades temporais, no caso dos seus atributos estendidos (SENTIDO, TEMPO_REF, VAL_NORM e VAL_DELTA) estarem corretamente preenchidos. Para levar em conta

esses atributos adicionais, os sistemas participantes foram avaliados de quatro modos distintos: clássico (considerando apenas os atributos CATEG, TIPO e SUBTIPO do HAREM clássico), estendido completo (considerando todos os atributos), estendido sem normalização (ignorando os atributos VAL_NORM e VAL_DELTA) e estendido somente com normalização (ignorando os atributos SENTIDO e TEMPO_REF).

No esquema geral de avaliação do Segundo HAREM (OLIVEIRA *et al.*, 2008), cada entidade mencionada pode receber uma de três pontuações, no tocante à sua identificação (isto é, a simples delimitação no texto, através das *tags*, de onde começa e termina a EM), ao serem emparelhadas com o que está na coleção dourada: correta, em falta e espúria. Essas pontuações são utilizadas para pontuar a classificação de cada atributo das EM. As diretivas de anotação adotadas definem uma hierarquia de quatro níveis: identificação da entidade e preenchimento dos atributos CATEGORIA, TIPO e SUBTIPO. Para esta avaliação, foi definida uma única medida (a classificação semântica combinada, ou, simplesmente, CSC), que é aplicada a cada entidade identificada corretamente, e engloba esses quatro níveis da hierarquia, possibilitando a atribuição de diferentes pesos a cada um dos níveis e ainda a penalização por classificações erradas. O valor total da medida é obtido somando as parcelas relativas aos níveis da classificação. A fórmula completa da medida CSC está ilustrada na Figura 6.3 abaixo.

$$1+\sum_{i=1}^{N}((1-\frac{1}{n_{cats}})\cdot cat_{certa_i}\cdot\alpha+(1-\frac{1}{n_{tipos}})\cdot tipo_{certo_i}\cdot\beta+(1-\frac{1}{n_{sub}})\cdot sub_{certo_i}\cdot\gamma)$$
$$-\sum_{i=0}^{M}(\frac{1}{n_{cats}}\cdot cat_{esp_i}\cdot\alpha+cat_{certa_i}\cdot\frac{1}{n_{tipos}}\cdot tipo_{esp_i}\cdot\beta+tipo_{certo_i}\cdot\frac{1}{n_{sub}}\cdot sub_{esp_i}\cdot\gamma)$$
$$+tr_{certo}\cdot\delta+s_{certo}\cdot\lambda$$
$$+\begin{cases} vd_{certo}\cdot\varepsilon \\ vn_{certo}\cdot\varepsilon \\ (E_{certa}+A_{certo}+D_{certo}+H_{certo}+M_{certo}+ES_{certa}+lim_{certo})\cdot\xi \\ (H_{certo}+M_{certo}+lim_{certo})\cdot\eta \end{cases}$$

$$K_{certo_i}=\begin{cases} 1 & \text{se o atributo } K_i \text{ estiver correcto,} \\ 0 & \text{se } K_i \text{ estiver incorrecto ou omisso} \end{cases}$$
$$K_{esp_i}=\begin{cases} 1-K_{certo_i} & \text{se o atributo } K_i \text{ estiver preenchido} \\ 0 & \text{se } K_i \text{ estiver omisso} \end{cases}$$

$K \in \{cat, tipo, sub\}$

$N$ = número de diferentes classificações vagas na CD, de acordo com o cenário selectivo.

$M$ = número de classificações espúrias na participação, de acordo com o cenário selectivo.

tr, s, vd, vn: classificação referente, respectivamente, aos atributos `TEMPO_REF`, `SENTIDO`, `VAL_DELTA`, e `VAL_NORM` (quando `TIPO="DURACAO"`).

*E*, *A*, *D*, *H*, *M*, *ES*, *lim*: classificação referente, respectivamente, aos campos era, ano, dia, mês, hora, minutos, estação e limite do atributo `VAL_NORM` (quando `TIPO="DATA"` e `TEMPO_REF="ABSOLUTO"`).

*H*, *M*, *lim*: classificação referente, respectivamente aos campos hora, mês e limite do atributo `VAL_NORM` (quando `TIPO="HORA"`).

α, β, γ = parâmetros correspondentes aos pesos das categorias, tipos e subtipos.

δ, λ, ε, ξ, η = parâmetros correspondentes aos pesos dos atributos estendidos.

**Figura 6.3. Fórmula da classificação semântica combinada (CSC)**

O peso dado a cada atributo é tão maior quanto mais difícil for acertar o seu valor, ou seja, é proporcional ao número de possibilidades que existem para preenchê-lo. À penalização, por outro lado, aplica-se o critério inverso, isto é, quanto mais difícil for acertar o preenchimento de um atributo, menor é a penalização por ter o atributo espúrio. A penalização resultante do preenchimento espúrio de um atributo nunca contém mais de uma parcela. Em outras palavras, para calcular a penalização, é contabilizada apenas a parcela do atributo espúrio no nível mais alto da hierarquia supracitada. Por exemplo, se uma classificação tiver o atributo CATEG correto e o atributo TIPO espúrio, o atributo SUBTIPO será, obrigatoriamente, também espúrio. No entanto, a penalização será dada apenas pelo atributo de tipo espúrio e não pelo subtipo. Para a avaliação oficial do Segundo HAREM, os organizadores optaram por dar maior importância à classificação da categoria do que à classificação do tipo, que, por sua vez, teve mais importância do que a classificação do subtipo. Assim, a esses atributos, foram conferidos os pesos $\alpha$ = 1, $\beta$ = 0,5 e $\gamma$ = 0,25, respectivamente.

Na avaliação em modo estendido, tem-se por objetivo dar uma pontuação adicional aos sistemas pelo preenchimento correto dos atributos estendidos da categoria TEMPO citados acima, sem os penalizar pela atribuição de valores espúrios. As primeiras duas linhas da fórmula acima correspondem à medida de avaliação do HAREM clássico, sendo as restantes parcelas usadas para avaliar os atributos estendidos das expressões temporais. Vale destacar ainda que, enquanto os atributos VAL_NORM para o tipo DURACAO e VAL_DELTA são avaliados como um todo, não sendo valorizado o preenchimento correto de cada um dos campos que compõe o valor desses atributos, a situação é diferente para o atributo VAL_NORM quando o atributo TIPO é DATA ou HORA. Nesse caso, cada um dos campos individuais do atributo contribui separadamente para o valor da medida de classificação da EM.

### 6.3.2. Plataforma de software

A plataforma de *software* para condução da avaliação disponibilizada pela organização do Segundo HAREM é composta por uma série de módulos implementados em Java, à exceção do gerador de relatórios individuais (implementado em R, uma linguagem de programação para cálculos estatísticos e gráficos) e do avaliador da pista TEMPO em modo estendido (implementado em Awk, uma linguagem de programação focada no processamento de *strings* e arquivos de texto). A biblioteca Java JDOM é utilizada para a manipulação de XML.

Cada módulo tem como entrada um (ou mais) arquivos de texto, que são processados de forma a produzir um resultado pronto para ser consumido pelo módulo seguinte. A Figura 6.4 representa todos os módulos de avaliação do Segundo HAREM, na ordem em que são utilizados para se chegar aos resultados da avaliação. Como nos alongarmos nestes pormenores foge do escopo desta dissertação, sugerimos a leitura de (OLIVEIRA *et al.*, 2008) para mais detalhes sobre o funcionamento e a implementação deste ferramental.

