Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Thompson Furtado

Interpretação automática de relatórios de operação de equipamentos

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio.

Orientador: Prof. Hélio Côrtes Vieira Lopes

Rio de Janeiro
Abril de 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Pedro Henrique Thompson Furtado

Interpretação automática de relatórios de operação de equipamentos

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Hélio Côrtes Vieira Lopes
Orientador
Departamento de Informática – PUC-Rio

Prof. Marco Antonio Casanova
Departamento de Informática – PUC-Rio

Profª. Simone Diniz Junqueira Barbosa
Departamento de Informática – PUC-Rio

Dr. Ismael Humberto Ferreira dos Santos
Petróleo Brasileiro - Rio de Janeiro – Matriz

Prof. Márcio da Silveira Carvalho
Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 20 de Abril de 2017

Pedro Henrique Thompson Furtado

Graduou-se em Engenharia Química pela Universidade Federal do Rio de Janeiro (UFRJ) em 2008. Completou, em 2010, o Curso de Formação de Engenheiros de Processamento da PETROBRAS, atuando desde então como pesquisador no Centro de Pesquisas e Desenvolvimento Leopoldo Américo Miguez de Mello (CENPES), nos temas Automação, Controle e Otimização de Processos e *Data Science* aplicados à área de Exploração e Produção de petróleo e gás natural.

Ficha Catalográfica

Furtado, Pedro Henrique Thompson

Interpretação automática de relatórios de operação de equipamentos / Pedro Henrique Thompson Furtado; orientador: Hélio Côrtes Vieira Lopes. – 2017.

130 f. : il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2017.

Inclui bibliografia

1. Informática – Teses. 2. Processamento de linguagem natural. 3. Aprendizado automático. 4. Ontologias. 5. Petróleo e gás natural. I. Lopes, Hélio Côrtes Vieira. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

É certo que meras palavras não são suficientes para agradecer a pessoas tão especiais e que contribuíram tanto para essa conquista.

Em primeiro lugar, agradeço Àquele que me deu a vida, a consciência e tudo o que foi necessário para chegar até aqui. A Sua Palavra inerrante afirma que Ele *"...é a luz que ilumina todos os homens."* (Evangelho de João 1:9b). A Jesus, o Filho Unigênito do Deus Eterno e Imortal, meu Amigo, meu Consolador, em quem *"estão escondidos todos os tesouros da sabedoria e do conhecimento"*, toda a honra e todo o louvor.

Agradeço à minha amada esposa Nathálya e ao meu pequeno Guilherme, que sofreram tantas ausências em razão deste grande esforço. Sem vocês, nenhuma conquista faria sentido, por maior que fosse.

Agradeço aos meus pais, Pedro e Elicéia, responsáveis por toda a base de exemplo, amor e cuidado; fundamentos que me permitiram ser, antes de pesquisador, um homem. Agradeço à minha querida irmã Amanda, participante de tudo isso, por seu carinho e admiração constantes.

Agradeço ao Prof. Hélio Lopes que, antes de orientador, é um verdadeiro amigo, que aceitou este desafio tão prontamente e foi fundamental em todas as etapas. Agradeço por acreditar em mim e me apoiar irrestritamente.

Agradeço especialmente ao meu grande amigo, mais que um irmão, Jonatas Grosman, sem o qual o resultado deste trabalho não seria alcançado, dado o tamanho do desafio que resolvemos enfrentar juntos.

Agradeço aos meus amigos/irmãos do Laboratório IDEIAS, Sônia, Cássio, Jefry e William, por me receberem de braços abertos e com um carinho tão sincero, perceptível em cada olhar. Nossas conversas tornaram esse caminho mais agradável.

Agradeço muito especialmente aos meus grandes amigos e companheiros do dia a dia de trabalho, Fábio Diehl, Thiago Anzai, Cristina Almeida, Tatiane Machado, Marcelo e Marcos Felipe.

Por fim, agradeço à PETROBRAS, orgulho dos brasileiros, motivadora, incentivadora e financiadora deste trabalho.

Resumo

Furtado, Pedro Henrique Thompson; Lopes, Hélio Côrtes Vieira. **Interpretação automática de relatórios de operação de equipamentos**. Rio de Janeiro, 2017. 130p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As unidades operacionais da área de Exploração e Produção (E&P) da PETROBRAS utilizam relatórios diários para o registro de situações e eventos em Unidades Estacionárias de Produção (UEPs), as conhecidas plataformas de produção de petróleo. Um destes relatórios, o SITOP (Situação Operacional das Unidades Marítimas), é um documento diário em texto livre que apresenta informações numéricas (índices de produção, algumas vazões, etc.) e, principalmente, informações textuais. A parte textual, apesar de não estruturada, encerra uma valiosíssima base de dados de histórico de eventos no ambiente de produção, tais como: quebras de válvulas, falhas em equipamentos de processo, início e término de manutenções, manobras executadas, responsabilidades etc. O valor destes dados é alto, mas o custo da busca de informações também o é, pois se demanda a atenção de técnicos da empresa na leitura de uma enorme quantidade de documentos. O objetivo do presente trabalho é o desenvolvimento de um modelo de processamento de linguagem natural para a identificação, nos textos dos SITOPs, de entidades nomeadas e extração de relações entre estas entidades, descritas formalmente em uma ontologia de domínio aplicada a eventos em unidades de processamento de petróleo e gás em ambiente *offshore*. Ter-se-á, portanto, um método de estruturação automática da informação presente nestes relatórios operacionais. Os resultados obtidos demonstram que a metodologia é útil para este caso, ainda que passível de melhorias em diferentes frentes. A extração de relações apresenta melhores resultados que a identificação de entidades, o que pode ser explicado pela diferença entre o número de classes das duas tarefas. Verifica-se também que o aumento na quantidade de dados é um dos fatores mais importantes para a melhoria do aprendizado e da eficiência da metodologia como um todo.

Palavras-chave

Processamento de linguagem natural; Aprendizado automático; Ontologias; Petróleo e gás natural.

Abstract

Furtado, Pedro Henrique Thompson; Lopes, Hélio Côrtes Vieira (Advisor). **Automatic interpretation of equipment operation reports**. Rio de Janeiro, 2017. 130p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The operational units at the Exploration and Production (E&P) area at PETROBRAS make use of daily reports to register situations and events from their Stationary Production Units (SPUs), the well-known petroleum production platforms. One of these reports, called SITOP (the Portuguese acronym for Offshore Unities' Operational Situation), is a daily document in free text format that presents numerical information and, mainly, textual information about operational situation of offshore units. The textual section, although unstructured, stores a valuable database with historical events in the production environment, such as: valve breakages, failures in processing equipment, beginning and end of maintenance activities, actions executed, responsibilities, etc. The value of these data is high, as well as the costs of searching relevant information, consuming many hours of attention from technicians and engineers to read the large number of documents. The goal of this dissertation is to develop a model of natural language processing to recognize named entities and extract relations among them, described formally as a domain ontology applied to events in offshore oil and gas processing units. After all, there will be a method for automatic structuring of the information from these operational reports. Our results show that this methodology is useful in SITOP's case, also indicating some possible enhancements. Relation extraction showed better results than named entity recognition, what can be explained by the difference in the amount of classes in these tasks. We also verified that the increase in the amount of data was one of the most important factors for the improvement in learning and methodology efficiency as a whole.

Keywords

Natural language processing; automatic learning; ontologies; oil and gas.

Sumário

Lista de figuras

Lista de tabelas

As únicas coisas que podemos conservar são as que entregamos a Deus. As que guardamos para nós são as que perderemos com certeza.

C.S. Lewis, *Cristianismo Puro e Simples.*

1
Introdução

As décadas recentes testemunharam uma expansão sem precedente no volume de dados não estruturados disponíveis em formato digital. Empresas estão começando a reconhecer o potencial valor econômico presente em grandes repositórios e fontes de dados textuais, incluindo externos ou de terceiros, como as redes sociais, e internos, como relatórios e coleções de documentos típicos de cada companhia (1). Informações extraídas destas fontes, muitas vezes subutilizadas, podem ter um elevado valor para uma série de aplicações.

Um relatório do ano de 2012 da *McKinsey Global Institute* (2) demonstrou claramente o potencial de retorno econômico de técnicas de mineração de dados textuais e não-textuais. Estimou-se, por exemplo, que um melhor aproveitamento de dados com as técnicas apropriadas de mineração textual e não-textual permitiria ao sistema de saúde americano a criação de mais de US$ 300 bilhões em valor anualmente. Semelhantemente, o uso eficiente de informações dos dados para melhorar operações e detectar fraudes poderia gerar para a administração do setor público europeu até US$ 250 bilhões de valor potencial anual.

1.1
Uso combinado de NLP e ontologias de domínio

Dados textuais podem ser organizados em diversos formatos, quase sempre em estruturas de anotação ou etiquetagem (XML, HTML etc.), que conferem aos textos significados objetivos e eficiência de acesso. O uso de ontologias de domínio leva este conceito a um nível mais alto, pois são poderosas ferramentas de representação de conhecimento, descrevendo um conjunto de relevantes conceitos específicos de domínios e suas relações em um sistema formal (3), (4). A união de tecnologias de *Natural Language Processing* (NLP) ao conceito de ontologias de domínio pode gerar resultados interessantes (5), transformando dados não estruturados, sem qualquer significado computacionalmente processável, em bases de conhecimentos descritíveis logicamente.

O uso destas técnicas inspira aplicações na maioria das áreas de negócios e indústrias. A área de Petróleo e Gás Natural, muito lucrativa e da maior

importância no cenário internacional, é um ambiente onde ferramentas que envolvam aprendizado de máquina encontram um "solo fértil", dados os muitos problemas a serem resolvidos em diversos aspectos. Há, por exemplo, trabalhos relacionados à aplicação de *Big Data Analytics* na produção de petróleo, como para detecção e predição de falhas em equipamentos de processo (6) e para análise e otimização de questões logísticas (7).

1.2 Uma possível aplicação na PETROBRAS

É comum, no âmbito dos processos produtivos desta indústria, a produção e uso de relatórios operacionais que descrevem o dia a dia das plantas, sistemas etc. A PETROBRAS, uma das maiores empresas de petróleo do mundo, não destoa das demais neste caso e utiliza, para o registro de informações, diversas bases de dados, estruturadas e não estruturadas. Uma das bases de dados textuais de sua área de Exploração e Produção (E&P), os relatórios SITOP (Situação Operacional das Unidades Marítimas), reúne informações diárias da operação de diversos sistemas das unidades marítimas da empresa. Estes documentos contêm informações numéricas (dados de produção óleo, dados de descarte de água, etc.) e textuais. A parte numérica, ilustrada na figura 1.1, apresenta uma certa estrutura e boa parte dos dados ali inseridos estão devidamente estruturados em outras bases de dados.

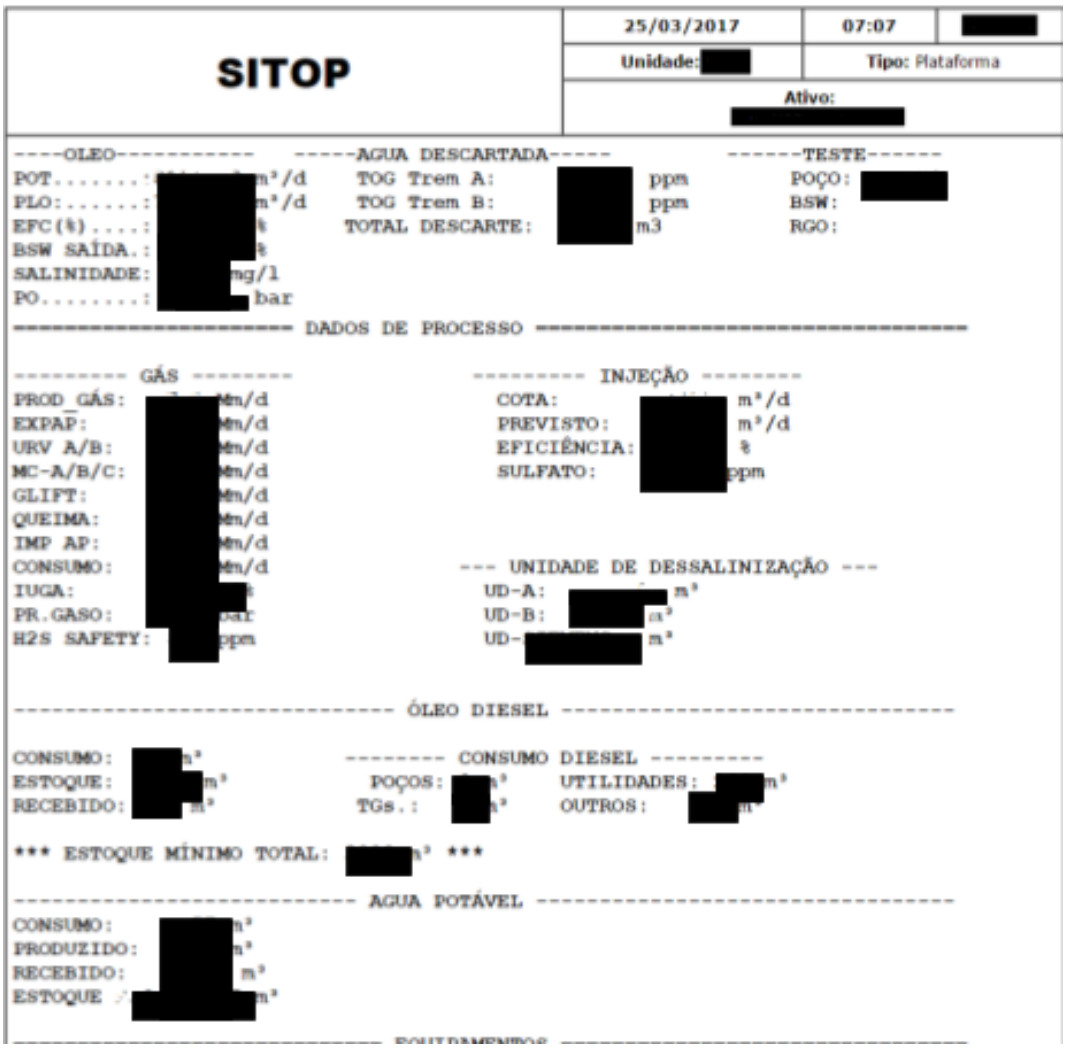
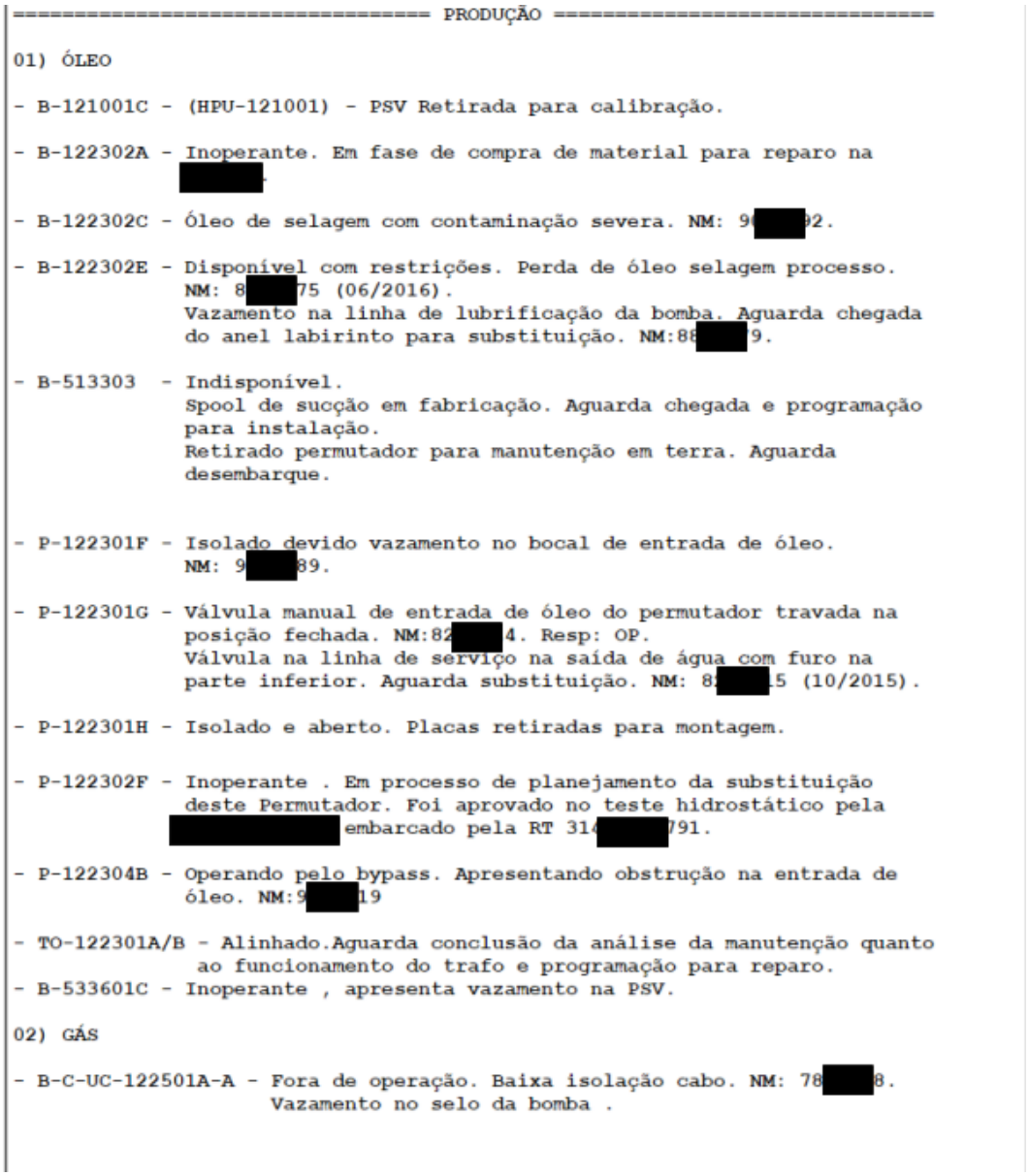


Figura 1.1: Exemplo da parte mais "estruturada", com dados numéricos, de um relatório SITOP. (Alguns dados foram corrompidos e descaracterizados propositalmente por razões de confidencialidade)

Entretanto, a parte textual, ilustrada na figura 1.2, diz respeito a eventos operacionais em muitos equipamentos e sistemas e é de escrita livre, isto é, operadores e responsáveis pelo gerenciamento das unidades registram descrições livres de situações relativas a sistemas e equipamentos como falhas, manutenções, trocas de equipamentos, atividades importantes, entre outras.



```
================================ PRODUÇÃO ================================

01) ÓLEO

- B-121001C - (HPU-121001) - PSV Retirada para calibração.

- B-122302A - Inoperante. Em fase de compra de material para reparo na
              [redacted].

- B-122302C - Óleo de selagem com contaminação severa. NM: 9[redacted]92.

- B-122302E - Disponível com restrições. Perda de óleo selagem processo.
              NM: 8[redacted]75 (06/2016).
              Vazamento na linha de lubrificação da bomba. Aguarda chegada
              do anel labirinto para substituição. NM:88[redacted]9.

- B-513303  - Indisponível.
              Spool de sucção em fabricação. Aguarda chegada e programação
              para instalação.
              Retirado permutador para manutenção em terra. Aguarda
              desembarque.


- P-122301F - Isolado devido vazamento no bocal de entrada de óleo.
              NM: 9[redacted]89.

- P-122301G - Válvula manual de entrada de óleo do permutador travada na
              posição fechada. NM:82[redacted]4. Resp: OP.
              Válvula na linha de serviço na saída de água com furo na
              parte inferior. Aguarda substituição. NM: 8[redacted]5 (10/2015).

- P-122301H - Isolado e aberto. Placas retiradas para montagem.

- P-122302F - Inoperante . Em processo de planejamento da substituição
              deste Permutador. Foi aprovado no teste hidrostático pela
              [redacted]embarcado pela RT 31[redacted]791.

- P-122304B - Operando pelo bypass. Apresentando obstrução na entrada de
              óleo. NM:9[redacted]19

- TO-122301A/B - Alinhado.Aguarda conclusão da análise da manutenção quanto
                 ao funcionamento do trafo e programação para reparo.
- B-533601C - Inoperante , apresenta vazamento na PSV.

02) GÁS

- B-C-UC-122501A-A - Fora de operação. Baixa isolação cabo. NM: 78[redacted]8.
                     Vazamento no selo da bomba .
```

Figura 1.2: Exemplo da parte em texto livre de um relatório SITOP. (Alguns dados foram corrompidos e descaracterizados propositalmente por razões de confidencialidade)

Trata-se, portanto, de um importante repositório de informações textuais históricas dos processos (são milhares documentos, de muitos anos), úteis para diversos fins: busca por soluções para falhas recentes pela consulta a casos similares no passado, análises de falhas para fins de manutenção, gerenciamento diário do processo, etc. As informações em texto livre apresentam a vantagem de permitirem um registro mais profundo e específico para cada caso registrado, mas também a desvantagem de não estarem em um formato facilmente tratável

por computadores. A já citada atividade de busca por soluções para problemas atuais pela consulta a casos semelhantes no passado, por exemplo, envolve a consulta manual, por especialistas, de relatório por relatório, buscando as informações e analisando as pertinentes ao objetivo do estudo. Isso, por si só, mostra o alto valor dos dados e, ao mesmo tempo, o alto custo de seu uso.

Assim, a estruturação automática dos relatórios SITOP e talvez de outros documentos em texto livre envolvidos no negócio da PETROBRAS, agregaria valor aos seus conteúdos tendo em vista as muitas aplicações que se viabilizariam a partir de então. Poder-se-ia, por exemplo, unir dados de falhas equipamentos registrados textualmente a dados de sensores de processo (pressões, temperaturas, vazões etc.) para a construção de modelos preditivos para falhas, geração de *dashboards* para cada equipamento com linhas de tempo informando seus eventos críticos, entre tantos outros usos nobres.

Esta dissertação apresenta os resultados iniciais de uma pesquisa cujo objetivo é aplicar métodos de NLP às informações de relatórios SITOP reais a fim de estruturar os dados textuais em um formato computacionalmente tratável. A contribuição desse trabalho se dá pela proposta de uma ontologia leve baseada na ISO 14224, modelando os eventos descritos nos textos do SITOP; pelo uso da ontologia proposta como estrutura para a extração de informações; pelo desenvolvimento, em conjunto com outro pesquisador, de um novo sistema em ambiente de nuvem para toda a cadeia de atividades necessárias para a preparação de dados e construção de modelos de aprendizado de máquina a serem usados na mineração de dados textuais; e melhorando a abordagem da literatura para a extração das relações com base na ontologia, pelo uso de um modelo alternativo de aprendizado estruturado.

A estrutura desta dissertação segue esta ordem: uma introdução ao processamento primário de petróleo, apresentando os equipamentos mais importantes e seus princípios de funcionamento; uma revisão bibliográfica relativa ao tema de *Text Analytics* (TA) e ao uso de ontologias para anotação de dados textuais de processos industriais; uma proposta de ontologia relacionada ao caso dos textos do SITOP; um descritivo geral do sistema projetado, implementado e usado no âmbito deste trabalho; os experimentos executados; uma discussão dos resultados alcançados e, por fim, uma conclusão e proposta de trabalhos futuros.

2
Introdução ao processamento primário de petróleo

A produção de petróleo e seus derivados envolve uma série de atividades e disciplinas especializadas. Em um modo esquemático geral, essa indústria está dividida em três grandes áreas, que refletem as fases de sua cadeia produtiva: *upstream*, *midstream* e *downstream* (8). *Upstream* é a fase que envolve as atividades de exploração e produção de petróleo e gás natural. *Midstream* é a fase que envolve as atividades de processamento intermediário, armazenamento e transporte das principais correntes produzidas. *Downstream*, por fim, diz respeito às atividades posteriores à produção, sendo o refino e produção de derivados o principal exemplo.

Esta pesquisa se dedica a uma questão relativa ao *Upstream*, à área de E&P, particularmente à produção de petróleo em ambiente *offshore*. A figura 2.1 ilustra um esquema produtivo *offshore* como um todo.

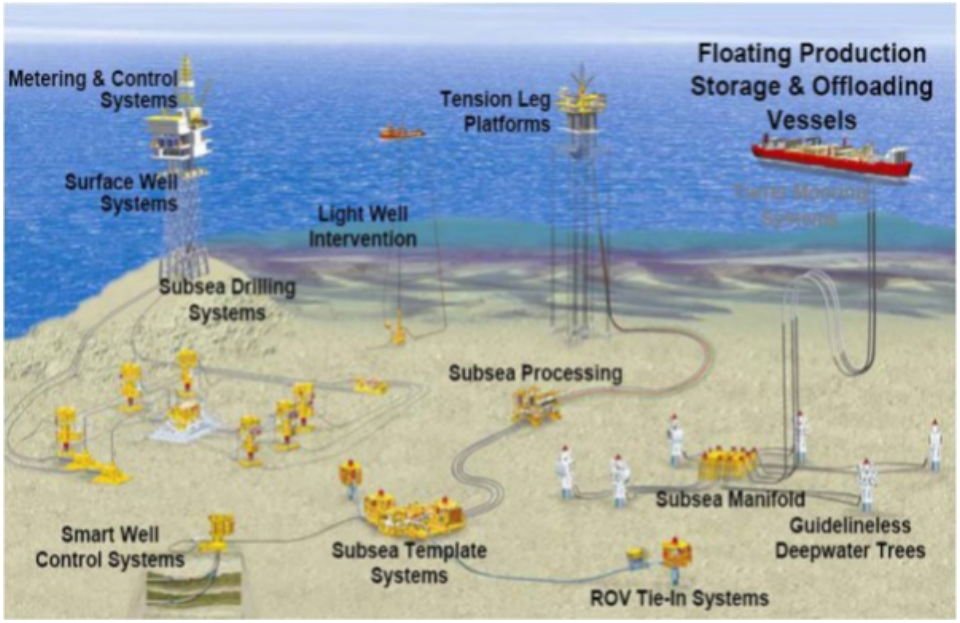


Figura 2.1: Esquema resumido de produção de petróleo *offshore*. (9)

Os poços de produção e linhas submarinas servem à elevação e escoamento dos fluidos dos reservatórios para as unidades estacionárias de produção (UEP) e, dependendo das condições destes fluidos e de questões

de viabilidade econômica, as unidades podem ser mais ou menos complexas. A mistura que chega às plataformas contém principalmente óleo, água, gás e diferentes espécies de sólidos e, uma vez que o interesse econômico está somente na produção de hidrocarbonetos (óleo e gás), as UEPs são dotadas de "facilidades de produção" (10) que são instalações destinadas a efetuar, sob condições controladas, o "processamento primário" dos fluidos. Esse processo envolve três etapas principais:

- A separação das correntes de óleo, água e gás com as impurezas ainda presentes;
- O tratamento ou condicionamento da corrente oleosa para seu enquadramento em termos de teores de água e gás, para o transporte eficiente e seguro para as facilidades *onshore* e atividades posteriores de refino;
- O tratamento ou condicionamento da corrente gasosa para seu enquadramento em termos de teores de algumas espécies e de condições físicas específicas, para então ser destinada ao transporte para o ambiente*onshore* (se economicamente viável), a procedimentos de elevação artificial de petróleo conhecidos como *gas lift*, à injeção em poços para a recuperação secundária de reservatórios ou ao consumo interno na unidade como combustível;
- O tratamento da corrente aquosa para reinjeção em poços (recuperação secundária) ou descarte sob condições reguladas.

A figura 2.2 apresenta um esquemático típico e geral deste processo.

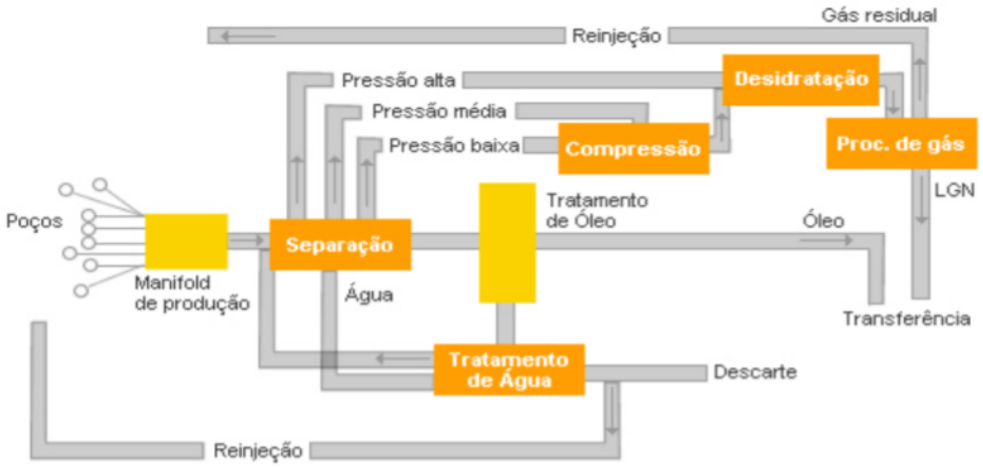


Figura 2.2: Esquema geral do processamento primário de petróleo. (10)

2.1
Separação das correntes principais

Logo que chega às unidades, a mistura produzida passa por uma primeira separação em vasos separadores, que podem ser bifásicos ou trifásicos, ilustrados nas figuras 2.3 e 2.4, atuando em série ou em paralelo. Os separadores bifásicos dividem a corrente multifásica em uma corrente gasosa e outra líquida, enquanto os separadores trifásicos também separam a corrente líquida em uma corrente aquosa (com algum teor de óleo) e uma corrente oleosa (com algum teor de água). Esses vasos podem ser fabricados nos formatos horizontal ou vertical, sendo os horizontais os normalmente mais eficientes por apresentarem uma maior área superficial de interface e permitirem, consequentemente, uma melhor separação entre fases.

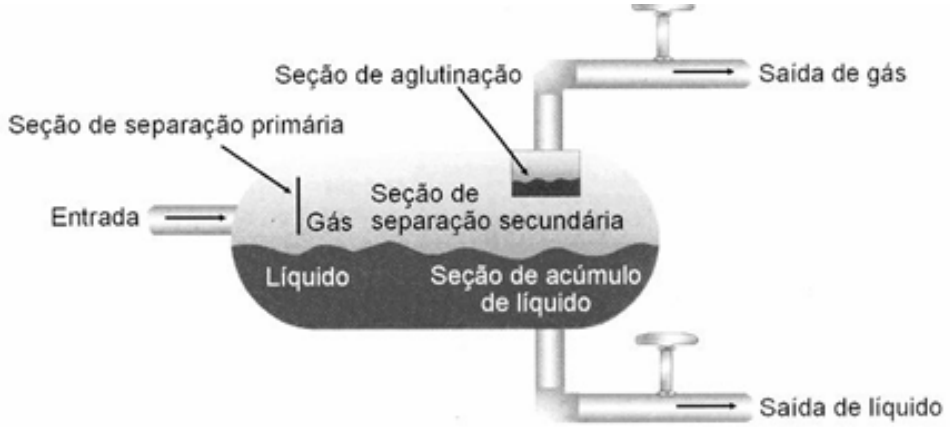


Figura 2.3: Esquema representativo de um separador bifásico horizontal. (10)

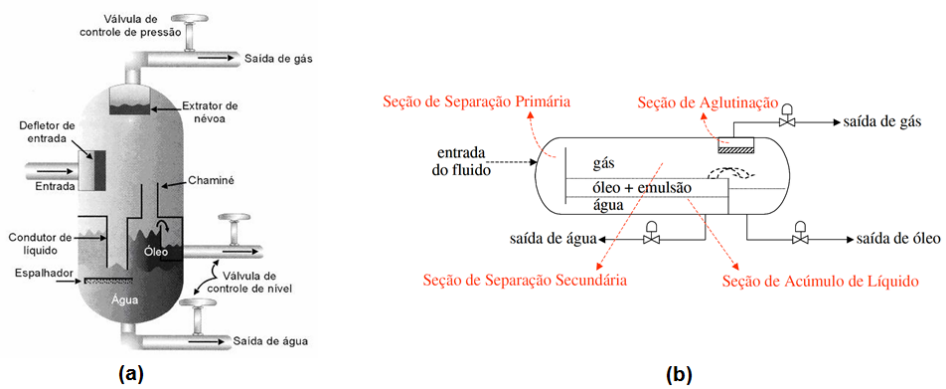


Figura 2.4: Esquemas representativos de (a) um separador trifásico vertical (10) e (b) um separador trifásico horizontal (11).

A separação de fases neste tipo de equipamento acontece pelos seguintes princípios:

- Ação da gravidade sobre as diferentes densidades;
- Desprendimento do gás sob as mudanças bruscas de velocidade e de direção do fluxo;
- Aglutinação de gotículas de água dispersas na fase oleosa com consequente coalescência e decantação;

Em vasos separadores, são comuns os seguintes problemas:

- Formação de espuma, causada principalmente por impurezas presentes na mistura, que dificulta o controle de nível do líquido em seu interior. A espuma ocupa um volume que poderia estar disponível para a coleta de líquido ou para a decantação e pode ser arrastada pelas correntes separadas, dificultando os processos a jusante;
- Obstrução pelo acúmulo de parafinas e de areia, diminuindo a eficiência de separação;
- Estabilização de emulsões, que são formadas na interface entre as fases oleosa e aquosa, problemáticas para a operação por motivos semelhantes aos da formação de espuma;
- Arraste de líquido pela corrente gasosa ou de gás pela corrente líquida.

2.2 Tratamento do gás

O condicionamento ou tratamento da corrente gasosa abrange um conjunto de processos físicos e químicos que visam a remoção ou redução dos teores de contaminantes (teores máximos de enxofre, de dióxido de carbono e de água) e enquadramento de condições físicas (ponto de orvalho e poder calorífico) para o atendimento de especificações de segurança, mercado ou de processamentos posteriores. O processo de desidratação, isto é, a remoção de água da corrente gasosa, pode ser feito através de absorção, com soluções de glicol, ou adsorção, com materiais como alumina, sílica gel e peneiras moleculares, que apresentam grande área superficial e afinidade pela molécula de água. A remoção de gases ácidos, (CO_2 e compostos de enxofre) pode ser efetuada através de processos de absorção química, geralmente com o uso de aminas, ou física, pelo uso de membranas.

2.3 Tratamento do óleo

O tratamento de óleo visa principalmente enquadrar o seu teor de água. A presença de água associada ao petróleo provoca uma série de problemas nas etapas de produção, transporte e refino. Na produção e transporte os maiores inconvenientes estão ligados:

- À necessidade de superdimensionamento das instalações de coleta, armazenamento e transferência, incluindo bombas, linhas, tanques etc.;
- Ao maior consumo de energia de um modo geral;
- À segurança operacional, pois a água pode, em determinadas condições, provocar problemas de corrosão e incrustação, causando danos às tubulações e equipamentos, potencialmente redundando em acidentes humanos e/ou ambientais.

No refino, outros problemas estão envolvidos e são ligados principalmente à presença de cloretos de cálcio e magnésio dissolvidos na água, pois estes compostos, sob a ação do calor, geram ácidos que causam corrosões em diversos equipamentos e acessórios, novamente potencializando acidentes, perdas de produtos, etc. Assim, a eliminação da água proporciona um tempo de operação mais longo e seguro de diversas unidades e equipamentos ao longo da cadeia produtiva do petróleo e reduz custos de manutenção em geral e custos de consumo de produtos químicos para neutralização dos ácidos. Trata-se, portanto, de uma etapa importantíssima do processamento primário.

Além do separador trifásico no início do processo, que já retira a maior quantidade de água livre em solução, o principal equipamento do tratamento de óleo é o tratador eletrostático, ilustrado na figura 2.5, frequentemente encontrado em sistemas marítimos de produção. Seu princípio de separação é a aplicação de campo elétrico de alta voltagem (15 kV a 50 kV) a uma emulsão, que faz com que as gotículas de água dispersas no óleo (meio de baixa constante dielétrica) adquiram uma forma elíptica (ver figura 2.6), se alinhem na direção do campo com pólos induzidos de sinais contrários que criam uma força atrativa e, consequentemente, se choquem e coalesçam, aumentando seus diâmetros e decantando mais rapidamente.

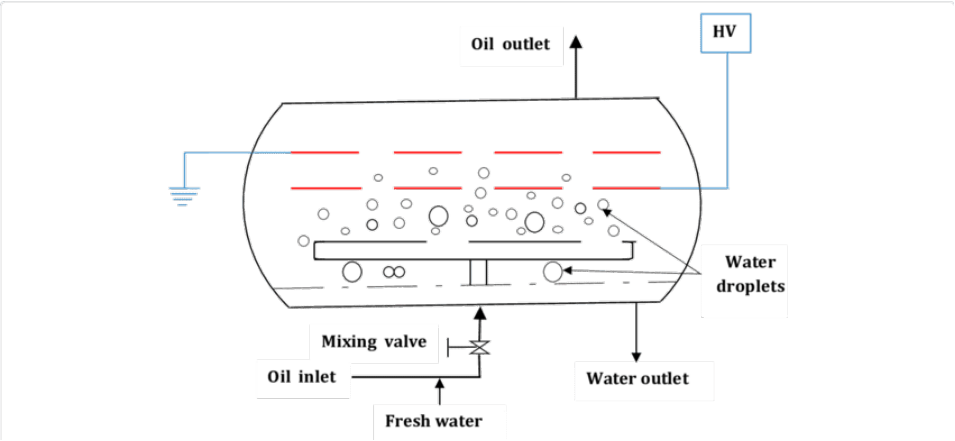


Figura 2.5: Esquema ilustrativo de um tratador eletrostático de óleo. (12)

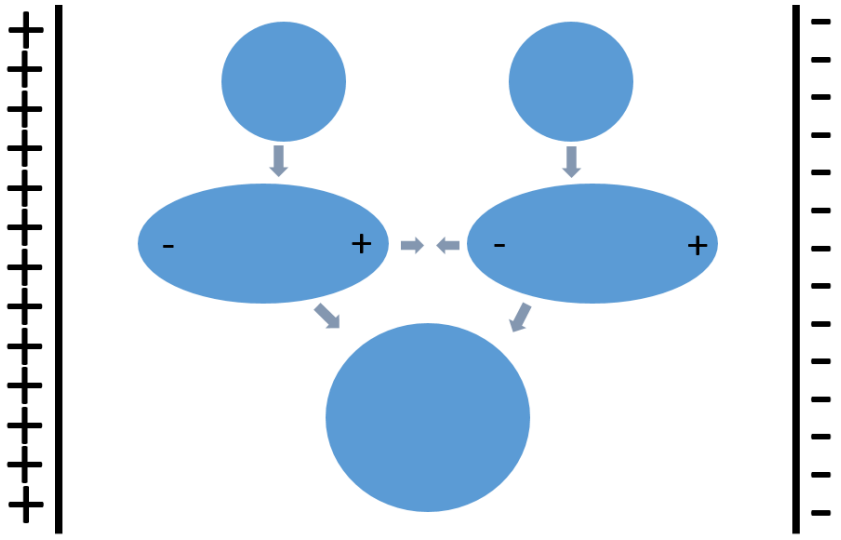


Figura 2.6: Mecanismo de separação de água em óleo por coalescência, através da aplicação de campo elétrico.

2.4 Tratamento da água

O tratamento de água produzida tem por finalidade recuperar parte do óleo nela presente em forma de emulsão e condicioná-la para reinjeção ou descarte, dentro de condições estabelecidas por normas. Tipicamente, a água proveniente dos separadores e tratadores de óleo é enviada para um vaso desgaseificador - que geralmente é um separador trifásico de baixa pressão – para a remoção de traços de gás ainda presentes no líquido. Em seguida, passa

a um sistema de remoção de óleo (enquadramento do teor de óleo em água), que pode contar com flotadores e/ou hidrociclones, indo em seguida para um tubo de despejo, no caso das unidades marítimas. Os flotadores recuperam o resíduo de óleo em água pela separação gravitacional, uma vez que as gotas de óleo são flotadas para a fase oleosa do equipamento e separadas.

Os hidrociclones, ilustrados pela figura 2.7, são equipamentos tubulares usados para acelerar essa separação através da aplicação força centrífuga: a água oleosa é introduzida tangencialmente, sob pressão, no trecho de maior diâmetro do hidrociclone, sendo direcionada internamente em fluxo espiral em direção ao trecho de menor diâmetro, o que gera uma aceleração que cria, enfim, um campo centrífugo que aumenta força de separação das fases, fazendo a água se concentrar na periferia e o óleo no centro. A água separada flui pelo lado de menor diâmetro e o rejeito oleoso pelo de maior diâmetro. Os hidrociclones são equipamentos especialmente importantes em ambiente *offshore* devido ao seu tamanho e peso baixos, uma vez que esses são parâmetros essenciais em um processo industrial que ocorre sobre uma unidade flutuante.

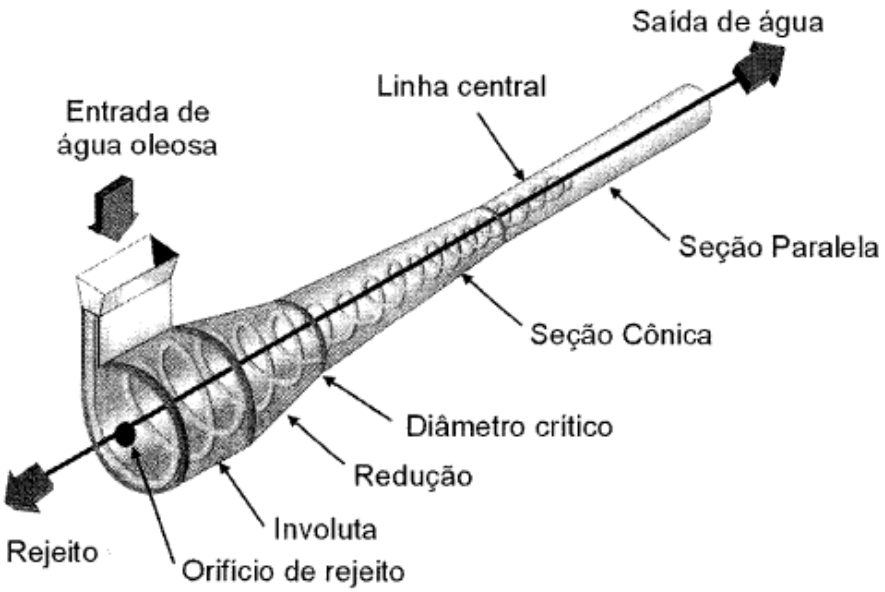


Figura 2.7: Esquema representativo de um hidrociclone, para redução do teor de óleo em água. (10)

Os destinos finais desta corrente aquosa são o lançamento no mar ou a reinjeção nos reservatórios para recuperação secundária de óleo. O descarte da água só pode ser feito dentro de determinadas especificações, regulamentadas por órgão de controle do meio ambiente que limita a quantidade de poluentes (teores de óleo, graxa, H_2S etc.) nos efluentes aquosos.

2.5 Outros equipamentos

Além dos equipamentos separadores e tratadores, são necessários aqueles que, pelo fornecimento de energia aos fluidos, os deslocam pelo sistema (bombas e compressores); os que regulam os caminhos e vazões (válvulas manuais e de controle) e os que trocam energia térmica entre correntes frias que precisam de aquecimento e correntes quentes que precisam de resfriamento (trocadores ou permutadores de calor). Há também equipamentos de apoio, motores, tanques, sistemas lógicos de controle etc. Todo este conjunto opera de forma controlada para que o processamento primário aconteça da melhor forma possível.

Os relatórios SITOP contêm informações sobre operações em unidades marítimas da área de E&P da PETROBRAS, tanto de sondas quanto de unidades de produção. O foco da presente pesquisa está nas UEPs, unidades flutuantes onde ocorre o processamento primário do petróleo produzido. Assim, apenas equipamentos e sistemas referentes à elevação e escoamento dos fluídos e ao processamento primário serão considerados neste trabalho.

3
Revisão bibliográfica

Este capítulo apresenta uma revisão bibliográfica dividida em duas seções: a primeira dedicada a trabalhos sobre mineração de textos para a estruturação de informação e a segunda dedicada a trabalhos sobre ontologias relacionadas a processos industriais.

3.1
Mineração de dados textuais para estruturação de informação

Em (1) são apresentados os principais desafios e tendências no campo de *Text Analytics* (TA), termo usado pelos autores para unir os conceitos de *Text Mining* e NLP. Eles apresentam os resultados recentes mais importantes nas pesquisas acadêmicas em TA sob três principais aspectos: 1 - o contexto geral, provendo informações como, por exemplo, o domínio da aplicação; 2 - métodos e técnicas, onde são descritos os principais métodos e técnicas empregadas, bem como suas implementações (*toolkits* e bibliotecas usados etc); 3 - avaliação, onde são descritos os experimentos usados para a avaliação da performance das técnicas. Uma das aplicações que merecem destaque, segundo os autores, é a descrita em (13). O trabalho propõe uma metodologia para a interpretação de *tweets* relacionados a eventos de trânsito, testando-a em casos de canais relacionados a trânsito na cidade do Rio de Janeiro. Para tanto, propõe e usa uma ontologia de eventos de trânsito denominada TEDO (*Traffic Event Domain Ontology*), que modela situações de trânsito como eventos, compostos por atores, locais e horários. A TEDO se baseia nas noções de evento (14) e relações entre eventos (15), (16). A figura 3.1 mostra o esquema resumido desta ontologia.

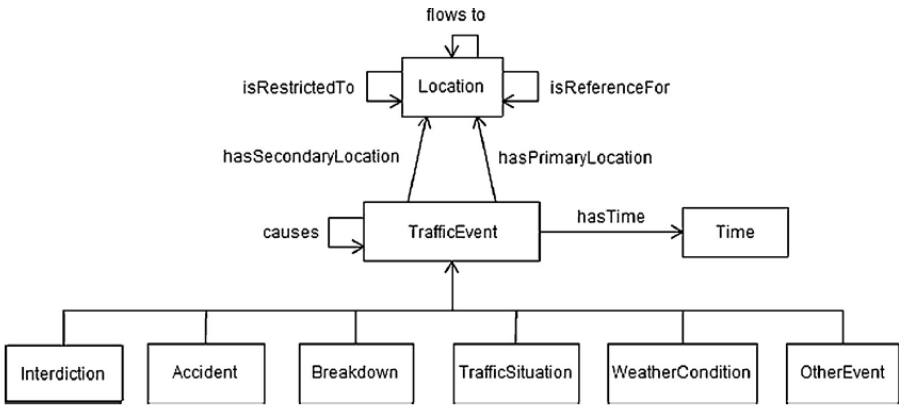


Figura 3.1: Classes da TEDO e *object properties* (*datatype properties* omitidas para legibilidade). (13)

A interpretação dos *tweets* envolve algumas etapas em série: (1) Tokenização dos *tweets* e etiquetagem morfossintática (*Part-of-speech tagging* – POS *tagging*); (2) Extração de entidades (*Named Entity Recognition* – NER); (3) Geolocalização; (4) Extração de relações entre entidades (*Relation Extraction* – RE); (5) Geração de triplas RDF (*Resource Description Framework*).

A etapa de tokenização e POS *tagging* gera atributos para o aprendizado de máquina que expressam as funções sintáticas dos *tokens* nas sentenças analisadas, usando para isso um algoritmo proprietário (formato *Web service*) chamado F-EXT (17). As informações dos *tokens* geradas por ele e usadas na construção de atributos neste trabalho são:

- Token (W_i): o conteúdo do *token* X_i;
- Simple Token (SMW_i): o conteúdo simplificado do *token* X_i (em caixa baixa, sem quaisquer caracteres especiais ou pontuações);
- Simplified Token (SW_i): o conteúdo simplificado do *token* X_i (em caixa baixa, sem quaisquer caracteres especiais, também removendo letras, pontuação e números de tamanho 1);
- Part-of-speech (POS_i): a etiqueta morfossintática do *token* X_i;
- Stemmed Word (STW_i): a raiz da palavra presente no *token* X_i. Exemplo: se X_i é *"blocked"*, STW_i é *"block"*.

Outros atributos foram construídos (*feature engineering*) e, somados ao POS *tagging*, permitem a aplicação de algoritmos de aprendizado de máquina. A primeira etapa de aprendizado é a do NER, onde as entidades descritas na TEDO são usadas como classes em uma tarefa de classificação. Os atributos construídos para esta tarefa no referido trabalho são:

- CurrT(X): o *token* da posição atual, X;
- PrevT(X, N): o *token* N posições antes do *token* X;
- NextT(X, N): o *token* N posições depois do *token* X;
- CurrWSC(X): indica se X começa em letra maiúscula e se há apenas uma letra maiúscula em todo o *token*;
- LocType(X): indica se a representação de X em letras minúsculas pertence a um determinado conjunto de palavras que designam localidades. Exemplos: "avenida", "av.", "rua", "estrada".

O classificador usado nesta tarefa é uma implementação SMO (*Sequential Minimal Optimization*) (18) da família de métodos SVM (*Support Vector Machine*) disponíveis no pacote Weka 3.6.5 (19). Uma das conclusões do trabalho foi a de que o método SVM apresenta os melhores resultados para NER, frente a outros algoritmos testados.

Uma subetapa de geolocalização determina as coordenadas das entidades nomeadas identificadas, usando para isso o algoritmo *SmartGeocode* (20). Em seguida, as relações entre entidades são extraídas na etapa de RE, que cria uma árvore de dependências G_T a partir de um *tweet* T com entidades nomeadas. Para a extração de G_T, computam-se atributos cada par de elementos textuais K e L, que são usados por um algoritmo para o aprendizado acerca da natureza das (possíveis) relações entre K e L. A implementação do referido trabalho usa primeiramente um perceptron estruturado de margem larga (21) que estabelece um peso para cada aresta de um grafo dirigido completo (todas as relações entre todos os nós e em todas as direções). Em seguida, um algoritmo de *Maximum Spanning Tree* define, no grafo completo, a árvore final, isto é, as relações mais importantes.

Os atributos de nó usados no trabalho são:

- Word(W): indica as palavras do nó K;
- Simplified Word(SW): as palavras simplificadas do nó K;
- Ruler Entity(RE): a entidade nomeada em K, em se tratando de entidade relevante. Se não relevante, usa-se a função morfossintática, POS;
- Named Entity(NE): a entidade nomeada em K, ignorada se não for uma entidade relevante;
- Punctuation(PUNCT): indica se há uma pontuação no texto em K.

Usando-se os atributos de nó, constrói-se os atributos das relações, que são:

- PossRel: indica se os nós K e L podem ter uma relação;
- ConcT(Y): as palavras concatenadas dos nós K e L;
- BetT(Y): a concatenação de todos os *tokens* entre K e L;
- Near(Y, N): a concatenação de todos os N *tokens* antes de K e N *tokens* depois de L;
- AbsLocPair: indica se K e L têm uma relação, onde a entidade em K é <restriction> e em L é <location-name>. Se não há relação, nenhum valor é usado;
- MetaWithLoc: indica se K e L têm uma relação, onde a entidade em K é <reference> ou <direction> e em L é <location-name>. Se não há relação, nenhum valor é usado.

Por fim, a informação de entidades e relações é traduzida como triplas RDF e, dessa forma, o *tweet* em texto livre ganha uma estrutura lógica com base em uma ontologia.

A título de exemplo, apresenta-se um *tweet* em sua versão original e os resultados, respectivamente nas figuras 3.2 e 3.3, do fluxo discutido acima em formato de grafo e em RDF:

Tweet: "Acidente entre 2 carros na Av das Américas na pista sentido Grota Funda próximo ao número 19880"

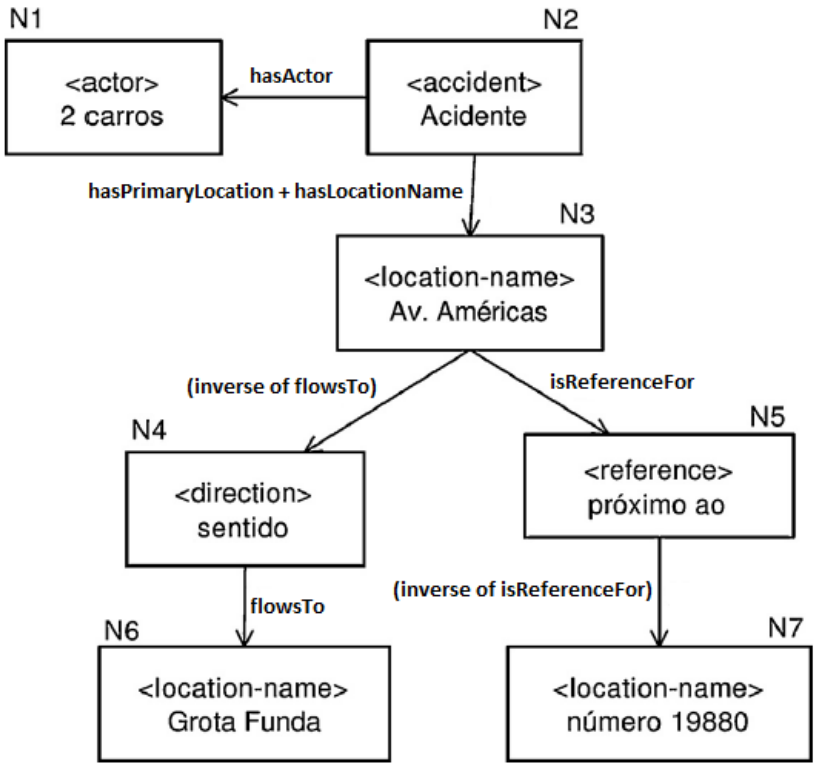


Figura 3.2: Resultado, em grafo, do fluxo proposto em (13) [Conforme figura do mesmo artigo, com adição das *tags* das relações, uma vez que no trabalho havia números que faziam referência a uma tabela não apresentada aqui.]

```
1.  <tedo:Accident
          rdf:about=‘‘http://www.ld.inf.puc-rio.br/tedo/trafficevent/100’’>
2.       <tedo:hasActor>2 carros</tedo:hasActor >
3.       <tedo:hasPrimaryLocation
           id="http://www.ld.inf.puc-rio.br/tedo/location/101">
4.       <tedo:hasSecondaryLocation
           id="http://www.ld.inf.puc-rio.br/tedo/location/102">
5.       <tedo:hasSecondaryLocation
           id="http://www.ld.inf.puc-rio.br/tedo/location/103">
6.       <tedo:hasTime id="http://www.ld.inf.puc-rio.br/time/104">
7.  </tedo:Accident>
8.  <!-- Primary Location -->
9.  <tedo:Location
          rdf:about="http://www.ld.inf.puc-rio.br/location/101">
10.       <tedo:hasLocationName>Av. Das Amricas</tedo:hasLocationName>
11.       <tedo:hasCoordinates>-22.998864, -43.365984</tedo:hasCoordinates>
12.       <tedo:flowsTo
                rdf:resource="http://www.ld.inf.puc-rio.br/location/102">
13.       <tedo:isReferenceFor
                rdf:resource="http://www.ld.inf.puc-rio.br/location/103">
14.  </tedo:Location>
15.  <!-- Secondary Location -->
16.  <tedo:Location
          rdf:about="http://www.ld.inf.puc-rio.br/location/102">
17.       <tedo:hasLocationName>Grota Funda</tedo:hasLocationName>
18.       <tedo:hasCoordinates>-23.015379, -43.521634</tedo:hasCoordinates>
19.  </tedo:Location> \\
20.  <!-- Secondary Location -->\\
21.  <tedo:Location
          rdf:about="http://www.ld.inf.puc-rio.br/location/103">
22.       <tedo:hasLocationName>Nmero 19880</tedo:hasLocationName>
23.       <tedo:hasCoordinates> -23.016279, -43.514426</tedo:hasCoordinates>
24.  </tedo:Location>
25.  <!-- Time -->
26.  <tedo:Time
          rdf:about="http://www.ld.inf.puc-rio.br/time/104">
27.       <tedo:hasPublicationTime>05/03/2012 07:07:01</tedo:hasPublicationTime>
28.  </tedo:Time>
```

Figura 3.3: Resultado, em RDF, do fluxo proposto em (13).

Um análise crítica importante deve ser feita. A referida implementação de RE usa a existência ou inexistência de relação entre nós como base de uma classificação binária. Após isso, tendo como referência as classes dos nós partícipes, consulta-se uma tabela de domínios e *ranges* para cada relação a fim de se obter o nome da relação correta. Desse modo, em caso de múltiplas classes de relações possíveis para uma mesma relação avaliada, ter-se-ia um problema a mais a ser resolvido: qual relação escolher? Outrossim, a implementação proposta estabelece uma árvore de dependências que, por definição, não permite ciclos ou mesmo a existência de múltiplas árvores em um mesmo documento. Entretanto, relações entre entidades nomeadas em documentos como *tweets* podem, eventualmente, apresentar ciclos e múltiplos conjuntos não conexos de relações em uma mesma porção de texto.

O trabalho discutido acima é a evolução de outro (22), cujo objetivo, muito semelhante, era o de reconhecer entidades para o monitoramento de

objetos móveis (carros, caminhões etc). A evolução se deu exatamente pela introdução da ontologia como estrutura lógica para extração e armazenamento da informação. Conclui-se, pela análise destes trabalhos, que o uso das ontologias fornece, além de um padrão coerente para a anotação dos textos (entidades e relações com significados definidos formalmente), a possibilidade de inferências, por mecanismos de *reasoning*, de outras informações mais detalhadas além daquilo que é expressado no texto antes do processamento. Assim, a união entre técnicas de processamento de linguagem natural e o uso ontologias potencializa ainda mais esse tipo de tarefa de TA.

Os relatórios SITOP podem ser vistos como conjuntos de *"tweets"* acerca do funcionamento, estado ou eventos relativos a equipamentos e sistemas da plataforma de produção de petróleo e gás. A presente dissertação propõe, portanto, o uso do fluxo de tarefas dos artigos discutidos acima para a estruturação da informação presente nos relatórios SITOP. Entretanto, uma ontologia relativa aos processos das plataformas torna-se necessária e deve modelar, ao máximo possível, as informações presentes nos diversos textos. Deste modo, o restante desta revisão bibliográfica discutirá algumas ontologias relativas a processos industriais encontradas na literatura.

3.2 Ontologias para processos industriais

Diversos trabalhos envolvendo ontologias para processos industriais foram publicados. Em (23) se apresenta o desenvolvimento de uma ontologia OWL (*Web Ontology Language*) superior baseada na ISO 15926 (*"Industrial automation systems and integration—Integration of life-cycle data for process plants including oil and gas production facilities"*). Ontologias superiores definem classes de alto nível tais como objetos, relações topológicas e mereológicas, de onde classes e relações mais específicas podem ser estabelecidas. Esta ISO define um padrão de integração, compartilhamento, troca e entrega de dados entre sistemas computacionais (24), (25), inicialmente com foco em um modelo de dados para informações sobre o ciclo de vida de uma instalação que se adequasse aos requisitos das indústrias de processo. Seu título mostrou-se inadequado por não refletir a abrangência que a especificação atingiu durante sua edição. O desenvolvimento de um modelo de dados genérico, em conjunto com a *Reference Data Library* para plantas de processo, acabou mostrando que, devido ao domínio da especificação ser muito amplo, o padrão poderia ser utilizado para modelar qualquer tipo de informação. Este padrão está descrito em sete partes e, em particular, a parte 2 (padronizada como ISO 15926-2:2003) especifica uma ontologia para

integração acesso e intercâmbio de dados a longo prazo (26). Essa ontologia superior contém duzentos conceitos, incluindo um meta-modelo para sua extensão através do que se chama *Reference Data Library* (aproximadamente vinte mil conceitos do domínio da engenharia).

A ISO 15926-2:2003 se baseia em uma visão metafísica explícita do mundo real conhecida como tetradimensionalidade (27). Nesta visão, os objetos se estendem no tempo tanto quanto no espaço, em vez de se limitarem completamente em cada ponto do tempo que passa. Além disso, este padrão também apresenta uma base extensional para a identidade de indivíduos. Assim, se dois objetos aparentemente distintos têm as mesmas partes (no espaço e no tempo), então são um mesmo objeto. Se uma barra de aço se transforma em um tubo, por exemplo, há uma parte temporal (estado) da barra de aço que coincide com o tubo e, por serem coincidentes, eles são o mesmo objeto. Em outras palavras, o tubo é um estado da barra de aço.

O artigo apresenta um exemplo industrial prático dessa característica descritiva da realidade. Suponha-se que uma bomba foi projetada e identificada como P-101. Algum tempo depois, o fabricante entrega uma bomba, com o número serial 1234, que atende perfeitamente às especificações de projeto da P-101. A bomba 1234 é instalada e, depois de um período de operação, falha. Então, a equipe de manutenção decide trocá-la pela bomba 9876. Esta situação pode ser facilmente modelada pelo uso do conceito de partes temporais (estados), conforme o esquema da figura 3.4.

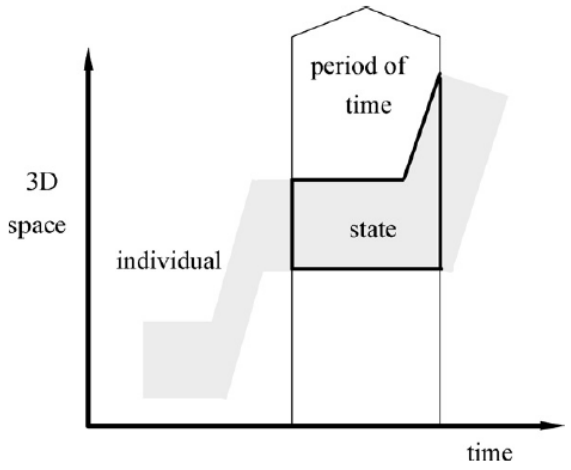


Figura 3.4: Um objeto (possível indivíduo) e sua parte temporal (estado). (23)

A ISO 15926-2:2003 inclui a classe *functional_physical_object* para definir coisas como a bomba P-101, que têm continuidade funcional, em vez de material, como suas bases de identidade. Em outras palavras, membros dessa

classe são partes substituíveis de um mesmo artefato. A P-101 permanece a mesma bomba com a mesma função na planta, mesmo que o equipamento concreto seja substituído, isto é, 1234 por 9876, como elucidado pela figura 3.5. Sob o ponto de vista da tetradimensionalidade, a P-101 consiste em equipamentos como partes temporais enquanto instalados como P-101. Assim, se S1 é o estado da bomba 1234 enquanto instalada como P-101 e S2 é o estado da bomba 9876 enquanto P-101, a P-101 tem as partes (estados) S1 e S2.

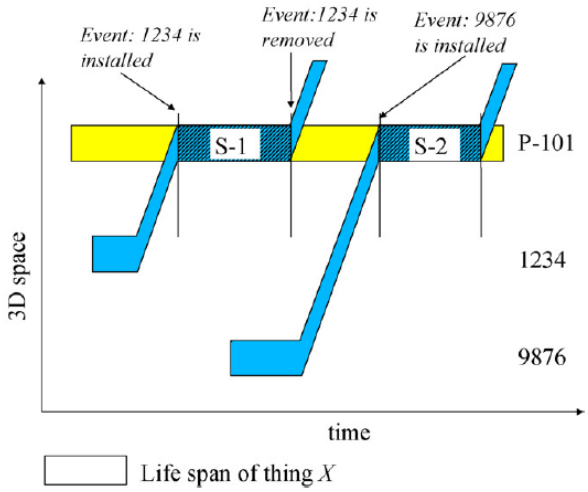


Figura 3.5: A bomba P-101 e suas partes 1234 e 9876. (23)

O conceito raiz na ontologia é *Thing*, o qual tem *abstract_object* e *possible_individual* como subtipos. Membros de *possible_individual* são entidades que existem no espaço e no tempo, incluindo objetos físicos como um compressor ou quaisquer indivíduos imaginários, como "Sherlock Holmes". Um membro de *possible_individual* tem um ciclo de vida limitado por eventos de início e fim. Já indivíduos que pertencem à classe *abstract_object* são do tipo que existem como conceitos, como entidades matemáticas, por exemplo, mas não concretamente no espaço e no tempo.

Assim as definições continuam em suas relações do tipo *is-a*, isto é, classes que contêm subclasses. Como exemplo, a figura 3.6 mostra essas definições para a classe *possible_individual*.

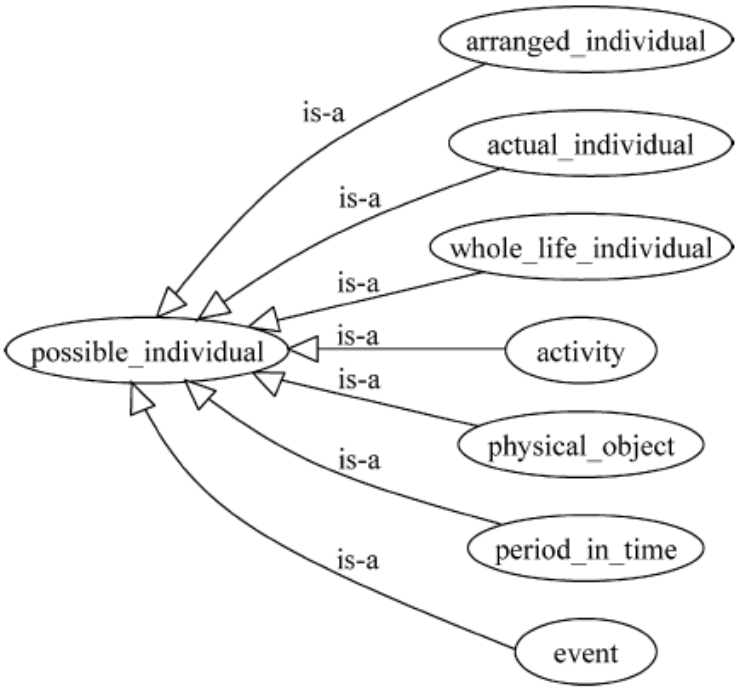


Figura 3.6: Subclasses de *possible_individual*. (23)

A classe *physical_object* compreende os membros de *possible_individual* que são distribuições de matéria, energia ou ambos. São exemplos: uma bomba, uma mesa, um pedaço de metal, etc. A figura 3.7 mostra as subclasses desta classe. Destaque-se o aparecimento da já discutida classe *functional_physical_object*.

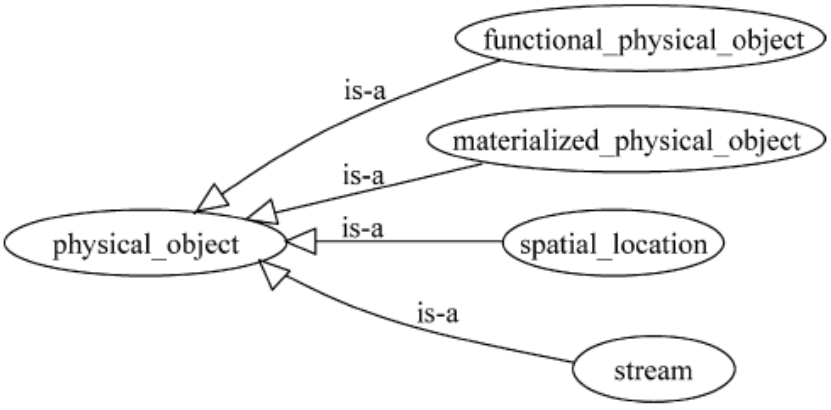


Figura 3.7: Subclasses de *physical_object*. (23)

O artigo descreve com profundidade cada um dos conceitos e relações envolvidos, mas, para fins de aplicação em descrições de eventos em processos,

pode-se destacar o conceito *activity*. Uma *activity* é um *possible_individual* que provoca mudanças que causam um event. Membros de *activity* têm membros de *event* como limites temporais. A classe *event* designa um *possible_individual* que tem extensão nula no tempo, o que significa que ele ocorre em um instante de tempo. São exemplo de membros de *activity* processos físico-químicos, operações na planta e situações anormais. Um *period_of_time* é um *possible_individual* que representa a totalidade da extensão do espaço para uma parte do tempo. O conceito de causalidade é descrito pela relação *cause_of_event*. Em resumo, a classe *activity* consiste em partes temporais dos membros *possible_individual* que participam da atividade. Assim, por exemplo, uma atividade de mistura compartilha das partes temporais do tanque e do agitador. A relação *participation* é usada para expressar a participação de um *possible_individual* em uma atividade.

A figura 3.8 resume toda a descrição feita acima.

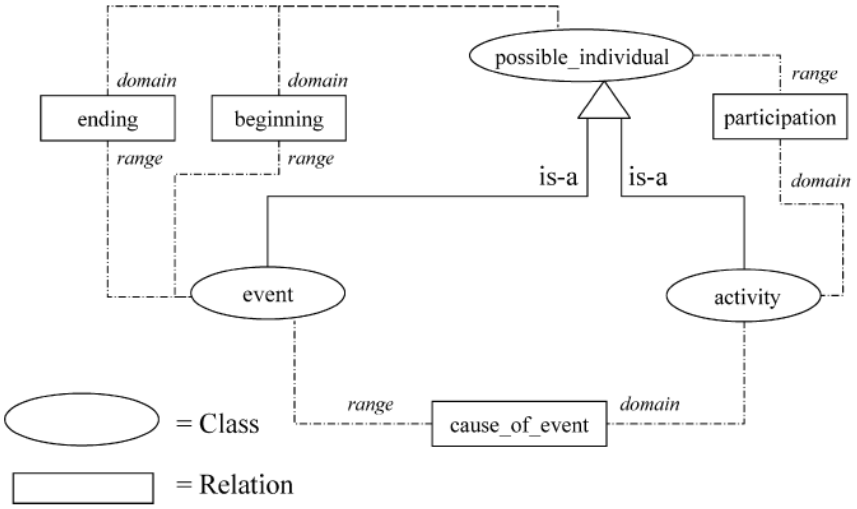


Figura 3.8: A classe *activity* e suas relações. (23)

O artigo ora discutido trata, portanto, de uma ontologia genérica, para extensão e concretização segundo as necessidades de cada uso. Os conceitos de partes temporais e atividades, incluindo a descrição de causalidades, são interessantes para o domínio de processos industriais, onde se busca descrever cadeias de eventos, relações de causa-efeito e ações corretivas.

Todavia, essa ISO apresenta algumas limitações. Algumas críticas são feitas quanto à sua demasiada complexidade e ao não provimento de elementos de construção e modelagem de alto nível, resultando em modelos de dados que não podem ser bem compreendidos e usados com facilidade (28). Em (29) se propõe o uso da ISO-15926 como uma base de informação para

análises de risco em processos industriais conhecidas como HAZOP (*Hazard and operability study*) auxiliadas por computador. Antes deste trabalho, alguns outros pesquisadores (30) também propuseram o uso de ontologias para expressar informações sobre riscos em processos. Contudo, seus estudos não se mostraram muito exitosos em gerar um grande interesse por outros pesquisadores ou industrias em geral. Mais uma vez, entre as principais razões está a modelagem complicada e de difícil uso.

Outros pesquisadores contribuíram com uma outra abordagem, chamada OntoCAPE (31), (32), uma ontologia formal e pesada para o domínio específico de processos de engenharia química. O termo CAPE corresponde a *"Computer-Aided Process Engineering"*, isto é, uma área de pesquisa, desenvolvimento e promoção de aplicações computacionais para o auxílio de atividades de engenharia química como, por exemplo, o projeto, a construção e as operações de plantas de processo. A OntoCAPE é, portanto, uma ontologia para CAPE. Seu desenvolvimento se deu inicialmente no projeto COGents (33), que explorava uma arquitetura baseada em agentes para a simulação numérica de processos químicos, sendo continuado então no projeto IMPROVE (34), cujo foco estava em novos conceitos e soluções de engenharia de software para apoiar processos de projeto de engenharia colaborativa.

A ontologia OntoCAPE é o resultado de um longo processo de melhoria e adequação e, em (31) são apresentadas e discutidas suas etapas de evolução que, segundo os autores, configuram uma robusta metodologia de desenvolvimento de ontologias em geral. Os estágios (ver figura 3.9) incluem: análise de requisitos; coleta de recursos reutilizáveis; especificação geral e informal completa; especificação formal e manutenção. Apesar de serem etapas em uma lógica sequencial, iterações entre diferentes etapas são admitidas e executadas conforme a necessidade.

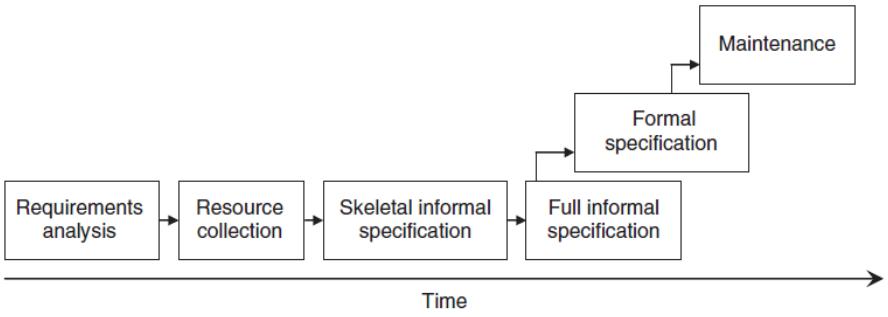


Figura 3.9: Estágios do desenvolvimento da OntoCAPE. (31)

A análise de requisitos trata do levantamento das necessidades da ontologia em termos de propósito, escopo, casos de uso e formalismos. A coleta

de recursos reutilizáveis é a etapa em que ontologias e modelos conceituais de dados já existentes são considerados e revisados para se identificar partes relevantes e passíveis de reutilização. A especificação geral e informal diz respeito à definição de uma representação informal do "esqueleto" da ontologia, em texto, por exemplo. A especificação formal e completa é, obviamente, a formalização da especificação informal em alguma linguagem ontológica, quando a ontologia pode, então, ser interpretada por sistemas de informação. Por fim, a etapa de manutenção visa a atualização, conforme a necessidade, das especificações informal e formal. Existem na literatura diversos princípios gerais que norteiam o estabelecimento de uma boa ontologia (35), (36), (37), (38), (39), (40) e (41). A OntoCAPE foi projetada (e é mantida e atualizada) seguindo alguns destes: coerência, concisão, inteligibilidade, adaptabilidade, comprometimento ontológico mínimo e eficiência.

A OntoCAPE está organizada em três tipos principais de elementos estruturais: camadas, módulos e modelos parciais. As camadas se subdividem em cinco níveis de abstração, separando, portanto, o conhecimento geral de conhecimentos particulares de domínios e aplicações específicos. Seguindo o princípio de "comprometimento ontológico mínimo" (35), conceitos e refinamentos não diretamente relacionados ao propósito de uma camada são destinados a outras camadas de níveis mais baixos. A camada mais alta e abstrata, *meta layer*, introduz os termos raízes e padrões fundamentais que serão usados nos níveis mais baixos. A camada *upper layer* descreve os paradigmas gerais de projeto para os quais a ontologia de domínio é organizada: a OntoCAPE se baseia na teoria geral de sistemas (42), (43), um princípio comum de organização de ontologias relacionadas à engenharia, como a YMIR (44) e outros exemplos semelhantes (45), (46). A próxima camada, *conceptual layer*, inicia a conceptualização formal do domínio CAPE. As camadas seguintes elaboram e particularizam o modelo conceitual pela adição de conceitos de relevância prática para certas aplicações. Assim, a *application-oriented layer* descreve diversas áreas gerais de aplicação, que são então concretizadas na *application-specific layer*. Os módulos são delimitadores que congregam um determinado número de classes que cobrem um tópico comum bem como as relações e restrições pertinentes. Por fim, modelos parciais são agrupamentos de módulos que se relacionam de modo mais próximo. A figura 3.10 apresenta a estrutura geral da OntoCAPE.

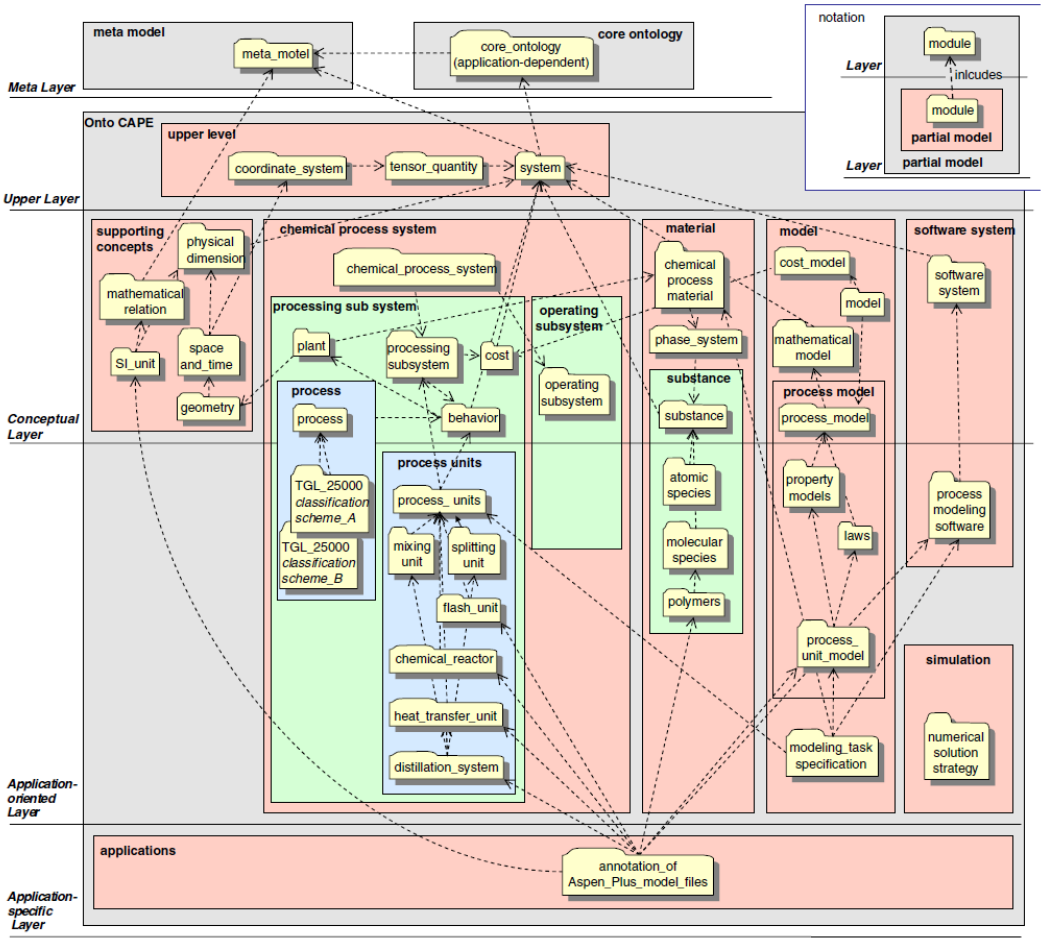


Figura 3.10: Estrutura da OntoCAPE. (32)

Alguns trabalhos mostram aplicações da OntoCAPE. Como anteriormente citado, no projeto COGents (47) a OntoCAPE é parte de um *framework* multiagente (ver figura 3.11), apoiando a seleção de componentes de modelagem de processos adequados a partir de bibliotecas de modelo distribuídas. Nessa aplicação, a ontologia serve como linguagem comum entre os agentes de *software* que interagem entre si e entre os agentes e os usuários humanos.

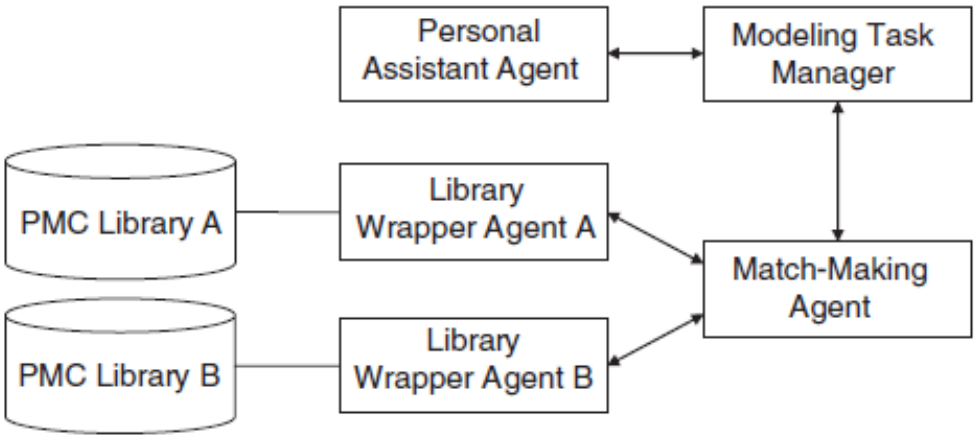


Figura 3.11: Representação simplificada da arquitetura do COGents, onde a OntoCAPE serve como linguagem comum entre os agentes de *software*. (32)

Em outro trabalho (48), esta ontologia dá suporte à construção, auxiliada por computadores, de modelos de processos, em duas etapas. Primeiramente, um modelador humano formula um modelo conceitual de um processo químico através da seleção, instanciação e conexão de conceitos ontológicos apropriados que reflitam as propriedades estruturais e fenomenológicas do referido processo químico. Em um próximo passo, tendo como base estas especificações, o modelo matemático é automaticamente criado por uma rotina que seleciona e entrega os componentes corretos a partir de uma biblioteca de blocos construtivos de modelos. Os modelos são integrados, com base na especificação ontológica, resultando em um modelo matemático completo. A figura 3.12 mostra um esquema que resume estas etapas.