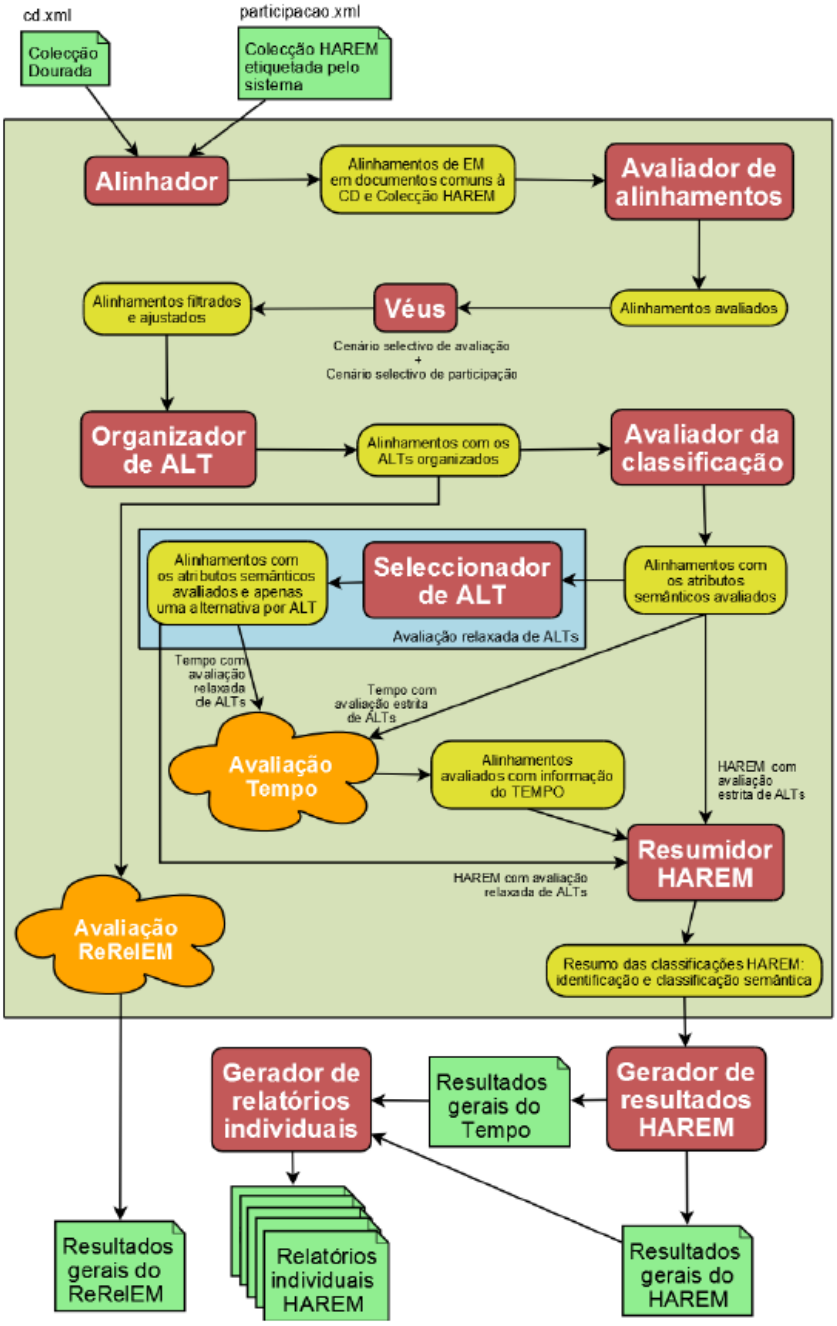



**Figura 6.4. Plataforma de *software* de avaliação do Segundo HAREM**

## 6.3.3. Avaliação do CoppeTER

A avaliação do sistema dentro do arcabouço do Segundo HAREM, apesar de não ser a ideal, uma vez que não prevê a medição de toda a gama de expressões e atributos que o sistema é capaz de anotar, tem, conforme mencionamos acima, os benefícios de ser facilmente reutilizável e de nos dar uma saída padronizada, possibilitando uma comparação direta com os outros sistemas participantes. Nesse sentido, submetemos o CoppeTER, em sua última versão de produção (de 19 de junho de 2011), à bateria de

testes automatizada levada a cabo pela plataforma de *software* descrita na seção anterior. Entre as diversas medidas listadas nos relatórios gerados para cada um dos modos de avaliação pelo sistema avaliador, consolidamos as principais na Tabela 6.2 a seguir.

**Tabela 6.2. Resultados do CoppeTER na avaliação do Segundo HAREM**

| Modo | Precisão | Revocação | Medida F | Sobregeração | Subgeração |
|---|---|---|---|---|---|
| Identificação | 96,21% | 86,75% | 91,24% | 3,79% | 13,25% |
| Classificação (clássico) | 96,03% | 87,09% | 91,34% | | |
| Classificação (completo) | **96,99%** | **89,34%** | **93,01%** | | |
| Classificação (apenas normalização) | 96,43% | 89,04% | 92,59% | | |
| Classificação (sem normalização) | 96,02% | 88,11% | 91,89% | | |

O primeiro modo de avaliação mede apenas a tarefa de identificação das entidades temporais no texto, isto é, verifica até que ponto o sistema foi capaz de corretamente delimitar, com as *tags* apropriadas, as ETs presentes nos documentos (de acordo com as anotadas da coleção dourada). Os quatro modos seguintes, também já citados anteriormente, correspondem, respectivamente, à classificação clássica do HAREM (aquela que considera apenas a identificação correta da categoria, tipo e subtipo da entidade mencionada) e às três variações do modo de avaliação estendido, quais sejam: estendido completo (a que considera todos os atributos adicionais usados para as entidades mencionadas do tipo temporal), estendido somente com normalização (que considera os atributos adicionais, à exceção dos atributos SENTIDO e TEMPO_REF) e estendido sem normalização (que considera os atributos adicionais, mas ignora os atributos VAL_NORM e VAL_DELTA).

No geral, os resultados do CoppeTER, nos diferentes modos de avaliação, são bem próximos, tendo o sistema sistematicamente alcançado uma precisão expressivamente alta, na casa dos 96-97%, o que já o coloca em par com o estado da arte para a tarefa de reconhecimento e normalização de expressões temporais em inglês, a língua para a qual há mais pesquisa no assunto. A revocação percebeu uma variação maior, estando todas as medições no intervalo entre 86,5% e 89,5%. Apesar de, como veremos mais adiante, bastante acima do melhor resultado obtido no Segundo HAREM (pelo sistema XIP), esses valores ainda estão um pouco distantes dos observados pelos melhores sistemas em língua inglesa (vide seção 2.3.3). A medida F, combinando as

duas métricas (que, em todos os casos desta avaliação é a chamada medida F balanceada padrão ou $F_1$, descrita na seção 2.1.5), para a avaliação ultrapassou os 93%.

As medidas de sobregeração e subgeração são aplicáveis apenas à identificação de entidades mencionadas (entre elas, obviamente, as temporais) no texto. Diferentemente dos outros, que utilizam a fórmula de pontuação detalhada anteriormente, neste modo de avaliação, as medidas de precisão e revocação são simplesmente as definições originais descritas na seção 2.1.5, adaptadas para o contexto da tarefa de reconhecimento de EMs. Assim, a precisão, neste caso, é a razão entre o número de EMs corretamente classificadas e o número de EMs classificadas por um sistema, enquanto a revocação é a proporção das EMs marcadas na coleção dourada recuperada pelo sistema, isto é, a razão entre o número de EMs corretamente classificadas e o número de EMs classificadas na CD.

Fazendo o contraponto, a sobregeração mede o excesso de resultados espúrios que o sistema produz, ou seja, a frequência com que produz resultados errados, enquanto a subgeração, por sua vez, é uma medida de quanto faltou ao sistema analisar, dada a solução conhecida (a coleção dourada, no caso). Essas medidas são calculadas conforme exposto na Figura 6.5. Destarte, entre as ETs marcadas pelo CoppeTER, apenas 3,79% delas (sobregeração) não eram realmente expressões temporais (ou, no mínimo, não constavam como tais na coleção dourada). Por outro lado, o sistema foi incapaz de identificar 13,25% (subgeração) das expressões temporais esperadas de acordo com a solução da CD, razão pela qual a revocação se mostrou aquém do desejado.

$$\text{Sobregeração} = \frac{\#(\text{EMs espúrias})}{\#(\text{EMs classificadas pelo sistema})} \qquad \text{Subgeração} = \frac{\#(\text{EMs em falta})}{\#(\text{EMs na CD})}$$

**Figura 6.5. Fórmulas de sobregeração e de subgeração**

Para os outros modos de avaliação, a precisão, que é a medida da qualidade da resposta do sistema, é calculada pela razão entre o somatório da pontuação obtida por cada EM (conforme a fórmula especificada anteriormente) e a pontuação máxima que seria obtida pelo sistema caso todas as EMs identificadas por ele estivessem completamente corretas (obviamente, de acordo, com a solução). Da mesma forma, a revocação, que mede a percentagem de soluções contidas na coleção dourada que o sistema conseguiu reconhecer e normalizar corretamente, é calculada pela razão entre o

somatório da pontuação obtida por cada EM e a pontuação máxima na CD (isto é, a pontuação que um sistema obteria se fosse capaz de reconhecer e normalizar corretamente todas as ETs presentes na coleção dourada).

Conforme podemos observar, o sistema alcançou excelentes resultados em todos os modos de avaliação, tendo seu melhor desempenho no modo estendido completo (destacado e em negrito na Tabela 6.2), que, conforme explicado na seção 6.3.1, leva em conta toda a pontuação adicional possível, considerando todos os atributos descritores das expressões temporais. Isso demonstra que o sistema, além de identificar bem as ETs, consegue preencher com precisão todos os atributos que as detalham. Em outras palavras, pode-se inferir do resultado que, quase sempre, quando o sistema é capaz de delimitar a expressão temporal no texto, ele também é capaz de corretamente reconhecê-la, interpretá-la e, por fim, normalizá-la, isto é, dificilmente, o CoppeTER identifica a ET perfeitamente, mas falha no preenchimento dos seus atributos específicos.

Essa inferência é fortalecida quando notamos que, igualmente, o pior resultado percebido pelo sistema se dá no modo de avaliação clássico, que considera apenas a classificação semântica básica (categoria, tipo e subtipo) das expressões temporais. A variação do modo de avaliação estendido que pontua apenas a normalização (também destacada na Tabela 6.2) aparece como o segundo melhor resultado do sistema, sendo também evidência adicional de que o CoppeTER, apesar de ainda não alcançar valores de revocação comparáveis aos sistemas do estado da arte para a língua inglesa, é extremamente preciso na normalização das ETs que de fato é capaz de reconhecer.

### 6.3.4. Comparação dos resultados

Dos dez sistemas participantes do Segundo HAREM (neste caso, do evento completo de REM), somente três não fizeram o reconhecimento de expressões temporais, o que demonstra um claro interesse em reconhecer este tipo de entidades (MOTA *et al.*, 2008). No entanto, apenas desses dois sistemas foram significativamente além do preenchimento dos atributos do HAREM clássico (CATEG, TIPO e SUBTIPO): o sistema XIP, do grupo L2F/Xerox, que foi capaz de preencher todos os atributos do modo estendido, e o sistema da Priberam, que conseguiu atribuir valores ao atributo TEMPO_REF.