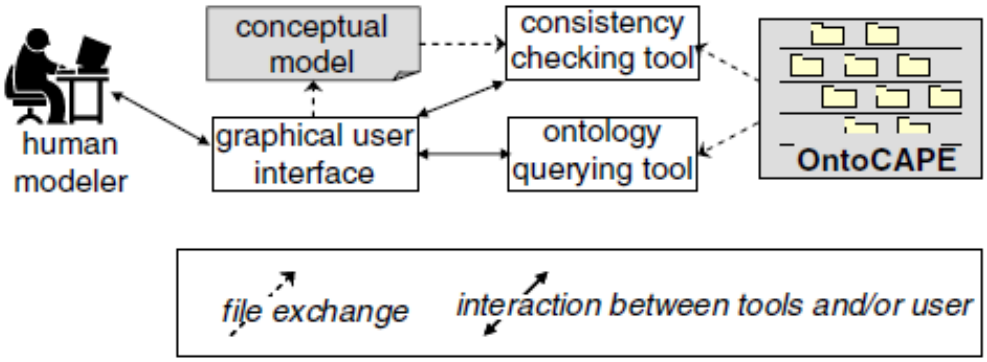


Figura 3.12: Estrutura da ferramenta de modelagem conceitual usando a OntoCAPE. (32)

A OntoCAPE também foi usada para a implementação de uma *Process Data Warehouse* (PDW) (49). Nesse caso, a ontologia é usada para a anotação de documentos eletrônicos e armazenamento de dados. Assim, a OntoCAPE tem se mostrado uma ontologia robusta, completa e com boas possibilidades de adaptação para casos específicos. Seguindo o já citado princípio da adaptabilidade, um trabalho merece destaque (50), visto que apresenta uma nova ontologia, a OntoSAFE, uma adaptação da OntoCAPE e, o mais importante, os autores do artigo não são os autores da OntoCAPE original, o que permite observar a facilidade ou não de seu uso apenas com base nas documentações, descrições formais etc. A OntoSAFE se propõe a ser uma ontologia para supervisão de processos, que é o conjunto de atividades e meios para garantir a operação segura de processos por operadores humanos que gerenciam as situações anormais e para aumentar a confiabilidade e disponibilidade do processo como um todo. Este conjunto de atividades é composto por mecanismos de identificação do estado do processo, de monitoramento e de diagnóstico de falhas. Diferente do projeto ou desenho do processo, a supervisão de processos não lida muito com a síntese de novas informações mas com a análise da informação já existente para realizar inferências sobre a condição da planta. Como a OntoCAPE originalmente não prevê classes e relações específicas desse domínio, a OntoSAFE foi construída herdando os conceitos aplicáveis da ontologia original e adição de outros de acordo com a necessidade. Como exemplo principal, destaca-se a adição do modelo parcial *CPS_condition*, que descreve formalmente conceitos, classes e relações para a representação do estado e condição da planta (ver figura 3.13).

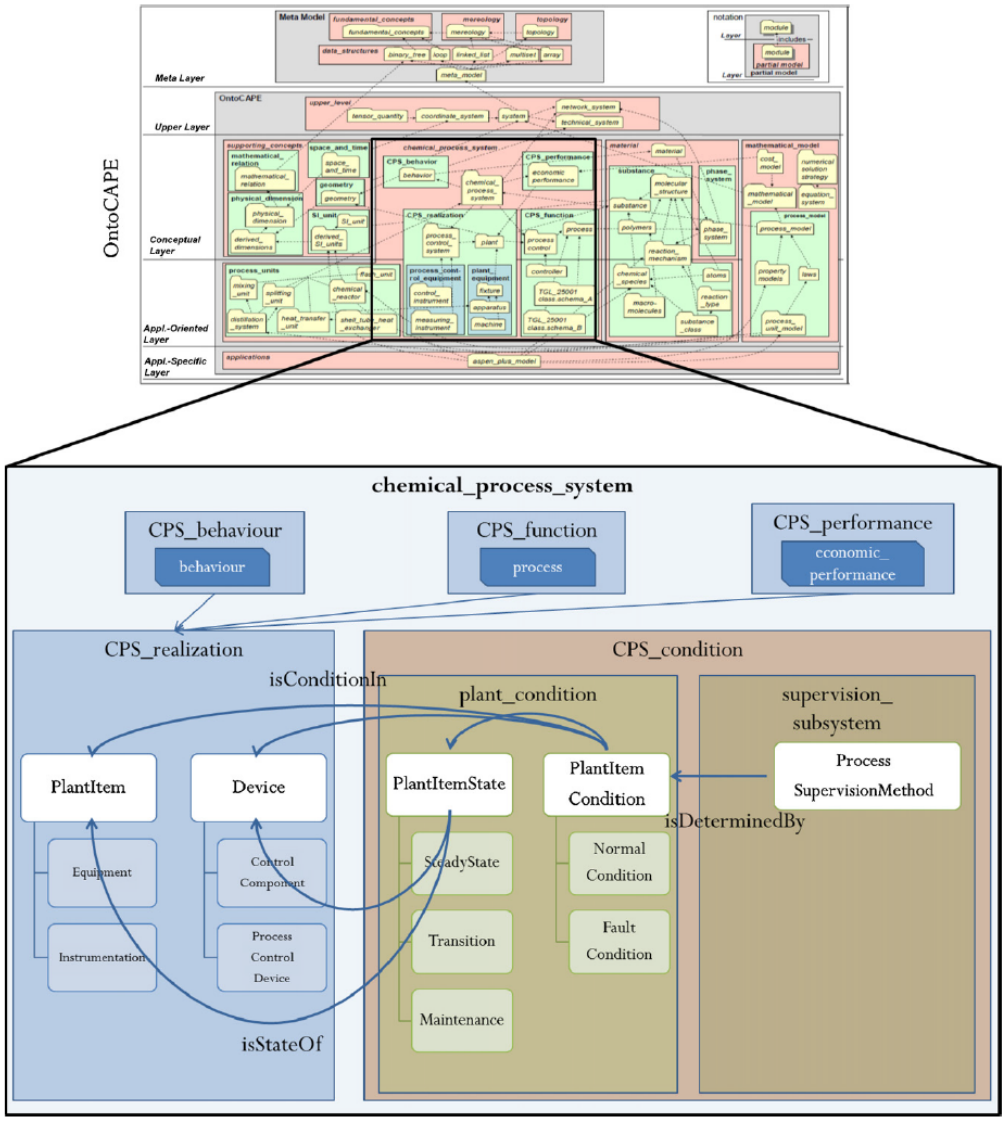


Figura 3.13: Estrutura da OntoSAFE, formada pela adição de um novo modelo parcial, *CPS_condition*, à OntoCAPE. (50)

O exemplo dos autores da OntoSAFE no referido trabalho é justamente o de uma aplicação em uma unidade *offshore* de processamento primário de petróleo. Para fins de ilustração, mostrando a coexistência de classes e relações da OntoCAPE original com as adições da OntoSAFE, a figura 3.14 destaca a representação de um esquema de monitoramento de um vaso separador de alta pressão (*HP-Separator-Section*), onde a condição do processo é determinada por um método matemático chamado PCA (*Principal Component Analysis*) com base em dados históricos deste equipamento. A realização física do equipamento já é descrita com base no modelo parcial *CPS_realization*, da OntoCAPE, bem como grandezas físicas e mesmo o modelo matemático usado

pelo método PCA. Entretanto, o método PCA e a descrição das condições e estados do vaso separador são descritos na modelo parcial adicionado pela OntoSAFE.

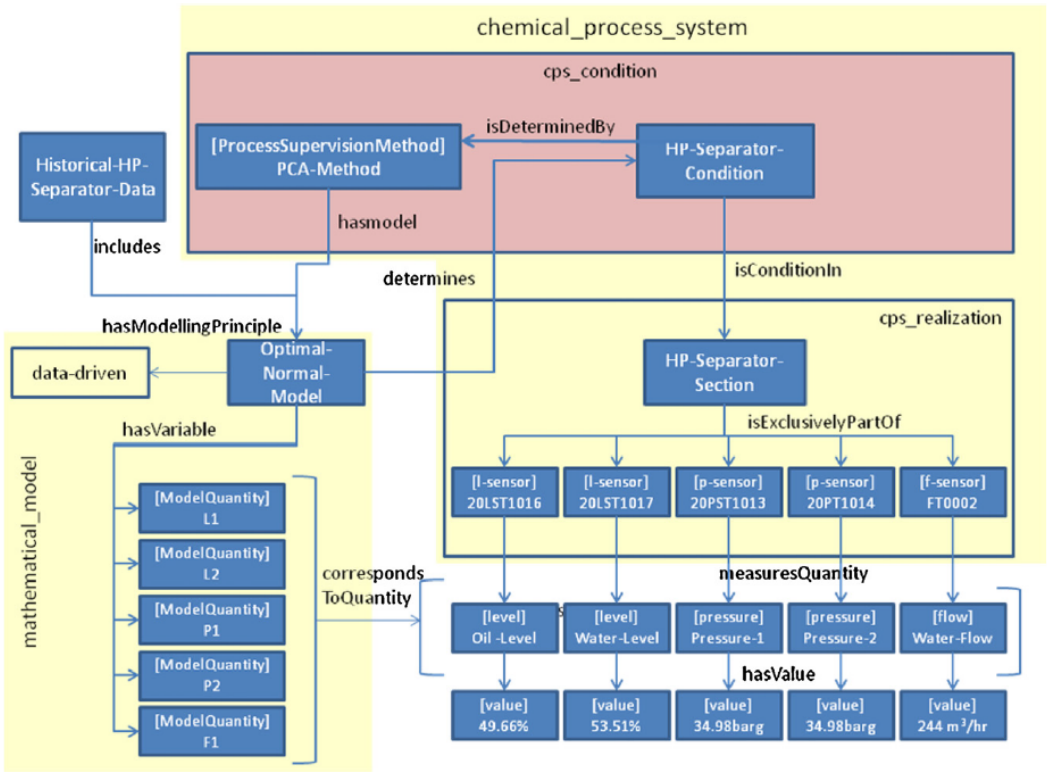


Figura 3.14: Representação do monitoramento baseado em PCA na OntoSAFE. (50)

Os resultados mostram, mais uma vez, que a OntoCAPE se mostra uma boa opção para uso geral em processos químicos. Maiores detalhes e uma descrição completa sobre esta podem ser consultados em (51).

Outros autores (52), (53) propuseram uma ontologia alternativa que, junto com regras de negócio, suporta a definição de uma estrutura lógica da operação de uma planta de processos (também aqui, como estudo de caso, um exemplo de aplicação em processamento primário de petróleo) com o objetivo de monitorar as relações de causa-efeito em ocorrências de *shutdowns* nas plantas. A figura 3.15 resume a ontologia proposta. Em linhas gerais, ela expressa as classes e relações importantes para o objetivo do referido trabalho, não sendo tão pesada e detalhada quanto a OntoCAPE.

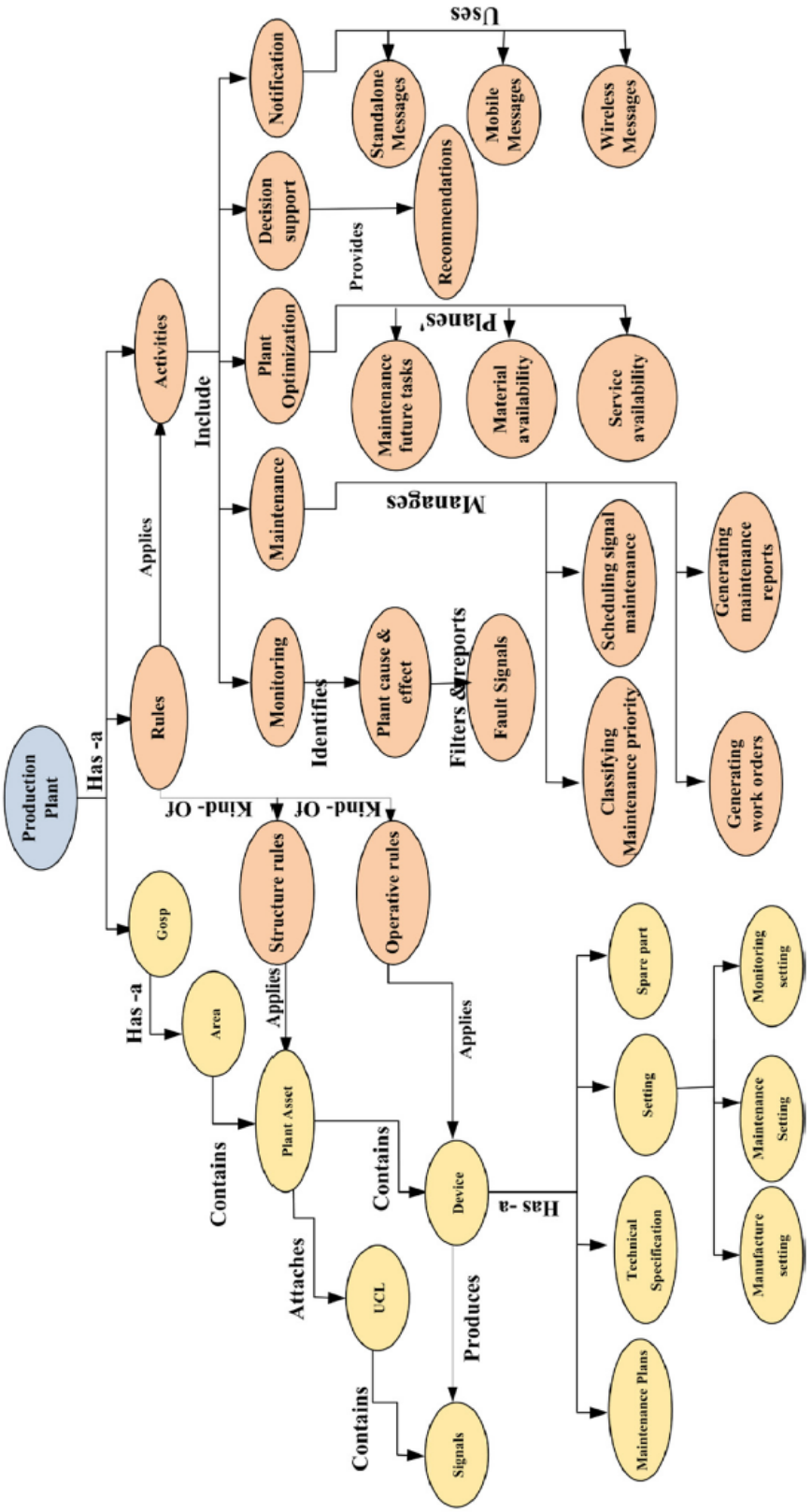


Figura 3.15: Ontologia para aplicação no monitoramento de causa-efeito em plantas de processamento de petróleo. (53)

Em um outro trabalho (54), pesquisadores propuseram e desenvolveram uma abordagem ontológica para a construção de uma base de conhecimento em falhas em equipamentos de processo (PEF – *Process Equipment Failures*). Em resumo, o objetivo foi a formação de uma base de conhecimento que permitisse não apenas o acesso aos dados brutos de falhas, posto que geralmente estão simplesmente armazenados em alguma base de dados, mas também a busca semântica para uma mais profunda investigação dos processos de falhas. O processo de construção dessa base de conhecimento seguiu uma sequência semelhante ao que foi descrito em (31) para o caso da OntoCAPE (ver figura 3.16).

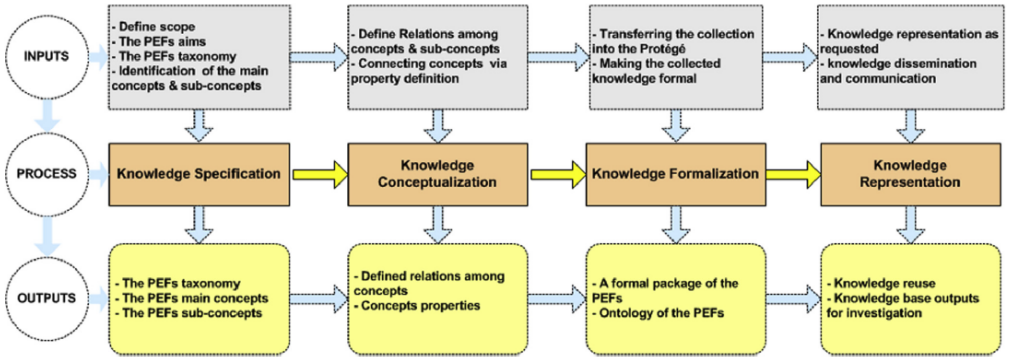


Figura 3.16: O processo de construção da base de conhecimento em PEF. (54)

A ontologia proposta define uma taxonomia para o conceito de "falha em equipamento de processo", desde uma generalização desde uma falha qualquer até falhas de equipamentos específicos. Um "mergulho" nessa taxonomia (ver figura 3.17) elucida o detalhamento de alguns tipos de falhas. As classes relações se baseiam na já discutida ISO 15926.

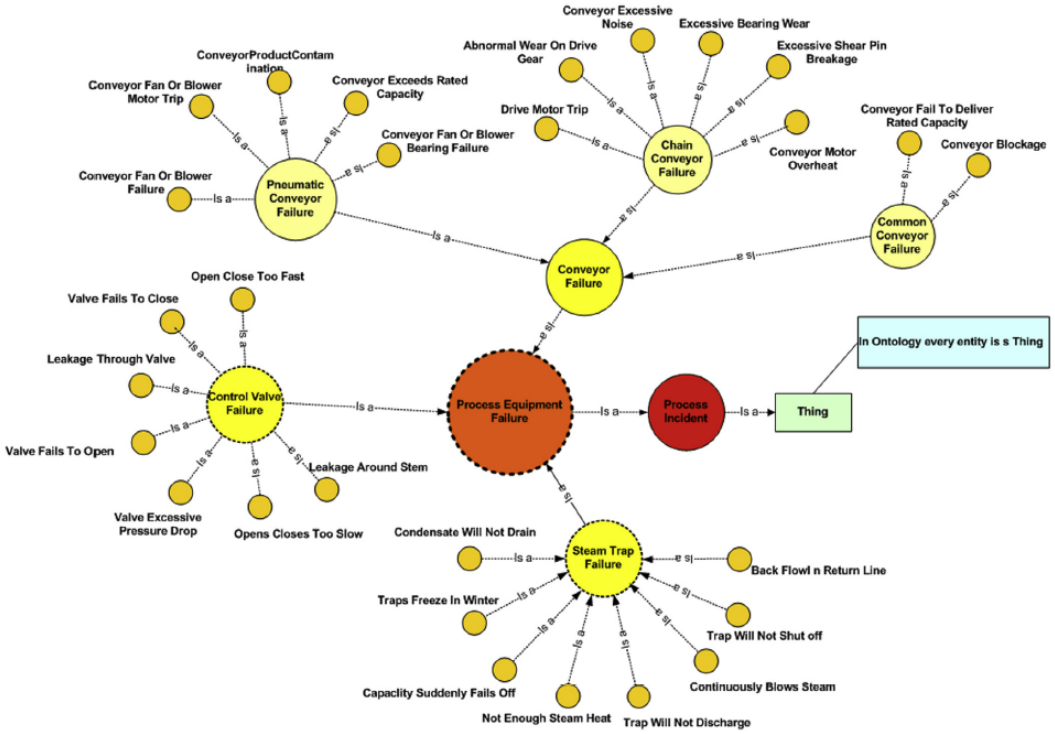


Figura 3.17: Algumas expansões da classe de PEF em falhas especiais, referentes a equipamentos específicos. (54)

Por fim, a figura 3.18 mostra o esquema geral da ontologia, com os principais conceitos e relações entre eles.

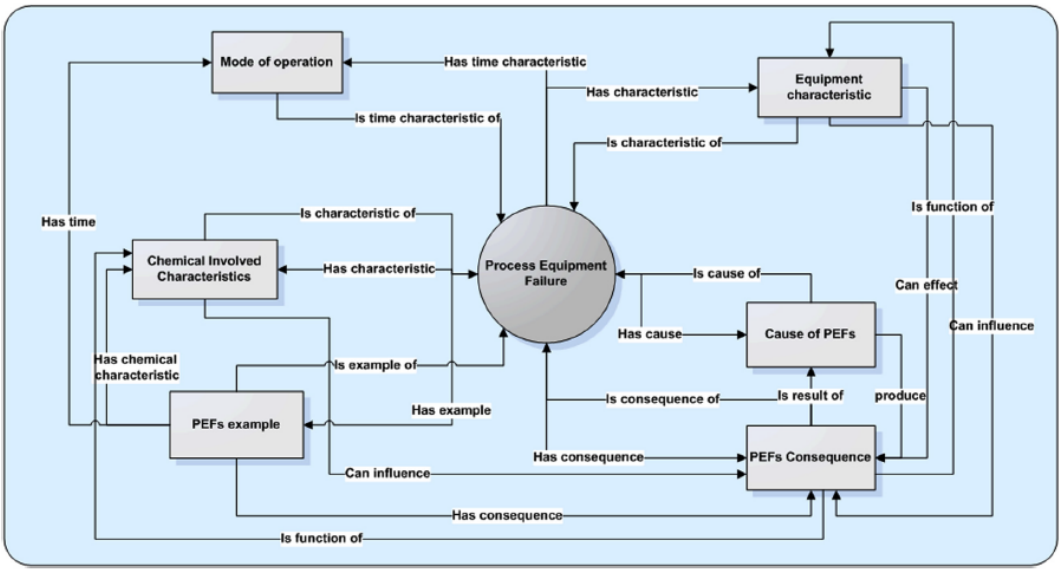


Figura 3.18: Esquema geral dos principais conceitos na base de conhecimento de PEF e algumas relações entre eles. [Figura obtida em (54).]

Há também exemplos de abordagens ontológicas relacionadas à manutenção (55), eventos disruptivos em sistemas de distribuição de óleo (56), entre outros, mostrando que o escopo da aplicação será sempre o determinante da melhor forma de descrever o domínio.

3.3 Considerações

Esta revisão bibliográfica, unindo a estratégia de estruturação de informações textuais com base em ontologias e aplicação de técnicas de processamento de linguagem natural às diversas abordagens ontológicas específicas de processos industriais, oferece à presente pesquisa algumas reflexões ou conclusões importantes:

- A estratégia de coletar informações textuais pelo uso de técnicas de processamento de linguagem natural mostrou bons resultados em aplicações recentes.
- A adição de ontologias como descrição lógica do domínio dos textos a serem processados contribui tanto para a estruturação formal de uma base de dados quanto para a inferência de informações que, sem as ontologias, não seria realizada.
- A modelagem proposta em (13) apresenta o resultado das relações entre entidades como uma árvore de dependências, o que pode trazer alguns

problemas em termos de extração de informações, conforme já discutido. Outrossim, a classificação de relações meramente binária, com posterior consulta a tabelas, pode gerar alguns problemas quando a ontologia usada apresentar múltiplas possibilidades de relações para um mesmo domínio e *range*.

- Ontologias podem ser leves ou pesadas, mais abstratas ou mais concretas. O escopo da aplicação determinará o nível de detalhamento e concretude da construção final.
- A estratégia de ontologias aliadas ao processamento de linguagem natural tende a dar melhores resultados com ontologias mais leves, pois mais classes e relações mais complexas demandam, proporcionalmente, maior quantidade de dados anotados para o aprendizado.
- Existem ontologias para processos industriais e mesmo para processos químicos. Dentre as estudadas, a OntoCAPE se mostra a mais robusta a adaptável, o que se comprova por trabalhos que mostram seu uso em diversas aplicações, inclusive com a expansão de seus conceitos em novos escopos, sendo a OntoSAFE um bom exemplo. Entretanto, essa ontologia é muito pesada e se destina a aplicações mais abrangentes em uma empresa de petróleo. Não seria razoável, quiçá impossível, a utilização do grau de detalhamento da OntoCAPE na aplicação direta sobre os relatórios de processos para fins de estruturação dos dados.
- Torna-se necessária, portanto, a construção de uma ontologia leve, mínima, que seja posteriormente conectada à ontologia geral usada na base de conhecimento do negócio E&P, seja ela a OntoCAPE ou outra qualquer. Esse processo de conexão é um tema para outros trabalhos.
- A aplicação da metodologia proposta em (13) envolve uma série de etapas para a coleta e preparação dos dados (anotação), treinamento e teste de modelos NER e RE e conversão de resultados para RDF. A fim de reduzir o tempo deste processo e torná-lo mais robusto, seria desejável o uso de um sistema que congregasse todos os passos em um único ambiente.

Tendo em vista estas diretrizes, os próximos dois capítulos apresentam, respectivamente, o uso e modelagem do problema em um sistema que abarca todas as etapas necessárias para a preparação de dados textuais e produção de modelos e, em seguida, a proposta de uma ontologia leve e simples para fins de prova de conceito no tocante à possibilidade de estruturar as informações dos relatórios de processo através da abordagem NLP + ontologia. Essa ontologia não deve ser considerada robusta e definitiva, pois o objetivo é, tão somente,

verificar a viabilidade do uso do aprendizado de máquinas para a mineração dos dados textuais dos relatórios.

4
Sistemas ERAS e LER

Este capítulo apresenta um resumo acerca dos sistemas ERAS e LER, que foram as principais ferramentas desenvolvidas, respectivamente, para a anotação dos dados textuais coletados dos relatórios SITOP e construção de modelos para a execução das tarefas NER e RE.

O processo de estruturação de dados textuais com base em uma ontologia, detalhadamente descrito na revisão bibliográfica, pode ser entendido como uma cadeia de atividades que demandam muito esforço e, dada a complexidade envolvida, pode conduzir a muitos erros. Em termos gerais, compreende-se uma grande divisão em duas partes principais: anotação de textos e treinamento de modelos por aprendizado de máquina. Assim, no âmbito deste projeto e em conjunto com a pesquisa de outro pesquisador (57), foram desenvolvidos os sistemas ERAS e LER. O sistema ERAS (*Entities and Relations Annotation System*) atende à primeira grande etapa, para anotação dos textos e organização de dados, tudo com base em uma ontologia de domínio. O sistema LER (*Learning Entities and Relations*) serve à segunda etapa, para uso dos dados anotados no treinamento de modelos de NER e RE, que posteriormente podem ser selecionados e alocados em uma cadeia NER-RE como um serviço *web* para estruturação automática de textos. A descrição profunda e detalhada de cada aspecto desse sistema pode ser vista em (57). O restante do capítulo apresenta apenas uma visão geral, mostrando a contribuição da presente pesquisa principalmente na etapa de aprendizado de máquina.

As figuras 4.1, 4.2 e 4.3 ilustram o processo geral realizado pelo conjunto ERAS-LER, que pode ser logicamente descrito em 7 etapas:

1. a manipulação e persistência de conjuntos de documentos textuais (fontes de dados);

2. o uso de bibliotecas de NLP com tokenizadores (*tokenizers*) e etiquetadores morfológicos (POS *taggers*) para a caracterização dos textos;

3. a anotação dos textos com base em ontologias de domínio em formato OWL;

4. a curadoria dos documentos, por parte do "dono" do projeto, pela comparação semi-automática das anotações;

5. o treinamento, a avaliação e a persistência de modelos de aprendizado de máquina para a tarefa de NER, que visam ao reconhecimento automático das entidades descritas nas ontologias;

6. o treinamento, a avaliação e a persistência de modelos de aprendizado de máquina para a tarefa de RE, que visam à construção automática de grafos de relações entre entidades, também baseada nas relações entre entidades descritas nas ontologias;

7. a geração e a disponibilização automáticas de serviços de leitura e estruturação de fontes textuais com base nos modelos gerados na etapa anterior.

As etapas de 1 a 4 compõem o ERAS e as etapas de 5 a 7 compõem o LER.

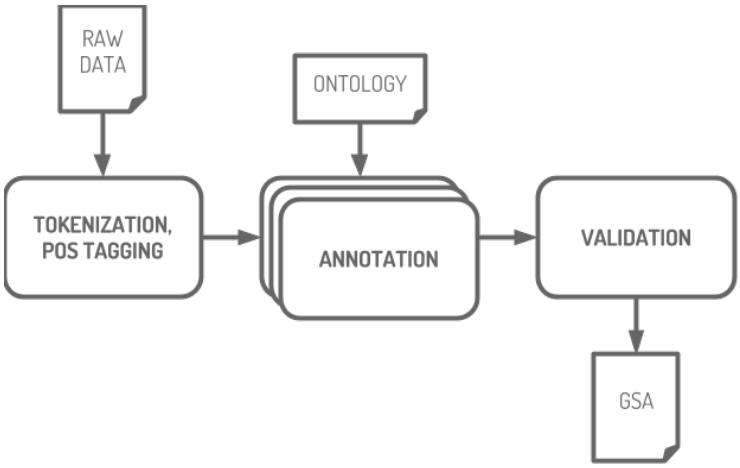


Figura 4.1: Preparação e curadoria de documentos (Etapas 1 a 4)

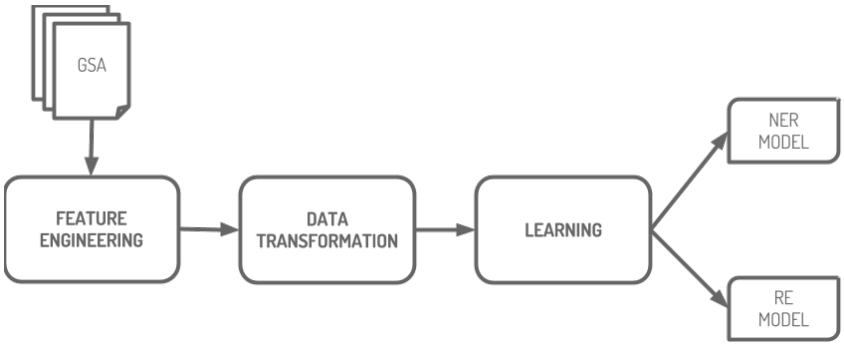


Figura 4.2: Treinamento, teste e persistência de modelos para NER e RE (Etapas 5 a 6)

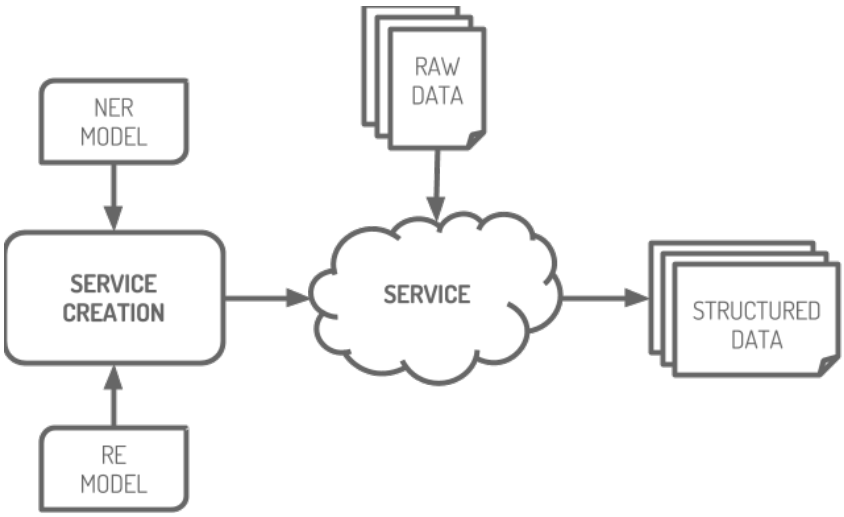


Figura 4.3: Disponibilização dos modelos integrados como um serviço (Etapa 7)

O sistema ERAS permite a importação de ontologias em formato OWL para a definição das classes e relações a serem extraídas dos textos na forma de triplas RDF. Os dados em texto são alimentados na forma de documentos (pequenas porções textuais), via *upload*. A seguir, passa-se à etapa de tokenização e POS *tagging*, para a geração e caracterização morfossintática das menores unidades de texto, conhecidas como *tokens*. O artigo que inspirou a criação do ERAS-LER (13) usa a já citada ferramenta F-EXT (17), que não está mais disponível para uso público e comercial, mas apenas para fins de pesquisa. Como o conjunto ERAS-LER se destina a uso geral, inclusive o uso industrial proposto nesta pesquisa, tornou-se necessária a adoção de uma outra ferramenta e, assim, o sistema Freeling 4.0 (58), explicado em resumo para uma versão anterior em (59), foi o escolhido para a atual configuração.

Comentários acerca das vantagens e desvantagens deste POS *tagger* alternativo podem ser consultados em (57). Aqui, cabe apenas a descrição das informações que o Freeling atribui a cada *token*:

– "FORM": a representação textual original do *token*;
– "LEMMA": a forma canônica do texto do *token*;
– "POS": a *tag* em si;
– "PROB": probabilidade da *tag* em "POS".

Segue-se então à etapa de anotação, onde os tokens já aparecem separados. A figura 4.4 mostra um exemplo de texto anotado no ERAS, com entidades e relações baseadas na ontologia carregada previamente.

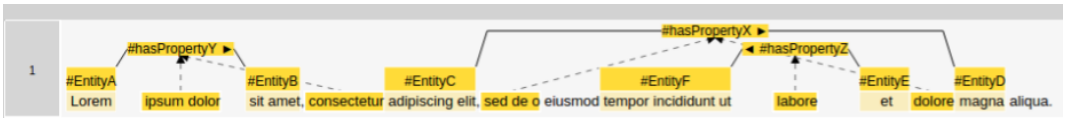


Figura 4.4: Exemplo de texto anotado no ERAS.

O ERAS introduz um conceito interessante para a engenharia de atributos a ser realizada no LER: os *connectors*. Trata-se de uma inovação deste sistema frente a outras propostas pesquisadas em (57). Durante o processo de anotação, identifica-se entidades nomeadas, etiquetadas segundo determinadas classes, e relações entre estas entidades. Os *connectors* são *tokens*, ou conjuntos de *tokens*, que o anotador registra como expressões de determinada relação identificada, funcionando como "dicas"para a engenharia de atributos. A figura 4.5 apresenta uma frase, "Susan é filha de Josh.", onde as entidades *Child* e *Father*, identificadas, são relacionadas pela relação "isChildOf". O *connector* neste caso é, evidentemente, a porção "é filha de". Essa inovação servirá, no contexto do sistema LER, a uma metodologia de criação de atributos automática, além de servir também à engenharia manual de atributos através dos gráficos, tabelas e nuvens de palavras fornecidos pelo ERAS.

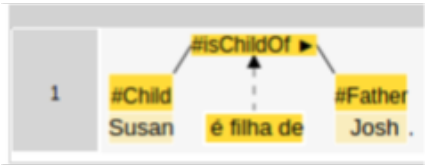


Figura 4.5: Exemplo da aplicação do conceito de *connector*.

O sistema ERAS prepara os pacotes de textos anotados para o sistema LER, cuja primeira etapa diz respeito à criação de tarefas (*tasks*) de NER e RE. Múltiplas tarefas podem ser criadas e executadas em paralelo. Algumas características do LER são iguais para os dois tipos de tarefa. Os pacotes de dados anotados no ERAS ficam disponíveis para importação em cada tarefa. Cada pacote, na configuração da execução, pode ser classificado com *"TRAIN"* (treino) ou *"TEST"* (teste).

Outra característica compartilhada por NER e RE neste sistema é o modo de busca pelo melhor modelo através da avaliação exaustiva da combinação dos hiperparâmetros, também conhecido como *Grid Search* (60). Outrossim, o método usado para a avaliação da generalização dos modelos construídos, comum aos dois tipos de tarefa, é a validação cruzada e, mais precisamente, o método K-*fold* (61). Ambos tipos de tarefa compartilham um mesmo esquema de construção de *features*, cada um com seus atributos pertinentes.

Os atributos, em geral, foram projetados de modo a generalizar a capacidade de construção de esquemas particulares. Para NER, há 6 tipos disponíveis:

- FORM(X, *step*, *regex*): aplica uma expressão regular (*regex*) sobre o conteúdo completo do parâmetro FORM (conforme resultado do POS *Tagger*) do *token* localizado a um número de passos (*step*, com valor nulo, positivo ou negativo) do *token* X. (Note-se que, se *step* é nulo, o atributo representará a aplicação da *regex* sobre a representação FORM do próprio *token* X.);
- LEMMA(X, *step*, *regex*): o mesmo conceito de FORM(X, *step*, *regex*), porém aplicado sobre a representação LEMMA;
- POS(X, *step*): o POS *tag* do *token* localizado a *step* passos do token X;
- PROB(X, *step*): o valor da probabilidade, segundo o POS *Tagger*, do conteúdo de POS do *token* localizado a *step* passos do *token* X;
- RANGE-FORM(X, *step*, *regex*): aplica uma expressão regular sobre o conteúdo da *string* formada pela concatenação – com espaços – dos parâmetros FORM dos *tokens* em um *range* de *step* a partir do *token* X. Se *step* é nulo, a *feature* usa uma *string* vazia e, portanto, gera resultado nulo;
- RANGE-LEMMA(X, *step*, *regex*): o mesmo conceito de RANGE-FORM(X, *step*, *regex*), porém aplicado sobre a representação LEMMA.

A figura 4.6 ilustra esses atributos em um conjunto de oito tokens.

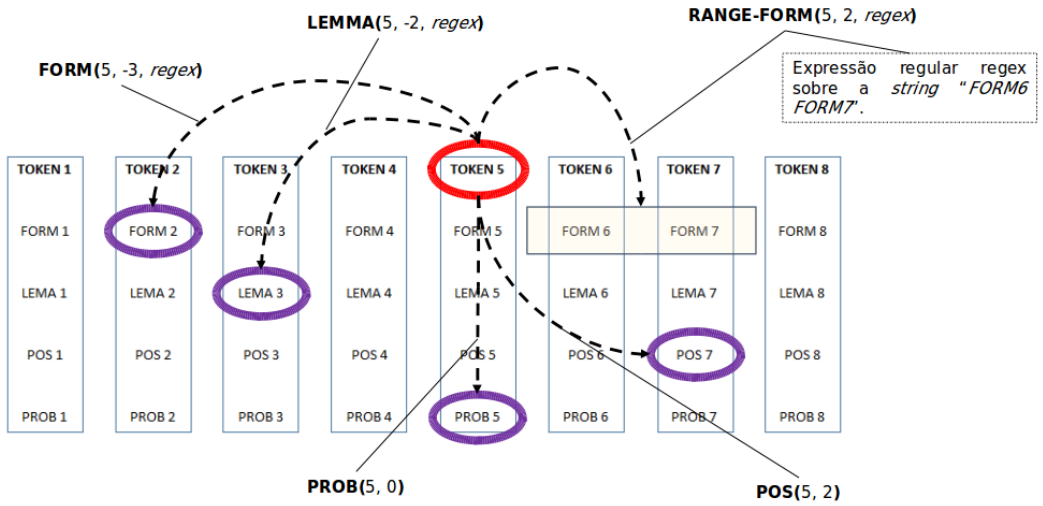


Figura 4.6: Exemplo ilustrativo dos atributos disponíveis para NER.

Há uma diferença entre NER e RE no tocante aos tipos de instância a serem avaliados. Em NER, avalia-se *token* a *token*. Em RE, avalia-se as relações entre "nós", isto é, entre entidades nomeadas segundo a ontologia, presentes no texto. São doze os atributos disponíveis:

- RANGE-FORM(X, *step*, *regex*): o mesmo conceito usado no NER, considerando, porém, que o parâmetro *step* se baseia em um dos nós da relação X, dependendo do sinal: se *step* é negativo, aplica-se à esquerda da relação; se positivo, à direita desta. Valores nulos de *step* não implementam esta *feature*;
- RANGE-LEMMA(X, *step*, *regex*): o mesmo conceito de RANGE-FORM, descrito no item anterior, aplicado sobre LEMMA;
- POSITIONAL-FORM(X, *step*, *regex*): o mesmo conceito usado no NER, considerando, porém, que o parâmetro *step* se baseia em um dos nós da relação X, dependendo do sinal: se *step* é negativo, aplica-se à esquerda da relação; se positivo, à direita desta. Valores nulos de *step* não implementam esta *feature*;
- POSITIONAL-LEMMA(X, *step*, *regex*): o mesmo conceito de POSITIONAL-FORM, descrito no item anterior, aplicado sobre LEMMA;
- POS(X, *step*): o mesmo conceito usado no NER, considerando, porém, que o parâmetro *step* se baseia em um dos nós da relação X, dependendo do sinal: se *step* é negativo, aplica-se à esquerda da relação; se positivo, à direita desta. Valores nulos de *step* não implementam esta *feature*;

- INTERIOR-RANGE-FORM(X, *regex*): aplica uma expressão regular sobre o conteúdo da *string* formada pela concatenação, com espaços, dos parâmetros FORM dos *tokens* entre os nós da relação X;
- INTERIOR-RANGE-LEMMA(X, *regex*): o mesmo conceito de INTERIOR-RANGE-FORM, descrito no item anterior, aplicado sobre LEMMA;
- NODE-TO-NODE-DISTANCE(X): calcula a distância, em *tokens*, entre os nós da relação X. Para fins de redução da magnitude desse atributo a uma ordem de grandeza próxima dos outros, que são binários, o valor é dividido pela máxima quantidade de *tokens* encontrada nos documentos de treino do modelo;
- NODE-TO-NODE-DISTANCE-WITH-SIGNAL(X): mesmo conceito do atributo descrito acima, porém possibilitando valores negativos. A ideia é que este atributo, além de representar a distância entre nós, carregue também a informação do sentido da relação: números negativos traduzem relações da direita para a esquerda e, positivos, da esquerda para a direita;
- CLASS-NODE-FROM(X): classe de NER do nó de saída da relação X;
- CLASS-NODE-TO(X): classe de NER do nó de chegada da relação X;
- POSSIBLE-RELATION(X): indica se a relação é ou não possível, com base nos dados do conjunto de treinamento.