**Tabela 6.3. Resultados da avaliação da identificação**

| Sistema | Precisão | Revocação | Medida F | Sobregeração | Subgeração |
|---|---|---|---|---|---|
| CoppeTER | 96,21% | 86,75% | 91,24% | 3,79% | 13,25% |
| XIP | 75,31% | 77,59% | 76,43% | 24,69% | 22,41% |
| PorTexTO | 74,70% | 53,45% | 62,31% | 25,30% | 46,55% |
| REMBRANDT | 61,76% | 45,26% | 52,24% | 38,24% | 54,74% |
| REMMA | 45,89% | 28,88% | 35,45% | 54,11% | 71,12% |
| Priberam | 7,34% | 20,26% | 10,78% | 92,66% | 79,74% |
| Cage2 | 6,67% | 2,59% | 3,73% | 93,33% | 97,41% |
| SeRELeP | 0,11% | 0,43% | 0,18% | 99,89% | 99,57% |

A Tabela 6.3 acima compara o desempenho do CoppeTER na tarefa de identificação (novamente, a simples delimitação das entidades no texto) de ETs com o alcançado pelos outros sete participantes do Segundo HAREM, conforme o resultado final reportado à época pelos organizadores. Todos esses resultados são ilustrados graficamente na Figura 6.6 a seguir.

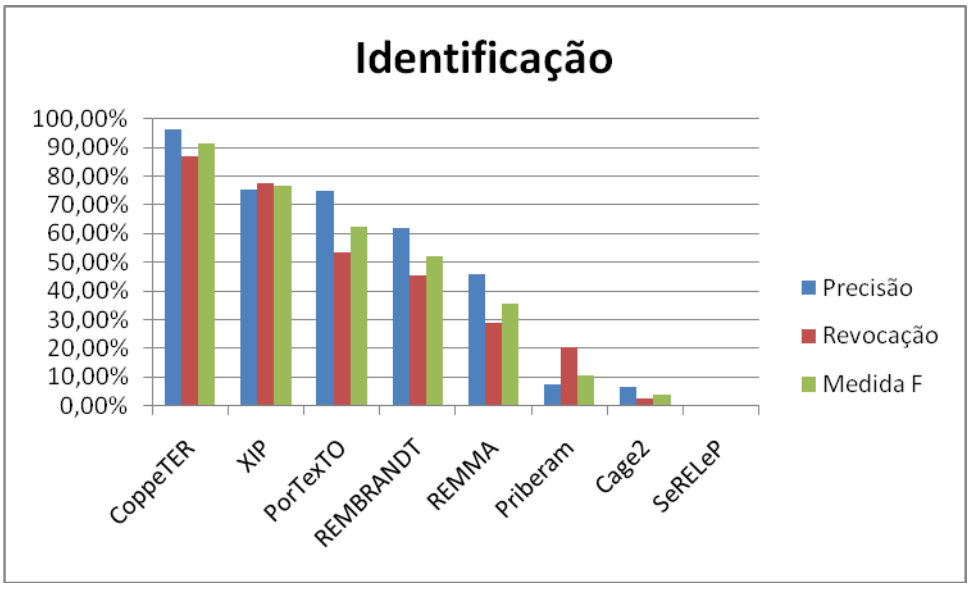



**Figura 6.6. Gráfico dos resultados da avaliação da identificação**

Em virtude de o Segundo HAREM ser um evento para o reconhecimento de diversas categorias de entidades mencionadas, é esperado que todos os sistemas participantes dividam seus esforços no reconhecimento de todos esses tipos de EMs, não podendo, portanto, focar-se em apenas um, como fizemos neste trabalho. Não é tão surpreendente, portanto, encontrar o CoppeTER com o melhor resultado em todas as comparações mostradas nesta seção.

**Tabela 6.4. Resultados da avaliação da classificação (clássico e completa)**

| | Classificação (clássico) | | | Classificação (completo) | | |
|---|---|---|---|---|---|---|
| **Sistema** | **Precisão** | **Revocação** | **Medida F** | **Precisão** | **Revocação** | **Medida F** |
| CoppeTER | 96,03% | 87,09% | 91,34% | 96,99% | 89,34% | 93,01% |
| XIP | 73,76% | 75,80% | 74,77% | 80,87% | 70,24% | 75,18% |
| PorTexTO | 73,50% | 53,27% | 61,77% | 73,50% | 27,08% | 39,57% |
| REMBRANDT | 60,28% | 44,81% | 51,40% | 60,28% | 22,77% | 33,06% |
| REMMA | 45,65% | 25,81% | 32,97% | 45,65% | 13,12% | 20,38% |
| Priberam | 9,50% | 19,46% | 12,77% | 11,15% | 11,86% | 11,50% |
| Cage2 | 6,67% | 1,66% | 2,66% | 6,67% | 0,84% | 1,50% |
| SeRELeP | 0,11% | 0,28% | 0,16% | 0,11% | 0,14% | 0,12% |

A Tabela 6.4 acima compara os resultados do CoppeTER com o desempenho dos participantes do Segundo HAREM para os dois modos de avaliação nos quais percebemos os extremos do resultado do sistema na seção anterior: clássico e estendido completo. A Figura 6.7 e a Figura 6.8 ilustram esses valores graficamente.

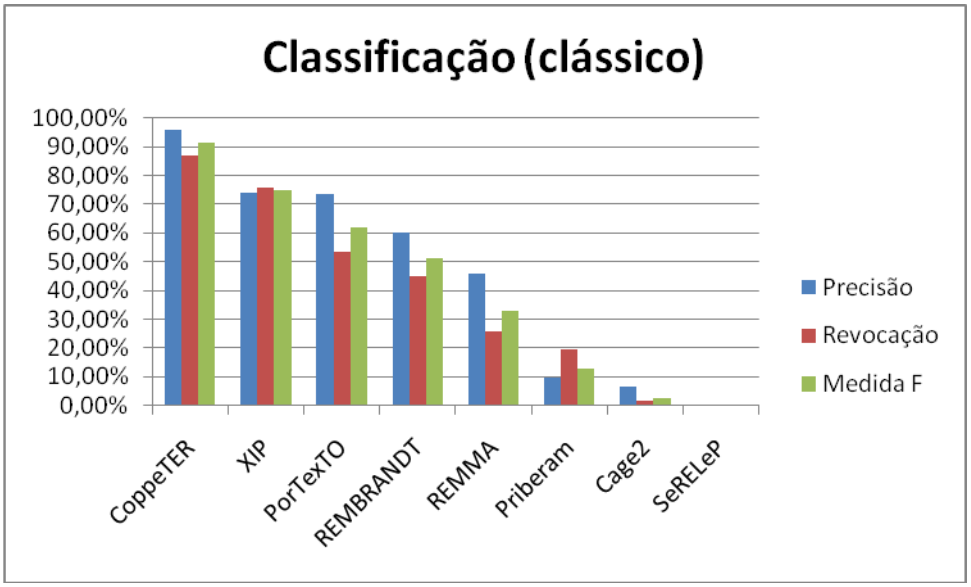



**Figura 6.7. Gráfico dos resultados da avaliação da classificação (clássico)**

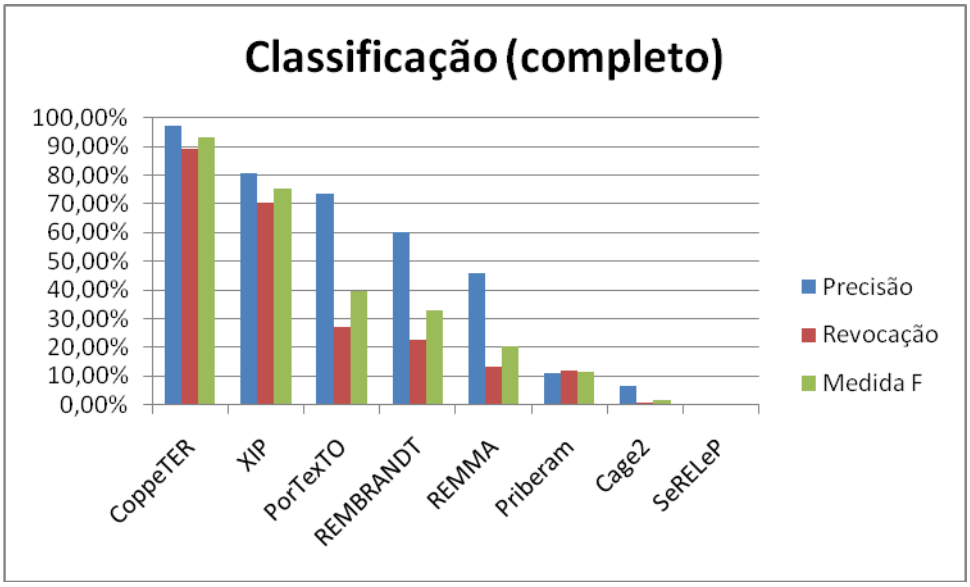



**Figura 6.8. Gráfico dos resultados da avaliação da classificação (completo)**

Vale notar que, tal qual verificamos para o CoppeTER na seção anterior, o XIP, diferentemente dos outros sistemas, vê a sua pontuação final (isto é, a sua medida F) aumentar quando é submetido à avaliação no modo estendido completo. Conforme explicamos na avaliação do sistema proposto nesta dissertação, isso se dá em função da pontuação adicional (dada à normalização) considerada neste modo de avaliação. Assim, também podemos concluir que o XIP, tendo identificado uma expressão temporal, é quase sempre capaz de normalizá-la com boa precisão, uma vez que a pontuação dada à normalização está melhorando seu desempenho em vez de piorá-lo.

**Tabela 6.5. Resultados da avaliação da classificação (apenas normalização e sem normalização)**

| | **Classificação (apenas normalização)** | | | **Classificação (sem normalização)** | | |
|---|---|---|---|---|---|---|
| **Sistema** | **Precisão** | **Revocação** | **Medida F** | **Precisão** | **Revocação** | **Medida F** |
| CoppeTER | 96,43% | 89,04% | 92,59% | 96,02% | 88,11% | 91,89% |
| XIP | 77,33% | 74,72% | 76,00% | 79,08% | 69,70% | 74,10% |
| PorTexTO | 73,50% | 37,30% | 49,49% | 73,50% | 34,61% | 47,06% |
| REMBRANDT | 60,28% | 31,37% | 41,27% | 60,28% | 29,11% | 39,26% |
| REMMA | 45,65% | 18,07% | 25,89% | 45,65% | 16,76% | 24,52% |
| Priberam | 11,15% | 16,34% | 13,26% | 9,50% | 12,64% | 10,85% |
| Cage2 | 6,67% | 1,16% | 1,98% | 6,67% | 1,08% | 1,86% |
| SeRELeP | 0,11% | 0,19% | 0,14% | 0,11% | 0,18% | 0,14% |

Por fim, listamos, na Tabela 6.5, os resultados do CoppeTER e dos demais participantes nos dois modos de avaliação restantes (estendido apenas com normalização e estendido sem normalização). Analogamente aos anteriores, os valores percebidos nestes modos de avaliação estão ilustrados de forma gráfica na Figura 6.9 e na Figura 6.10.

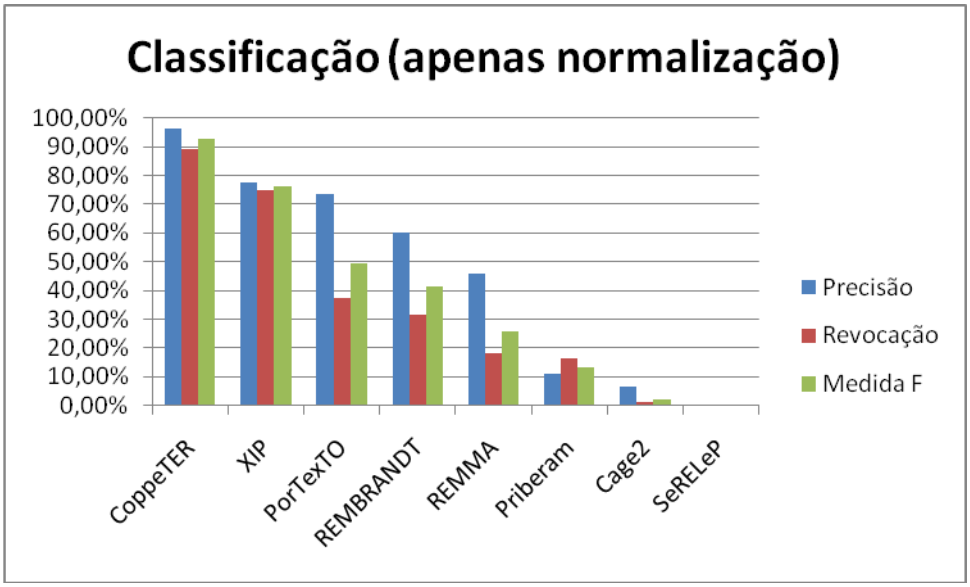



**Figura 6.9. Gráfico dos resultados da avaliação da classificação (apenas normalização)**

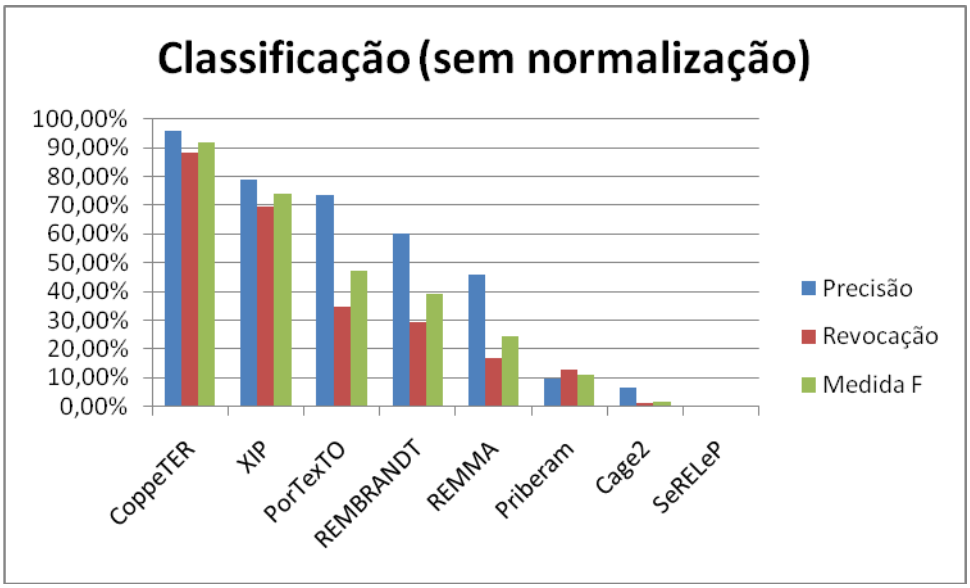



**Figura 6.10. Gráfico dos resultados da avaliação da classificação (sem normalização)**

Em conclusão, comparativamente com os outros sistemas, e esquecendo o contexto maior da avaliação conjunta proposta pelo Segundo HAREM para o

reconhecimendo de entidades mencionadas, os resultados do CoppeTER são excelentes. Devemos, contudo, conforme citado acima, levar em consideração que todos esses sistemas buscavam identificar uma gama de entidades muito maior que o conjunto restrito de expressões temporais, que é o foco desde trabalho. Em uma situação na qual o XIP pudesse ter investido mais esforços no desenvolvimento do seu módulo de processamento temporal, teríamos uma comparação mais justa.

### 6.3.5. Análise de erros

Analisando manualmente as falhas cometidos pelo CoppeTER na tarefa de reconhecimento e normalização de expressões temporais do Segundo HAREM, observamos dois tipos de erros repetidas vezes. O primeiro deles tem origem em uma diferença conceitual entre o que considermos como uma ET de frequência e o que as diretivas de anotação do evento classificavam como tal. Em nossa proposta, fazendo jus ao quesito temporal que motiva e permeia todo este trabalho, optamos por marcar apenas frequências que tivessem um componente temporal explícito.

Os organizadores da pista TEMPO do Segundo HAREM, por outro lado, decidiram englobar todo o conceito de frequência em sua proposta e solicitar a marcação também de expressões de frequência como “ocasionalmente” ou “frequentemente”, que, além de serem completamente subjetivas, não fazem referência a nenhuma unidade temporal. Além dessas, entre as entidades temporais marcadas como frequências na coleção dourada (que, obviamente, o sistema falhou em reconhecer), constavam as expressões “a maior parte das vezes”, “algumas vezes” e “por diversas vezes”, que, em nossa opinião, sequer deveriam ser consideradas frequências (independentemente do que consideramos como anotavéis ou não).

O outro tipo de erro comumente cometido pelo CoppeTER advém do fato de o sistema ser completamente focado e especializado no reconhecimento de expressões temporais (que considerávamos ser um benefício e uma das principais razões de termos obtido resultados tão bons), e diferentemetne dos outros participantes da avaliação conjunta, não levar em conta a existência de nenhum outra categoria de entidades mencionadas. Como, a princípio, pensavamos haver pouca, ou não haver nenhuma, interseção entre as construções características de entidades temporais e os outros tipos de EMs, não vislumbramos um procedimento de desambiguação para os casos em que uma expressão que poderia ser considerada uma ET é, na realidade, parte de um outro

tipo de entidade mencionada maior. Entre os exemplos desse tipo de falha cometida pelo sistema estão as expressões “Campo 24 de Agosto” e “Decreto de 28 de Maio”, que deveriam se classificadas como local e obra, respectivamente, mas foram interpretados como ETs de data relativas pelo CoppeTER.

### 6.3.6. Relevância do experimento

Além da já citada injustiça em compararmos sistemas generalistas (como a maioria dos participantes do Segundo HAREM, que buscam reconhecer várias categorias de entidades mencionadas) com especialistas (como o CoppeTER, que trata somente de entidades temporais), devemos ressaltar outros dois pontos que consideramos ameaças à relevância deste experimento.

Um deles diz respeito ao tamanho do corpus utilizado na avaliação. Por mais que a coleção dourada contenha uma boa quantidade (mais de mil) de expressões temporais marcadas e classificadas, acreditamos que a avaliação, em especial da normalização, perde um pouco em validade haja vista a disponibilidade de apenas uma pequena quantidade (pouco mais de 230) de ETs completamente normalizadas e com todos os atributos estendidos preenchidos. Gostaríamos de ter conduzido o experimento tendo à disposição a totalidade das mais de mil entidades temporais presentes na CD do Segundo HAREM, contudo, realizar (e depois validar) a normalização do restante das expressões manualmente comandaria esforço expressivo, suficiente para ocupar uma equipe de pessoas.