A figura 4.7 ilustra esses atributos em uma relação avaliava em um conjunto de dez *tokens*.

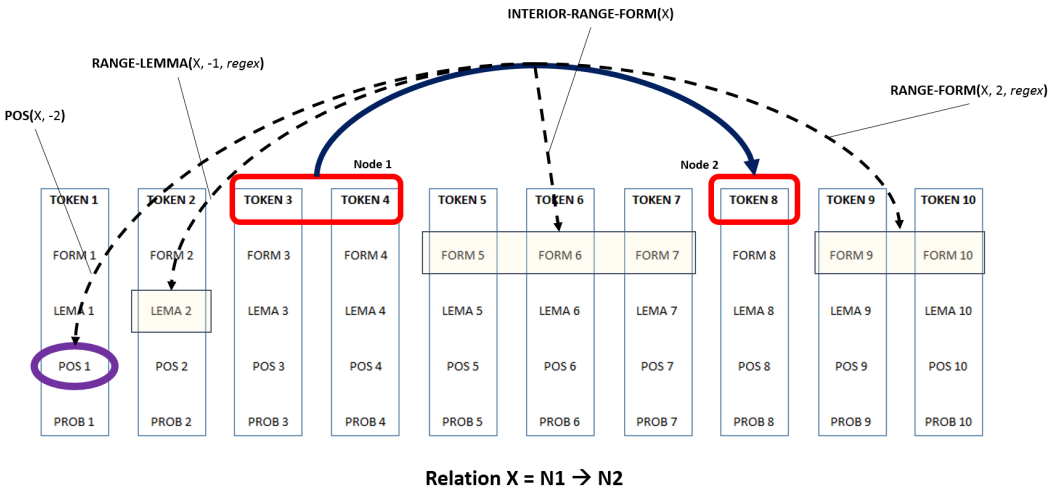


Figura 4.7: Exemplo ilustrativo dos atributos disponíveis para RE.

O POS *tagger* Freeling adota uma abordagem multiníveis para as *tags* de POS, o que permite a caracterização morfossintática dos *token* em mais subcategorias do que em outros POS *taggers*. Isso confere ao LER mais possibilidades de separação entre classes, uma vez que mais dimensões para a separação/classificação são geradas. Assim, como ilustrado na figura 4.8, as configurações das tarefas de NER e RE permitem a definição de quais e quantos aspectos de POS devem ou não ser usados.

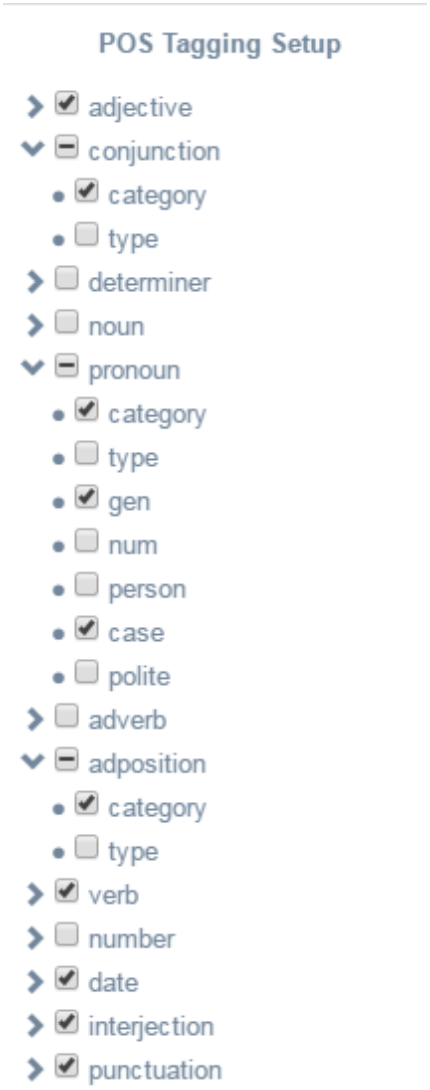


Figura 4.8: Árvore de opções de níveis das categorias de POS a serem consideradas (com apenas algumas categorias expandidas, para reduzir a imagem).

As tarefas de NER fazem uso de alguns classificadores disponíveis no pacote Sklearn 0.18.1 (62), (63) e as de RE, de classificadores baseados em aprendizado estruturado disponíveis no pacote PyStruct 0.2.4 (64). Para NER,

as instâncias são passadas ao classificador divididas por *token*, como ilustrado na figura 4.9.

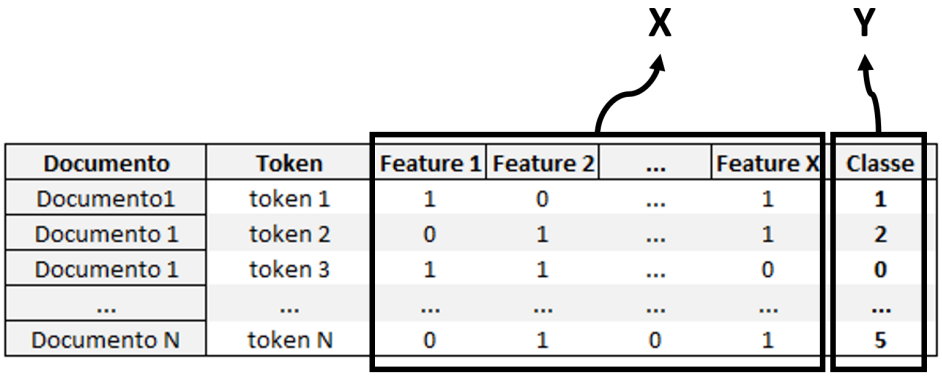


Figura 4.9: Esquemático da modelagem adotada pelo LER para a tarefa NER. "X"e "Y"se referem, respectivamente, aos *inputs* e aos *outputs* do modelo.

Seguindo uma das reflexões levantadas na revisão bibliográfica, implementou-se a tarefa de RE no LER com uma abordagem diferente da adotada em (13). No referido artigo, usa-se um método que, no fim, gera somente árvores, com relações extraídas por uma classificação binária e então etiquetadas com base nas classes dos nós envolvidos. Isso é um problema, pois as árvores impedem que se modele relações cíclicas, ou relações não conexas em um mesmo documento. Além disso, o fato de se fazer uma classificação binária com etiquetamento posterior trará problemas quando houver múltiplas relações possíveis para um mesmo par de classes, pois algum outro método para decidir a relação correta se fará necessário. Destarte, optou-se, na implementação de RE no LER, pelo uso de uma modelagem que evitasse esses problemas, através do pacote PyStruct, que disponibiliza algoritmos de aprendizado estruturado. Em resumo, o aprendizado estruturado visa a aprender uma estrutura, um grafo, em vez de simples classes, de modo que o aprendizado de uma classe acontece acoplado ao aprendizado das outras que fazem parte da mesma estrutura. Um dos modelos disponibilizados pelo pacote, chamado Linear-chain CRF (*Conditional Random Fields* (65), (66)), adota a premissa de que se quer aprender e predizer as classes de uma cadeia de nós, considerando também que as arestas têm igual significado e potencial. Essa modelagem foi escolhida por permitir o aprendizado simultâneo de todas as relações entre entidades já identificadas em um documento, permitindo a diferenciação natural de ambiguidades de relações e, o mais importante, não esperando uma árvore como saída do modelo. Para tal, considerou-se que, para fins de classificação, as relações entre pares de entidades são nós

de uma cadeia linear, o que obviamente não é a realidade prática, mas serve simplesmente pela possibilidade de se considerar, de forma integrada, um conjunto de atributos para cada relação da cadeia. A figura 4.10 ilustra a ideia.

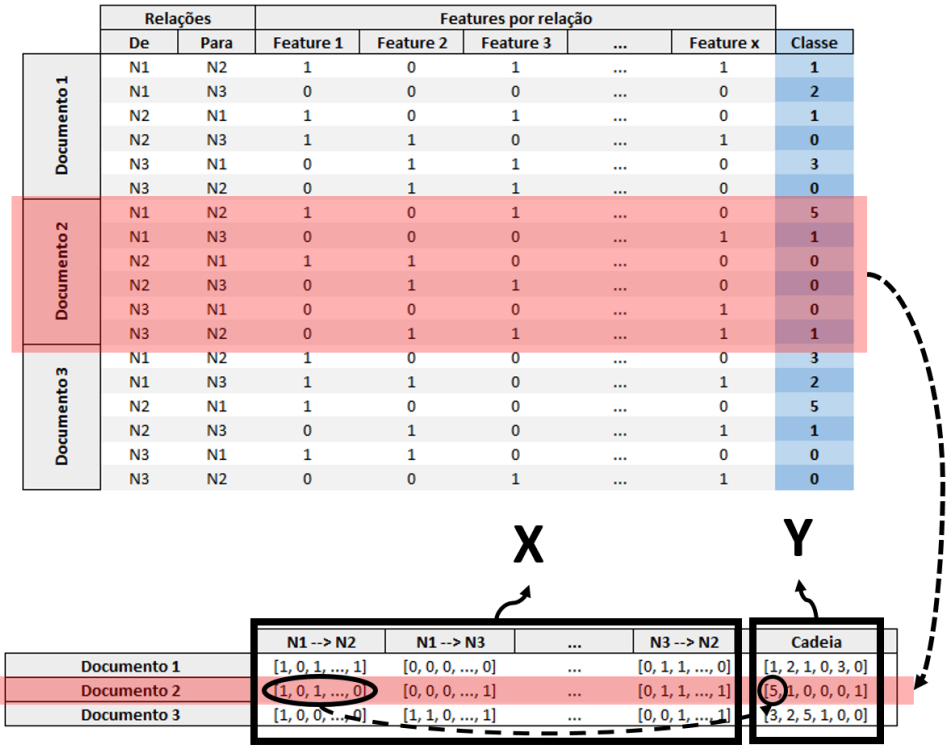


Figura 4.10: Esquemático da modelagem adotada pelo LER para a tarefa RE, pelo uso do modelo Linear-chain CRF do pacote PyStruct. "X"e "Y"se referem, respectivamente, aos *inputs* e aos *outputs* do modelo.

Uma vez estabelecida a ferramenta usada neste trabalho, faz-se necessária uma ontologia que modele, para determinado fim, os textos dos relatórios SITOP. O próximo capítulo trata de uma proposta de ontologia leve para este fim.

5
Proposta de ontologia para estruturação de dados do SITOP

Seguindo as reflexões ao fim da revisão bibliográfica, o presente trabalho propõe uma ontologia leve para fins de estruturação das informações dos relatórios SITOP. Cabe ressaltar que o processo de produção da referida ontologia não buscará um nível de "perfeição" e robustez para uma primeira aplicação, seguindo um rigor metodológico como, por exemplo, o usado para a OntoCAPE - que tem muitos anos de desenvolvimento, diversos participantes, muitos ciclos de melhoria, etc. - pois o foco desta pesquisa está em analisar a viabilidade do uso da metodologia NLP + ontologias proposto em trabalhos recentes para *tweets* de trânsito. Evidentemente, quaisquer trabalhos futuros neste mesmo tipo de dados deverão melhorar a ontologia, propor mecanismos de ligação ou correspondência desta com ontologias gerais mais pesadas, entre outros avanços.

Ontologias como OntoCAPE e OntoSAFE, pesadas e complexas, estão excluídas do horizonte de aplicação para a modelagem proposta. Poder-se-ia propor a ontologia usada em (54) para a estruturação da base de conhecimento de PEFs. Como discutido na revisão bibliográfica, esta foi construída seguindo um certo padrão de especificação, conceptualização, formalização e representação, à semelhança do que foi feito no caso da OntoCAPE. Além disso, há a vantagem de a falha em equipamentos de processos, assunto dos mais importantes nos SITOPs, ser o foco, representando os equipamentos mais importantes em um processo industrial e, consequentemente, em uma plataforma de processamento primário de petróleo. Outra candidata possível seria a proposta em (53), ainda mais em se tratando de um enfoque total em plantas de processamento de petróleo. Outro ponto positivo é a classificação hierárquica de entidades que considera a real subdivisão de unidades de produção, áreas de processamento, sistemas, subsistemas e equipamentos, modelando o modo em que engenheiros e operadores enxergam as plataformas. Mesmo uma evolução ou, por assim dizer, "concretização" da ontologia superior proposta segundo a ISO 15926-2:2003 (23) poderia ser considerada. Talvez seu aspecto mais importante seja a tetradimensionalidade que trata, de certa forma, os casos de troca de equipamentos reais em uma mesma designação funcional. Todas as ontologias aqui citadas consideram, de formas diferentes,

as relações de causa-efeito em eventos em processos industriais.

5.1
Definição de um padrão para a ontologia

Entretanto, para fins de aplicação prática nos relatórios SITOP, outra abordagem mais adequada pode ser adotada. A PETROBRAS é participante de uma organização chamada OREDA (*Offshore & Onshore Reliability Data*), um projeto compartilhado por oito empresas de petróleo e gás com operações a nível mundial, cujo objetivo principal é a coleta e o compartilhamento de dados de confiabilidade entre as organizações envolvidas. Confiabilidade é a capacidade de sistemas e equipamentos em desempenhar satisfatoriamente suas funções de acordo com especificações e condições preestabelecidas.

A OREDA também atua como um fórum de coordenação e gerenciamento de coleções de dados de confiabilidade na indústria de petróleo e gás. O projeto teve seu início em 1981 como uma iniciativa da antiga "Diretoria de Petróleo Norueguesa" (atual "Autoridade de Segurança de Petróleo"), com o objetivo primário de coletar dados de confiabilidade para a segurança de equipamentos, aumentando sua capacidade de concentração de dados através da participação de outras grandes empresas do setor (67). A metodologia adotada pelo OREDA tem sua representação formal (taxonomia e especificações) na ISO 14224 (*Petroleum, petrochemical and natural gas industries—Collection and exchange of reliability and maintenance data for equipment*).

Para uma aplicação prática, o presente trabalho testará a viabilidade do uso da técnica NLP + ontologia para a estruturação das informações dos relatórios SITOP sobre equipamentos e sistemas em plantas de processamento primário de petróleo e gás, propondo, para isso, uma ontologia leve tendo como base a ISO 14224 (com algumas inserções úteis, quando necessárias).

5.2
Descrição geral e destaques importantes da ISO 14224

Este trabalho usa a versão de 2006 do referido padrão (68). O escopo da ISO 14224 provê uma base taxonômica, uma linguagem padronizada comum, para a coleta e o armazenamento de dados de confiabilidade e manutenção de equipamentos em todo tipo de processos produtivos e operações da indústria de petróleo, gás natural e petroquímicos. Ela descreve princípios gerais, termos formais e definições que constituem uma "linguagem da confiabilidade", útil para o compartilhamento de experiências operacionais neste tema.

Alguns termos e definições disponíveis no capítulo III deste padrão merecem destaque para seu uso no presente trabalho:

- "falha" (*failure*): perda da capacidade de um item de executar uma função requerida.
- "causa da falha" (*failure cause*): circunstâncias associadas ao projeto, manufatura, instalação, uso e manutenção que conduziram à falha.
- "mecanismo da falha" (*failure mechanism*): Processos físicos, químicos ou de outra natureza que conduziram à falha.
- "modo da falha" (*failure mode*): efeito pelo qual a falha é observada no item em falha.
- "item" (*item*): qualquer parte, componente, dispositivo, subsistema, unidade funcional, equipamento ou sistema que pode ser considerado de modo individual.
- "manutenção" (*maintenance*): combinação de todas as ações técnicas e administrativas, incluindo ações de supervisão, que objetivem reter um item em um estado, ou restaurá-lo a um estado, em que este possa exercer uma função requerida.
- "estado operacional" (*operating state*): estado em que um item está exercendo sua função requerida.
- "*up state*": estado de um item caracterizado pelo fato de que este pode exercer uma função requerida, assumindo-se que os recursos externos, se requeridos, são providos.
- "*down state*": estado desabilitado interno de um item caracterizado por uma falha ou por uma possível incapacidade de executar uma função necessária durante uma manutenção preventiva.
- "*up time*": intervalo de tempo durante o qual um item está em "*up state*".
- "tempo ocioso" (*idle time*): parte do "*up time*" em que um item não está funcionando.
- "*tag number*": número que identifica a localização física do equipamento.

O capítulo V da ISO 14224, que trata da cobertura do padrão em termos de tipos de equipamentos, indica que este padrão se aplica a equipamentos e sistemas usados na indústria de petróleo, gás natural e petroquímica, incluindo, mas não limitado a categorias de equipamentos tais como equipamentos de processo e tubulações, equipamentos de segurança, equipamentos *subsea*, equipamentos de carregamento / descarregamento, equipamentos de poços e equipamentos de perfuração. Os relatórios SITOP abordam eventos relacionados a todas essas classes de equipamentos, o que

mais uma vez corrobora o uso deste padrão para a estruturação das informações operacionais em formato de texto.

O capítulo VII, que trata da qualidade dos dados, cita alguns problemas e limitações na obtenção dos dados de confiabilidade:

Quanto às fontes: podem faltar dados suficientes e podem se tratar de fontes espalhadas em muitos e diferentes sistemas (computadores, arquivos, livros, desenhos etc).

Quanto à interpretação: os dados são comumente armazenados em bases de dados padronizadas. Todavia, no processo de alimentação das bases, as fontes podem ser interpretadas de modos diferentes por indivíduos diferentes. É exatamente neste ponto que a padronização na forma de definições pode reduzir os erros ou imprecisões.

Quanto ao formato: para fins de limitação e de correta e facilitada análise dos dados, a informação codificada é preferível ao formato em texto livre. Entretanto, recomenda-se o cuidado com a seleção e aplicação dos termos e nomenclaturas apropriados para a informação requerida. Algo que também requer atenção é o fato de que, apesar de o uso de códigos promover a redução do tamanho das bases de dados, também resulta na "perda" de parte das informações, como detalhes de cada caso que são descritos por texto livre. Assim, as informações textuais devem ser incluídas a fim de detalhar situações não tão claras e inesperadas (não cobertas em detalhes pelo padrão).

Quanto ao método de coleta dos dados: a maioria dos dados requeridos para esta área são hoje armazenados em sistemas computadorizados e, através do uso de algoritmos avançados de conversão, é possível a transferência de dados, em um modo (semi)automatizado, entre as diversas bases estabelecidas, reduzindo custos.

Quanto à competência e à motivação: a coleta de dados do modo manual "usual" pode se tornar repetitiva e tediosa. Desse modo, indica-se que o pessoal empregado nesta tarefa deve ter um *know-how* razoável e não deve ser inexperiente.

O uso da técnica de TA proposta nesta dissertação, se bem aplicada, pode ajudar a evitar a maior parte dos problemas citados acima. Quanto às fontes, possibilita a coleta de dados antes não computacionalmente tratáveis (relatórios operacionais em texto livre) tendo como base uma ontologia do referido padrão.

Quanto à interpretação, partindo do princípio de que o processo de anotação de textos para treinamento dos modelos de aprendizado de máquina é feito de forma correta e coerente, empregando-se também metodologias posteriores que impeçam o armazenamento de informações erradas geradas

pelos mesmos modelos, tem-se que esta técnica tende a padronizar a interpretação das fontes textuais.

Quanto ao formato, o uso de uma ontologia promove uma formalização real dos conceitos e definições do padrão, além de permitir, eventualmente, a inferência de informações não presentes nas fontes textuais.

Quanto ao método de coleta, a técnica ora proposta se apresenta como uma forma avançada de estruturação e disponibilização dos dados.

Por fim, quanto à motivação, o uso de um método de "leitura" automática dos dados textuais dispensa a necessidade da presença humana na estruturação dos dados de todos os milhares de relatórios disponíveis.

O capítulo VIII da ISO define os conceitos de fronteiras de equipamento, taxonomia e relativos a tempo. As fronteiras de equipamento são formas claras e imperativas de estabelecer "onde começam e terminam" os domínios de um equipamento, sistema etc. Isto serve à padronização da interpretação para a coleta de dados. A figura 5.1, por exemplo, mostra em um diagrama a fronteira definida para o conceito de "bombas".

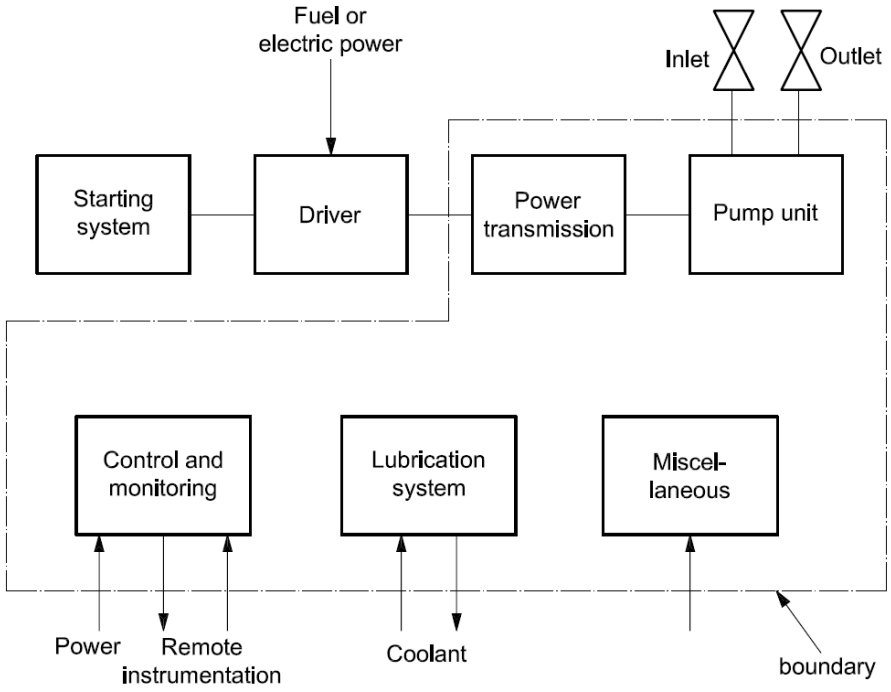


Figura 5.1: Exemplo de um diagrama que define a fronteira de bombas. (68)

O conceito de taxonomia é definido como uma classificação sistemática de itens em grupos genéricos tendo como base fatores possivelmente comuns a muitos os itens (localização, uso, subdivisão de equipamentos etc.). A figura 5.2 apresenta a taxonomia geral, em níveis, da ISO 14224.

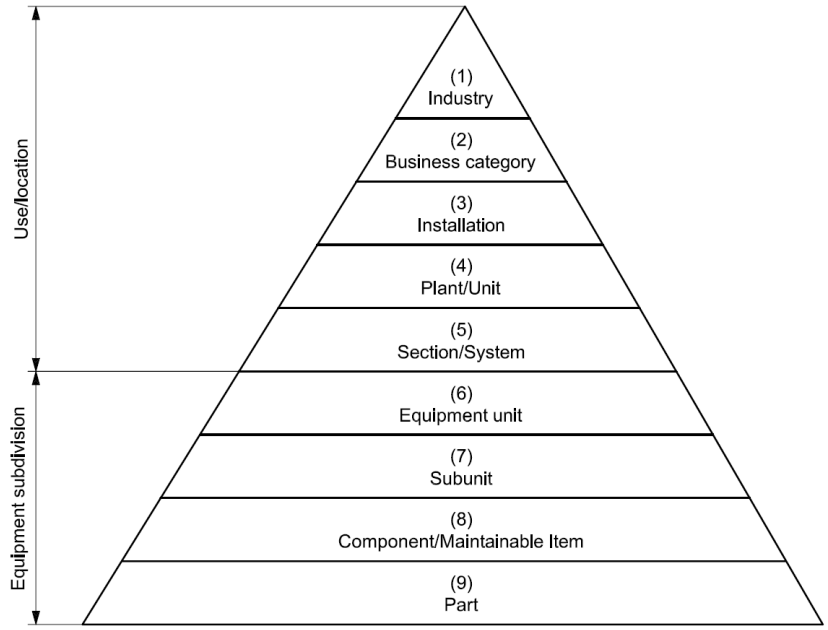


Figura 5.2: Taxonomia geral da ISO 14224. (68)

Os níveis 1 a 5 representam uma categorização de alto nível que se relaciona a indústrias e plantas de processo a despeito das unidades e equipamentos envolvidos. Esses níveis existem para lidar com equipamentos que podem ser usados em diversas indústrias e configurações de plantas de processo. Assim, para a análise da confiabilidade de equipamentos similares (ex.: bombas em plataformas *offshore* e bombas em refinarias) é necessária a especificação do contexto operacional.

Os níveis 6 a 9 correspondem às unidades de equipamento, às subdivisões de mais baixo nível, em uma estrutura de relacionamentos pais-filhos. Para a coleta de dados, este padrão tem um foco maior nas unidades do nível 6 (ex.: trocadores de calor, compressores, tubulações, bombas, aquecedores, turbinas a gás etc.) fazendo referências indiretas às subunidades e aos componentes dos níveis mais baixos. O número de níveis de subdivisões necessário para a coleta dos dados de um determinado equipamento depende da complexidade deste e do uso a que se destina o conjunto de informações.

O documento é extenso, com muitas definições e diretrizes. Para a construção da ontologia-exemplo aplicada no caso dos relatórios SITOP, algumas partes deste padrão desempenham um papel mais fundamental além daquilo que já foi comentado acima. Em primeiro lugar, destaca-se a tabela A-4 (do anexo A) do padrão, que lista os principais exemplos de unidades e equipamentos (quanto ao nível 6 da taxonomia) presentes na indústria de petróleo, gás natural e petroquímica, categorizados quanto a princípios gerais

de funcionamento. Bombas e compressores, por exemplo, são equipamentos rotativos (*Rotating*) enquanto vasos e filtros são equipamentos mecânicos (*Mechanical*).

Outro destaque deve ser dado à seção A.2 (*Equipment-specific data*), que abre as definições dos equipamentos categorizados na tabela A-4 em termos das já comentadas definições de fronteira e das subdivisões dos níveis mais baixos conforme a necessidade de cada caso. As tabelas da referida seção são usadas para elucidar as taxonomias e definições de entidades na ontologia no tocante à representação dos equipamentos de processo e subitens existentes no processamento primário citados nos relatórios SITOP.

Além das definições de equipamentos concretos, a ISO 14224 prevê uma nomenclatura para a interpretação de falhas, detalhada no Anexo B. Os textos presentes no SITOP descrevem falhas em geral, em diferentes graus de detalhamento e pontos de vista, posto que se trata de um texto livre. Nos textos usados neste trabalho, as descrições das falhas quase sempre se referem aos mecanismos ou aos modos destas. Pode-se dizer que, pelo conteúdo dos textos, seria mais adequada a classificação da falha quanto ao modo. Todavia, para fins de simplificação (redução do número de classes para as atividades de NER dada a pouca quantidade de textos anotados em tempo hábil) as falhas são classificadas na ontologia-exemplo apenas quanto aos seus mecanismos (descritos na tabela B-2 do padrão) e colocadas em uma classe geral que representa a causa de um determinado estado do equipamento. Isso evidentemente não representa completamente o que se determina no Anexo B, isto é, a descrição completa da falha quanto à causa, ao modo e ao mecanismo, mas adaptações desta ontologia poderão ser feitas em trabalhos posteriores. Além disso, a técnica a ser empregada tem como objetivo realizar um primeiro tratamento apenas do dado textual, o que não significa que toda a estruturação da informação deve ser feita em uma única etapa, que o processo total deva se basear apenas em aprendizado de máquinas ou mesmo que os relatórios textuais sejam a única fonte de informação. Em uma aplicação prática, uma ontologia mais geral deste padrão poderia prever todos os aspectos das falhas e informar os modos e causas com base em fontes além dos textos dos relatórios SITOP, como notas de manutenção e outros documentos. Para representar este *link* entre evento de falha (representada pelo mecanismo) e outras fontes que detalhem os casos, inserir-se-á uma classe chamada *Document*, que representará documentos citados nos textos, relativos a cada caso de falha. As figuras 5.3 e 5.4 apresentam em duas partes a tabela B-2 retirada do texto da ISO 14224 de 2006.

Failure mechanism		Subdivision of the failure mechanism		Description of the failure mechanism
Code number	Notation	Code number	Notation	
1	Mechanical failure	1.0	General	A failure related to some mechanical defect but where no further details are known
		1.1	Leakage	External and internal leakage, either liquids or gases: If the failure mode at equipment unit level is coded as "leakage", a more causally oriented failure mechanism should be used wherever possible.
		1.2	Vibration	Abnormal vibration: If the failure mode at equipment level is vibration, which is a more causally oriented failure mechanism, the failure cause (root cause) should be recorded wherever possible.
		1.3	Clearance/ alignment failure	Failure caused by faulty clearance or alignment
		1.4	Deformation	Distortion, bending, buckling, denting, yielding, shrinking, blistering, creeping, etc.
		1.5	Looseness	Disconnection, loose items
		1.6	Sticking	Sticking, seizure, jamming due to reasons other than deformation or clearance/alignment failures
2	Material failure	2.0	General	A failure related to a material defect but no further details known
		2.1	Cavitation	Relevant for equipment such as pumps and valves
		2.2	Corrosion	All types of corrosion, both wet (electrochemical) and dry (chemical)
		2.3	Erosion	Erosive wear
		2.4	Wear	Abrasive and adhesive wear, e.g. scoring, galling, scuffing, fretting
		2.5	Breakage	Fracture, breach, crack
		2.6	Fatigue	If the cause of breakage can be traced to fatigue, this code should be used.
		2.7	Overheating	Material damage due to overheating/burning
		2.8	Burst	Item burst, blown, exploded, imploded, etc.
3	Instrument failure	3.0	General	Failure related to instrumentation but no details known
		3.1	Control failure	No, or faulty, regulation
		3.2	No signal/ indication/alarm	No signal/indication/alarm when expected
		3.3	Faulty signal/ indication/alarm	Signal/indication/alarm is wrong in relation to actual process. Can be spurious, intermittent, oscillating, arbitrary
		3.4	Out of adjustment	Calibration error, parameter drift
		3.5	Software failure	Faulty, or no, control/monitoring/operation due to software failure
		3.6	Common cause/ mode failure	Several instrument items failed simultaneously, e.g. redundant fire and gas detectors; also failures related to a common cause.

Figura 5.3: Recorte da tabela B-2 da ISO 14224:2006, parte 1/2. (68)

Failure mechanism		Subdivision of the failure mechanism		Description of the failure mechanism
Code number	Notation	Code number	Notation	
4	Electrical failure	4.0	General	Failures related to the supply and transmission of electrical power, but where no further details are known
		4.1	Short circuiting	Short circuit
		4.2	Open circuit	Disconnection, interruption, broken wire/cable
		4.3	No power/voltage	Missing or insufficient electrical power supply
		4.4	Faulty power/voltage	Faulty electrical power supply, e.g. overvoltage
		4.5	Earth/isolation fault	Earth fault, low electrical resistance
5	External influence	5.0	General	Failure caused by some external events or substances outside the boundary but no further details are known
		5.1	Blockage/plugged	Flow restricted/blocked due to fouling, contamination, icing, flow assurance (hydrates), etc.
		5.2	Contamination	Contaminated fluid/gas/surface, e.g. lubrication oil contaminated, gas-detector head contaminated
		5.3	Miscellaneous external influences	Foreign objects, impacts, environmental influence from neighbouring systems
6	Miscellaneous [a]	6.0	General	Failure mechanism that does not fall into one of the categories listed above
		6.1	No cause found	Failure investigated but cause not revealed or too uncertain
		6.2	Combined causes	Several causes: If there is one predominant cause this should be coded.
		6.3	Other	No code applicable: Use free text.
		6.4	Unknown	No information available

[a] The data acquirer should judge which is the most important failure mechanism descriptor if more than one exist, and try to avoid the 6.3 and 6.4 codes.

Figura 5.4: Recorte da tabela B-2 da ISO 14224:2006, parte 2/2. (68)

É comum, nos textos do SITOP, a citação de atividades de manutenção relacionadas às falhas descritas. A ISO 14224:2006 define e resume as categorias gerais das atividades de manutenção na tabela B-5 (*Maintenance activity*). A figura 5.5 apresenta um recorte desta tabela retirado diretamente do texto do padrão.

Code Number	Activity	Description	Examples	Use [a]
1	Replace	Replacement of the item by a new or refurbished item of the same type and make	Replacement of a worn-out bearing	C, P
2	Repair	Manual maintenance action performed to restore an item to its original appearance or state	Repack, weld, plug, reconnect, remake, etc.	C
3	Modify [b]	Replace, renew or change the item, or a part of it, with an item/part of a different type, make, material or design	Install a filter with smaller mesh diameter, replace a lubrication oil pump with another type, reconfiguration etc.	C, P
4	Adjust	Bringing any out-of-tolerance condition into tolerance	Align, set and reset, calibrate, balance	C, P
5	Refit	Minor repair/servicing activity to bring back an item to an acceptable appearance, internal and external	Polish, clean, grind, paint, coat, lube, oil change, etc.	C, P
6	Check [c]	The cause of the failure is investigated, but no maintenance action performed, or action is deferred. Able to regain function by simple actions, e.g. restart or resetting.	Restart, resetting, no maintenance action, etc. Particularly relevant for functional failures, e.g. fire and gas detectors, subsea equipment	C
7	Service	Periodic service tasks: Normally no dismantling of the item	e.g. cleaning, replenishment of consumables, adjustments and calibrations	P
8	Test	Periodic test of function or performance	Function test of gas detector, accuracy test of flow meter	P
9	Inspection	Periodic inspection/check: a careful scrutiny of an item carried out with or without dismantling, normally by use of senses	All types of general check. Includes minor servicing as part of the inspection task	P
10	Overhaul	Major overhaul	Comprehensive inspection/overhaul with extensive disassembly and replacement of items as specified or required	C, P
11	Combination	Several of the above activities are included	If one activity dominates, this may alternatively be recorded	C, P
12	Other	Maintenance activity other than specified above	may dominates	C, P

[a] C: used typically in corrective maintenance; P: used typically in preventive maintenance.

[b] Modification is not defined as a maintenance category, but is often performed by persons trained in the maintenance disciplines. Modification to a major extent can have influence on the operation and reliability of an equipment unit.

[c] "Check" includes the circumstances both where a failure cause was revealed but maintenance action was considered either not necessary or not possible to carry out and where no failure cause could be found.

Figura 5.5: Recorte da tabela B-5 da ISO 14224:2006. (68)

Um comentário deve ser feito quanto ao uso da tabela B-5 na etapa de anotação das atividades de manutenção citadas nos textos do SITOP. Percebe-se, ao longo do processo, que nem sempre é simples a classificação da atividade descrita como uma ou outra atividade. Isso representa, de certa forma, a limitação referente à interpretação e comentada no início deste capítulo, conforme previsto pelo próprio padrão. A anotação foi feita por apenas uma pessoa, o autor da presente dissertação e mesmo assim não foi incomum a incoerência de interpretação variável no tempo, ora considerando que a atividade era de um tipo, ora de outro. Essa dificuldade poderia aumentar ainda mais em um cenário real, onde diversos anotadores participariam da preparação dos dados. Talvez uma análise mais cuidadosa de históricos de

manutenção, um estudo mais profundo do próprio texto da ISO 14224 e a preparação dos dados em conjunto com outros profissionais defina mais claramente a adequação de cada classe a cada caso e isso deverá ser feito em trabalhos que sejam a continuidade deste.

Após comentários gerais sobre o texto do padrão, destacando os pontos mais importantes, descreve-se a seguir a estrutura da ontologia-exemplo usada na aplicação da técnica de TA proposta.

5.3 Definição da ontologia-exemplo baseada na ISO 14224

A ontologia apresentada a seguir foi produzida com o auxílio do *software* Protégé (69). A classe mais abstrata para quaisquer projetos neste *software* é *Thing*. Dela, derivam-se oito classes que definem conceitos gerais. A figura 5.6 mostra as classes de nível mais elevado.

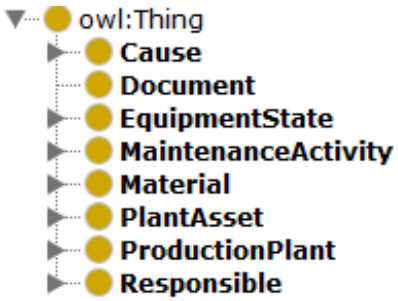


Figura 5.6: Primeiro nível da ontologia-exemplo proposta para teste.

Como declarado anteriormente, na medida do possível buscou-se a adequação desta ontologia-exemplo às nomenclaturas e conceitos expressos na ISO 14224:2006 que se mostraram, pela observação dos dados, os mais passíveis de extração através da técnica de TA a ser empregada.

As classes gerais diretamente ligadas ao padrão são *EquipmentState*, *MaintenanceActivity*, *PlantAsset* e *ProductionPlant*. A classe *Cause*, ilustrada pela figura 5.7, representa as causas de paradas ou mau funcionamento de equipamentos e sistemas em geral. Como comentado anteriormente, as causas de falha foram analisadas apenas pelos seus mecanismos, com base na tabela B-2 do padrão, antes ilustrada nas figuras 5.3 e 5.4.

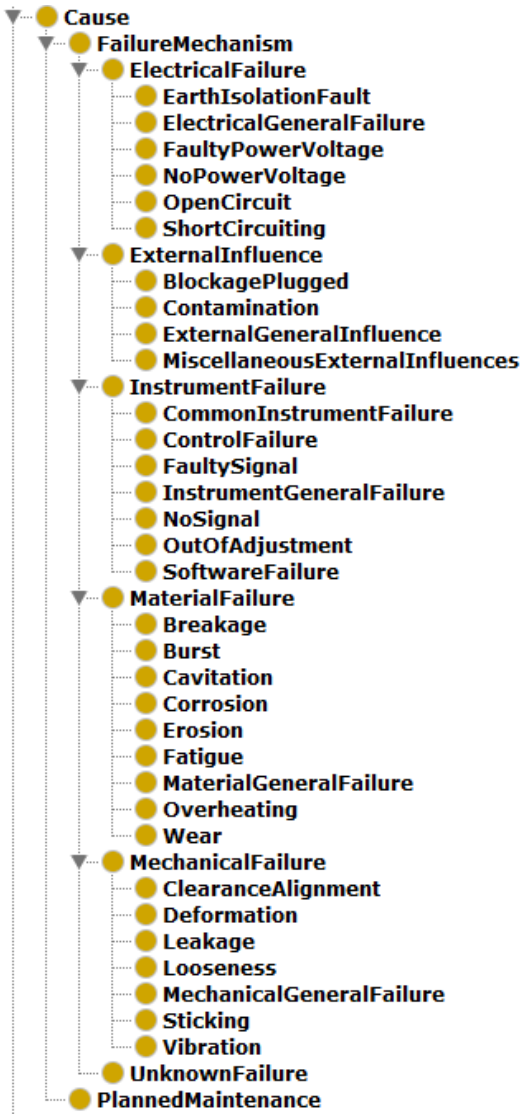


Figura 5.7: Subclasses de *Cause*.

A classe EquipmentState, detalhada na figura 5.8, representa os estados de funcionamento (*UpState*) ou não funcionamento (*DownState*) dos equipamentos, processos, sistemas etc., representadas como classes disjuntas. Essa subdivisão pode ser entendida pelas definições de "*up state*" e "*down state*" feitas no início da última seção.

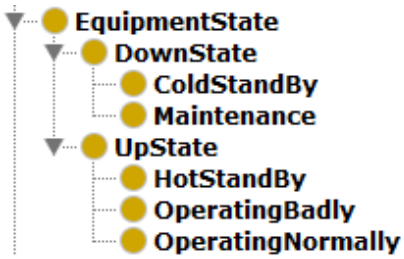


Figura 5.8: Subclasses de *EquipmentState*.

É importante transcrever aqui uma parte do texto do capítulo VIII do padrão, seção VIII.3.1, que embasará algumas definições do parágrafo seguinte:

> *"When equipment is in an idle state or in "hot" standby, i.e. being ready for immediate operation when started, it is considered to be operating (or "in-service") by the definitions in this International Standard. Equipment on standby, which would require some activities to be performed before being ready for operation ("cold" standby), is not considered to be in an operating state."* (68)

Essas definições motivaram a divisão dos estados de *stand-by* em duas classes: *HotStandBy*, que representa o estado em que o item está em *stand-by*, mas parado por algum motivo operacional normal e *ColdStandBy*, que representa o estado em que o item está em *stand-by*, mas sem condições de um funcionamento correto em caso de partida. Seguindo essa lógica, *HotStandBy* e *ColdStandBy* foram colocados como subclasses de *UpState* e *DownState*, respectivamente.