O outro ponto que pensamos diminuir a relevância deste experimento é a impossibilidade de avaliarmos as melhorias que fizemos em cima dos esquemas de anotação do TIMEX2 e do Segundo HAREM. Além de termos introduzido a normalização de mais tipos de expressões temporais (intervalos de datas e durações, frequências, expressões genéricas e indefinidas, idades, etc.), estendemos a normalização já conhecida com novos atributos para reconhecer, capturar e interpretar imprecisão, vagueza, limites e modificadores dessas ETs. Consideramos, portanto, que esse conjunto de contribuições deste trabalho ainda carece de avaliação qualitativa e quantitativa, pois imaginamos que a maior parte dos erros e falhas cometidas pelo sistema (e, portanto, os indicadores mais precisos das direções a serem tomadas em trabalhos futuros) surgiriam exatamente nesse contexto.

# 7. Conclusão

Neste trabalho, demos alguns passos na direção de viabilizar uma completa cadeia de processamento temporal em língua portuguesa, atacando, especificamente, os problemas de reconhecimento e normalização de expressões temporais no idioma. Nesse sentido, conduzimos uma extensa análise da área de RET e das áreas correlatas de extração de informação e reconhecimento de entidades, a partir da qual contextualizamos o problema e apresentados o atual estado da arte para diversas línguas.

Com base nisso, propusemos um esquema de anotação temporal, acompanhado de uma sugestão de arquitetura, para o português, mais próximo do padrão adotado internacionalmente (TIMEX2/TimeML), mas sem se distanciar tanto das diretivas desenvolvidas para o Segundo HAREM. Tal esquema foi implementado no CoppeTER – Coppe Temporal Expression Recognizer, um sistema híbrido para reconhecimento e normalização de expressões temporais em português, que combina uma gramática de regras e aprendizado de máquina (mais especificamente, etiquetadores gramaticais e classificadores de máxima entropia).

## 7.1. Resultados alcançados

Intuitivamente, acreditamos que o padrão de anotação proposto nesta dissertação, resultado de melhorias que fizemos em cima dos esquemas de anotação do TIMEX2 e do Segundo HAREM, mais bem traduz a forma como as expressões temporais ocorrem naturalmente nos textos em língua portuguesa. Essa percepção está alierçada na facilidade com que mapeamos os termos para a estrutura interna do sistema utilizada pelo motor de regras na implementação e na rapidez com que fomos capazes de traduzir essa estrutura para os padrões existentes. Não conseguimos ainda, contudo, avaliá-lo de forma qualitativa e quantitativa.

No geral, os resultados do CoppeTER foram muito bons, tendo o sistema sistematicamente alcançado uma precisão expressivamente alta, em par com o estado da arte para a tarefa de reconhecimento e normalização de expressões temporais em inglês, a língua para a qual há mais pesquisa no assunto. Apesar de sermos precisos,

percebemos, contudo, que ainda estamos um pouco distantes da abrangência observada pelos melhores sistemas em língua inglesa.

Comparativamente com os outros sistemas, e esquecendo o contexto maior da avaliação conjunta proposta pelo Segundo HAREM para o reconhecimendo de entidades mencionadas, os resultados do CoppeTER são excelentes, ficando bem acima do melhor desempenho reportado até então.

Como contribuições, podemos citar, ainda, a composição de um corpus de mais três mil notícias, a partir do qual foram derivados dois conjuntos de dados para treinamento e avaliação de classificadores de desambiguação de expressões temporais relativas. Tais conjuntos de dados podem ser úteis para nortear outros trabalhos que utilizem classificação estatística para desambiguação.

## 7.2. Trabalhos futuros

Como continuidade deste trabalho, podemos citar, de imediato, o fechamento dos pontos deixados em aberto por esta dissertação, quais sejam: avaliar o padrão de anotação, atacar as fontes de erros mais comuns cometidos pelo sistema e conduzir experimentos com maior abrangência, relevância e validade.

Uma das falhas comumente cometidas pelo CoppeTER é a confusão entre entidades temporais e outros tipos de entidades mencionadas, quando estas são compostas por expressões que poderiam, sozinhas, ser consideradas ETs (como as expressões “Campo 24 de Agosto” e “Decreto de 28 de Maio”, que deveriam ser classificadas como local e obra, respectivamente, mas são interpretadas como ETs de data relativas pelo sistema). Um possível trabalho futuro poderia levar em conta a existência de outros tipos de EMs na desambiguação para tentar resolver o problema.

No tocante aos experimentos, gostaríamos de ver como os classificadores de máxima entropia (e outros classificadores estatísticos aplicáveis) se comportariam com conjuntos de dados mais heterogêneos, envolvendo mais variações de ETs relativas e outros gêneros textuais. Seria importante também a construção de um corpus significativamente maior com para avaliar o desempenho do sistema como um tudo. A título de comparação, o corpus do TIDES para o inglês contém oito vezes mais ETs classificadas e trinta vezes mais ETs normalizadas que o corpus do Segundo HAREM.

Outra continuidade deste trabalho poderia ser a investigação de erros advindos da linearidade no tratamento do texto. Notícias de eventos que estão se desenrolando

costumam trazer uma sequência cronológica fortemente não-linear e supomos que isso deve afetar significativamente o desempenho da interpretação de expressões temporais relativas. Citações também se mostram como desafios interessantes para sistemas de RET, uma vez que o foco temporal é completamente alterado dentro desse tipo de passagem textual.

Outra direção que poderia ser investigada é quanto à possibilidade de desenvolvimento de uma ontologia de expressões temporais. Tal ontologia permitiria um melhor tratamento das diversas unidades temporais, provavelmente facilitando a construção das regras, ao provê-las com uma maior capacidade de expressão.

Também seria interessante desenvolver aplicações para este trabalho e seus derivados, especialmente nas áreas que mais podem se beneficiar do conhecimento temporal extraído, como recuperação de informação, processamento de linguagem natural em português e visualização de dados (esta para, por exemplo, gerar linhas do tempo, calendários, gráficos e afins representando a estrutura temporal de documentos textuais).

Por fim, uma extensão natural deste trabalho, seria seguir com a implementação de uma completa cadeia de processamento temporal, no sentido de reconhecer eventos e relacioná-los entre si e com outras expressões temporais no texto. Para isso, provavelmente, também se mostrará necessária uma investigação da necessidade de adaptação da TimeML e suas diretivas de anotação para a língua portuguesa.

.

# Referências Bibliográficas


ACE, 2004. Disponível em: http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf. Acesso em: 6 jun. 2011.

ACE, 2005. Disponível em: http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v2a.pdf. Acesso em: 8 jun. 2011.

ACE, 2008. Disponível em: http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf. Acesso em: 9 jun. 2011.

AFONSO, S., BICK, E., HABER, R., *et al.*, 2001, "Floresta sintá(c)tica: um treebank para o português". In: *Actas do XVII Encontro Nacional da Associação Portuguesa de Linguística (APL 2001)*, pp. 533-545, Lisboa, Portugal, Outubro.

AHN, D., ADAFRE, S. F., DE RIJKE, M., "Towards Task-Based Temporal Extraction and Recognition". In: *Annotating, Extracting and Reasoning about Time and Events*, v. 5151, *Dagstuhl Seminar Proceedings*, Begegnungs- und Forschungszentrum für Informatik (IBFI), pp. 1-20, 2005.

AHN, D., VAN RANTWIJK J., DE RIJKE, M., 2007, "A Cascaded Machine Learning Approach to Interpreting Temporal Expressions". In: *Proceedings of NAACL HLT 2007*, pp. 420-427, Rochester, NY, EUA, Agosto.

ALONSO, O. R., 2008, *Temporal Information Retrieval*. Ph.D. dissertation, University of California, Davis, CA, EUA.

ALONSO, O., GERTZ, M., BAEZA-YATES, R., 2007, "On the Value of Temporal Information in Information Retrieval", *ACM SIGIR Forum*, v. 41, n. 2 (Dez), pp. 35-41.

AMARAL, C., FIGUEIRA, H., MENDES, A., *et al.*, "Adaptação do Sistema de Reconhecimento de Entidades Mencionadas da Priberam ao HAREM". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento*

*de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 9, Porto, Portugal, Linguateca, 2008

APPELT, D. E., HOBBS, J. R., BEAR, J., *et al.*, 1993, "FASTUS: A Finite-state Processor for Information Extraction from Real-world Text". In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993)*, pp. 1172-1178, Chambéry, França, Setembro.

ASAHARA, M., MATSUMOTO, Y., 2003, "Japanese Named Entity Extraction with Redundant Morphological Analysis". In: *Proceedings of the 7th Conference on Natural Language Learning (CoNLL 2003) at HLT-NAACL 2003*, pp. 8-15, Edmonton, Canadá, Junho.

BAEZA-YATES, R., RIBEIRO-NETO, B., 1999, *Modern Information Retrieval*. 1 ed. Boston, MA, EUA, Addison-Wesley Longman Publishing Company.

BALDWIN, J. A., 2002, *Learning Temporal Annotation of French News*, M.Sc. thesis, Georgetown University, Washington, DC, EUA.

BANKO, M., CAFARELLA, M. J., SODERLAND, S., *et al.*, 2008, "Open Information Extraction from the Web", *Communications of the ACM*, v. 51, n. 12 (Dez), pp 68-74.

BAPTISTA, J., HAGÈGE, C., MAMEDE, N., "Identificação, classificação e normalização de expressões temporais do português: A experiência do Segundo HAREM e o futuro". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 2, Porto, Portugal, Linguateca, 2008.

BASILI, R., CAMMISA, M., DONATI, E., 2005, "RitroveRAI: A Web Application for Semantic Indexing and Hyperlinking of Multimedia News". In: *Proceedings of the International Semantic Web Conference (ISWC 2005)*, pp 97-111, Galway, Irlanda, Novembro.

BERGHEL, H., 1997, "Cyberspace 2000: Dealing with Information Overload", *Communications of the ACM*, v. 40, n. 2 (Fev), pp. 19-24.

BERGLUND, A., 2004, *Extracting Temporal Information and Ordering Events for Swedish*, M.Sc. thesis, Lund University, Lund, Suécia.

BERGLUND, A., JOHANSSON, R., NUGUES, P., 2006a, "Extraction of Temporal Information from Texts in Swedish", In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 259-264, Genova, Itália, Maio.

BERGLUND, A., JOHANSSON, R., NUGUES, P., 2006b, "A Machine Learning Approach to Extract Temporal Information from Texts in Swedish and Generate Animated 3D Scenes", In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pp. 76-86, Trento, Itália, Abril.

BICK, E., 2000, *The Parsing System Palavras - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Famework*. Ph.D. dissertation, Aarhus University, Aarhus, Dinamarca.

BICK, E., 2004, "A Named Entity Recognizer for Danish", In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 305-308, Lisboa, Portugal, Maio.

BICK, E., 2006, "Functional Aspects in Portuguese NER". In: *Proceedings of the 7th International Workshop on Computational Processing of the Portuguese Language (PROPOR 2006)*, pp. 80-89, Itatiaia, MG, Brasil, Maio.

BIKEL, D. M., MILLER, S., SCHWARTZ, R., *et al.*, 1997, "Nymble: a High-Performance Learning Name-Finder". In: *Proceedings of the 5th Conference on Applied Natural Language Processing*, pp. 194-201, Washington, DC, EIA, Abril.

BIRD, S., KLEIN, E., LOPER, E., 2009, *Natural Language Processing with Python*. 1 ed. Sebastopol, CA, EUA, O'Reilly Media.

BIRD, S., LOPER, E., 2006, "NLTK: the Natural Language Toolkit". In: *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 69-72, Sydney, Australia, Julho.

BISHOP, C. M., 2006, *Pattern Recognition and Machine Learning*. 1 ed. New York, NY, EUA, Springer.

BITTAR, A., 2009, "Annotation of Events and Temporal Expressions in French Texts". In: *Proceedings of the Third Linguistic Annotation Workshop of the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 48-51, Suntec, Cingapura, Agosto.

BODENREIDER, O., ZWEIGENBAUM, P., 2000, "Identifying Proper Names in Parallel Medical Terminologies", *Stud Health Technol Inform.*, v. 77, pp. 443-447.

BOND, F., OGURA, K., UCHINO, H., 1997, "Temporal Expressions in Japanese-to-English Machine Translation". In: *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1997)*, pp. 55-62, Santa Fe, NM, EUA, Julho.

BORTHWICK, A., STERLING, J., AGICHTEIN, E., *et al.*, 1998, "NYU: Description of the MENE Named Entity System as Used in MUC-7". In: *Proceedings of the 7th Message Understanding Conference (MUC-7)*, pp. 1-6, Fairfax, VA, EUA, Abril.

BR-ISPELL, 2011. Disponível em: http://www.ime.usp.br/~ueda/br.ispell/. Acesso em: 16 jun. 2011.

BRILL, E., 1995, "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging", *Computational Linguistics*, v. 21, n. 4 (Dez), pp. 543-565.

CALIFF, M. E., MOONEY, R. J., 1999, "Relational Learning of Pattern-Match Rules for Information Extraction". In: *Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence*, pp. 328-334, Orlando, FL, EUA, Julho.

CARDOSO, N., "REMBRANDT - Reconhecimento de Entidades Mencionadas Baseado em Relações e ANálise Detalhada do Texto". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades*

*mencionadas: O Segundo HAREM*, 1 ed., capítulo 11, Porto, Portugal, Linguateca, 2008

CARDOSO, N., SANTOS, D., "Directivas para a identificação e classificação semântica na colecção dourada do HAREM". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 16, Lisboa, Portugal, Linguateca, 2007a.

CARDOSO, N., SANTOS, D., VILELA, R., "Directivas para a identificação e classificação morfológica na colecção dourada do HAREM". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 17, Lisboa, Portugal, Linguateca, 2007b.

CARVALHO, P., OLIVEIRA, H. G., SANTOS, D., *et al.*, "Segundo HAREM: Modelo geral, novidades e avaliação". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 1, Porto, Portugal, Linguateca, 2008.

CASELLI, T., DELL'ORLETTA, F., PRODANOF, I., 2009, "TETI: a TimeML Compliant TimEx Tagger for Italian". In: *Proceedings of the International Multiconference on Computer Science and Information Technology*, pp. 185-192, Mrągowo, Polônia, Outubro.

CHANG, C. H., KAYED, M., GIRGIS, M. R., *et al.*, 2006, "A Survey of Web Information Extraction Systems", *IEEE Transactions on Knowledge and Data Engineering*, v. 18, n. 10 (Out), pp. 1411-1428.

CHINCHOR, N. A., 1998, "Overview of MUC-7/MET-2". In: *Proceedings of the 7th Message Understanding Conference (MUC-7)*, pp. 1-4, Fairfax, VA, EUA, Abril.

COHEN, W. W., HURST, M., JENSEN, L. S., 2002, "A Flexible Learning System for Wrapping Tables and Lists in HTML Documents". In: *Proceedings of the 11th International Conference on World Wide Web*, pp. 232-241, Honolulu, Hawaii, EUA, Maio.

CONJUGUE, 2011. Disponível em: http://www.ime.usp.br/~ueda/br.ispell/conjugue.html. Acesso em: 16 jun. 2011.

COWIE, J., LEHNERT, W., 1996, "Information Extraction", *Communications of the ACM*, v. 39, n. 1 (Jan), pp. 80-91.

CRAVEIRO, O., MACEDO, J., MADEIRA, H., "PorTexTO: Sistema de Anotação/Extracção de Expressões Temporais". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 8, Porto, Portugal, Linguateca, 2008

DELLA PIETRA, S., DELLA PIETRA, V., LAFFERTY, J., 1997, "Inducing Features of Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 4 (Abr), pp. 380-393.

DOAN, A., RAMAKRISHNAN, R., CHEN, F., *et al.*, 2006, "Community Information Management", *IEEE Data Engineering Bulletin*, v. 29, n. 1 (Mar), pp. 64-72.

DODDINGTON, G., MITCHELL, A., PRZYBOCKI, M., *et al.*, 2004, "The Automatic Content Extraction (ACE) Program – Tasks, Data, & Evaluation". In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pp 837-840, Lisboa, Portugal, 2004, Maio.

EMBLEY, D. W., JIANG, Y. S., NGY, Y.-K., 1999, "Record-Boundary Discovery in Web Documents". In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pp. 467-478, Philadephia, PA, EUA, Junho.

FERRÁNDEZ, Ó., KOZAREVA, Z., TORAL, A., *et al.*, "Tackling HAREM's Portuguese Named Entity Recognition Task with Spanish Resources". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 11, Lisboa, Portugal, Linguateca, 2007.

FERREIRA, L., TEIXEIRA, A., CUNHA, J. P. S., "REMMA - Reconhecimento de Entidades Mencionadas do MedAlert". In: Mota, C., Santos, D. (eds), *Desafios*

*na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 12, Porto, Portugal, Linguateca, 2008

FERRO, L., GERBER, L., MANI, I., *et al.*, 2004, *TIDES 2003 Standard for the Annotation of Temporal Expressions*. In: Technical Report, MITRE.

FERRO, L., GERBER, L., MANI, I., *et al.*, 2005, *TIDES 2005 Standard for the Annotation of Temporal Expressions*. In: Technical Report, MITRE.

FERRO, L., KOZIEROK, R., GERBER, L., *et al.*, 2002, "Annotating Temporal Information - From Theory to Practice". In: *Proceedings of the 2nd International Conference on Human Language Technology (HLT 2002)*, pp. 226-230, San Diego, CA, EUA, Março.

FERRO, L., MANI, I., SUNDHEIM, B., *et al.*, 2001, *TIDES Temporal Annotation Guidelines*. In: Technical Report, MITRE.

FLEISCHMAN, M., 2001, "Automated Subcategorization of Named Entities". In: *Proceedings of the Student Research Workshop and Tutorial Abstracts*, pp. 25-30, Toulouse, França, Julho.

FLEISCHMAN, M., HOVY, E. H., 2002, "Fine Grained Classification of Named Entities". In: *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 1-7, Taipei, Taiwan, Setembro.

FREITAG, D., 1998, "Multistrategy Learning for Information Extraction". In: *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)*, pp. 161-169, Madison, WI, EUA, Julho.

FREITAS, C., CARVALHO, P., OLIVEIRA, H. G., *et al.*, 2010, "Second HAREM: Advancing the State of the Art of Named Entity Recognition in Portuguese", In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pp. 3630-3637, Valetta, Malta, Maio.