Outras subclasses de *UpState* são *OperatingNormally*, que representa o estado em que o item está operando sem problemas e *OperatingBadly*, que representa o estado em que o item está operando com algum tipo de problema. Já *DownState* tem mais uma subclasse, *Maintenance*, que representa o estado não operacional do item para a execução de alguma ação de manutenção.

A classe *MaintenanceActivity*, ilustrada na figura 5.9, representa as classes de atividade de manutenção da tabela B-5 do padrão, já apresentada na figura 5.5.

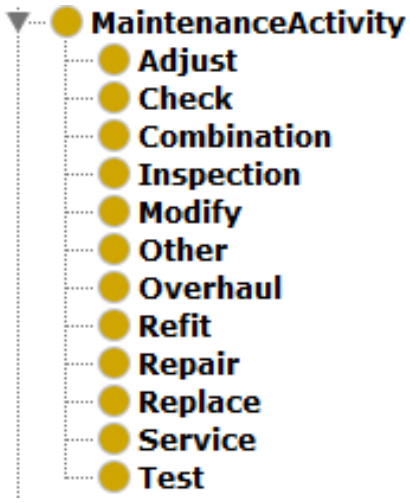


Figura 5.9: Subclasses de *MaintenanceActivity*.

A classe *PlantAsset*, ilustrada na figura 5.10, representa de uma forma muito simplificada os níveis 5 a 9 da taxonomia geral do padrão, conforme já descrito e ilustrado na figura 5.2.

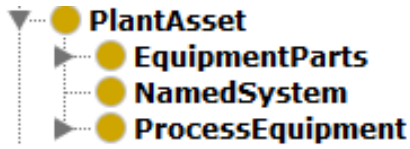


Figura 5.10: Subclasses de *PlantAsset*.

Como é evidente, não há cinco subclasses em *PlantAsset* que representem exatamente os níveis 5, 6, 7, 8 e 9 da referida taxonomia. Para fins de simplificação, a representação hierárquica dos conceitos de equipamentos da ontologia-exemplo foi definida em apenas dois diferentes níveis: um representando os equipamentos divididos em categorias (*ProcessEquipment*) e outro representando os níveis mais baixos como subitens em geral (*EquipmentPart*). Além destes, foi definida também a classe *NamedSystem*, para representar as citações de grandes sistemas especiais do processo (ex.: Sistema de Tratamento de Água), relacionando-a mais ao nível 5, ou pequenos sistemas compostos por pequenas partes (ex.: sistema de medição de nível de um vaso separador), relacionando-a mais ao nível 7. Obviamente, esse esquema não parece muito elegante e seria mais robusta a definição exata da taxonomia, o que deverá ser feito em trabalhos posteriores. Entretanto, como já comentado, a quantidade de dados coletados e anotados em tempo hábil para os experimentos deste trabalho levantou a necessidade de reduzir a quantidade

de classes para a tarefa NER, que mesmo assim já será demasiadamente grande. Deste modo, como são muito variáveis as possibilidades de "sistemas", incluindo seus diferentes tamanhos, em uma só classe foram congregados todos eles.

Descendo um nível em *PlantAsset*, a classe *EquipmentParts*, ilustrada pela figura 5.11, representa os níveis mais baixos da taxonomia, as partes e peças menores que geralmente fazem parte dos equipamentos. A lista não é exaustiva em relação ao padrão, mas apenas os principais itens foram colocados e outros adicionados conforme a necessidade dos textos dos relatórios, quando pequenos itens citados não apareciam formalmente na ISO 14224:2006.

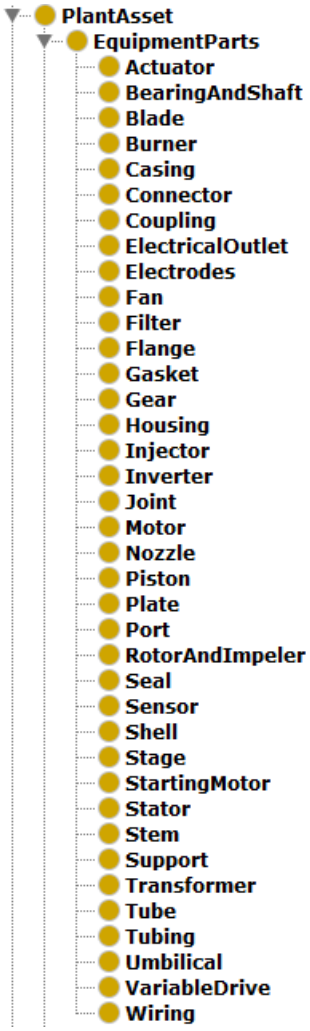


Figura 5.11: Subclasses de *EquipmentParts*.

A classe *ProcessEquipment*, ilustrada pela figura 5.12, representa a maior parte das categorias e equipamentos (os mais importantes para os relatórios SITOP) em conformidade com a tabela A-4 da ISO 14224:2006, já comentada anteriormente.

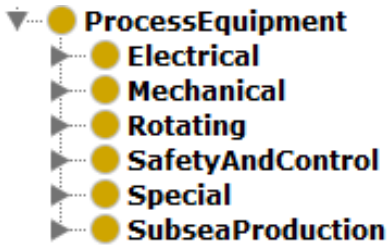


Figura 5.12: Subclasses de *ProcessEquipment*.

Alguns textos dos relatórios SITOP fazem referência a equipamentos não formalmente descritos na versão de 2006 da ISO 14224. Para equipamentos não encontrados no texto do padrão, uma categoria chamada "*Special*" foi adicionada às já previstas na tabela A-4. Uma exceção foi feita no caso do tratador eletrostático (*ElectrostaticSeparator*), que foi colocado abaixo da classe de vasos (*Vessel*). Separadores trifásicos e bifásicos não foram inseridos como classes novas, mas representadas dentro da classe *Vessel*. A fim de resumir a presente descrição da ontologia, não serão abertas cada uma das categorias de *ProcessEquipment*.

Voltando ao primeiro nível, a classe *ProductionPlant*, ilustrada pela figura 5.13, serve apenas para representar a unidade de produção inteira e a subdivisão do processamento primário em suas três partes principais (níveis 3, 4 e 5 da taxonomia): tratamentos de óleo (*ProductionOilArea*), de água (*ProductionWaterArea*) e de gás (*ProductionGasArea*). Essas classes não foram utilizadas, mas eventualmente o serão no futuro, principalmente para a organização dos dados de equipamentos em seus sistemas corretos, através de outras metodologias. Talvez seja interessante a inserção de outras grandes áreas da planta de processos (ex.: sistema de utilidades), a depender de cada aplicação.

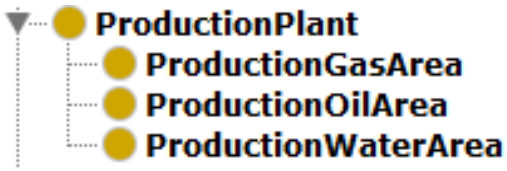


Figura 5.13: Subclasses de *ProductionPlant*.

As outras classes do primeiro nível não estão relacionadas exatamente à ISO 14224:2006, mas a citações peculiares dos relatórios SITOP passíveis de extração pela metodologia proposta e tratamento posterior para detalhamento da informação. A classe *Document* representa documentos representados por

códigos especiais que aparecem nos textos, como notas de manutenção, requisições de transporte de equipamentos e materiais, entre outros. A classe *Material* representa alguns materiais que aparecem nos mecanismos de falha (água, gás, óleo lubrificante, sólidos). Por último, a classe *Responsible* representa alguma equipe, divisão, fabricante etc., responsável por alguma ação de manutenção pela resolução de algum problema operacional.

Algumas propriedades também são definidas nesta ontologia. As *object properties* são:

1. *hasActionRelated*: liga algum estado ou problema operacional a uma ação de manutenção específica.
   Domain: *EquipmentState*, *Cause*
   Range: *MaintenanceActivity*

2. *hasCause*: liga algum estado não operacional, mau funcionamento ou mesmo uma causa de falha a uma outra causa, possibilitando a construção de cadeias de causas e efeitos.
   Domain: *DownState*, *OperatingBadly*, *Cause*
   Range: *Cause*

3. *hasFailure*: liga algum item da planta a um mecanismo de falha que este esteja apresentando.
   Domain: *PlantAsset*
   Range: *FailureMechanism*

4. *hasDocumentRelated*: liga alguma atividade de manutenção, algum estado operacional ou alguma causa de falha a um documento relacionado.
   Domain: *MaintenanceActivity*, *EquipmentState*, *Cause*
   Range: *Document*

5. *hasPlantAsset*: liga a unidade de produção e grandes áreas desta a itens de planta, como equipamentos e subsistemas.
   Domain: *ProductionPlant*
   Range: *PlantAsset*

6. *hasEquipmentPart*: liga equipamentos de processo aos seus itens constituintes e peças em geral.
   Domain: *ProcessEquipment*
   Range: *EquipmentParts*

7. *hasPositionalReferenceToOtherAsset*: liga itens da planta e causas de falhas ou paradas de itens a itens que servem como referências posicionais

para o domínio (ex.: "falha X ocorreu na tubulação ENTRE os equipamentos Y e Z").
Domain: *PlantAsset*, *Cause*
Range: *PlantAsset*

8. *hasResponsible*: liga algum estado não operacional, mau funcionamento ou mesmo um mecanismo de falha a um responsável pelo tratamento do caso.
Domain: *DownState*, *OperatingBadly*, *Cause*
Range: *Responsible*

9. *hasState*: liga a unidade operacional ou algum item de planta a um estado de operação.
Domain: *ProductionPlant*, *PlantAsset*
Range: *EquipmentState*

10. *hasStateAsCause*: liga algum estado de operação ou causa de falha a um estado que seja a fonte do evento (ex.: "equipamento X está parado em razão da parada do equipamento Y.").
Domain: *Cause*, *EquipmentState*
Range: *EquipmentState*

11. *isMaterialOf*: liga algum material a uma causa de falha.
Domain: *Material*
Range: *Cause*

12. *isPartOfOtherEquipment*: liga algum item de planta a outro, do qual é parte.
Domain: *PlantAsset*
Range: *PlantAsset*

13. *isResponsibleForAnAction*: liga um responsável a alguma ação de manutenção.
Domain: *Responsible*
Range: *MaintenanceActivity*

14. *isTargetOfAnAction*: liga um item de planta a uma ação de manutenção realizada ou a ser realizada sobre ele.
Domain: *PlantAsset*
Range: *MaintenanceActivity*

As *data properties* são:

1. *hasDateTime*: liga ações de manutenção, causas de falha e estados de operação a um tempo (data).
   Domain: *MaintenanceActivity*, *Cause*, *EquipmentState*
   Range: *xsd:dateTime*

2. *hasExpectedDateTime*: liga ações de manutenção a uma data esperada.
   Domain: *MaintenanceActivity*
   Range: *xsd:dateTime*

3. *hasExpectedSolutionDateTime*: liga causas de falha e estados não operacionais a uma data esperada para a solução do problema.
   Domain: *Cause*, *DownState*
   Range: *xsd:dateTime*

4. *hasOrdinalNumber*: liga itens de planta a uma designação numérica (aqui, no formato de *string*) que representem sua posição ordinal em algum sistema (ex.: "... 2º estágio do separador...").
   Domain: *PlantAsset*
   Range: *xsd:string*

5. *isAlsoCalled*: liga um item de planta a uma *string* que represente, no texto, um outro nome pelo qual ele é chamado.
   Domain: *PlantAsset*
   Range: *xsd:string*

Descritos o sistema para preparação dos dados e construção de modelos e a ontologia-exemplo proposta, o próximo capítulo apresentará os experimentos realizados.

6
Descrição dos experimentos

Este capítulo descreve os experimentos executados no ambiente LER, que funciona em um servidor Intel® Core™i5, com CPU 650 @ 3.20GHz e 12GB de RAM.

6.1
Coleta e anotação de textos do SITOP

Foram coletadas porções textuais de relatórios SITOP de quatro diferentes plataformas de produção, cobrindo um período de dez anos, isto é, da data mais antiga à mais recente entre todos os relatórios. O procedimento de coleta de "documentos"– conforme a nomenclatura adotada no ERAS-LER – foi feito de modo automático, pelo uso de expressões regulares, tentando a máxima adaptação para cada "estilo" de relatório. Outrossim, a maioria dos textos estava originalmente em caixa-alta, o que é um problema para o uso do Freeling, pelo que também foi implementado um *script* para colocar os caracteres em caixa-alta ou caixa-baixa de acordo com a necessidade.

Uma vez que o procedimento como um todo não estava livre de falhas – as diferenças de estilo eram grandes entre os anos e entre as plataformas – os documentos finais não tiveram uma configuração sempre perfeita. Portanto, a seleção última dos documentos para os testes foi feita manualmente, buscando textos com menos erros no formato final. Esse comentário já elucida que, para uma aplicação industrial da metodologia ora proposta e avaliada, faz-se necessária uma automação inteligente e robusta para a coleta e transformação dos dados originais do SITOP e, só então, alimentação no sistema ERAS-LER, tanto para anotação e treinamento quanto para uso do serviço de estruturação em tempo real.

A limitação de tempo possibilitou a anotação de apenas 551 documentos, com apenas um anotador: o autor do presente trabalho. A quantidade de *tokens* nos documentos para treino e validação variou segundo o demonstrado no histograma da figura 6.1.

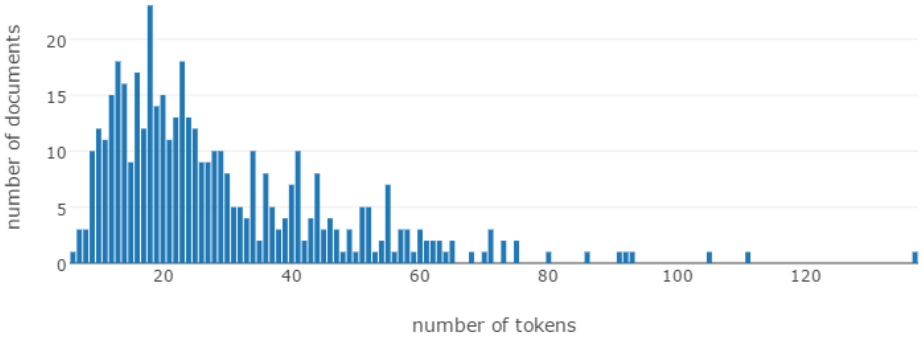


Figura 6.1: Distribuição do número de *tokens* nos documentos de treino e validação.

As figuras 6.2 e 6.3 mostram exemplos de documentos considerados, respectivamente, muito curto e muito extenso.

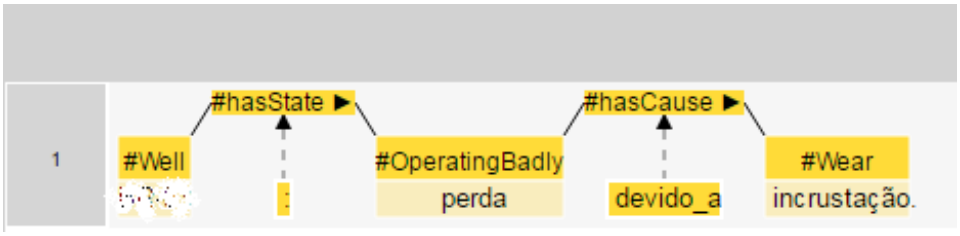


Figura 6.2: Exemplo de documento muito curto, dentre os presentes no pacote de treino.

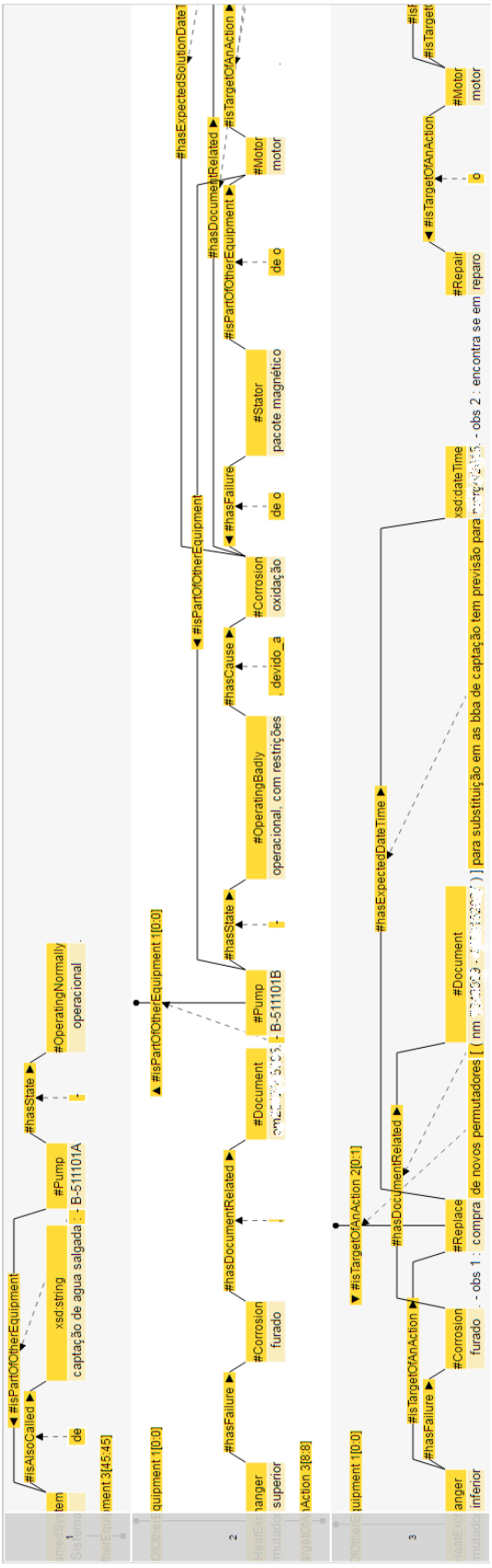


Figura 6.3: Exemplo de documento muito extenso, dentre os presentes no pacote de treino.

Pode-se dizer que, em comparação com a aplicação original para *tweets*, que naturalmente são limitados em 140 caracteres, os mais de 500 documentos anotados neste trabalho representam, na verdade, muito mais de 500 *tweets*.

As estatísticas finais da anotação dos documentos de treino e validação mostram a diferenciada complexidade das tarefas NER e RE para o caso ora estudado, se comparado à aplicação original desta metodologia para eventos em *tweets* de condições de trânsito (13). Apenas uma parcela das classes da ontologia foi efetivamente identificada e anotada, já resultando, entretanto, em 191 classes para NER (tenha-se em mente que cada classe anotada gera duas classes efetivas, em razão dos indicativos de início e meio da entidade, respectivamente "B-" e "I-", além da classe nula ou vazia, que é a mais presente nos dados) e 17 classes para RE.

Após o processo de anotação, dividiu-se aleatoriamente os documentos em três diferentes pacotes, para carregamento no sistema LER: *TRAIN* (treino), *VALIDATION* (validação) e *TEST* (teste). A quantidade de documentos em cada pacote seguiu uma proporção [treino]/[validação] (para construção de modelos e busca por melhores parâmetros) e [treino + validação]/[teste] (para avaliação final dos modelos) de 80%/20%. Seguindo a correta metodologia de avaliação de modelos de aprendizado de máquina, o conjunto de teste não foi usado em nenhum momento para iterar tentativas de construção ou melhoria de modelos, mas somente ao fim do processo para uma única e definitiva avaliação do modelo escolhido no processo de treino-validação. Destarte, os pacotes *TRAIN*, *VALIDATION* e *TEST* contêm, respectivamente, 353, 88 e 110 documentos.

6.2 Experimentos de NER

Os pacotes foram então disponibilizados para a construção de modelos de NER e RE no LER. Para a tarefa NER, foram avaliados inicialmente quatro conjuntos de atributos, variando em termos de detalhamento ou profundidade de descrição do texto. A maior parte foi extraída pelo procedimento automático disponível no LER, considerando o LEMMA dos *tokens* das 50 expressões mais frequentes em cada classe de entidade nomeada. Os LEMMAs que faziam referência a códigos de equipamentos específicos foram generalizados em expressões regulares mais abrangentes.

O primeiro e mais profundo nível de detalhamento, correspondendo aos experimentos com nome NER-SITOP-1, considerou os LEMMAs das posições -2, -1, 0, 1 e 2 de cada *token*, os POS das posições -2, -1, 0, 1 e 2 de cada *token* e o nível máximo de detalhamento da árvore de POS *Tagging*. O segundo nível

de detalhamento, correspondendo aos experimentos com nome NER-SITOP-2, considerou os LEMMAs das posições -2, -1, 0, 1 e 2 de cada *token*, os POS das posições -2, -1, 0, 1 e 2 de cada token e apenas as categorias da árvore de POS *Tagging*. O terceiro nível, correspondendo aos experimentos com nome NER-SITOP-3, considerou os LEMMAs das posições -1, 0 e 1 de cada *token*, os POS das posições -2, -1, 0, 1 e 2 de cada *token* e o nível máximo de detalhamento da árvore de POS *Tagging*. O quarto e último nível, de menor detalhamento, correspondendo aos experimentos com nome NER-SITOP-4, considerou os LEMMAs das posições -1, 0 e 1 de cada *token*, os POS das posições -2, -1, 0, 1 e 2 de cada *token* e apenas as categorias da árvore de POS *Tagging*. As figuras 6.4 e 6.5 ilustram essas configurações.

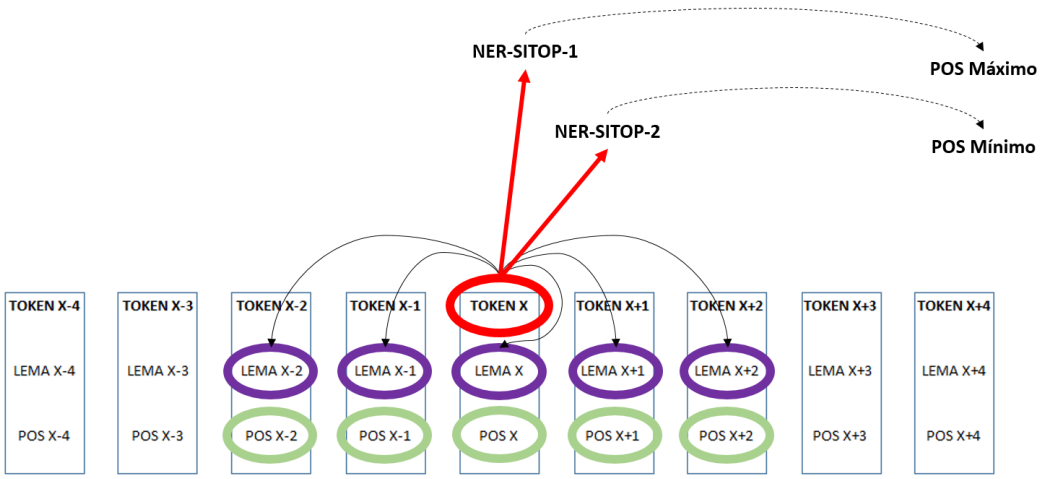


Figura 6.4: Configurações de atributos NER-SITOP-1 e NER-SITOP-2.

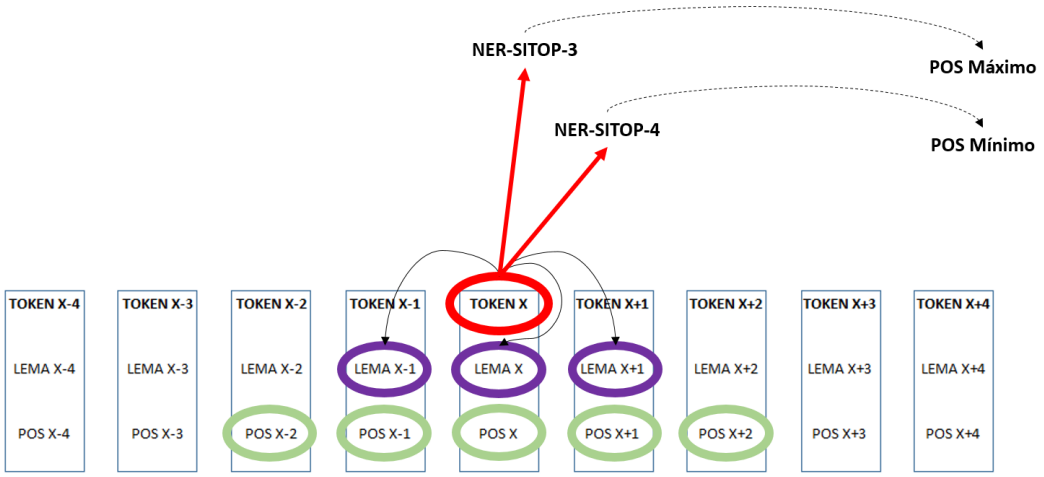


Figura 6.5: Configurações de atributos NER-SITOP-3 e NER-SITOP-4

Os conjuntos de atributos descritos acima foram usados nas buscas de hiperparâmetros de três diferentes algoritmos de classificação do Sklearn: svm.SVC, linear_model.SGDClassifier e ensemble.RandomForestClassifier.

O algoritmo svm.SVC implementa uma classificação por SVM baseada no pacote *libsvm* (70), sendo possível o uso de quatro diferentes *kernels*: linear, polinomial, sigmoide e função de base radial (RBF), sendo esta última o *default*. Alguns testes prévios mostraram que o *kernel* RBF gerava os piores resultados – talvez explicado pelo fato de os valores dos atributos não serem normalmente distribuídos – e os outros não lineares não melhoravam o desempenho em relação ao linear. Além disso, na própria documentação do pacote se diz que a complexidade temporal do algoritmo é mais que quadrática com o número de amostras, o que foi plena e arduamente confirmado nestes testes iniciais: fica evidente que, para um uso industrial, com um número de documentos ainda maior do que os anotados neste estudo, este classificador é simplesmente inviável. Assim, para fins meramente comparativos, adotou-se apenas o uso do *kernel* linear, iterando entre as combinações dos parâmetros *class_weight* (que promove o balanceamento das classes pela compensação de alguns parâmetros internos) e *OneVsRest* (que indica se a estratégia de fazer a classificação binária de cada classe separadamente deve ser adotada).

O algoritmo linear_model.SGDClassifier implementa classificadores lineares, com aprendizado por *stochastic gradient descent* (SGD) (71), (72), onde o gradiente da função objetivo (*loss function*) é estimado a cada nova amostra ou instância avaliada e, diferente do SVC original, permite um aprendizado "*on-line*", não necessitando avaliar o resultado sobre todo o

conjunto de dados em cada iteração da otimização. A estratégia resulta em uma complexidade linear com o número de amostras. O algoritmo foi avaliado apenas na classe *default* "hinge" para o parâmetro *loss* (o que, segundo a documentação, resulta em um SVM linear), iterando entre as combinações dos parâmetros *class_weight* e *OneVsRest*. Para um determinado conjunto de documentos do SITOP e atributos, o tempo de treinamento deste tipo de classificador foi quinze vezes menor que o do svm.SVC e esta diferença certamente aumentaria com um número crescente de documentos.

O algoritmo ensemble.RandomForestClassifier implementa uma classificação por *Random Forest* (73), que é um meta-estimador por árvores de decisão construídas por amostras randômicas do conjunto de treinamento, com reposição. Como resultado da aleatoriedade, o viés da floresta geralmente aumenta (em relação ao viés de uma única árvore não randômica sobre todos os dados) mas com uma queda da variância mais que compensadora do aumento de viés, gerando um modelo melhor em termos gerais. Testes iniciais com os parâmetros *default* (dez árvores, etc.) apresentaram um tempo de treinamento, para um mesmo conjunto de documentos SITOP e atributos, da mesma ordem de grandeza do verificado no linear_model.SGDClassifier. O algoritmo foi avaliado iterando-se combinações dos parâmetros *criterion* (que indica o critério a ser usado para o *split* dos nós de cada árvore), *class_weight* e *OneVsRest*.

6.3 Experimentos de RE

Para RE, foram avaliados inicialmente seis conjuntos de atributos, variando em termos de detalhamento ou profundidade de descrição do texto e das relações entre entidades identificadas em etapa anterior. A maior parte foi extraída pelo procedimento automático disponível no LER, considerando o LEMMA dos *tokens* das cinquenta expressões concatenadas mais frequentes nos *connectors* ligados a todas as relações. Os LEMMAs que faziam referência a códigos e números (datas e documentos, por exemplo) foram generalizados em expressões regulares mais abrangentes.

O primeiro e mais profundo nível de detalhamento, correspondendo aos experimentos com nome RE-SITOP-1, considerou os atributos POSSIBLE-RELATION, CLASS-NODE-FROM, CLASS-NODE-TO, NODE-TO-NODE-DISTANCE, NODE-TO-NODE-DISTANCE-WITH-SIGNAL e, com as expressões regulares automáticas, INTERIOR-RANGE-LEMMA e RANGE-LEMMA, este último com *ranges* de busca -10 e 10. O segundo nível de detalhamento, RE-SITOP-2, conta com todos os atributos de RE-SITOP-1

menos os de RANGE-LEMMA. O terceiro nível, RE-SITOP-3, com todos os atributos de RE-SITOP-2 menos os de INTERIOR-RANGE-LEMMA. O quarto nível, RE-SITOP-4, com todos os atributos de RE-SITOP-3 menos o NODE-TO-NODE-DISTANCE-WITH-SIGNAL. O quinto nível, RE-SITOP-5, com todos os atributos de RE-SITOP-4 menos o NODE-TO-NODE-DISTANCE. O sexto e último nível, RE-SITOP-6, apenas com POSSIBLE-RELATION (um atributo certamente dos mais importantes). A figura 6.6 ilustra essas configurações.

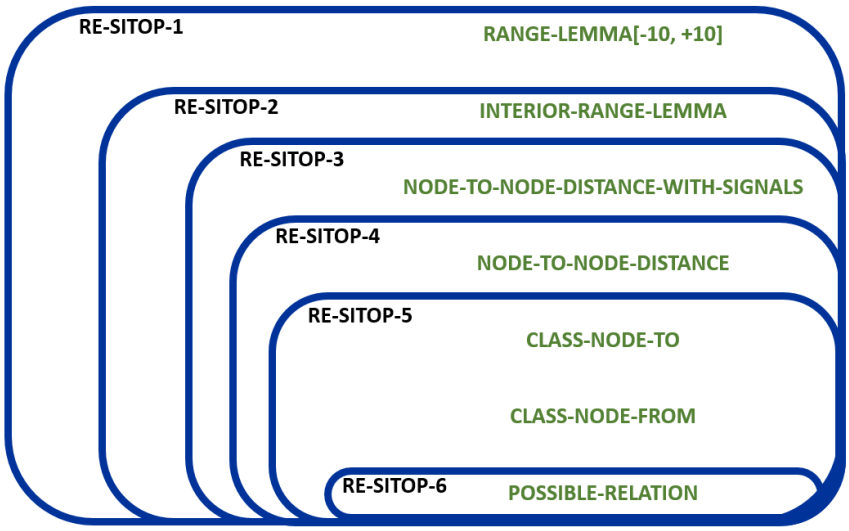


Figura 6.6: Configurações dos pacotes de atributos para a tarefa RE.

Para cada pacote, foram avaliados dois algoritmos de classificação estruturada no modelo ChainCRF do pacote PyStruct: learners.FrankWolfeSSVM (74) e learners.StructuredPerceptron.

Após os testes iniciais seguindo uma validação cruzada de 3-*fold*, foram realizados, para cada tarefa, os treinamentos definitivos com base no conjunto *TRAIN* + *VALIDATION*, com os melhores parâmetros e atributos dos melhores modelos. Os modelos definitivos de NER e RE foram então testados uma única e definitiva vez, sem iterações ou melhorias, sobre o conjunto *TEST*.

6.4 Experimentos de generalização para NER

Tendo em vista a enorme complexidade do problema de aprendizado de máquinas que se busca resolver, isto é, quase 200 classes de entidades para NER, 17 classes de relações para RE, documentos com uma variabilidade grande de número de *tokens* e a quantidade limitada de dados para treinamento, foram testados também alguns casos de exclusão e aglutinação

de entidades da ontologia em nós genéricos, como atividades de manutenção, por exemplo. O objetivo destes testes foi avaliar se uma abordagem de extração de informações em níveis menos detalhados melhora os resultados. Os resultados dessas abordagens diferenciadas podem inspirar novas pesquisas sobre estratégias de extração de informações em textos deste nível de complexidade.

6.5 Métricas para avaliação dos modelos

O sistema LER avalia os resultados de todos os modelos – de NER e RE – e escolhe os melhores parâmetros para cada um dos clássicos índices *Accuracy*, *Precision*, *Recall* e *F1-measure* (75):

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{6-1}$$

$$Precision = \frac{tp}{tp + fp} \tag{6-2}$$

$$Recall = \frac{tp}{tp + fn} \tag{6-3}$$

$$F1\text{-}measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6-4}$$

Sendo:

tp — "true positive"

fp — "false positive"

tn — "true negative"

fn — "false negative"

O próximo capítulo apresenta os resultados resumidos de todos estes experimentos e as discussões pertinentes.

7
Resultados e discussões

O presente capítulo apresenta os resultados dos experimentos propostos no capítulo anterior e algumas discussões pertinentes. Em seguida, para cada tarefa são escolhidos os melhores modelos que são então avaliados uma única e definitiva vez sobre os dados de teste. São também avaliadas, pelo uso dos mesmos modelos, algumas generalizações de classes de acordo com a ontologia. Os resultados detalhados de cada experimento, no formato de tabelas, podem ser consultados no anexo A.

7.1
Tarefa NER

As tabelas A.1, A.2, A.3 e A.4 apresentam os resultados médios para cada experimento inicial sobre o conjunto *TRAIN*, com validação cruzada de 3-*fold* e "teste" intermediário sobre o conjunto *VALIDATION*.

Os valores de *Accuracy* não indicam satisfatoriamente a qualidade dos modelos nos casos estudados. Isso se deve principalmente ao fato de não ser este um caso de classificação binária, porém, ao contrário, com mais de uma centena de classes diferentes. Uma vez que esta métrica é calculada (ver fórmulas do capítulo anterior) pela razão entre o somatório de *true positives* e *true negatives* sobre o total, muitos *true negatives* são, na verdade, outras classes que não foram identificadas como a classe-alvo. Assim, o valor de *Accuracy* será sempre altíssimo, próximo de 1.0, mais em razão de *true negatives* do que pela boa qualidade do modelo para a classe que se busca identificar. Esta métrica não será, portanto, utilizada para a seleção dos modelos para teste.

Escolheu-se a métrica *F1-measure* para a seleção dos melhores modelos por ser uma média harmônica entre *Recall* e *Precision*, fornecendo, portanto, uma informação "equilibrada" e mais robusta da capacidade de extração de informação.

Estabelecidos os critérios de escolha, o conjunto de atributos associado a classificador com maior valor de F1 médio para o treinamento apenas sobre o pacote *TRAIN* é NER-SITOP-2 com *Random Forest*, conforme o destaque feito na tabela A.2. Vale lembrar que o algoritmo SVC não foi escolhido em

razão da inviabilidade de uso com uma quantidade maior de dados, o que fica claro pela grande diferença entre tempos de treinamento.

Mais alguns experimentos foram feitos, agora com os pacotes *TRAIN* e *VALIDATION* como conjunto de treinamento, a fim de melhorar a capacidade preditiva do conjunto NER-SITOP-2 *Random Forest*. Inicialmente, para avaliar a capacidade de aumento do poder preditivo da estratégia pelo simples aumento do número de dados, executou-se um experimento simples com a mesma configuração anterior, sendo a única diferença a quantidade de dados. O resultado de F1 médio aumentou de 0.354 (desvio padrão = 0.313) para 0.455 (desvio padrão = 0.309), demonstrando que, de fato, o aumento da quantidade de dados anotados tende a aperfeiçoar a capacidade da estratégia de identificar as entidades nos textos sem piorar a generalização, visto que o desvio padrão não variou significativamente. Cabe observar que esses testes foram feitos com um número baixo de *folds* na validação cruzada e sem fixação de sementes para a geração de números randômicos e, portanto, novas rodadas destes experimentos podem mostrar valores diferentes, mas o aumento do F1 médio é inegável.

A geração de um modelo definitivo, para teste e construção do serviço, foi feito iterando-se agora sobre o número de árvores da floresta randômica: 5, 10, 12, 15, 20, 25, 30, 40 e 50. O modelo com maior valor de F1 neste caso apresenta os seguintes parâmetros (sendo o restante os valores *default*):

- *Criterion*: *entropy*
- *class_weight*: *'balanced'*
- *One Vs Rest*: true
- *Number of trees in the forest*: 25

A tabela A.5 apresenta os resultados para o conjunto de treinamento (*TRAIN + VALIDATION*) com validação cruzada 5-*fold*, com os parâmetros descritos acima. Os resultados de F1 médio nesta tabela mostram que um aumento no número de árvores melhorou o modelo. Houve também - apesar de não apresentado - um pequeno e correspondente aumento do tempo de treinamento, o que não inviabiliza a estratégia, como o observado para o SVC. Entretanto, algumas classes mostraram um resultado muito ruim, o que na maioria dos casos se deve principalmente ao elevadíssimo e já comentado número de classes diferentes. Além disso, as expressões regulares nem sempre separam perfeitamente alguns casos, sobretudo quando essas expressões são idênticas: *Housing* e *Coupling*, por exemplo, têm o mesmo LEMMA "carcaça", segundo a anotação feita. Como *Coupling* tem um

suporte menor que o de *Housing*, acabou sendo preterido no processo de treinamento do modelo. Outros casos, mesmo com expressões regulares bem definidas, como *Fan* e *Filter*, também não foram detectados no treinamento. Alguns outros casos, entretanto, apresentaram resultados de F1 nulos ou estranhamente baixos por erro na construção do conjunto de atributos (deleção inadequada das expressões regulares especiais dessas classes), o que só foi percebido pela visualização da tabela de *scores*. São eles: *Compressor* e *VapourRecoveryUnit*. Foi construído, então, um conjunto corrigido adicionando esses atributos faltantes. O experimento foi então repetido, sendo os resultados os apresentados na tabela A.6.

Percebe-se que, mesmo com o aumento dos valores de F1 das classes que motivaram a correção ou adição de atributos, o valor médio de F1 caiu de 0.463 para 0.451, o que é um comportamento normal, visto que um novo cenário se estabeleceu para o classificador. Todavia, é importante ressaltar que a média de F1 apresentada no LER não traduz a real escala de importância de algumas classes em relação a outras e em alguns casos talvez seja razoável a perda de qualidade para uma classe menos frequente e importante por uma melhoria na capacidade preditiva de outra classe mais importante. O caso da classe *Compressor*, por exemplo, reflete bem essa questão: antes seu F1 era 0.228, subindo para 0.711 com a correção cujo alvo era exatamente esta classe, mas pagando por isso em perdas da qualidade em outros casos. É evidente que o melhor cenário é a predição aperfeiçoada de todas as classes, mas isso dependerá de uma quantidade maior de dados, melhor e mais cuidadosa engenharia de atributos etc.

Assim, estabeleceu-se o modelo corrigido como o definitivo para teste sobre o pacote *TEST*. A tabela A.7 apresenta os respectivos resultados de teste. O modelo de NER definitivo, para este trabalho, apresenta um F1 médio de 0.536 e desvio padrão entre classes de 0.377, o que não é um resultado perfeito, mas considerado bom para a complexidade do caso. Por uma simples análise da probabilidade de acerto de quaisquer classes via sorteio aleatório, é fácil verificar que o modelo treinado representa um avanço.

7.2 Tarefa RE

Os resultados detalhados de cada experimento podem ser consultados no Anexo. As tabelas A.8, A.9, A.10, A.11, A.12 e A.13 apresentam os resultados médios para cada experimento inicial sobre o conjunto *TRAIN*, com validação cruzada de 3-*fold* e "teste" intermediário sobre o conjunto *VALIDATION*.

Seguindo o mesmo critério de escolha da tarefa NER, o conjunto de

atributos associado a classificador com maior valor de F1 médio para o treinamento apenas sobre o pacote *TRAIN* é RE-SITOP-2. O algoritmo escolhido foi o Structured Perceptron em razão de sua muito maior velocidade em relação ao Frank Wolf SSVM, como destacado na tabela A.9.

O treinamento do modelo definitivo foi feito apenas pela adição do pacote *VALIDATION*, sem mudar em nada o algoritmo original, onde todos os parâmetros continuam com os valores *default*, à exceção de *Average*, que passa a ter o valor *true*. As tabelas A.14 e A.15 apresentam, respectivamente, os resultados para treino e teste deste modelo.

Primeiramente, os atributos RANGE-LEMMA fora das regiões entre as entidades (sob os arcos das relações) pioraram o resultado máximo atingido pelo conjunto de todas as *features*, o que talvez possa ser explicado por conflitos entre expressões regulares ligadas a diferentes relações, quando observados os contextos à esquerda e à direita das relações. Isso não significa que em outros casos a conclusão seria a mesma e, mesmo no caso aqui estudado, é possível que a mudanças nos *ranges* à esquerda e à direita ou um melhor desenvolvimento das expressões regulares melhorem os resultados.