FREITAS, C. ROCHA, P., BICK, E., 2008, "Floresta Sintá(c)tica: Bigger, Thicker and Easier". In: *Proceedings of the 8th International Conference on Computational Processing of the Portuguese Language (PROPOR 2008)*, pp. 216-219, Aveiro, Portugal, Setembro.

GAGNON, M., LAPALME, G., 1996, "From Conceptual Time to Linguistic Time", *Computational Linguistics*, v. 22, n. 1 (Mar), pp. 91-127.

GCONJUGUE, 2011. Disponível em: http://jalvesaq.googlepages.com/gconjugue.html. Acesso em: 16 jun. 2011.

GORALWALLA, I. A., LEONTIEV, Y., ÖZSU, M. T., *et al.*, 2001, "Temporal Granularity: Completing the Puzzle, *Journal of Intelligent Information Systems*, v. 16, n. 1 (Jan), pp. 41-63.

GRISHMAN, R. SUNDHEIM, B., 1996, "Message Understanding Conference - 6: A Brief History". In: *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 466-471, Copenhague, Dinamarca, Junho.

HACIOGLU, K., CHEN, Y., DOUGLAS, B., 2005, "Automatic Time Expression Labeling for English and Chinese Text". In: *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2005)*, pp. 548-559, Cidade do México, México, Fevereiro.

HAGÈGE, C., BAPTISTA, J., MAMEDE, N., 2009, "Portuguese Temporal Expressions Recognition: from TE characterization to an effective TER module implementation". In: *Proceedings of the 7th Brazilian Symposium in Information and Human Language Technology (STIL 2009)*, pp. 36-43, São Carlos, SP, Brasil, Setembro.

HAGÈGE, C., BAPTISTA, J., MAMEDE, N., 2010, "Caracterização e Processamento de Expressões Temporais em Português", *Linguamatica*, v. 2, n. 1 (Abr), pp. 63-77.

HAGÈGE, C., BAPTISTA, J., MAMEDE, N., "Reconhecimento de entidades mencionadas com o XIP: Uma colaboração entre a Xerox e o L2F do INESC-ID Lisboa". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 15, Porto, Portugal, Linguateca, 2008a.

HAGÈGE, C., BAPTISTA, J., MAMEDE, N., "Proposta de anotação e normalização de expressões temporais da categoria TEMPO para o Segundo HAREM". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., apêndice B, Porto, Portugal, Linguateca, 2008b.

HOBBS, J. R., RILOFF, E. "Information Extraction". In: Indurkhya, N., Damerau, F. J. (eds), *Handbook of Natural Language Processing*, 2 ed., capítulo 21, Goshen, CT, EUA, 2010.

HOUAISS, A., VILLAR, M. S., 2001, *Dicionário Houaiss da lingua portuguesa*. 1 ed. Rio de Janeiro, RJ, Brasil, Objetiva.

JANG, S. B., BALDWIN, J., MANI, I., 2004, "Automatic TIMEX2 Tagging of Korean News", *ACM Transactions on Asian Language Information Processing*, v. 3, n. 1 (Mar), pp. 51-65.

JAYRAM, T. S., KRISHNAMURTHY, R., RAGHAVAN, S., *et al.*, 2006, "Avatar Information Extraction System", *IEEE Data Engineering Bulletin*, v. 29, n. 1 (Mar), pp. 40-48.

JURAFSKY, D., MARTIN, J. H., 2000, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1 ed. Upper Saddle River, NJ, EUA, Prentice Hall.

KISS, T., STRUNK, J., 2006, "Unsupervised Multilingual Sentence Boundary Detection", *Computational Linguistics*, v. 32, n. 4 (Dez), pp. 485-525.

KRIPKE, S., 1980, *Naming and Necessity*. 1 ed. Cambridge, MA, EUA, Harvard University Press.

KUSHMERICK, N., 1997, *Wrapper Induction for Information Extraction*. Ph.D. dissertation, University of Washington, Seattle, WA, EUA.

LAFFERTY, J. D., McCALLUM, A., PEREIRA, F. C. N., 2001, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*, pp. 282-289, Williamstown, MA, EUA, Julho.

LAVELLI, A., CALIFF, M., CIRAVEGNA, F., *et al.*, 2004, "A Critical Survey of the Methodology for IE Evaluation". In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pp. 1-4, Lisboa, Portugal, Maio.

LAVELLI, A., MAGNINI, B., NEGRI, M., *et al.*, 2005, *Italian Content Annotation Bank (I-CAB): Temporal Expressions (v. 1.0)*. In: Technical Report T05-05-12, ITC-irst.

LEEK, T. R., 1997, *Information Extraction Using Hidden Markov Models*, M.Sc. thesis, University of California, San Diego, CA, EUA.

LI, W., WONG, K.-F., CAO, G., *et al.*, 2004, "Applying Machine Learning to Chinese Temporal Relation Resolution". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pp. 582-588, Barcelona, Espanha, Julho.

LI, W., WONG, K.-F. YUAN, C., 2001, "Toward Automatic Chinese Temporal", *Journal of the American Society for Information Science*, v. 52, n. 9 (Jul), pp. 748-762.

LINGPIPE, 2011. Disponível em: http://alias-i.com/lingpipe/. Acesso em: 6 ago. 2011.

LINGUATECA, 2011. Disponível em: http://www.linguateca.pt/. Acesso em: 2 ago. 2011.

LOPER, E., BIRD, S., 2002, "NLTK: The Natural Language Toolkit". In: *Proceedings of the Association for Computational Linguistics Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 62-69, Somerset, NJ, EUA, Julho.

LOUREIRO, J. M. S., 2007, *Reconhecimento de Entidades Mencionadas (Obra, Valor, Relações de Parentesco e Tempo) e Normalização de Expressões Temporais*, Dissertação de M.Sc., Instituto Superior Técnico (Universidade Técnica de Lisboa), Lisboa, Portugal.

MAGNINI, B., PIANTA, E., GIRARDI, C., *et al.*, 2006, "I-CAB: the Italian Content Annotation Bank", In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 963-968, Genova, Itália, Maio.

MAKHOUL, J., KUBALA, F., SCHWARTZ, R., *et al.*, 1999, "Performance Measures for Information Extraction". In: *Proceedings of DARPA Broadcast News Workshop*, pp. 249-252, Herndon, VA, EUA, Fevereiro.

MAKKONEN, J., AHONEN-MYKA, H., SALMENKIVI, M., 2003a, "Topic Detection and Tracking with Spatio-Temporal Evidence". In: *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR 2003)*, pp. 251–265, Pisa, Itália, Abril.

MAKKONEN, J., AHONEN-MYKA, H., 2003b, "Extraction of Temporal Expressions from Finnish News-feed". In: *Proceedings of 14th Nordic Conference of Computational Linguistics*, pp 1-12, Reykjavik, Islândia, Maio.

MANI, I., "Chronoscopes: A theory of underspecified temporal representations". In: *Annotating, Extracting and Reasoning about Time and Events*, v. 5151, *Dagstuhl Seminar Proceedings*, Begegnungs- und Forschungszentrum für Informatik (IBFI), pp. 127-139, 2005.

MANI, I., WILSON, G., 2000, "Robust Temporal Processing of News". In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69-76, Hong Kong, Outubro.

MANNING, C. D., RAGHAVAN, P., SCHÜTZE, H., 2008, *Introduction to Information Retrieval*. 1 ed. New York, NY, EUA, Cambridge University Press.

MARTINS, B., "O Sistema CaGE no Segundo HAREM". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 7, Porto, Portugal, Linguateca, 2008

McCALLUM, A., LI, W., 2003, "Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons". In: *Proceedings of the 7th Conference on Natural Language Learning (CoNLL 2003) at HLT-NAACL 2003*, pp. 188-191, Edmonton, Canadá, Junho.

MERCHANT, R., OKUROWSKI, M. E., CHINCHOR, N., 1996, "The Multilinguial Entity Task (MET) Overview". In: *Proceedings of the 1996 TIPSTER Workshop*, pp. 445-447, Vienna, VA, Maio.

MIKHEEV, A., MOENS, M., GROVER, C., 1999, "Named Entity Recognition without Gazetteers". In: *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, pp. 1-8, Bergen, Noruega, Junho.

MITCHELL, T., 1997, *Machine Learning*. 1 ed. New York, NY, EUA, McGraw-Hill.

MOTA, C., CARVALHO, P., FREITAS, C., *et al.*, "É Tempo de Avaliar o TEMPO". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 3, Porto, Portugal, Linguateca, 2008.

MOTA, C., SILBERZTEIN, M., "Em Busca da Máxima Precisão sem Almanaques: O Stencil/NooJ no HAREM". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 15, Lisboa, Portugal, Linguateca, 2007.

MYSQL, 2011. Disponível em: http://www.mysql.com/. Acesso em: 12 jul. 2011.

NADEAU, D., SEKINE, S., 2007, "A Survey of Named Entity Recognition and Classification", *Lingvisticae Investigationes*, v. 30, n. 1 (Jan), pp. 3-26.

NIGEM, K., LAFFERTY, J., McCALLUM, A., 1999, "Using Maximum Entropy for Text Classification". In: *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61-67, Estocolmo, Suécia, Agosto.

NING, P., WANG, X. S, JAJODIA, S., 2002, "An Algebraic Representation of Calendars", *Annals of Mathematics and Artificial Intelligence*, v. 36, n. 1 (Set), pp. 5-38.

NLTK, 2011. Disponível em: http://www.nltk.org/. Acesso em: 3 jun. 2011.

OLIVEIRA, H. G., MOTA, C., FREITAS, C., *et al.*, "Avaliação à medida no Segundo HAREM". In: Mota, C., Santos, D. (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*, 1 ed., capítulo 5, Porto, Portugal, Linguateca, 2008.

OPENNLP, 2011. Disponível em: http://maxent.sourceforge.net/. Acesso em: 3 ago. 2011.

PHP, 2011. Disponível em: http://www.php.net/. Acesso em: 13 jul. 2011.

PYTHON, 2011. Disponível em: http://www.python.org/. Acesso em: 12 jun. 2011.

PINTO, D., McCALLUM, A., WEI, X., *et al.*, 2003, "Table Extraction Using Conditional Random Fields". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 232-242, Toronto, Canada, Agosto.

PUSTEJOVSKY, J., HANKS, P., SAURI, R., *et al.*, 2003, "The TIMEBANK Corpus". In: *Proceedings of the 2003 Corpus Linguistics Conference (CL 2003)*, pp 647-656, Lancaster, Inglaterra, Março.

PUSTEJOVSKY, J., KNIPPEN, R., LITTMAN, J., *et al.*, 2005, "Temporal and Event Information in Natural Language Text", *Language Resources and Evaluation*, v. 39, n. 2 (Mai), pp. 123-164.

PUSTEJOVSKY, J., VERHAGEN, M., 2009, “SemEval-2010 Task 13 - Evaluating Events, Time Expressions, and Temporal Relations (TempEval-2)”. In: *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW 2009)*, pp 112-116, Boulder, CO, EUA, Junho.

RATNAPARKHI, A., 1998, *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, EUA.

RAU, L. F., 1991, “Extracting Company Names from Text”. In: *Proceedings of the 7th Conference on Artificial Intelligence Applications*, pp. 29-32, Miami Beach, FL, EUA, Fevereiro.

RUSSEL, S., NORVIG, P., 2002, *Artificial Intelligence: A Modern Approach*. 2 ed. Upper Saddle River, NJ, EUA, Prentice Hall.

SANG, E. F. T. K., DE MEULDER, F., 2003, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition”. In: *Proceedings of the 7th Conference on Natural Language Learning (CoNLL 2003) at HLT-NAACL 2003*, pp. 142-147, Edmonton, Canadá, Junho.

SANTOS, D., CARDOSO, N., 2006b, “A Golden Resource for Named Entity Recognition in Portuguese”. In: *Proceedings of the 7th International Workshop on Computational Processing of the Portuguese Language (PROPOR 2006)*, pp. 69-79, Itatiaia, MG, Brasil, Maio.

SANTOS, D., CARDOSO, N., “Breve introdução ao HAREM”. In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 1, Lisboa, Portugal, Linguateca, 2007a.

SANTOS, D., CARDOSO, N., SECO, N., “Avaliação no HAREM: Métodos e Medidas”. In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 18, Lisboa, Portugal, Linguateca, 2007b.

SANTOS, D., FREITAS, C., OLIVEIRA, H. G., *et al.*, 2008, "Second HAREM: New Challenges and Old Wisdom". In: *Proceedings of the 8th International Conference on Computational Processing of the Portuguese Language (PROPOR 2008)*, pp. 212-215, Aveiro, Portugal, Setembro.

SANTOS, D., SECO, N., CARDOSO, N., *et al.*, 2006a, "HAREM: An Advanced NER Evaluation Contest in Portuguese", In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 1986-1991, Genova, Itália, Maio.

SAQUETE, E., MARTÍNEZ-BARCO, P., 2000, "Grammar Specification for the Recognition of Temporal Expressions", In: *Proceedings of the 2000 International Conference on Machine Translation and Multilingual Applications in the New Millennium (MT 2000)*, pp. 21-27, Exeter, Inglaterra, Novembro.

SAQUETE, E., MARTÍNEZ-BARCO, P., MUÑOZ, R., 2002, "Recognizing and Tagging Temporal Expressions in Spanish", In: *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pp. 1-8, Las Palmas, Espanha, Maio.

SARAWAGI, S., 2008, "Information Extraction", *Foundations and Trends in Databases*, v. 1, n. 3 (Mar), pp. 261-377.

SARMENTO, L., "O SIEMÊS e a sua participação no HAREM e no Mini-HAREM". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 14, Lisboa, Portugal, Linguateca, 2007.

SAURÍ, R., SAQUETE, E., PUSTEJOVSKY, J., 2010, *Annotating Time Expressions in Spanish*. In: TimeML Annotation Guidelines, TempEval-2010.

SCHILDER, F., HABEL, C., 2001, "From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages". In: *Proceedings of the Workshop on Temporal and Spatial Information Processing at the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, pp. 65-72, Toulouse, França, Julho.

SCHMID, H., 1994, “Probabilistic Part-of-Speech Tagging Using Decision Trees”. In: *Proceedings of the International Conference on New Methods in Language Processing*, pp. 44-49, Manchester, Inglaterra, Setembro.

SECO, N., “MUC vs HAREM - A Contrastive Perspective”. In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 3, Lisboa, Portugal, Linguateca, 2007.

SEKINE, S., 1998, “NYU: Description of the Japanese NE System Used for MET-2”. In: *Proceedings of the 7th Message Understanding Conference (MUC-7)*, pp. 1-6, Fairfax, VA, EUA, Abril.

SEKINE, S., ISAHARA, H., 2000, “IREX: IR and IE Evaluation Project in Japanese”. In: *Proceedings of the 2000 Language Resources and Evaluation Conference (LREC 2000)*, pp. 1-6, Atenas, Grécia, Junho.

SEKINE, S., NOBATA, C., 2004, “Definition, Dictionary and Tagger for Extended Named Entities”, In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 1977-1980, Lisboa, Portugal, Maio.

SETZER, A., 2001, *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. Ph.D. thesis, University of Sheffield, Sheffield, Inglaterra.

SETZER, A., GAIZAUSKAS, R., 2000a, “Annotating Events and Temporal Information in Newswire Texts”. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, pp. 1-7, Atenas, Grécia, Junho.

SETZER, A., GAIZAUSKAS, R., 2000b, “Annotating Events and Temporal Information in Newswire Texts”. In: *Proceedings of the Information Extraction Meets Corpus Linguistics Pre-Conference Workshop at the 2nd International Conference on Language Resources and Evaluation*, pp. 1-6, Atenas, Grécia, Maio.

SETZER, A., GAIZAUSKAS, R., 2001, “A Pilot Study on Annotating Temporal Relations in Text”. In: *Proceedings of the Workshop on Temporal and Spatial*

*Information Processing at the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, pp. 65-72, Toulouse, França, Julho.

SOLORIO, T., "MALINCHE: A NER System for Portuguese that Reuses Knowledge from Spanish". In: Santos, D., Cardoso, N. (eds), *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 1 ed., capítulo 10, Lisboa, Portugal, Linguateca, 2007.

TERN, 2004. Disponível em: http://timex2.mitre.org/tern_evalplan-2004.29apr04.pdf. Acesso em: 6 jun. 2011.

THIELEN, C., 1995, "An Approach to Proper Name Tagging for German". In: *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1995)*, pp. 1-6, Dublin, Irlanda, Março.

TREUMUTH, M., 2008, "Normalization of Temporal Information in Estonian". In: *Proceedings of the 11th International Conference on Text, Speech and Dialogue (TSD 2008)*, pp. 211-218, Brno, República Tcheca, Setembro.

VAZOV, N., 2001a, "A System for Extraction of Temporal Expressions from French Texts". In: *Proceedings of the 8ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN 2001)*, pp. 315-324, Tours, França, Julho.

VAZOV, N., 2001b, "A System for Extraction of Temporal Expressions from French Texts Based on Semantic and Syntactic Constraints". In: *Proceedings of the Workshop on Temporal and Spatial Information Processing at the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, pp. 96-103, Toulouse, França, Julho.

VERHAGEN, M., GAIZAUSKAS, R., SCHILDER, F., *et al.*, 2007, "SemEval-2007 Task 15: TempEval Temporal Relation Identification", In: *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 75-80, Praga, República Tcheca, Junho.

VERHAGEN, M., GAIZAUSKAS, R., SCHILDER, F., *et al.*, 2009, "The TempEval Challenge: Identifying Temporal Relations in Text", *Language Resources and Evaluation*, v. 43, n. 2 (Jun), pp. 161-179.

VERHAGEN, M., SAURI, R., CASELLI, T., *et al.*, 2010, "SemEval-2010 Task 13 - TempEval-2", In: *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010)*, pp. 57-62, Uppsala, Suécia, Julho.

VICENTE-DÍEZ, M. T., SAMY, D., MARTÍNEZ, P., 2008, "An Empirical Approach to a Preliminary Successful Identification and Resolution of Temporal Expressions in Spanish News Corpora", In: *Proceedings of the 6th International Language Resources and Evaluation (LREC 2008)*, pp. 2153-2158, Marrakech, Marrocos, Maio.

WIEBE, J., O'HARA, T. P., ÖHRSTRÖM-SANDGREN, T., *et al.*, 1998, "An Empirical Approach to Temporal Reference Resolution", *Journal of Artificial Intelligence Research*, v. 9, n. 1 (Ago), pp. 247-293.

XSLT, 2011. Disponível em: http://www.w3.org/TR/xslt. Acesso em: 17 jun. 2011.