Além disso, também é perceptível a qualidade da abordagem estruturada, com classificação direta das relações, quando comparada ao que foi feito em (13). A estratégia aqui proposta e testada lida satisfatoriamente com ciclos, grafos desconexos e ambiguidades com um único algoritmo.

7.3 Serviço NER-RE

Esta seção mostra alguns exemplos, bons e ruins, dos resultados entregues pelo serviço web construído a partir dos melhores modelos NER e RE apresentados nas seções anteriores. Todos os exemplos são de documentos do pacote *TEST*. A figura 7.1 ilustra a criação do serviço.

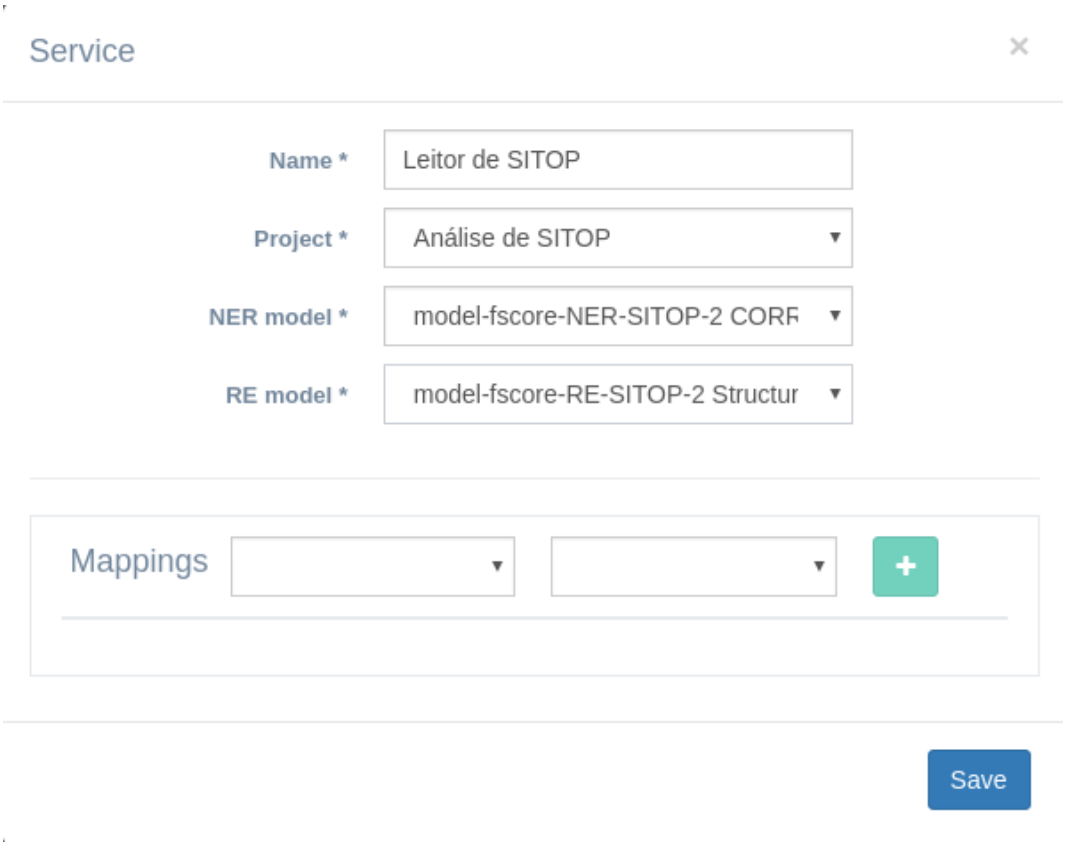


Figura 7.1: Criação do serviço de leitura de SITOP com os modelos NER e RE desenvolvidos e testados.

O exemplo de mau funcionamento da figura 7.2 mostra a identificação correta e incompleta (o *token* "vistoria" deveria ser identificado), mas sem identificação das relações.

Figura 7.2: Exemplo de identificação correta de entidades, mas sem qualquer ligação entre elas

O exemplo da figura 7.3 mostra a identificação indevida de alguns *tokens*, uma vez que o "defeito na tomada elétrica" não foi detalhado, mas erroneamente colocado sob a etiqueta *DownState*. Além disso, falta a relação

hasState entre *Compressor* e *DownState*, o que daria alguma informação útil, apesar de incompleta.

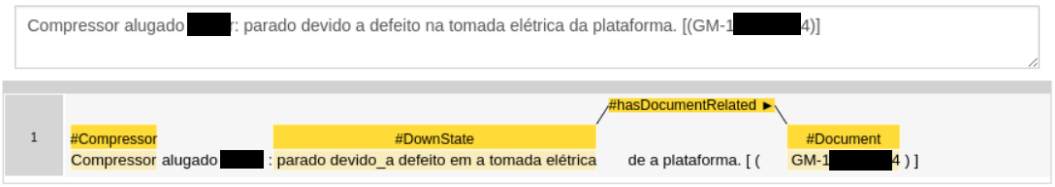


Figura 7.3: Exemplo de problema de etiquetação errônea e consequente perda do detalhamento da informação.

O exemplo da figura 7.4 mostra outro caso de informação incompleta, pois o compressor não foi identificado e ligado ao estado *Maintenance*.

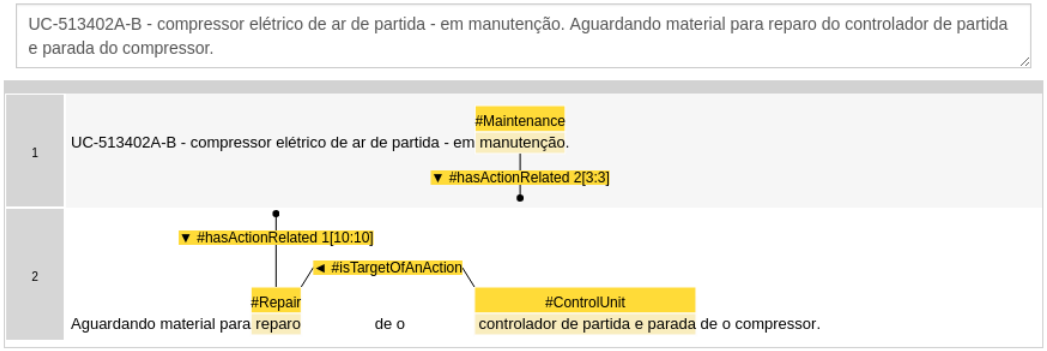


Figura 7.4: Exemplo de não extração de informação útil em razão da não identificação da entidade essencial. Neste caso, o compressor.

O exemplo da figura 7.5 mostra um caso de informação quase completa. O tratador eletrostático tinha seu trafo inoperante, mas como a entidade *Transformer* não foi identificada, o estado *DownState* foi imprecisamente atribuído ao tratador. A informação não é completamente errada, posto que o separador eletrostático com transformadores desligados realmente não exerce ou opera sua função. O motivo e a data do estado inoperante foram correta e precisamente extraídos.

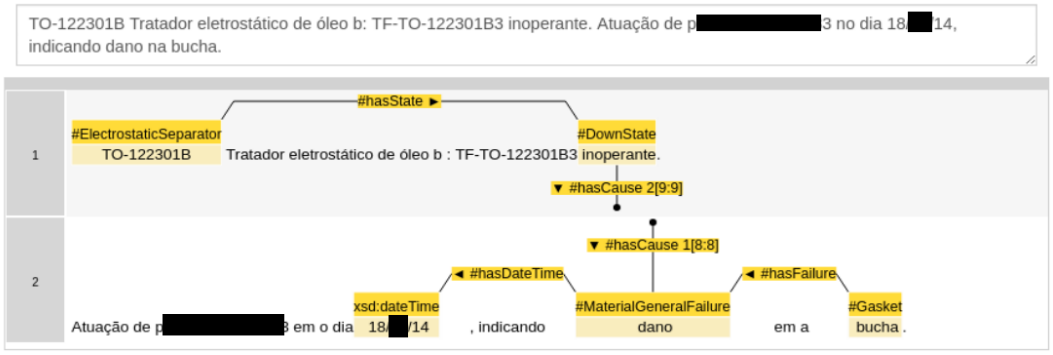


Figura 7.5: Exemplo de informação extraída de forma imprecisa em razão da não identificação de entidade.

O exemplo da figura 7.6 mostra uma informação mais densa e quase perfeitamente extraída, tendo em vista a falta da relação entre o estado *DownState* dos ventiladores e o documento que trata da manutenção, este último corretamente identificado. Aqui fica claro um problema gerado pela própria metodologia de anotação adotada: na maioria dos casos, quando havia um estado e uma falha identificados no texto, optava-se pela ligação entre a falha e os documentos relacionados, não conectando aos documentos, nestes casos, os estados ou as atividades de manutenção. Assim, como neste exemplo só estados são citados e o modelo não "viu" muitos casos de estados ligados a documentos no conjunto de treinamento, não foi inferida a ligação entre *Document* e *DownState*.

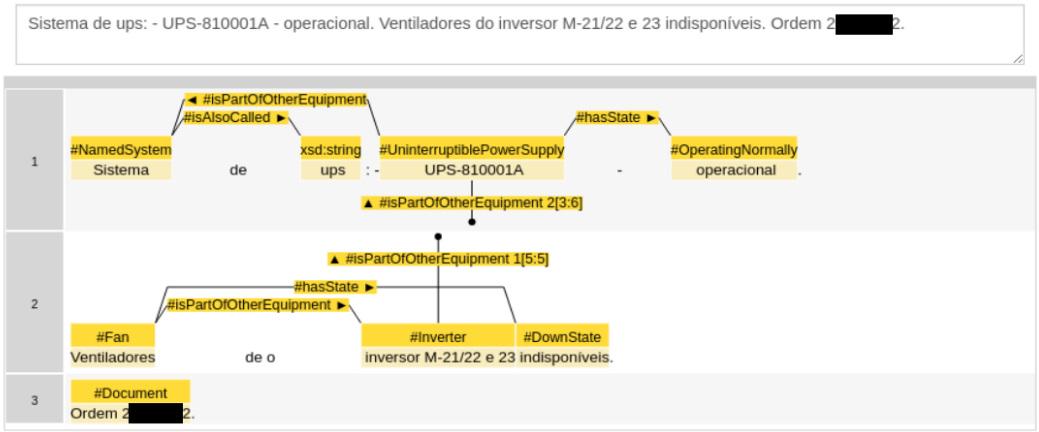


Figura 7.6: Exemplo de informação extraída de forma incompleta por falhas na estratégia de anotação.

O exemplo da figura 7.7 mostra um comportamento semelhante ao anterior, onde falta apenas a ligação entre o evento e o documento. Mais uma vez, como o modelo não viu muitos casos de atividades de manutenção ligados aos documentos, não inferiu a ligação entre *Service* e *Document*.

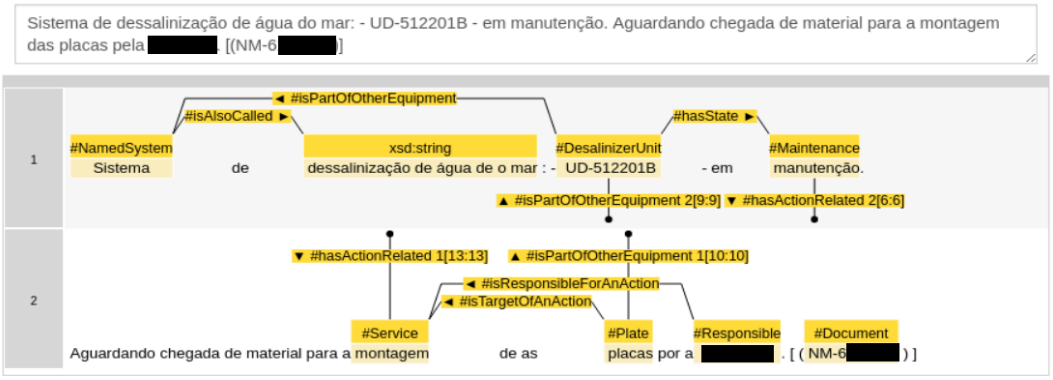


Figura 7.7: Outro exemplo de informação extraída de forma incompleta por falhas na estratégia de anotação.

O exemplo da figura 7.8 mostra a informação correta e precisa, mas sem o importante detalhamento do estado "bloqueado"(*DownState*) do permutador de calor em razão do vazamento e a intervenção a este relacionada.

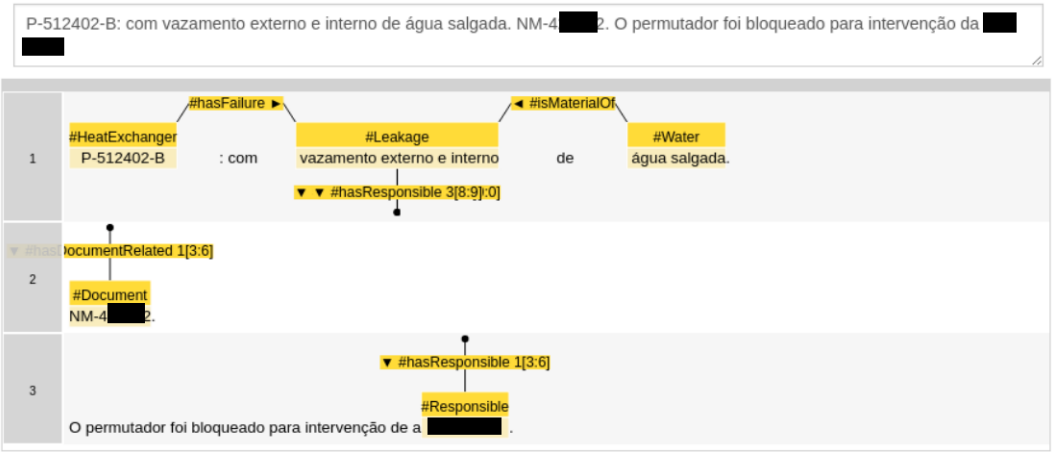


Figura 7.8: Outro exemplo de informação correta e precisa, mas com possibilidades de maiores detalhamentos.

O exemplo da figura 7.9 mostra a indevida identificação do termo “Bomba” no início da sentença como uma entidade, como se duas bombas

estivessem em manutenção. A ação de reparo, entretanto, foi corretamente identificada.

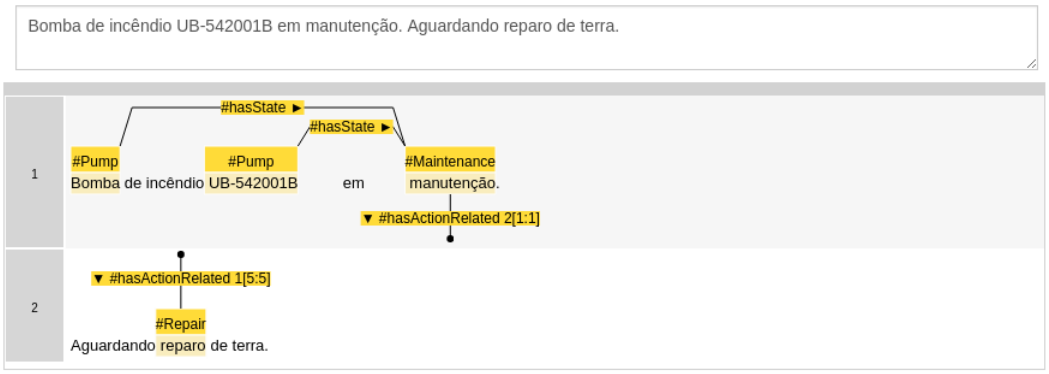


Figura 7.9: Exemplo da mesma entidade sendo identificada duas vezes.

Como o fim de todo este processo é a estruturação dos dados como triplas RDF, as figuras 7.10 e 7.11 mostram um exemplo completo da saída do serviço, com as triplas geradas.

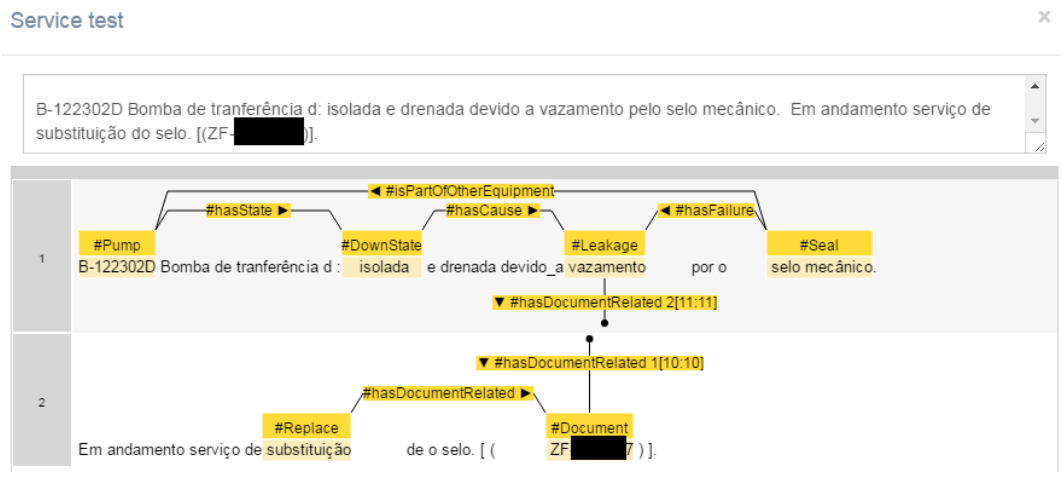


Figura 7.10: Exemplo de saída do serviço.

```
1  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Pump/0> <
       http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/hasState
       > <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       DownState/0> .
2  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Replace/0>
       <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       hasDocumentRelated> <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled
       -ontology-17/Document/0> .
3  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Seal/0> <
       http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       hasFailure> <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-
       ontology-17/Leakage/0> .
4  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Leakage/0>
       <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       hasDocumentRelated> <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled
       -ontology-17/Document/0> .
5  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Document/0>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.org/
       phtf/ontologies/2016/11/untitled-ontology-17/Document> .
6  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Seal/0> <
       http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.org/
       phtf/ontologies/2016/11/untitled-ontology-17/Seal> .
7  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Leakage/0>
       <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.org/
       phtf/ontologies/2016/11/untitled-ontology-17/Leakage> .
8  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Seal/0> <
       http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       isPartOfOtherEquipment> <http://www.semanticweb.org/phtf/ontologies/2016/11/
       untitled-ontology-17/Pump/0> .
9  <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/DownState
       /0> <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/
       hasCause> <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology
       -17/Leakage/0> .
10 <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Pump/0> <
       http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.org/
       phtf/ontologies/2016/11/untitled-ontology-17/Pump> .
11 <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/Replace/0>
       <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.org/
       phtf/ontologies/2016/11/untitled-ontology-17/Replace> .
12 <http://www.semanticweb.org/phtf/ontologies/2016/11/untitled-ontology-17/DownState
       /0> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.semanticweb.
       org/phtf/ontologies/2016/11/untitled-ontology-17/DownState> .
```

Figura 7.11: Triplas RDF retornadas pelo serviço, no exemplo da figura 7.10

7.4 Teste de generalizações

Durante a etapa de anotação, em alguns momentos os mecanismos de falhas (*Cause*) e atividades de manutenção (*MaintenanceActivity*) pareceram

mais difíceis de se discernir, de modo que o próprio modelo final provavelmente será impreciso nas classificações específicas desses dois grupos de entidades. O sistema ERAS-LER disponibiliza um menu de seleções das classes a serem usadas, excluídas e até mesmo generalizadas de acordo com a taxonomia da ontologia. A título de avaliação de abordagens alternativas, foi realizado um teste com a generalização de todos os mecanismos de falha nas suas maiores classes e das atividades de manutenção em uma única classe *MaintenanceActivity*. Os resultados do teste são apresentados na tabela A.16.

A generalização proposta causou um pequeno aumento do valor de F1 médio, indicando que uma abordagem deste tipo, com classes de níveis mais altos, pode melhorar um pouco a qualidade classificação. A partir disso, pode-se pensar, por exemplo, em uma metodologia multi-níveis, com classificadores NER em série, onde as classes de níveis mais altos sejam identificadas primeiro e cada categoria seja detalhada por classificadores mais focados e específicos.

8
Conclusão e trabalhos futuros

Este trabalho propôs o uso de uma metodologia para estruturação automática de dados textuais dos relatórios SITOP das unidades de produção *offshore* da PETROBRAS. Dentre as muitas aplicações de NLP disponíveis, uma pareceu mais interessante para esta empreitada: o uso de ontologias de domínio como bases lógicas que, associadas a técnicas de NLP (NER e RE), transformam dados de texto livre em bancos de triplas RDF.

Um obstáculo ao uso deste tipo de tecnologia é o enorme esforço necessário para a coleta e preparação de dados para a construção de modelos, visto que os textos precisam ser anotados para a identificação das entidades e das relações presentes, tudo isso tendo como referência à ontologia. Além do dispêndio de tempo, é possível que erros sejam cometidos neste processo. Com o objetivo de dar robustez a essa metodologia, desenvolveu-se, em conjunto com outro pesquisador, o sistema ERAS-LER. Trata-se de um sistema *web* que, em um único ambiente, cobre toda a cadeia de atividades, desde a tokenização/POS *Tagging*, passando pelas etapas de construção dos modelos de NER e RE, até a disponibilização de serviços na nuvem para a estruturação automática dos textos como triplas RDF. Trata-se, portanto, de uma contribuição para a área de *Text Analytics*, possibilitando uma maior velocidade na geração de dados e construção de modelos.

A pesquisa na literatura buscou um aprofundamento no tema de ontologias de domínio para processos industriais, sobretudo na área de produção de petróleo. Há muitas iniciativas para diversos objetivos. Em resumo, entendeu-se que a aplicação desta metodologia demanda uma ontologia leve, que não gere um número exagerado de classes e relações. Tendo como motivação a participação da PETROBRAS no projeto OREDA e o foco dos textos do SITOP em questões relacionadas a falhas em equipamentos de processo, o presente trabalho propôs uma ontologia-exemplo totalmente baseada na ISO-14224. Mesmo sendo uma ontologia simples, a quantidade de classes resultante foi muito maior que o da TEDO (ontologia para *tweets* de trânsito), o que fez deste caso um grande desafio para o uso da metodologia implementada no LER.

A principal contribuição do presente trabalho para o ERAS-LER foi

demandar dele uma série de melhorias: no caso do SITOP são maiores os textos, muitas são as entidades presentes e mais complexas são suas inter-relações. Os trabalhos anteriores executavam a extração de relações através de aprendizado estruturado e inferência da estrutura final como uma árvore, que por definição não contém ciclos e precisa ser conexa. Além disso, essa abordagem adotava uma classificação binária (há ou não há relação entre os nós), decidindo qual seria a exata relação por uma consulta a algum tipo de tabela de domínios e *ranges*, o que traz outros problemas quando da presença de ambiguidades de relações entre um mesmo par de classes. A nova estratégia implementada no LER executa um aprendizado estruturado direto, com a classificação conjunta da existência e da classe da relação.

Foram executados diversos experimentos em um conjunto de 551 documentos (441 para treino e 110 para teste) extraídos de relatórios SITOP de 4 diferentes plataformas marítimas, cobrindo um período de 10 anos. Em resumo, para a tarefa NER (191 classes) alcançou-se no conjunto de teste um valor médio de *F1* de 0.536, com *Precision* médio de 0.626 e *Recall* médio de 0.504. Para a tarefa RE (18 classes) alcançou-se no conjunto de teste um valor médio de *F1* de 0.752, com *Precision* médio de 0.844 e *Recall* médio de 0.702. Como esperado, os resultados de NER se mostraram piores que os de RE, se tornando, portanto, o gargalo da metodologia neste caso. Esse comportamento não é estranho, uma vez que a quantidade de documentos anotados não foi grande o suficiente para cobrir o aprendizado de quase 200 classes. Outrossim, não são apenas muitas classes, mas um expressivo desbalanceamento da participação de cada uma no conjunto de dados, mesmo se a classe vazia for desconsiderada. Para fins de avaliação do salto na qualidade do modelo para NER através do fornecimento de mais dados, testou-se um caso de treino apenas com uma fração do conjunto de treino original, comparando os resultados com os do treino de todo o conjunto de treino original. Houve uma melhora expressiva: um aumento de aproximadamente 25% na quantidade de dados causou um aumento de quase 30% em F1. Resta evidente, portanto, que mais dados precisam ser anotados.

Foi testada também a estratégia de generalizar algumas classes, com vistas à redução da complexidade do problema. Houve alguma melhora de 5% no valor de F1 para o conjunto de teste pela redução de 33% na quantidade de classes, mas que não foi tão expressiva quanto a gerada pelo aumento da quantidade de dados.

A partir destes resultados, pode-se concluir que a metodologia é adequada para o caso dos relatórios SITOP, mesmo sendo este um problema mais complexo do que os que originalmente foram usados. Conclui-se também que há

a necessidade de maior quantidade de dados anotados, pois a qualidade final do modelo de NER se mostra muito mais sensível a esta variável do que a qualquer outra. A qualidade destes dados também pode aumentar se mais anotadores estiverem envolvidos e análises de concordância entre estes forem realizadas. Também é possível que melhorias nos atributos e aplicação de técnicas para indução e seleção automáticas de *features* deem bons resultados, mesmo com o conjunto de dados atual. Não obstante, recomenda-se também novos estudos de outras alternativas de abordagem deste tema.

Há trabalhos (76) no sentido de lidar com o problema de um modo direto, *"end-to-end"*, realizando NER e RE em uma única etapa. Essa abordagem pode ser interessante, tendo em vista que as características de NER poderiam servir como informação para RE e vice-versa. Há também trabalhos (77) que tratam da aplicação de técnicas de *Deep Learning* sobre a tarefa NER, o que pode ser muito interessante no sentido de não limitar as possibilidades de aprendizado por *features* tão simples, como as usadas neste trabalho. Há também pesquisas (78) para extração de árvores de dependência também pelo uso de *Deep Learning*, o que pode servir, com alguma adaptação, para uma abordagem "end-to-end"de mais alto nível. Todos esses caminhos podem melhorar muito os resultados alcançados neste trabalho.

Referências bibliográficas

[1] ITTOO, A.; NGUYEN, L. M. ; VAN DEN BOSCH, A.. **Text analytics in industry: Challenges, desiderata and trends**. Computers in Industry, 78:96–107, 2016.

[2] FILIPPOV, S.. **Mapping text and data mining in academic and research communities in Europe**. Lisbon Council, 2014.

[3] GUARINO, N.. **Formal ontology and information systems**. In: PROCEEDINGS OF FOIS, volumen 98, p. 81–97, 1998.

[4] AMAR, F. B. B.; GARGOURI, B. ; HAMADOU, A. B.. **Generating core domain ontologies from normalized dictionaries**. Engineering Applications of Artificial Intelligence, 51:230–241, 2016.

[5] REDLICH, L. R.. **Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da web**. Master's thesis, PUC-Rio, 2013.

[6] SANTOS, I.; MACHADO, M.; RUSSO, E.; MANGUINHO, D.; ALMEIDA, V.; WO, R.; BAHIA, M.; CONSTANTINO, D.; SALOMONE, D. ; PESCE, M.. **Big data analytics for predictive maintenance modeling: Challenges and opportunities**. In: OTC BRASIL. Offshore Technology Conference, 2015.

[7] SOUSA, C.; SANTOS, I.; ALMEIDA, V.; ALMEIDA, A.; SILVA, G.; CIARLINI, A.; PRADO, A.; SENRA, R.; GOTTIN, V. ; BHAYA, A.. **Applying big data analytics to logistics processes of oil and gas exploration and production through a hybrid modeling and simulation approach**. In: OTC BRASIL. Offshore Technology Conference, 2015.

[8] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Petroleum, petrochemical and natural gas industries–production assurance and reliability management**. International Organization for Standardization Geneva, 2008.

[9] CENTRO SVILUPPO MATERIALLI - CSM. **Centro sviluppo materialli - CSM**. http://www.c-s-m.it/en/business_sectors/oil_and_gas.html, 2017. [Acesso em 03 de Abril de 2017].

[10] THOMAS, J. E.. **Fundamentos de engenharia de petróleo**. Interciência, 2001.

[11] FILHO, J. E. S.. **Processamento primário de fluidos: Separação e tratamento**. Universidade Corporativa Petrobras, 2004.

[12] SELLAMI, M.; NAAM, R. ; TEMMAR, M.. **Optimization of operating parameters of oil desalting in southern treatment unit (HMD/Algeria)**. J Pet Environ Biotechnol, 7(271):2, 2016.

[13] ALBUQUERQUE, F. C.; CASANOVA, M. A.; LOPES, H.; REDLICH, L. R.; DE MACEDO, J. A. F.; LEMOS, M.; DE CARVALHO, M. T. M. ; RENSO, C.. **A methodology for traffic-related twitter messages interpretation**. Computers in Industry, 78:57–69, 2016.

[14] SOWA, J. F.. **Knowledge representation: logical, philosophical, and computational foundations**, volumen 13. MIT Press, 2000.

[15] KANEIWA, K.; IWAZUME, M. ; FUKUDA, K.. **An upper ontology for event classifications and relations**. In: AUSTRALASIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, p. 394–403. Springer, 2007.

[16] WORBOYS, M.; HORNSBY, K.. **From objects to events: Gem, the geospatial event model**. In: INTERNATIONAL CONFERENCE ON GEOGRAPHIC INFORMATION SCIENCE, p. 327–343. Springer, 2004.

[17] MOTTA, E. N.; FERNANDES, E. R. ; MILIDIÚ, R. L.. **F-ext-ws-2.0: A web service for natural language processing**. 2010.

[18] PLATT, J. C.. **12 fast training of support vector machines using sequential minimal optimization**. Advances in kernel methods, p. 185–208, 1999.

[19] HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P. ; WITTEN, I. H.. **The weka data mining software: an update**. ACM SIGKDD explorations newsletter, 11(1):10–18, 2009.

[20] DA COSTA ALBUQUERQUE, F.; CASANOVA, M. A.; DE MACEDO, J. A. F.; DE CARVALHO, M. T. M. ; RENSO, C.. **A proactive application to monitor truck fleets**. In: MOBILE DATA MANAGEMENT (MDM), 2013 IEEE 14TH INTERNATIONAL CONFERENCE ON, volumen 1, p. 301–304. IEEE, 2013.

[21] FERNANDES, E. R.; DOS SANTOS, C. N. ; MILIDIÚ, R. L.. **Latent structure perceptron with feature induction for unrestricted coreference resolution**. In: JOINT CONFERENCE ON EMNLP AND CONLL-SHARED TASK, p. 41–48. Association for Computational Linguistics, 2012.

[22] DA COSTA ALBUQUERQUE, F.. **Environment changes detection: A proactive system to monitor moving objects**. Master's thesis, PUC-Rio, 2012.

[23] BATRES, R.; WEST, M.; LEAL, D.; PRICE, D.; MASAKI, K.; SHIMADA, Y.; FUCHINO, T. ; NAKA, Y.. **An upper ontology based on iso 15926**. Computers & Chemical Engineering, 31(5):519–534, 2007.

[24] WIKIPEDIA. **ISO 15926 — Wikipedia, the free encyclopedia**. https://en.wikipedia.org/w/index.php?title=ISO_15926oldid=737691065, 2016. [Acesso em 7 de Março de 2017].

[25] YOGUI, R.. **ISO 15926-padrão internacional para integração e automação no PLM (Plant Lifecycle Management)**. In: V CONGRESSO RIO AUTOMAÇÃO, INSTITUTO BRASILEIRO DE PETRÓLEO, GÁS E BIOCOMBUSTÍVEIS–IBP. RIO DE JANEIRO, 2009.

[26] WEST, M.; SULLIVAN, J. ; TEIJGELER, H.. **ISO/FDIS 15926-2: Lifecycle integration of process plant data including oil and gas production facilities**, 2003.

[27] SIDER, T.. **Four-dimensionalism: An ontology of persistence and time**. Oxford University Press on Demand, 2001.

[28] WU, C.-G.; XU, X.; ZHANG, B.-K. ; NA, Y.-L.. **Domain ontology for scenario-based hazard evaluation**. Safety science, 60:21–34, 2013.

[29] BATRES, R.; SUZUKI, T.; SHIMADA, Y. ; FUCHINO, T.. **A graphical approach for hazard identification**. In: 18TH EUROPEAN SYMPOSIUM ON COMPUTER AIDED PROCESS ENGINEERING IFP, CAPE WORKING PARTY OF THE EUROPEAN FEDERATION OF CHEMICAL ENGINEERING, FRANCE, p. 197–202, 2008.

[30] LEE, B. H.. **Using FMEA models and ontologies to build diagnostic models**. AI EDAM, 15(04):281–293, 2001.

[31] MORBACH, J.; YANG, A. ; MARQUARDT, W.. **OntoCAPE: A large-scale ontology for chemical process engineering**. Engineering applications of artificial intelligence, 20(2):147–161, 2007.

[32] MORBACH, J.; WIESNER, A. ; MARQUARDT, W.. **OntoCAPE: A (re) usable ontology for computer-aided process engineering**. Computers & Chemical Engineering, 33(10):1546–1556, 2009.

[33] BRAUNSCHWEIG, B.; FRAGA, E.; GUESSOUM, Z.; MARQUARDT, W.; NADJEMI, O.; PAEN, D.; PIÑOL, D.; ROUX, P.; SAMA, S. ; SERRA, M.. **CAPE web services: The COGents way**. Computer Aided Chemical Engineering, 18:1021–1026, 2004.

[34] MARQUARDT, W.; NAGL, M.. **Workflow and information centered support of design processes: the improve perspective**. Computers & Chemical Engineering, 29(1):65–82, 2004.

[35] GRUBER, T. R.. **Toward principles for the design of ontologies used for knowledge sharing?** International journal of human-computer studies, 43(5-6):907–928, 1995.

[36] FOX, M. S.; GRUNINGER, M.. **Enterprise modeling**. AI magazine, 19(3):109, 1998.

[37] AZPÍREZ, J.; GÓMEZ-PÉREZ, A.; LOZANO-TELLO, A. ; PINTO, S.. **(ONTO) 2 agent: an ontology-based www broker to select ontologies**. 1998.

[38] CHANDRASEKARAN, B.; JOSEPHSON, J. R. ; BENJAMINS, V. R.. **What are ontologies, and why do we need them?** IEEE Intelligent Systems and their applications, 14(1):20–26, 1999.

[39] GOMEZ-PEREZ, A.; FERNÁNDEZ-LÓPEZ, M. ; CORCHO, O.. **Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web**. Springer Science & Business Media, 2006.

[40] RECTOR, A.; DRUMMOND, N.; HORRIDGE, M.; ROGERS, J.; KNUBLAUCH, H.; STEVENS, R.; WANG, H. ; WROE, C.. **OWL pizzas: Practical experience of teaching owl-dl: Common errors & common patterns**. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT, p. 63–81. Springer, 2004.

[41] SMITH, B.. **Against idiosyncrasy in ontology development**. Frontiers in Artificial Intelligence and Applications, 150:15, 2006.

[42] BUNGE, M.. **Treatise on basic philosophy, volume 4, ontology ii: A world of systems**, 1982.

[43] VAN GIGCH, J. P.. **System Design Modeling and Metamodeling**. Springer Science & Business Media, 1991.

[44] ALBERTS, L. K.. **YMIR: a sharable ontology for the formal representation of engineering design knowledge**. Formal Design Methods for Computer-Aided Design, Elsevier/IFIP, 1994.

[45] BORST, W. N.. **Construction of engineering ontologies for knowledge sharing and reuse**. Universiteit Twente, 1997.

[46] BAYER, B.; MARQUARDT, W.. **Towards integrated information models for data and documents**. Computers & chemical engineering, 28(8):1249–1266, 2004.

[47] YANG, A.; BRAUNSCHWEIG, B.; FRAGA, E. S.; GUESSOUM, Z.; MARQUARDT, W.; NADJEMI, O.; PAEN, D.; PINOL, D.; ROUX, P. ; SAMA, S.. **A multi-agent system to facilitate component-based process modeling and design**. Computers & Chemical Engineering, 32(10):2290–2305, 2008.

[48] YANG, A.; MARQUARDT, W.. **An ontology-based approach to conceptual process modelling**. Computer Aided Chemical Engineering, 18:1159–1164, 2004.

[49] BRANDT, S. C.; MORBACH, J.; MIATIDIS, M.; THEISSEN, M.; JARKE, M. ; MARQUARDT, W.. **An ontology-based approach to knowledge management in design processes**. Computers & Chemical Engineering, 32(1):320–342, 2008.

[50] NATARAJAN, S.; GHOSH, K. ; SRINIVASAN, R.. **An ontology for distributed process supervision of large-scale chemical plants**. Computers & Chemical Engineering, 46:124–140, 2012.

[51] MARQUARDT, W.; MORBACH, J.; WIESNER, A. ; YANG, A.. **OntoCAPE: A re-usable ontology for chemical process engineering**. Springer Science & Business Media, 2009.

[52] ELHDAD, R.; CHILAMKURTI, N. ; TORABI, T.. **A novel design for the production process using ontology and business rules**. In: INDUSTRIAL ELECTRONICS AND APPLICATIONS (ICIEA), 2011 6TH IEEE CONFERENCE ON, p. 1525–1530. IEEE, 2011.

[53] ELHDAD, R.; CHILAMKURTI, N. ; TORABI, T.. **An ontology-based framework for process monitoring and maintenance in petroleum plant**. Journal of Loss Prevention in the Process Industries, 26(1):104–116, 2013.

[54] MOHAMMADFAM, I.; KALATPOUR, O.; GOLMOHAMMADI, R. ; KHOTANLOU, H.. **Developing a process equipment failure knowledge base using ontology approach for process equipment related incident investigations**. Journal of Loss Prevention in the Process Industries, 26(6):1300–1307, 2013.

[55] MATSOKIS, A.; KARRAY, H. M.; CHEBEL-MORELLO, B. ; KIRITSIS, D.. **An ontology-based model for providing semantic maintenance**. IFAC Proceedings Volumes, 43(3):12–17, 2010.

[56] JUN SUN, L.; LI, F. ; HU, X.. **An ontology-based model for typical-context awareness in the oil products distribution system**. Procedia Computer Science, 96:1156–1165, 2016.

[57] GROSMAN, J.. **LER: Anotação e classificação automática de entidades e relações**. Master's thesis, PUC-Rio, 2017.

[58] TALP RESEARCH CENTER. **Welcome | Freeling Homepage**. http://nlp.cs.upc.edu/freeling/node/1, 2017. [Acesso em 20 de Março de 2017].

[59] PADRÓ, L.; STANILOVSKY, E.. **Freeling 3.0: Towards wider multilinguality**. In: PROCEEDINGS OF THE LANGUAGE RESOURCES AND EVALUATION CONFERENCE (LREC 2012), Istanbul, Turkey, May 2012. ELRA.

[60] WIKIPEDIA. **Hyperparameter optimization — wikipedia, the free encyclopedia**. https://en.wikipedia.org/w/index.php?title=Hyperparameter_optimizationoldid=767641399, 2017. [Acesso em 26 de Março de 2017].

[61] WIKIPEDIA. **Cross-validation (statistics) — wikipedia, the free encyclopedia**. https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)oldid=772325199, 2017. [Acesso em 26 de Março de 2017].

[62] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS,

R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M. ; DUCHESNAY, E.. **Scikit-learn: Machine learning in Python**. Journal of Machine Learning Research, 12:2825–2830, 2011.

[63] BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B. ; VAROQUAUX, G.. **API design for machine learning software: experiences from the scikit-learn project**. In: ECML PKDD WORKSHOP: LANGUAGES FOR DATA MINING AND MACHINE LEARNING, p. 108–122, 2013.

[64] MÜLLER, A. C.; BEHNKE, S.. **pystruct - learning structured prediction in python**. Journal of Machine Learning Research, 15:2055–2060, 2014.

[65] VISHWANATHAN, S.; SCHRAUDOLPH, N. N.; SCHMIDT, M. W. ; MURPHY, K. P.. **Accelerated training of conditional random fields with stochastic gradient methods**. In: PROCEEDINGS OF THE 23RD INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 969–976. ACM, 2006.

[66] NOWOZIN, S.; LAMPERT, C. H.. **Structured learning and prediction in computer vision**. Foundations and Trends® in Computer Graphics and Vision, 6(3–4):185–365, 2011.

[67] OREDA. **Oreda -» about**. http://www.oreda.com/about-us/, 2016. [Acesso em 16 de Março de 2017].

[68] ISO, TC AND SC, N. **Petroleum, petrochemical and natural gas industriescollection and exchange of reliability and maintenance data for equipment**. 2006.

[69] MUSEN, M. A.. **The protégé project: A look back and a look forward**. AI matters, 1(4):4–12, 2015.

[70] , C.-C.; LIN, C.-J.. **LIBSVM: A library for support vector machines**. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[71] ZHANG, T.. **Solving large scale linear prediction problems using stochastic gradient descent algorithms**. In: PROCEEDINGS OF

THE TWENTY-FIRST INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 116. ACM, 2004.

[72] BOTTOU, L.. **Large-scale machine learning with stochastic gradient descent**. In: PROCEEDINGS OF COMPSTAT'2010, p. 177–186. Springer, 2010.

[73] BREIMAN, L.. **Random forests**. Machine learning, 45(1):5–32, 2001.

[74] LACOSTE-JULIEN, S.; JAGGI, M.; SCHMIDT, M. ; PLETSCHER, P.. **Block-coordinate frank-wolfe optimization for structural svms**. arXiv preprint arXiv:1207.4747, 2012.

[75] WIKIPEDIA. **Precision and recall — wikipedia, the free encyclopedia**. https://en.wikipedia.org/w/index.php?title=Precision_and_recalloldid=772411404, 2017. [Acesso em 31 de Março de 2017].

[76] MIWA, M.; BANSAL, M.. **End-to-end relation extraction using lstms on sequences and tree structures**. arXiv preprint arXiv:1601.00770, 2016.

[77] CHIU, J. P.; NICHOLS, E.. **Named entity recognition with bidirectional LSTM-CNNs**. arXiv preprint arXiv:1511.08308, 2015.

[78] CHEN, D.; MANNING, C. D.. **A fast and accurate dependency parser using neural networks.** In: EMNLP, p. 740–750, 2014.

A
Tabelas de resultados das tarefas de aprendizado de máquina

Este apêndice apresenta as tabelas detalhadas dos resultados discutidos no capítulo 7 desta dissertação.

A.1
Tarefa NER

As tabelas A.1, A.2, A.3 e A.4 apresentam, para cada grupo de atributos, os resultados médios para cada experimento inicial sobre o conjunto *TRAIN*, com validação cruzada de 3-*fold* e "teste" intermediário sobre o conjunto *VALIDATION*.

Tabela A.1: Resultados médios dos algoritmos para NER, sobre o pacote de atributos NER-SITOP-1.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
SVC	*Accuracy*	Treino (TRAIN)	0.997, 0.011	0.464, 0.332	0.468, 0.330	0.437, 0.310	432.8
		Teste (VALIDATION)	0.997, 0.010	0.521, 0.410	0.534, 0.403	0.505, 0.384	
	Recall	Treino (TRAIN)	0.997, 0.011	0.464, 0.332	0.468, 0.330	0.437, 0.310	432.8
		Teste (VALIDATION)	0.997, 0.010	0.521, 0.410	0.534, 0.403	0.505, 0.384	
	Precision	Treino (TRAIN)	0.997, 0.012	0.383, 0.330	0.494, 0.371	0.403, 0.327	185.5
		Teste (VALIDATION)	0.997, 0.010	0.486, 0.404	0.566, 0.434	0.500, 0.395	
	F1	Treino (TRAIN)	0.997, 0.011	0.464, 0.332	0.468, 0.330	0.437, 0.310	432.8
		Teste (VALIDATION)	0.997, 0.010	0.521, 0.410	0.534, 0.403	0.505, 0.384	
Linear SGDC	*Accuracy*	Treino (TRAIN)	0.997, 0.013	0.289, 0.296	0.332, 0.328	0.279, 0.281	104.2
		Teste (VALIDATION)	0.997, 0.011	0.387, 0.378	0.461, 0.427	0.395, 0.367	
	Recall	Treino (TRAIN)	0.996, 0.018	0.344, 0.311	0.330, 0.306	0.298, 0.279	51.7
		Teste (VALIDATION)	0.996, 0.016	0.419, 0.394	0.429, 0.414	0.389, 0.364	
	Precision	Treino (TRAIN)	0.997, 0.013	0.289, 0.296	0.332, 0.328	0.279, 0.281	104.2
		Teste (VALIDATION)	0.997, 0.011	0.387, 0.378	0.461, 0.427	0.395, 0.367	
	F1	Treino (TRAIN)	0.996, 0.018	0.344, 0.311	0.330, 0.306	0.298, 0.279	51.7
		Teste (VALIDATION)	0.996, 0.016	0.419, 0.394	0.429, 0.414	0.389, 0.364	
Random Forest	*Accuracy*	Treino (TRAIN)	0.997, 0.014	0.293, 0.285	0.394, 0.335	0.313, 0.287	66.8
		Teste (VALIDATION)	0.997, 0.011	0.410, 0.377	0.502, 0.429	0.433, 0.378	
	Recall	Treino (TRAIN)	0.997, 0.014	0.293, 0.285	0.394, 0.335	0.313, 0.287	66.8
		Teste (VALIDATION)	0.997, 0.011	0.410, 0.377	0.502, 0.429	0.433, 0.378	
	Precision	Treino (TRAIN)	0.997, 0.014	0.293, 0.285	0.394, 0.335	0.313, 0.287	66.8
		Teste (VALIDATION)	0.997, 0.011	0.410, 0.377	0.502, 0.429	0.433, 0.378	
	F1	Treino (TRAIN)	0.997, 0.014	0.293, 0.285	0.394, 0.335	0.313, 0.287	66.8
		Teste (VALIDATION)	0.997, 0.011	0.410, 0.377	0.502, 0.429	0.433, 0.378	

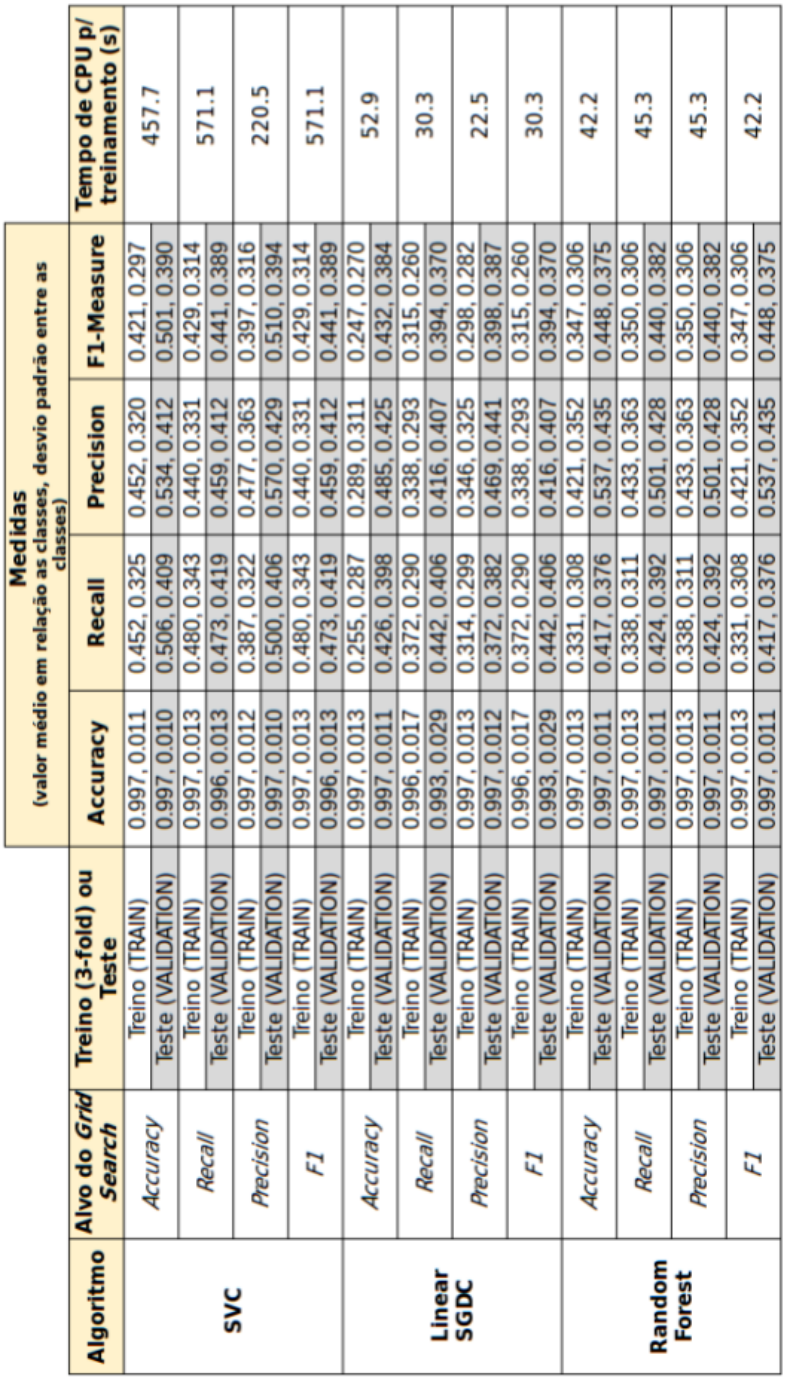
Tabela A.2: Resultados médios dos algoritmos para NER, sobre o pacote de atributos NER-SITOP-2, com destaque sobre o melhor valor de F1 entre todos os experimentos iniciais.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
SVC	*Accuracy*	Treino (TRAIN)	0.997, 0.012	0.419, 0.335	0.507, 0.359	0.429, 0.322	280.6
		Teste (VALIDATION)	0.997, 0.010	0.499, 0.408	0.571, 0.435	0.511, 0.399	
	Recall	Treino (TRAIN)	0.997, 0.011	0.492, 0.323	0.478, 0.320	0.453, 0.300	818.8
		Teste (VALIDATION)	0.997, 0.012	0.500, 0.413	0.506, 0.413	0.475, 0.384	
	Precision	Treino (TRAIN)	0.997, 0.012	0.419, 0.335	0.507, 0.359	0.429, 0.322	280.6
		Teste (VALIDATION)	0.997, 0.010	0.499, 0.408	0.571, 0.435	0.511, 0.399	
	F1	Treino (TRAIN)	0.997, 0.011	0.492, 0.323	0.478, 0.320	0.453, 0.300	818.8
		Teste (VALIDATION)	0.997, 0.012	0.500, 0.413	0.506, 0.413	0.475, 0.384	
Linear SGDC	*Accuracy*	Treino (TRAIN)	0.997, 0.012	0.319, 0.302	0.373, 0.325	0.312, 0.279	31.6
		Teste (VALIDATION)	0.997, 0.011	0.439, 0.407	0.468, 0.422	0.432, 0.388	
	Recall	Treino (TRAIN)	0.996, 0.016	0.381, 0.321	0.335, 0.304	0.313, 0.279	15.4
		Teste (VALIDATION)	0.995, 0.018	0.409, 0.390	0.407, 0.407	0.378, 0.363	
	Precision	Treino (TRAIN)	0.997, 0.012	0.319, 0.302	0.373, 0.325	0.312, 0.279	31.6
		Teste (VALIDATION)	0.997, 0.011	0.439, 0.407	0.468, 0.422	0.432, 0.388	
	F1	Treino (TRAIN)	0.996, 0.016	0.381, 0.321	0.335, 0.304	0.313, 0.279	15.4
		Teste (VALIDATION)	0.995, 0.018	0.409, 0.390	0.407, 0.407	0.378, 0.363	
Random Forest	*Accuracy*	Treino (TRAIN)	0.997, 0.013	0.332, 0.315	0.454, 0.376	0.354, 0.313	61.1
		Teste (VALIDATION)	0.997, 0.011	0.434, 0.390	0.534, 0.440	0.459, 0.391	
	Recall	Treino (TRAIN)	0.997, 0.014	0.333, 0.307	0.449, 0.364	0.344, 0.292	76.7
		Teste (VALIDATION)	0.997, 0.011	0.416, 0.382	0.515, 0.430	0.440, 0.378	
	Precision	Treino (TRAIN)	0.997, 0.013	0.332, 0.315	0.454, 0.376	0.354, 0.313	61.1
		Teste (VALIDATION)	0.997, 0.011	0.434, 0.390	0.534, 0.440	0.459, 0.391	
	F1	Treino (TRAIN)	0.997, 0.013	0.332, 0.315	0.454, 0.376	**0.354, 0.313**	61.1
		Teste (VALIDATION)	0.997, 0.011	0.434, 0.390	0.534, 0.440	**0.459, 0.391**	

Tabela A.3: Resultados médios dos algoritmos para NER, sobre o pacote de atributos NER-SITOP-3.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
SVC	*Accuracy*	Treino (TRAIN)	0.997, 0.011	0.440, 0.326	0.441, 0.323	0.414, 0.306	580.6
		Teste (VALIDATION)	0.997, 0.010	0.513, 0.413	0.524, 0.407	0.500, 0.389	
	Recall	Treino (TRAIN)	0.997, 0.012	0.452, 0.337	0.433, 0.333	0.417, 0.319	890.5
		Teste (VALIDATION)	0.997, 0.011	0.493, 0.411	0.497, 0.409	0.469, 0.381	
	Precision	Treino (TRAIN)	0.997, 0.012	0.376, 0.315	0.481, 0.349	0.394, 0.309	273.9
		Teste (VALIDATION)	0.997, 0.009	0.479, 0.400	0.544, 0.430	0.486, 0.390	
	F1	Treino (TRAIN)	0.997, 0.012	0.452, 0.337	0.433, 0.333	0.417, 0.319	890.5
		Teste (VALIDATION)	0.997, 0.011	0.493, 0.411	0.497, 0.409	0.469, 0.381	
Linear SGDC	*Accuracy*	Treino (TRAIN)	0.997, 0.013	0.251, 0.282	0.294, 0.299	0.246, 0.265	49.9
		Teste (VALIDATION)	0.997, 0.011	0.384, 0.387	0.425, 0.420	0.380, 0.370	
	Recall	Treino (TRAIN)	0.995, 0.018	0.320, 0.301	0.309, 0.284	0.278, 0.258	24.2
		Teste (VALIDATION)	0.994, 0.028	0.396, 0.398	0.381, 0.410	0.357, 0.369	
	Precision	Treino (TRAIN)	0.997, 0.013	0.251, 0.282	0.294, 0.299	0.246, 0.265	49.9
		Teste (VALIDATION)	0.997, 0.011	0.384, 0.387	0.425, 0.420	0.380, 0.370	
	F1	Treino (TRAIN)	0.995, 0.018	0.320, 0.301	0.309, 0.284	0.278, 0.258	24.2
		Teste (VALIDATION)	0.994, 0.028	0.396, 0.398	0.381, 0.410	0.357, 0.369	
Random Forest	*Accuracy*	Treino (TRAIN)	0.997, 0.014	0.312, 0.309	0.409, 0.367	0.327, 0.310	36.7
		Teste (VALIDATION)	0.997, 0.010	0.411, 0.388	0.499, 0.434	0.430, 0.384	
	Recall	Treino (TRAIN)	0.997, 0.014	0.312, 0.309	0.409, 0.367	0.327, 0.310	36.7
		Teste (VALIDATION)	0.997, 0.010	0.411, 0.388	0.499, 0.434	0.430, 0.384	
	Precision	Treino (TRAIN)	0.997, 0.014	0.312, 0.309	0.409, 0.367	0.327, 0.310	36.7
		Teste (VALIDATION)	0.997, 0.010	0.411, 0.388	0.499, 0.434	0.430, 0.384	
	F1	Treino (TRAIN)	0.997, 0.014	0.312, 0.309	0.409, 0.367	0.327, 0.310	36.7
		Teste (VALIDATION)	0.997, 0.010	0.411, 0.388	0.499, 0.434	0.430, 0.384	

Tabela A.4: Resultados médios dos algoritmos para NER, sobre o pacote de atributos NER-SITOP-4.



Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes) Accuracy	Recall	Precision	F1-Measure	Tempo de CPU p/ treinamento (s)
SVC	*Accuracy*	Treino (TRAIN)	0.997, 0.011	0.452, 0.325	0.452, 0.320	0.421, 0.297	457.7
		Teste (VALIDATION)	0.997, 0.010	0.506, 0.409	0.534, 0.412	0.501, 0.390	
	Recall	Treino (TRAIN)	0.997, 0.013	0.480, 0.343	0.440, 0.331	0.429, 0.314	571.1
		Teste (VALIDATION)	0.996, 0.013	0.473, 0.419	0.459, 0.412	0.441, 0.389	
	Precision	Treino (TRAIN)	0.997, 0.012	0.387, 0.322	0.477, 0.363	0.397, 0.316	220.5
		Teste (VALIDATION)	0.997, 0.010	0.500, 0.406	0.570, 0.429	0.510, 0.394	
	F1	Treino (TRAIN)	0.997, 0.013	0.480, 0.343	0.440, 0.331	0.429, 0.314	571.1
		Teste (VALIDATION)	0.996, 0.013	0.473, 0.419	0.459, 0.412	0.441, 0.389	
Linear SGDC	*Accuracy*	Treino (TRAIN)	0.997, 0.013	0.255, 0.287	0.289, 0.311	0.247, 0.270	52.9
		Teste (VALIDATION)	0.997, 0.011	0.426, 0.398	0.485, 0.425	0.432, 0.384	
	Recall	Treino (TRAIN)	0.996, 0.017	0.372, 0.290	0.338, 0.293	0.315, 0.260	30.3
		Teste (VALIDATION)	0.993, 0.029	0.442, 0.406	0.416, 0.407	0.394, 0.370	
	Precision	Treino (TRAIN)	0.997, 0.013	0.314, 0.299	0.346, 0.325	0.298, 0.282	22.5
		Teste (VALIDATION)	0.997, 0.012	0.372, 0.382	0.469, 0.441	0.398, 0.387	
	F1	Treino (TRAIN)	0.996, 0.017	0.372, 0.290	0.338, 0.293	0.315, 0.260	30.3
		Teste (VALIDATION)	0.993, 0.029	0.442, 0.406	0.416, 0.407	0.394, 0.370	
Random Forest	*Accuracy*	Treino (TRAIN)	0.997, 0.013	0.331, 0.308	0.421, 0.352	0.347, 0.306	42.2
		Teste (VALIDATION)	0.997, 0.011	0.417, 0.376	0.537, 0.435	0.448, 0.375	
	Recall	Treino (TRAIN)	0.997, 0.013	0.338, 0.311	0.433, 0.363	0.350, 0.306	45.3
		Teste (VALIDATION)	0.997, 0.011	0.424, 0.392	0.501, 0.428	0.440, 0.382	
	Precision	Treino (TRAIN)	0.997, 0.013	0.338, 0.311	0.433, 0.363	0.350, 0.306	45.3
		Teste (VALIDATION)	0.997, 0.011	0.424, 0.392	0.501, 0.428	0.440, 0.382	
	F1	Treino (TRAIN)	0.997, 0.013	0.331, 0.308	0.421, 0.352	0.347, 0.306	42.2
		Teste (VALIDATION)	0.997, 0.011	0.417, 0.376	0.537, 0.435	0.448, 0.375	

A tabela A.5 apresenta os resultados para o conjunto de treinamento (*TRAIN + VALIDATION*) com validação cruzada 5-*fold*.

Tabela A.5: Conjunto NER-SITOP-2 com Random Forest, *TRAIN+VALIDATION*. (Com problemas em alguns atributos)

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
O	8389	0.872, 0.011	0.962, 0.005	0.860, 0.021	0.908, 0.009
B-#Actuator	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Adjust	6	0.999, 0.000	0.125, 0.216	0.250, 0.433	0.166, 0.288
B-#BearingAndShaft	35	0.998, 0.001	0.554, 0.179	0.763, 0.136	0.633, 0.164

Continuação da Tabela A.5

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
B-#Blade	2	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#BlockagePlugged	21	0.998, 0.001	0.286, 0.230	0.600, 0.374	0.373, 0.276
B-#Breakage	18	0.999, 0.000	0.659, 0.377	0.777, 0.391	0.699, 0.367
B-#Burner	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Casing	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Centrifuge	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Check	20	0.998, 0.000	0.290, 0.190	0.546, 0.394	0.352, 0.220
B-#Chloride	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#ClearanceAlignment	4	0.999, 0.000	0.666, 0.471	0.666, 0.471	0.666, 0.471
B-#ColdStandBy	9	0.999, 0.000	0.333, 0.408	0.500, 0.500	0.375, 0.414
B-#Combination	31	0.996, 0.001	0.128, 0.112	0.290, 0.369	0.163, 0.152
B-#Compressor	42	0.996, 0.001	0.191, 0.194	0.294, 0.320	0.228, 0.236
B-#Contamination	25	0.999, 0.000	0.688, 0.275	0.777, 0.306	0.709, 0.274
B-#ControlFailure	8	0.999, 0.000	0.250, 0.433	0.250, 0.433	0.250, 0.433
B-#ControlUnit	11	0.999, 0.000	0.250, 0.250	0.500, 0.500	0.333, 0.333
B-#Cooler	12	0.999, 0.000	0.750, 0.276	1.000, 0.000	0.825, 0.204
B-#Corrosion	17	0.999, 0.000	0.587, 0.368	0.687, 0.409	0.630, 0.382
B-#Coupling	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Deformation	4	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#DesalinizerUnit	43	0.999, 0.000	0.866, 0.266	0.925, 0.106	0.857, 0.188
B-#Document	273	0.996, 0.000	0.905, 0.063	0.937, 0.046	0.918, 0.023
B-#DownState	250	0.996, 0.000	0.844, 0.050	0.955, 0.026	0.895, 0.021
B-#EarthIsolationFault	6	0.999, 0.000	0.385, 0.472	0.376, 0.462	0.380, 0.467
B-#ElectricGenerator	7	0.999, 0.000	0.125, 0.216	0.250, 0.433	0.166, 0.288
B-#ElectricMotor	3	0.999, 0.000	0.250, 0.000	1.000, 0.000	0.400, 0.000
B-#ElectricalGeneralFailure	3	0.999, 0.000	0.166, 0.166	0.500, 0.500	0.250, 0.250
B-#ElectricalOutlet	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#ElectrochlorinationUnit	4	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#ElectrostaticSeparator	18	0.999, 0.000	0.428, 0.440	0.500, 0.500	0.458, 0.462
B-#Fan	5	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#FaultySignal	6	0.999, 0.000	0.187, 0.324	0.250, 0.433	0.214, 0.371
B-#Filter	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Flange	4	0.999, 0.000	0.166, 0.235	0.333, 0.471	0.222, 0.314
B-#Flare	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#FloterUnit	8	0.999, 0.000	0.375, 0.414	0.500, 0.500	0.416, 0.433
B-#Flowline	2	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#Gas	4	0.999, 0.000	0.666, 0.471	0.666, 0.471	0.666, 0.471
B-#GasGenerator	2	0.999, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
B-#Gasket	11	0.999, 0.000	0.400, 0.374	0.433, 0.388	0.393, 0.334
B-#HeatExchanger	88	0.997, 0.001	0.352, 0.346	0.571, 0.469	0.413, 0.376
B-#HeaterOrBoiler	6	1.000, 0.000	0.923, 0.108	0.933, 0.094	0.928, 0.101
B-#HotStandBy	5	0.999, 0.000	0.288, 0.408	0.309, 0.437	0.298, 0.422
B-#Housing	9	0.999, 0.000	0.666, 0.408	0.750, 0.433	0.700, 0.412
B-#HydraulicPowerUnit	2	0.999, 7.686	0.833, 0.166	1.000, 0.000	0.900, 0.099
B-#Hydrocyclone	6	0.999, 0.000	0.234, 0.405	0.208, 0.360	0.220, 0.382
B-#Inspection	5	0.999, 0.000	0.750, 0.433	0.541, 0.360	0.616, 0.375
B-#InstrumentGeneralFailure	11	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Inverter	3	0.999, 0.000	0.250, 0.250	0.250, 0.250	0.250, 0.250
B-#Joint	7	0.999, 0.000	0.750, 0.250	0.625, 0.375	0.666, 0.333
B-#Leakage	156	0.996, 0.001	0.372, 0.458	0.366, 0.452	0.369, 0.455
B-#Looseness	5	0.999, 0.000	0.343, 0.343	0.392, 0.392	0.366, 0.366
B-#Lubricant	32	0.998, 0.000	0.393, 0.353	0.545, 0.448	0.442, 0.374
B-#Maintenance	75	0.998, 0.000	0.718, 0.174	0.866, 0.048	0.775, 0.120
B-#MaterialGeneralFailure	14	0.998, 0.000	0.465, 0.439	0.443, 0.372	0.431, 0.375
B-#MechanicalGeneralFailure	11	0.998, 0.000	0.515, 0.155	0.833, 0.235	0.611, 0.157
B-#Membrane	5	1.000, 0.000	0.312, 0.312	0.312, 0.312	0.312, 0.312
B-#Modify	2	0.999, 0.000	1.000, 0.000	0.750, 0.250	0.833, 0.166
B-#Motor	54	0.998, 0.001	0.590, 0.368	0.738, 0.377	0.630, 0.355
B-#NamedSystem	69	0.999, 0.000	0.701, 0.377	0.773, 0.390	0.732, 0.380
B-#NoSignal	5	0.999, 0.000	0.305, 0.432	0.318, 0.450	0.312, 0.441
B-#Nozzle	4	0.999, 0.000	0.833, 0.166	0.928, 0.071	0.875, 0.125
B-#Oil	9	0.999, 0.000	0.570, 0.360	0.722, 0.419	0.631, 0.378
B-#OperatingBadly	111	0.994, 0.001	0.683, 0.290	0.800, 0.244	0.673, 0.213
B-#OperatingNormally	13	0.999, 0.000	0.662, 0.211	0.885, 0.097	0.744, 0.169
B-#Other	3	0.999, 0.000	0.333, 0.000	0.500, 0.000	0.400, 0.000
B-#Overhaul	2	0.999, 0.000	0.681, 0.000	0.882, 0.000	0.769, 0.000
B-#Overheating	12	0.999, 0.000	0.466, 0.359	0.800, 0.400	0.551, 0.353
B-#Pipeline	20	0.998, 0.000	0.285, 0.234	0.450, 0.400	0.342, 0.284
B-#Piping	5	0.999, 0.000	0.112, 0.113	0.500, 0.500	0.183, 0.184
B-#Piston	5	0.999, 0.000	0.333, 0.408	0.375, 0.414	0.350, 0.409
B-#Plate	36	0.998, 0.001	0.200, 0.400	0.100, 0.200	0.133, 0.266
B-#ProductionPlant	1	1.000, 0.000	0.500, 0.000	0.500, 0.000	0.500, 0.000
B-#Pump	235	0.998, 0.000	0.559, 0.335	0.725, 0.390	0.626, 0.352
B-#Refit	12	0.999, 0.000	0.972, 0.024	0.910, 0.127	0.934, 0.072
B-#Repair	90	0.997, 0.001	0.466, 0.323	0.795, 0.397	0.564, 0.323
B-#Replace	67	0.996, 0.001	0.436, 0.356	0.513, 0.426	0.469, 0.384
B-#Reservoir	2	0.999, 7.686	0.291, 0.125	0.916, 0.083	0.420, 0.134
B-#Responsible	103	0.995, 0.001	0.365, 0.309	0.542, 0.449	0.435, 0.363
B-#Riser	2	1.000, 0.000	0.625, 0.041	0.752, 0.025	0.681, 0.014

Continuação da Tabela A.5

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
B-#RotorAndImpeler	5	0.999, 0.000	0.711, 0.204	0.875, 0.050	0.775, 0.139
B-#Seal	32	0.999, 0.000	0.647, 0.408	0.746, 0.387	0.661, 0.378
B-#Sensor	43	0.996, 0.002	0.754, 0.170	0.820, 0.201	0.778, 0.178
B-#Service	18	0.998, 0.001	0.339, 0.249	0.606, 0.328	0.425, 0.279
B-#Shell	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#ShortCircuiting	1	1.000, 0.000	0.571, 0.000	1.000, 0.000	0.727, 0.000
B-#Solids	8	0.998, 0.000	0.047, 0.067	0.166, 0.235	0.074, 0.104
B-#Stage	15	0.998, 0.000	0.300, 0.412	0.500, 0.500	0.333, 0.408
B-#Stator	10	1.000, 0.000	0.250, 0.353	0.250, 0.353	0.250, 0.353
B-#Stem	1	0.999, 0.000	0.833, 0.000	1.000, 0.000	0.909, 0.000
B-#Sticking	17	0.998, 0.000	0.511, 0.413	0.688, 0.406	0.559, 0.393
B-#StorageTank	12	0.998, 0.000	0.270, 0.180	0.666, 0.408	0.367, 0.220
B-#SubseaLine	8	0.998, 0.000	0.444, 0.415	0.444, 0.415	0.444, 0.415
B-#Test	11	0.999, 0.000	0.400, 0.489	0.240, 0.387	0.266, 0.388
B-#Transformer	10	0.999, 0.000	0.250, 0.433	0.083, 0.144	0.125, 0.216
B-#Tube	2	0.999, 0.000	0.750, 0.250	0.750, 0.250	0.750, 0.250
B-#Tubing	1	0.999, 0.000	0.400, 0.000	1.000, 0.000	0.571, 0.000
B-#Umbilical	10	0.998, 0.000	0.222, 0.314	0.222, 0.314	0.222, 0.314
B-#UninterruptiblePowerSupply	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#UnknownFailure	13	0.998, 0.000	0.428, 0.177	0.800, 0.244	0.497, 0.138
B-#UpState	1	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Valve	76	0.995, 0.002	0.100, 0.200	0.05, 0.100	0.066, 0.133
B-#VapourRecoveryUnit	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#VariableDrive	2	0.999, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
B-#Vessel	14	0.998, 0.000	0.468, 0.357	0.400, 0.234	0.407, 0.247
B-#Vibration	35	0.998, 0.000	0.509, 0.316	0.547, 0.322	0.525, 0.317
B-#Water	34	0.998, 0.000	0.305, 0.259	0.433, 0.416	0.348, 0.309
B-#Wear	6	0.999, 0.000	0.399, 0.432	0.500, 0.408	0.428, 0.420
B-#Well	63	0.999, 0.000	0.554, 0.264	0.858, 0.133	0.635, 0.203
B-#Wiring	1	1.000, 0.000	0.900, 0.000	1.000, 0.000	0.947, 0.000
B-#XmasTree	5	0.999, 0.000	0.333, 0.471	0.333, 0.471	0.333, 0.471
B-xsd:dateTime	109	0.999, 0.000	0.752, 0.244	0.831, 0.207	0.766, 0.199
B-xsd:string	79	0.998, 0.000	0.809, 0.242	0.833, 0.210	0.782, 0.189
I-#Actuator	5	0.998, 0.000	0.809, 0.000	1.000, 0.000	0.894, 0.000
I-#Adjust	3	0.999, 0.000	0.375, 0.375	0.500, 0.500	0.428, 0.428
I-#BearingAndShaft	45	0.998, 0.000	0.743, 0.308	0.888, 0.149	0.762, 0.257
I-#BlockagePlugged	4	0.999, 0.000	0.812, 0.187	0.879, 0.046	0.837, 0.123
I-#Centrifuge	1	1.000, 0.000	0.533, 0.000	1.000, 0.000	0.695, 0.000
I-#Check	13	0.999, 0.000	0.171, 0.342	0.171, 0.342	0.171, 0.342
I-#ClearanceAlignment	3	0.999, 0.000	0.742, 0.075	1.000, 0.000	0.850, 0.049
I-#ColdStandBy	15	0.998, 0.001	0.955, 0.076	1.000, 0.000	0.975, 0.041
I-#Combination	7	0.999, 0.000	0.400, 0.489	0.384, 0.471	0.392, 0.480
I-#Compressor	26	0.997, 0.001	0.500, 0.500	0.291, 0.414	0.321, 0.408
I-#Contamination	14	0.998, 0.000	0.666, 0.380	0.633, 0.371	0.618, 0.335
I-#ControlFailure	10	0.998, 0.000	0.333, 0.408	0.375, 0.414	0.291, 0.297
I-#ControlUnit	20	0.998, 0.000	0.050, 0.086	0.250, 0.433	0.083, 0.144
I-#Corrosion	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#DesalinizerUnit	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Document	156	0.997, 0.001	0.266, 0.388	0.400, 0.489	0.300, 0.399
I-#DownState	58	0.998, 0.001	0.500, 0.447	0.461, 0.409	0.445, 0.372
I-#EarthIsolationFault	8	0.999, 0.000	0.278, 0.269	0.577, 0.473	0.360, 0.313
I-#ElectricMotor	3	0.999, 0.000	0.375, 0.000	1.000, 0.000	0.545, 0.000
I-#ElectricalGeneralFailure	1	1.000, 0.000	0.562, 0.000	1.000, 0.000	0.720, 0.000
I-#ElectrostaticSeparator	7	0.997, 0.000	1.000, 0.000	0.968, 0.000	0.984, 0.000
I-#FaultySignal	9	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Filter	3	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Flowline	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#GasGenerator	11	0.995, 0.000	0.875, 0.000	1.000, 0.000	0.933, 0.000
I-#Gasket	2	0.999, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
I-#HeatExchanger	5	0.998, 0.000	0.350, 0.150	1.000, 0.000	0.500, 0.166
I-#HotStandBy	12	0.998, 0.000	0.291, 0.412	0.291, 0.412	0.291, 0.412
I-#InstrumentGeneralFailure	3	0.998, 0.000	0.666, 0.000	1.000, 0.000	0.800, 0.000
I-#Inverter	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Leakage	67	0.997, 0.001	0.722, 0.372	0.791, 0.396	0.753, 0.381
I-#Looseness	8	0.998, 0.001	0.383, 0.241	1.000, 0.000	0.509, 0.259
I-#Lubricant	16	0.999, 0.000	0.429, 0.412	0.500, 0.447	0.412, 0.359
I-#Maintenance	9	0.999, 0.000	0.187, 0.324	0.250, 0.433	0.214, 0.371
I-#MechanicalGeneralFailure	9	0.999, 0.000	0.603, 0.286	0.904, 0.134	0.690, 0.220
I-#Motor	10	0.999, 0.000	0.216, 0.272	0.375, 0.414	0.226, 0.241
I-#NoSignal	10	0.999, 0.000	0.444, 0.415	0.333, 0.235	0.355, 0.273
I-#Nozzle	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#OperatingBadly	187	0.991, 0.003	0.300, 0.266	0.496, 0.419	0.357, 0.296
I-#OperatingNormally	5	0.999, 0.000	0.239, 0.338	0.247, 0.349	0.243, 0.344
I-#Overheating	6	0.999, 0.000	0.483, 0.366	0.687, 0.409	0.526, 0.330
I-#Pipeline	32	0.996, 0.001	0.250, 0.433	0.250, 0.433	0.250, 0.433
I-#Piping	12	0.999, 0.000	0.500, 0.372	0.625, 0.414	0.491, 0.303
I-#Piston	6	0.999, 0.000	0.722, 0.055	0.804, 0.104	0.753, 0.016
I-#Plate	6	0.998, 0.000	0.702, 0.297	0.731, 0.131	0.650, 0.099
I-#Pump	119	0.992, 0.002	0.375, 0.353	0.800, 0.400	0.457, 0.348

Continuação da Tabela A.5					
Tag	**Sup.**	**Accuracy, SD**	**Recall, SD**	**Precision, SD**	**F-score, SD**
I-#Refit	5	0.999, 0.000	0.280, 0.280	0.250, 0.250	0.264, 0.264
I-#Repair	7	0.999, 1.537	0.458, 0.462	0.375, 0.379	0.408, 0.408
I-#Replace	10	0.998, 0.000	0.595, 0.299	0.818, 0.129	0.643, 0.184
I-#Responsible	52	0.997, 0.001	0.166, 0.210	0.300, 0.399	0.200, 0.244
I-#Seal	16	0.999, 0.000	0.357, 0.312	0.437, 0.424	0.360, 0.318
I-#Sensor	46	0.997, 0.002	0.345, 0.431	0.360, 0.446	0.352, 0.438
I-#Service	13	0.998, 0.000	0.275, 0.277	0.500, 0.500	0.354, 0.355
I-#Shell	3	0.999, 0.000	0.142, 0.000	1.000, 0.000	0.250, 0.000
I-#ShortCircuiting	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Solids	4	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
I-#Stage	2	0.999, 0.000	0.375, 0.375	0.333, 0.333	0.352, 0.352
I-#Stator	10	1.000, 0.000	0.333, 0.471	0.333, 0.471	0.333, 0.471
I-#Sticking	2	0.999, 0.000	0.400, 0.000	1.000, 0.000	0.571, 0.000
I-#StorageTank	10	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#SubseaLine	11	0.997, 0.001	0.928, 0.071	0.928, 0.071	0.928, 0.071
I-#Test	11	0.999, 0.000	0.511, 0.378	0.583, 0.381	0.483, 0.295
I-#Transformer	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Tube	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Umbilical	9	0.998, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
I-#UnknownFailure	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Valve	50	0.996, 0.001	0.366, 0.371	0.485, 0.448	0.408, 0.387
I-#VariableDrive	4	0.998, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
I-#Vessel	11	0.997, 0.001	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Vibration	18	0.999, 0.000	0.464, 0.467	0.500, 0.500	0.480, 0.481
I-#Water	24	0.998, 0.000	0.247, 0.258	0.500, 0.500	0.327, 0.334
I-#Wear	3	0.999, 0.000	0.625, 0.000	0.576, 0.000	0.600, 0.000
I-#Well	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Wiring	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-xsd:string	190	0.995, 0.001	0.200, 0.400	0.200, 0.400	0.200, 0.400
Average, SD		**0.998, 0.009**	**0.436, 0.287**	**0.552, 0.334**	**0.463, 0.292**

A tabela A.6 apresenta os resultados do experimento anterior repetido, com correções nos atributos.

Tabela A.6: Conjunto NER-SITOP-2 com Random Forest, *TRAIN+VALIDATION*. (Problemas de atributos corrigidos.)

Tag	**Sup.**	**Accuracy, SD**	**Recall, SD**	**Precision, SD**	**F-score, SD**
O	8389	0.876, 0.010	0.965, 0.005	0.862, 0.019	0.910, 0.008
B-#Actuator	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Adjust	6	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#BearingAndShaft	35	0.998, 0.001	0.554, 0.179	0.830, 0.153	0.652, 0.161
B-#Blade	2	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#BlockagePlugged	21	0.998, 0.001	0.306, 0.217	0.733, 0.388	0.421, 0.263
B-#Breakage	18	0.999, 0.000	0.860, 0.195	0.977, 0.044	0.899, 0.123
B-#Burner	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Casing	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Centrifuge	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Check	20	0.998, 0.000	0.250, 0.223	0.533, 0.452	0.327, 0.280
B-#Chloride	1	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#ClearanceAlignment	4	0.999, 0.000	0.666, 0.471	0.555, 0.415	0.6, 0.432
B-#ColdStandBy	9	0.999, 0.000	0.333, 0.408	0.500, 0.500	0.375, 0.414
B-#Combination	31	0.997, 0.001	0.178, 0.186	0.206, 0.193	0.190, 0.188
B-#Compressor	42	0.998, 0.000	0.681, 0.325	0.793, 0.244	0.711, 0.288
B-#Contamination	25	0.999, 0.000	0.722, 0.238	0.950, 0.099	0.803, 0.175
B-#ControlFailure	8	0.999, 0.000	0.250, 0.433	0.250, 0.433	0.25, 0.433
B-#ControlUnit	11	0.999, 0.000	0.250, 0.250	0.500, 0.500	0.333, 0.333
B-#Cooler	12	1.000, 0.000	0.916, 0.144	1.000, 0.000	0.95, 0.086
B-#Corrosion	17	0.999, 0.000	0.587, 0.368	0.687, 0.409	0.630, 0.382
B-#Coupling	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Deformation	4	1.000, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#DesalinizerUnit	43	0.999, 0.000	0.866, 0.266	0.963, 0.072	0.880, 0.193
B-#Document	273	0.997, 0.000	0.918, 0.067	0.951, 0.040	0.932, 0.028
B-#DownState	250	0.995, 0.000	0.831, 0.094	0.955, 0.025	0.885, 0.050
B-#EarthIsolationFault	6	0.999, 0.000	0.375, 0.460	0.378, 0.464	0.376, 0.462
B-#ElectricGenerator	7	0.999, 0.000	0.125, 0.216	0.250, 0.433	0.166, 0.288
B-#ElectricMotor	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#ElectricalGeneralFailure	3	0.999, 0.000	0.166, 0.166	0.500, 0.500	0.250, 0.250
B-#ElectricalOutlet	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#ElectrochlorinationUnit	4	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#ElectrostaticSeparator	18	0.999, 0.000	0.351, 0.363	0.500, 0.500	0.409, 0.414
B-#Fan	5	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#FaultySignal	6	0.999, 0.000	0.187, 0.324	0.250, 0.433	0.214, 0.371
B-#Filter	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Flange	4	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Flare	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000

Continuação da Tabela A.6

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
B-#FloterUnit	8	0.999, 0.000	0.375, 0.414	0.500, 0.500	0.416, 0.433
B-#Flowline	2	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#Gas	4	1.000, 0.000	0.666, 0.471	0.666, 0.471	0.666, 0.471
B-#GasGenerator	2	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Gasket	11	0.999, 0.000	0.400, 0.374	0.433, 0.388	0.393, 0.334
B-#HeatExchanger	88	0.997, 0.001	0.352, 0.346	0.600, 0.489	0.423, 0.385
B-#HeaterOrBoiler	6	1.000, 0.000	0.897, 0.145	0.939, 0.085	0.916, 0.117
B-#HotStandBy	5	0.999, 0.000	0.311, 0.439	0.311, 0.439	0.311, 0.439
B-#Housing	9	0.999, 0.000	0.666, 0.408	0.750, 0.433	0.700, 0.412
B-#HydraulicPowerUnit	2	0.999, 7.686	0.833, 0.166	1.000, 0.000	0.900, 0.099
B-#Hydrocyclone	6	0.999, 0.000	0.484, 0.484	0.447, 0.453	0.464, 0.467
B-#Inspection	5	0.999, 0.000	0.750, 0.433	0.541, 0.360	0.616, 0.375
B-#InstrumentGeneralFailure	11	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Inverter	3	0.999, 0.000	0.250, 0.250	0.500, 0.500	0.333, 0.333
B-#Joint	7	0.999, 0.000	0.750, 0.250	0.625, 0.375	0.666, 0.333
B-#Leakage	156	0.996, 0.001	0.165, 0.331	0.171, 0.342	0.168, 0.336
B-#Looseness	5	0.999, 0.000	0.625, 0.125	0.928, 0.071	0.733, 0.066
B-#Lubricant	32	0.998, 0.000	0.418, 0.378	0.528, 0.439	0.447, 0.377
B-#Maintenance	75	0.998, 0.000	0.718, 0.174	0.879, 0.052	0.779, 0.117
B-#MaterialGeneralFailure	14	0.999, 0.000	0.490, 0.407	0.627, 0.344	0.484, 0.318
B-#MechanicalGeneralFailure	11	0.999, 0.000	0.468, 0.099	0.799, 0.282	0.557, 0.122
B-#Membrane	5	1.000, 0.000	0.479, 0.145	0.916, 0.083	0.607, 0.107
B-#Modify	2	0.999, 0.000	1.000, 0.000	0.750, 0.250	0.833, 0.166
B-#Motor	54	0.998, 0.001	0.698, 0.246	0.955, 0.088	0.776, 0.181
B-#NamedSystem	69	0.999, 0.000	0.920, 0.111	0.966, 0.066	0.942, 0.089
B-#NoSignal	5	0.999, 0.000	0.305, 0.432	0.318, 0.450	0.312, 0.441
B-#Nozzle	4	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#Oil	9	0.999, 0.000	0.534, 0.311	0.629, 0.385	0.572, 0.335
B-#OperatingBadly	111	0.994, 0.001	0.750, 0.316	0.800, 0.244	0.713, 0.249
B-#OperatingNormally	13	0.999, 0.000	0.595, 0.154	0.890, 0.092	0.703, 0.142
B-#Other	3	0.999, 0.000	0.333, 0.000	0.500, 0.000	0.400, 0.000
B-#Overhaul	2	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
B-#Overheating	12	0.999, 0.000	0.453, 0.349	0.766, 0.388	0.530, 0.337
B-#Pipeline	20	0.998, 0.000	0.235, 0.208	0.538, 0.453	0.319, 0.273
B-#Piping	5	0.999, 0.000	0.354, 0.383	0.583, 0.433	0.405, 0.371
B-#Piston	5	0.999, 0.000	0.375, 0.414	0.500, 0.500	0.416, 0.433
B-#Plate	36	0.998, 0.001	0.240, 0.387	0.400, 0.489	0.266, 0.388
B-#ProductionPlant	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Pump	235	0.998, 0.000	0.333, 0.421	0.400, 0.489	0.360, 0.445
B-#Refit	12	0.998, 0.000	0.798, 0.216	0.782, 0.232	0.762, 0.177
B-#Repair	90	0.997, 0.001	0.437, 0.302	0.698, 0.386	0.516, 0.306
B-#Replace	67	0.996, 0.001	0.598, 0.339	0.647, 0.341	0.619, 0.338
B-#Reservoir	2	0.999, 7.686	0.383, 0.216	0.875, 0.125	0.476, 0.190
B-#Responsible	103	0.996, 0.002	0.525, 0.313	0.685, 0.364	0.589, 0.326
B-#Riser	2	1.000, 0.000	0.375, 0.375	0.409, 0.409	0.391, 0.391
B-#RotorAndImpeler	5	0.999, 0.000	0.590, 0.428	0.590, 0.418	0.589, 0.421
B-#Seal	32	0.999, 0.000	0.625, 0.351	0.649, 0.341	0.629, 0.336
B-#Sensor	43	0.996, 0.002	0.673, 0.203	0.853, 0.188	0.723, 0.175
B-#Service	18	0.998, 0.001	0.464, 0.283	0.809, 0.156	0.566, 0.243
B-#Shell	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#ShortCircuiting	1	1.000, 0.000	0.714, 0.000	1.000, 0.000	0.833, 0.000
B-#Solids	8	0.998, 0.001	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Stage	15	0.998, 0.000	0.535, 0.467	0.625, 0.414	0.555, 0.451
B-#Stator	10	1.000, 0.000	0.250, 0.353	0.333, 0.471	0.285, 0.404
B-#Stem	1	0.999, 0.000	0.833, 0.000	1.000, 0.000	0.909, 0.000
B-#Sticking	17	0.999, 0.000	0.434, 0.376	0.638, 0.379	0.475, 0.354
B-#StorageTank	12	0.998, 0.000	0.437, 0.369	0.666, 0.408	0.492, 0.358
B-#SubseaLine	8	0.999, 0.000	0.444, 0.415	0.666, 0.471	0.500, 0.408
B-#Test	11	0.999, 0.000	0.306, 0.369	0.366, 0.371	0.266, 0.249
B-#Transformer	10	0.999, 0.000	0.375, 0.414	0.333, 0.408	0.291, 0.297
B-#Tube	2	0.999, 0.000	0.166, 0.166	0.500, 0.500	0.250, 0.250
B-#Tubing	1	0.999, 0.000	0.250, 0.000	1.000, 0.000	0.400, 0.000
B-#Umbilical	10	0.999, 0.000	0.355, 0.273	0.666, 0.471	0.457, 0.336
B-#UninterruptiblePowerSupply	4	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
B-#UnknownFailure	13	0.998, 0.000	0.447, 0.392	0.483, 0.409	0.450, 0.372
B-#UpState	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
B-#Valve	76	0.995, 0.001	0.200, 0.244	0.250, 0.387	0.200, 0.266
B-#VapourRecoveryUnit	1	1.000, 0.000	0.500, 0.000	1.000, 0.000	0.666, 0.000
B-#VariableDrive	2	0.999, 0.000	0.375, 0.000	0.500, 0.000	0.428, 0.000
B-#Vessel	14	0.998, 0.000	0.500, 0.500	0.375, 0.414	0.416, 0.433
B-#Vibration	35	0.998, 0.000	0.385, 0.203	0.552, 0.325	0.446, 0.237
B-#Water	34	0.999, 0.000	0.246, 0.208	0.480, 0.448	0.313, 0.269
B-#Wear	6	0.998, 0.000	0.238, 0.336	0.333, 0.471	0.277, 0.392
B-#Well	63	0.999, 0.000	0.594, 0.337	0.643, 0.332	0.611, 0.328
B-#Wiring	1	1.000, 0.000	1.000, 0.000	0.800, 0.000	0.888, 0.000
B-#XmasTree	5	0.999, 0.000	0.366, 0.385	0.666, 0.471	0.426, 0.392
B-xsd:dateTime	109	0.999, 0.000	0.905, 0.189	0.820, 0.151	0.845, 0.141
B-xsd:string	79	0.999, 0.000	0.742, 0.387	0.725, 0.390	0.733, 0.388
I-#Actuator	5	0.998, 0.000	0.920, 0.000	1.000, 0.000	0.958, 0.000

Continuação da Tabela A.6

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
I-#Adjust	3	0.999, 0.000	0.615, 0.329	0.805, 0.138	0.672, 0.272
I-#BearingAndShaft	45	0.998, 0.000	0.803, 0.160	0.874, 0.189	0.827, 0.156
I-#BlockagePlugged	4	0.999, 0.000	0.312, 0.312	0.500, 0.500	0.384, 0.384
I-#Centrifuge	1	1.000, 0.000	0.533, 0.000	1.000, 0.000	0.695, 0.000
I-#Check	13	0.999, 0.000	0.203, 0.316	0.400, 0.489	0.246, 0.351
I-#ClearanceAlignment	3	1.000, 0.000	0.428, 0.428	0.428, 0.428	0.428, 0.428
I-#ColdStandBy	15	0.998, 0.001	0.916, 0.144	0.916, 0.144	0.916, 0.144
I-#Combination	7	0.999, 0.000	0.400, 0.489	0.384, 0.471	0.392, 0.480
I-#Compressor	26	0.998, 0.000	0.691, 0.410	0.583, 0.433	0.591, 0.387
I-#Contamination	14	0.998, 0.000	0.600, 0.489	0.600, 0.489	0.600, 0.489
I-#ControlFailure	10	0.998, 0.000	0.600, 0.424	0.615, 0.417	0.550, 0.357
I-#ControlUnit	20	0.998, 0.000	0.333, 0.408	0.500, 0.500	0.375, 0.414
I-#Corrosion	1	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#DesalinizerUnit	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Document	156	0.997, 0.001	0.200, 0.400	0.100, 0.200	0.133, 0.266
I-#DownState	58	0.998, 0.001	0.299, 0.163	0.800, 0.400	0.433, 0.226
I-#EarthIsolationFault	8	0.999, 0.000	0.550, 0.458	0.556, 0.455	0.551, 0.452
I-#ElectricMotor	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#ElectricalGeneralFailure	1	1.000, 0.000	0.937, 0.000	0.937, 0.000	0.937, 0.000
I-#ElectrostaticSeparator	7	0.997, 0.000	0.875, 0.000	1.000, 0.000	0.933, 0.000
I-#FaultySignal	9	0.998, 0.000	0.534, 0.145	0.962, 0.052	0.670, 0.103
I-#Filter	3	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Flowline	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#GasGenerator	11	0.996, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Gasket	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#HeatExchanger	5	0.998, 0.000	0.416, 0.083	1.000, 0.000	0.583, 0.083
I-#HotStandBy	12	0.998, 0.000	0.666, 0.471	0.500, 0.408	0.555, 0.415
I-#InstrumentGeneralFailure	3	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Inverter	3	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Leakage	67	0.997, 0.001	0.341, 0.240	0.800, 0.400	0.457, 0.292
I-#Looseness	8	0.998, 0.001	0.327, 0.327	0.475, 0.475	0.387, 0.387
I-#Lubricant	16	0.999, 0.000	0.125, 0.250	0.200, 0.400	0.153, 0.307
I-#Maintenance	9	0.999, 0.000	0.398, 0.359	0.619, 0.409	0.410, 0.351
I-#MechanicalGeneralFailure	9	0.999, 0.000	0.166, 0.235	0.111, 0.157	0.133, 0.188
I-#Motor	10	0.999, 7.686	0.353, 0.293	0.750, 0.433	0.454, 0.318
I-#NoSignal	10	0.999, 0.000	0.388, 0.283	0.555, 0.415	0.444, 0.314
I-#Nozzle	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#OperatingBadly	187	0.991, 0.004	0.304, 0.402	0.337, 0.425	0.318, 0.411
I-#OperatingNormally	5	0.999, 0.000	0.444, 0.257	0.833, 0.235	0.540, 0.247
I-#Overheating	6	0.999, 0.000	0.625, 0.414	0.579, 0.379	0.571, 0.361
I-#Pipeline	32	0.996, 0.001	0.083, 0.144	0.250, 0.433	0.125, 0.216
I-#Piping	12	0.999, 0.000	0.175, 0.303	0.228, 0.395	0.198, 0.343
I-#Piston	6	0.999, 0.000	0.500, 0.500	0.500, 0.500	0.500, 0.500
I-#Plate	6	0.998, 0.000	0.666, 0.333	0.666, 0.333	0.500, 0.000
I-#Pump	119	0.992, 0.002	0.480, 0.265	0.667, 0.350	0.548, 0.288
I-#Refit	5	0.999, 1.537	0.166, 0.166	0.500, 0.500	0.250, 0.250
I-#Repair	7	0.999, 0.000	0.416, 0.433	0.500, 0.500	0.450, 0.455
I-#Replace	10	0.998, 0.001	0.722, 0.207	0.866, 0.188	0.738, 0.054
I-#Responsible	52	0.997, 0.001	0.368, 0.315	0.627, 0.404	0.373, 0.274
I-#Seal	16	0.999, 0.000	0.119, 0.147	0.329, 0.418	0.174, 0.217
I-#Sensor	46	0.997, 0.002	0.204, 0.249	0.292, 0.396	0.231, 0.288
I-#Service	13	0.998, 0.000	0.181, 0.314	0.201, 0.348	0.190, 0.330
I-#Shell	3	0.999, 0.000	0.750, 0.000	0.666, 0.000	0.705, 0.000
I-#ShortCircuiting	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Solids	4	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Stage	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Stator	10	0.999, 0.000	0.383, 0.306	0.643, 0.456	0.467, 0.347
I-#Sticking	2	0.999, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#StorageTank	10	0.998, 0.000	0.366, 0.033	1.000, 0.000	0.535, 0.035
I-#SubseaLine	11	0.997, 0.001	1.000, 0.000	1.000, 0.000	1.000, 0.000
I-#Test	11	0.999, 0.000	0.358, 0.227	0.666, 0.408	0.455, 0.278
I-#Transformer	4	0.999, 0.000	0.357, 0.357	0.454, 0.454	0.400, 0.400
I-#Tube	3	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
I-#Umbilical	9	0.998, 0.000	0.357, 0.357	0.500, 0.500	0.416, 0.416
I-#UnknownFailure	4	0.999, 0.000	0.333, 0.471	0.333, 0.471	0.333, 0.471
I-#Valve	50	0.996, 0.001	0.300, 0.399	0.350, 0.435	0.304, 0.378
I-#VariableDrive	4	0.998, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Vessel	11	0.997, 0.001	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-#Vibration	18	0.999, 0.000	0.250, 0.433	0.125, 0.216	0.166, 0.288
I-#Water	24	0.998, 0.000	0.214, 0.371	0.250, 0.433	0.230, 0.399
I-#Wear	3	0.999, 0.000	0.600, 0.000	1.000, 0.000	0.749, 0.000
I-#Well	2	0.999, 0.000	1.000, 0.000	1.000, 0.000	1.000, 0.000
I-#Wiring	1	1.000, 0.000	0.000, 0.000	0.000, 0.000	0.000, 0.000
I-xsd:string	190	0.995, 0.001	0.269, 0.240	0.415, 0.379	0.307, 0.256
Average, SD		**0.998, 0.008**	**0.427, 0.299**	**0.531, 0.329**	**0.451, 0.299**

A tabela A.7 apresenta os resultados de teste único e definitivo, sobre os

dados *TEST*, do conjunto de atributos, algoritmo e parâmetros escolhido nas etapas anteriores.

Tabela A.7: Conjunto NER-SITOP-2 com Random Forest, *TEST*. (Problemas de atributos corrigidos.)

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
O	2236	0.891	0.961	0.887	0.923
B-#Adjust	1.000	1.000	0.000	0.000	0.000
B-#BearingAndShaft	5	0.999	1.000	1.000	1.000
B-#BlockagePlugged	5	0.998	0.600	1.000	0.749
B-#Breakage	4	0.999	0.200	1.000	0.333
B-#Centrifuge	3	0.999	0.750	0.750	0.750
B-#Check	4	0.999	0.333	0.500	0.400
B-#ColdStandBy	1.000	1.000	0.500	1.000	0.666
B-#Combination	6	0.998	1.000	1.000	1.000
B-#Compressor	10	0.998	0.166	0.500	0.250
B-#Contamination	5	0.999	0.700	0.700	0.700
B-#ControlFailure	2	0.999	0.400	1.000	0.571
B-#ControlUnit	2	0.999	0.500	1.000	0.666
B-#Corrosion	8	0.999	0.500	1.000	0.666
B-#Coupling	2	0.999	0.750	1.000	0.857
B-#DesalinizerUnit	7	0.999	0.000	0.000	0.000
B-#Document	70	0.996	0.857	0.857	0.857
B-#DownState	72	0.995	0.900	0.954	0.926
B-#ElectricGenerator	2	1.000	0.847	0.953	0.897
B-#ElectricalGeneralFailure	2	0.999	1.000	1.000	1.000
B-#ElectricalOutlet	1.000	0.999	0.000	0.000	0.000
B-#ElectrostaticSeparator	3	0.999	0.000	0.000	0.000
B-#Fan	2	1.000	0.666	1.000	0.800
B-#FaultySignal	2	0.999	1.000	1.000	1.000
B-#Flange	1.000	1.000	0.500	1.000	0.666
B-#FloterUnit	1.000	0.999	1.000	1.000	1.000
B-#Gas	1.000	1.000	0.000	0.000	0.000
B-#GasGenerator	1.000	0.999	1.000	1.000	1.000
B-#Gasket	3	0.999	0.000	0.000	0.000
B-#HeatExchanger	22	0.999	0.333	1.000	0.500
B-#Housing	3	0.999	0.909	0.952	0.930
B-#Hydrocyclone	3	0.999	0.666	1.000	0.800
B-#Inspection	3	0.999	0.333	1.000	0.500
B-#InstrumentGeneralFailure	2	0.999	0.000	0.000	0.000
B-#Inverter	1.000	1.000	0.000	0.000	0.000
B-#Leakage	43	0.995	1.000	1.000	1.000
B-#Lubricant	10	0.998	0.790	0.829	0.809
B-#Maintenance	14	0.997	0.000	0.000	0.000
B-#MaterialGeneralFailure	3	0.999	0.600	0.750	0.666
B-#MechanicalGeneralFailure	2	1.000	0.857	0.705	0.774
B-#Membrane	1.000	1.000	0.666	1.000	0.800
B-#Motor	14	0.997	1.000	1.000	1.000
B-#NamedSystem	22	0.998	1.000	1.000	1.000
B-#NoSignal	1.000	0.999	0.785	0.687	0.733
B-#Nozzle	1.000	1.000	0.909	0.909	0.909
B-#Oil	1.000	0.999	1.000	0.500	0.666
B-#OperatingBadly	22	0.996	1.000	1.000	1.000
B-#OperatingNormally	4	0.999	0.000	0.000	0.000
B-#Overheating	4	0.999	0.681	0.833	0.749
B-#Pipeline	7	0.996	0.750	1.000	0.857
B-#Piping	1.000	0.999	0.500	1.000	0.666
B-#Piston	1.000	0.999	0.000	0.000	0.000
B-#Plate	7	0.997	0.000	0.000	0.000
B-#ProductionPlant	1.000	0.999	0.000	0.000	0.000
B-#Pump	57	0.997	0.285	0.500	0.363
B-#Refit	3	0.999	0.000	0.000	0.000
B-#Repair	20	0.998	0.912	0.962	0.936
B-#Replace	23	0.996	0.333	0.500	0.400
B-#Responsible	28	0.995	0.800	0.888	0.842
B-#RotorAndImpeler	1.000	0.999	0.521	0.857	0.648
B-#Seal	14	0.998	0.678	0.730	0.703
B-#Sensor	9	0.997	0.000	0.000	0.000
B-#Service	5	0.998	0.714	1.000	0.833
B-#Stage	3	0.999	0.555	0.625	0.588
B-#Stator	6	1.000	0.000	0.000	0.000
B-#Sticking	4	0.998	0.666	1.000	0.800
B-#StorageTank	5	0.998	1.000	1.000	1.000
B-#SubseaLine	2	0.999	0.500	0.400	0.444
B-#Test	5	0.998	0.200	0.500	0.285
B-#Transformer	1.000	0.999	0.500	1.000	0.666
B-#Tube	1.000	0.999	0.200	1.000	0.333
B-#Umbilical	1.000	1.000	0.000	0.000	0.000

Continuação da Tabela A.7

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
B-#UninterruptiblePowerSupply	1.000	1.000	0.000	0.000	0.000
B-#UnknownFailure	3	0.999	0.000	0.000	0.000
B-#Valve	26	0.997	1.000	1.000	1.000
B-#Vessel	6	0.998	1.000	1.000	1.000
B-#Vibration	6	0.998	0.666	0.666	0.666
B-#Water	13	0.999	0.807	0.840	0.823
B-#Wear	2	1.000	0.333	0.500	0.400
B-#Well	9	1.000	0.500	0.500	0.500
B-#XmasTree	2	0.999	0.846	1.000	0.916
B-xsd:dateTime	24	0.998	1.000	1.000	1.000
B-xsd:string	23	0.999	1.000	1.000	1.000
I-#BearingAndShaft	6	0.999	0.500	1.000	0.666
I-#BlockagePlugged	8	0.997	0.875	0.954	0.913
I-#Centrifuge	1.000	0.999	0.913	0.954	0.933
I-#Check	1.000	0.999	0.000	0.000	0.000
I-#ColdStandBy	1.000	0.999	0.833	1.000	0.909
I-#Compressor	9	0.998	0.250	0.666	0.363
I-#ControlFailure	3	0.999	1.000	0.500	0.666
I-#ControlUnit	4	1.000	0.000	0.000	0.000
I-#DesalinizerUnit	2	1.000	0.000	0.000	0.000
I-#Document	34	0.997	0.000	0.000	0.000
I-#DownState	15	0.997	0.333	1.000	0.500
I-#ElectricalGeneralFailure	1.000	0.999	0.000	0.000	0.000
I-#ElectrostaticSeparator	2	0.999	1.000	1.000	1.000
I-#FaultySignal	3	0.999	1.000	1.000	1.000
I-#Gasket	5	0.998	0.823	0.903	0.861
I-#InstrumentGeneralFailure	1.000	0.999	0.400	1.000	0.571
I-#Inverter	3	1.000	0.000	0.000	0.000
I-#Leakage	19	0.997	0.000	0.000	0.000
I-#Lubricant	7	0.999	0.000	0.000	0.000
I-#Maintenance	2	0.999	0.333	1.000	0.500
I-#MechanicalGeneralFailure	2	0.999	0.000	0.000	0.000
I-#Motor	3	0.999	0.000	0.000	0.000
I-#NoSignal	2	0.999	0.000	0.000	0.000
I-#OperatingBadly	40	0.993	0.000	0.000	0.000
I-#Pipeline	6	0.997	1.000	1.000	1.000
I-#Plate	2	0.998	0.631	0.857	0.727
I-#Pump	21	0.996	0.857	1.000	0.923
I-#Refit	1.000	1.000	0.000	0.000	0.000
I-#Repair	1.000	0.999	1.000	0.666	0.800
I-#Replace	3	0.999	0.666	0.666	0.666
I-#Responsible	14	0.996	1.000	0.500	0.666
I-#Seal	7	0.998	0.675	0.794	0.729
I-#Sensor	11	0.995	0.000	0.000	0.000
I-#Service	3	0.999	0.000	0.000	0.000
I-#Stage	2	0.999	0.619	0.764	0.684
I-#Stator	6	1.000	1.000	1.000	1.000
I-#StorageTank	8	0.998	0.000	0.000	0.000
I-#SubseaLine	3	0.999	0.666	0.666	0.666
I-#Test	4	0.999	0.500	0.700	0.583
I-#Tube	1.000	0.999	0.428	1.000	0.600
I-#Umbilical	2	0.999	0.727	0.421	0.533
I-#Valve	19	0.996	0.000	0.000	0.000
I-#Vessel	5	0.998	0.500	1.000	0.666
I-#Vibration	4	0.998	1.000	1.000	1.000
I-#Water	5	0.999	0.250	1.000	0.400
I-xsd:string	49	0.997	0.333	1.000	0.500
Average, SD		**0.998, 0.009**	**0.504, 0.382**	**0.626, 0.419**	**0.536, 0.377**

A.2 Tarefa RE

As tabelas A.8, A.9, A.10, A.11, A.12 e A.13 apresentam, para cada grupo de atributos, os resultados médios para cada experimento inicial sobre o conjunto *TRAIN*, com validação cruzada de 3-*fold* e "teste" intermediário sobre o conjunto *VALIDATION*.

Tabela A.8: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-1.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.991, 0.012	0.619, 0.226	0.638, 0.214	0.607, 0.223	2759.5
		Teste (VALIDATION)	0.994, 0.011	0.598, 0.266	0.733, 0.294	0.652, 0.269	
	Recall	Treino (TRAIN)	0.991, 0.012	0.620, 0.223	0.642, 0.211	0.611, 0.221	2437.3
		Teste (VALIDATION)	0.994, 0.011	0.598, 0.266	0.722, 0.299	0.647, 0.270	
	Precision	Treino (TRAIN)	0.991, 0.012	0.620, 0.223	0.642, 0.211	0.611, 0.221	2437.3
		Teste (VALIDATION)	0.994, 0.011	0.598, 0.266	0.722, 0.299	0.647, 0.270	
	F1	Treino (TRAIN)	0.991, 0.012	0.620, 0.223	0.642, 0.211	0.611, 0.221	2437.3
		Teste (VALIDATION)	0.994, 0.011	0.598, 0.266	0.722, 0.299	0.647, 0.270	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.991, 0.013	0.600, 0.184	0.609, 0.220	0.581, 0.204	2437.3
		Teste (VALIDATION)	0.993, 0.012	0.571, 0.263	0.696, 0.285	0.620, 0.263	
	Recall	Treino (TRAIN)	0.991, 0.013	0.600, 0.184	0.609, 0.220	0.581, 0.204	141.1
		Teste (VALIDATION)	0.993, 0.012	0.571, 0.263	0.696, 0.285	0.620, 0.263	
	Precision	Treino (TRAIN)	0.991, 0.013	0.584, 0.241	0.613, 0.257	0.578, 0.256	135.4
		Teste (VALIDATION)	0.991, 0.015	0.604, 0.257	0.614, 0.312	0.594, 0.270	
	F1	Treino (TRAIN)	0.991, 0.013	0.584, 0.241	0.613, 0.257	0.578, 0.256	135.4
		Teste (VALIDATION)	0.991, 0.015	0.604, 0.257	0.614, 0.312	0.594, 0.270	

Tabela A.9: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-2, com destaque sobre o melhor valor de F1 entre todos os experimentos iniciais.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes) Accuracy	Recall	Precision	F1-Measure	Tempo de CPU p/ treinamento (s)
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.993, 0.011	0.717, 0.231	0.647, 0.247	0.662, 0.245	2844.6
		Teste (VALIDATION)	0.994, 0.011	0.627, 0.274	0.739, 0.289	0.672, 0.273	
	Recall	Treino (TRAIN)	0.993, 0.011	0.717, 0.231	0.647, 0.247	0.662, 0.245	2844.6
		Teste (VALIDATION)	0.994, 0.011	0.627, 0.274	0.739, 0.289	0.672, 0.273	
	Precision	Treino (TRAIN)	0.993, 0.011	0.717, 0.231	0.647, 0.247	0.662, 0.245	2844.6
		Teste (VALIDATION)	0.994, 0.011	0.627, 0.274	0.739, 0.289	0.672, 0.273	
	F1	Treino (TRAIN)	0.993, 0.011	0.717, 0.231	0.647, 0.247	0.662, 0.245	2844.6
		Teste (VALIDATION)	0.994, 0.011	0.627, 0.274	0.739, 0.289	0.672, 0.273	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.992, 0.013	0.678, 0.268	0.657, 0.267	0.653, 0.267	157.0
		Teste (VALIDATION)	0.993, 0.012	0.614, 0.280	0.756, 0.298	0.668, 0.276	
	Recall	Treino (TRAIN)	0.989, 0.017	0.730, 0.212	0.580, 0.260	0.625, 0.238	164.3
		Teste (VALIDATION)	0.993, 0.013	0.640, 0.252	0.724, 0.277	0.669, 0.251	
	Precision	Treino (TRAIN)	0.992, 0.013	0.678, 0.268	0.657, 0.267	0.653, 0.267	157.0
		Teste (VALIDATION)	0.993, 0.012	0.614, 0.280	0.756, 0.298	0.668, 0.276	
	F1	Treino (TRAIN)	0.992, 0.013	0.678, 0.268	0.657, 0.267	**0.653, 0.267**	157.0
		Teste (VALIDATION)	0.993, 0.012	0.614, 0.280	0.756, 0.298	**0.668, 0.276**	

Tabela A.10: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-3.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.992, 0.013	0.676, 0.245	0.661, 0.227	0.654, 0.232	2528.7
		Teste (VALIDATION)	0.993, 0.012	0.576, 0.293	0.748, 0.283	0.629, 0.281	
	Recall	Treino (TRAIN)	0.992, 0.013	0.676, 0.245	0.661, 0.227	0.654, 0.232	2528.7
		Teste (VALIDATION)	0.993, 0.012	0.576, 0.293	0.748, 0.283	0.629, 0.281	
	Precision	Treino (TRAIN)	0.992, 0.013	0.675, 0.245	0.663, 0.223	0.655, 0.229	2429.0
		Teste (VALIDATION)	0.993, 0.012	0.576, 0.293	0.748, 0.283	0.629, 0.281	
	F1	Treino (TRAIN)	0.992, 0.013	0.675, 0.245	0.663, 0.223	0.655, 0.229	2429.0
		Teste (VALIDATION)	0.993, 0.012	0.576, 0.293	0.748, 0.283	0.629, 0.281	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.992, 0.013	0.675, 0.234	0.625, 0.203	0.635, 0.213	156.6
		Teste (VALIDATION)	0.993, 0.013	0.580, 0.288	0.706, 0.278	0.626, 0.273	
	Recall	Treino (TRAIN)	0.985, 0.025	0.685, 0.266	0.548, 0.239	0.562, 0.244	163.8
		Teste (VALIDATION)	0.989, 0.019	0.628, 0.292	0.529, 0.300	0.554, 0.278	
	Precision	Treino (TRAIN)	0.992, 0.013	0.675, 0.234	0.625, 0.203	0.635, 0.213	156.6
		Teste (VALIDATION)	0.993, 0.013	0.580, 0.288	0.706, 0.278	0.626, 0.273	
	F1	Treino (TRAIN)	0.992, 0.013	0.675, 0.234	0.625, 0.203	0.635, 0.213	156.6
		Teste (VALIDATION)	0.993, 0.013	0.580, 0.288	0.706, 0.278	0.626, 0.273	

Tabela A.11: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-4.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.992, 0.013	0.665, 0.247	0.658, 0.217	0.647, 0.227	2117.7
		Teste (VALIDATION)	0.993, 0.013	0.561, 0.277	0.725, 0.280	0.616, 0.263	
	Recall	Treino (TRAIN)	0.992, 0.013	0.665, 0.247	0.658, 0.217	0.647, 0.227	2117.7
		Teste (VALIDATION)	0.993, 0.013	0.561, 0.277	0.725, 0.280	0.616, 0.263	
	Precision	Treino (TRAIN)	0.992, 0.013	0.665, 0.247	0.658, 0.217	0.647, 0.227	2117.7
		Teste (VALIDATION)	0.993, 0.013	0.561, 0.277	0.725, 0.280	0.616, 0.263	
	F1	Treino (TRAIN)	0.992, 0.013	0.665, 0.247	0.658, 0.217	0.647, 0.227	2117.7
		Teste (VALIDATION)	0.993, 0.013	0.561, 0.277	0.725, 0.280	0.616, 0.263	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.991, 0.014	0.677, 0.231	0.632, 0.216	0.639, 0.220	128.6
		Teste (VALIDATION)	0.992, 0.014	0.573, 0.275	0.702, 0.253	0.618, 0.254	
	Recall	Treino (TRAIN)	0.983, 0.028	0.686, 0.229	0.519, 0.208	0.555, 0.203	116.0
		Teste (VALIDATION)	0.992, 0.015	0.604, 0.281	0.671, 0.260	0.613, 0.247	
	Precision	Treino (TRAIN)	0.991, 0.014	0.677, 0.231	0.632, 0.216	0.639, 0.220	128.6
		Teste (VALIDATION)	0.992, 0.014	0.573, 0.275	0.702, 0.253	0.618, 0.254	
	F1	Treino (TRAIN)	0.991, 0.014	0.677, 0.231	0.632, 0.216	0.639, 0.220	128.6
		Teste (VALIDATION)	0.992, 0.014	0.573, 0.275	0.702, 0.253	0.618, 0.254	

Tabela A.12: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-5.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.989, 0.018	0.578, 0.271	0.599, 0.263	0.547, 0.243	3695.3
		Teste (VALIDATION)	0.990, 0.018	0.446, 0.308	0.587, 0.325	0.492, 0.299	
	Recall	Treino (TRAIN)	0.989, 0.018	0.578, 0.271	0.599, 0.263	0.547, 0.243	3695.3
		Teste (VALIDATION)	0.990, 0.018	0.446, 0.308	0.587, 0.325	0.492, 0.299	
	Precision	Treino (TRAIN)	0.989, 0.018	0.578, 0.271	0.599, 0.263	0.547, 0.243	3695.3
		Teste (VALIDATION)	0.990, 0.018	0.446, 0.308	0.587, 0.325	0.492, 0.299	
	F1	Treino (TRAIN)	0.989, 0.018	0.578, 0.271	0.599, 0.263	0.547, 0.243	3695.3
		Teste (VALIDATION)	0.990, 0.018	0.446, 0.308	0.587, 0.325	0.492, 0.299	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.988, 0.018	0.604, 0.238	0.469, 0.242	0.496, 0.242	277.7
		Teste (VALIDATION)	0.984, 0.029	0.626, 0.250	0.418, 0.264	0.480, 0.252	
	Recall	Treino (TRAIN)	0.988, 0.018	0.604, 0.238	0.469, 0.242	0.496, 0.242	277.7
		Teste (VALIDATION)	0.984, 0.029	0.626, 0.250	0.418, 0.264	0.480, 0.252	
	Precision	Treino (TRAIN)	0.988, 0.021	0.581, 0.252	0.553, 0.247	0.533, 0.242	301.0
		Teste (VALIDATION)	0.990, 0.019	0.475, 0.287	0.616, 0.285	0.515, 0.263	
	F1	Treino (TRAIN)	0.988, 0.021	0.581, 0.252	0.553, 0.247	0.533, 0.242	301.0
		Teste (VALIDATION)	0.990, 0.019	0.475, 0.287	0.616, 0.285	0.515, 0.263	

Tabela A.13: Resultados médios dos algoritmos para RE, sobre o pacote de atributos RE-SITOP-6.

Algoritmo	Alvo do *Grid Search*	Treino (3-fold) ou Teste	Medidas (valor médio em relação as classes, desvio padrão entre as classes)				Tempo de CPU p/ treinamento (s)
			Accuracy	Recall	Precision	F1-Measure	
Frank Wolf SSVM	*Accuracy*	Treino (TRAIN)	0.985, 0.027	0.096, 0.263	0.105, 0.253	0.097, 0.254	268.4
		Teste (VALIDATION)	0.988, 0.021	0.084, 0.243	0.104, 0.250	0.089, 0.240	
	Recall	Treino (TRAIN)	0.985, 0.027	0.096, 0.263	0.105, 0.253	0.097, 0.254	268.4
		Teste (VALIDATION)	0.988, 0.021	0.084, 0.243	0.104, 0.250	0.089, 0.240	
	Precision	Treino (TRAIN)	0.985, 0.027	0.096, 0.263	0.105, 0.253	0.097, 0.254	268.4
		Teste (VALIDATION)	0.988, 0.021	0.084, 0.243	0.104, 0.250	0.089, 0.240	
	F1	Treino (TRAIN)	0.985, 0.027	0.096, 0.263	0.105, 0.253	0.097, 0.254	268.4
		Teste (VALIDATION)	0.988, 0.021	0.084, 0.243	0.104, 0.250	0.089, 0.240	
Structured Perceptron	*Accuracy*	Treino (TRAIN)	0.984, 0.027	0.071, 0.233	0.064, 0.205	0.067, 0.218	19.0
		Teste (VALIDATION)	0.988, 0.022	0.080, 0.244	0.084, 0.243	0.081, 0.240	
	Recall	Treino (TRAIN)	0.983, 0.027	0.072, 0.227	0.074, 0.210	0.067, 0.214	18.1
		Teste (VALIDATION)	0.988, 0.021	0.107, 0.289	0.083, 0.220	0.081, 0.228	
	Precision	Treino (TRAIN)	0.983, 0.027	0.072, 0.227	0.074, 0.210	0.067, 0.214	18.1
		Teste (VALIDATION)	0.988, 0.021	0.107, 0.289	0.083, 0.220	0.081, 0.228	
	F1	Treino (TRAIN)	0.984, 0.027	0.071, 0.233	0.064, 0.205	0.067, 0.218	19.0
		Teste (VALIDATION)	0.988, 0.022	0.080, 0.244	0.084, 0.243	0.081, 0.240	

As tabelas A.14 e A.15 apresentam, respectivamente, os resultados para treino e teste único e definitivo, sobre os dados *TEST*, do modelo e atributos escolhidos nas etapas anteriores.

Tabela A.14: Conjunto RE-SITOP-2 com Structured Perceptron, *TRAIN + VALIDATION*.

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
O	19413	0.955, 0.008	0.977, 0.003	0.970, 0.012	0.973, 0.005
#hasActionRelated	209	0.994, 0.003	0.681, 0.132	0.742, 0.078	0.701, 0.078
#hasCause	264	0.997, 0.000	0.883, 0.023	0.897, 0.061	0.889, 0.040
#hasDateTime	59	0.999, 0.000	0.795, 0.162	0.888, 0.099	0.818, 0.057
#hasDocumentRelated	302	0.992, 0.003	0.720, 0.112	0.796, 0.068	0.750, 0.067
#hasExpectedDateTime	10	0.999, 0.000	0.250, 0.433	0.062, 0.108	0.100, 0.173
#hasExpectedSolutionDateTime	38	0.998, 0.000	0.777, 0.139	0.760, 0.256	0.714, 0.127
#hasFailure	343	0.994, 0.001	0.863, 0.047	0.839, 0.041	0.849, 0.010
#hasOrdinalNumber	9	0.999, 0.000	0.791, 0.216	0.916, 0.144	0.833, 0.166
#hasPositionalReferenceToOtherAsset	34	0.996, 0.002	0.233, 0.137	0.219, 0.139	0.212, 0.119
#hasResponsible	38	0.998, 0.000	0.517, 0.374	0.438, 0.255	0.459, 0.290
#hasState	442	0.994, 0.001	0.912, 0.009	0.860, 0.060	0.884, 0.029
#hasStateAsCause	47	0.998, 0.000	0.625, 0.258	0.689, 0.165	0.643, 0.199
#isAlsoCalled	70	0.999, 0.000	1.000, 0.000	0.923, 0.129	0.955, 0.078
#isMaterialOf	86	0.999, 0.000	0.972, 0.039	0.953, 0.055	0.960, 0.027
#isPartOfOtherEquipment	541	0.985, 0.002	0.674, 0.104	0.739, 0.043	0.699, 0.055
#isResponsibleForAnAction	69	0.999, 0.000	0.837, 0.051	0.878, 0.107	0.854, 0.068
#isTargetOfAnAction	156	0.994, 0.001	0.535, 0.091	0.588, 0.059	0.552, 0.040
Average, SD		**0.994, 0.010**	**0.725, 0.219**	**0.731, 0.247**	**0.714, 0.239**

Tabela A.15: Conjunto RE-SITOP-2 com Structured Perceptron, *TEST*.

Tag	Support	Accuracy	Recall	Precision	F-score
O	4959	0.946	0.979	0.960	0.969
#hasActionRelated	56	0.992	0.535	0.625	0.576
#hasCause	76	0.996	0.802	0.924	0.859
#hasDateTime	8	0.999	0.625	0.833	0.714
#hasDocumentRelated	75	0.994	0.733	0.820	0.774
#hasExpectedDateTime	3	0.999	0.333	1.000	0.500
#hasExpectedSolutionDateTime	13	0.999	0.769	0.909	0.833
#hasFailure	90	0.992	0.677	0.824	0.743
#hasOrdinalNumber	1.000	1.000	1.000	1.000	1.000
#hasPositionalReferenceToOtherAsset	12	0.998	0.250	0.750	0.375
#hasResponsible	7	0.998	0.285	0.400	0.333
#hasState	108	0.994	0.824	0.864	0.843
#hasStateAsCause	8	0.999	0.875	0.777	0.823
#isAlsoCalled	22	1.000	1.000	1.000	1.000
#isMaterialOf	25	0.999	0.960	1.000	0.979
#isPartOfOtherEquipment	126	0.984	0.595	0.681	0.635
#isResponsibleForAnAction	22	0.999	0.818	1.000	0.900
#isTargetOfAnAction	41	0.996	0.585	0.827	0.685
Average, SD		**0.993, 0.012**	**0.702, 0.231**	**0.844, 0.154**	**0.752, 0.198**

A.3 Teste de generalizações

Os resultados do teste de generalização de algumas classes são apresentados na tabela A.16.

Tabela A.16: Conjunto NER-SITOP-2 com Random Forest, *TEST*, com generalização de algumas classes.

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
O	2234	0.894	0.962	0.890	0.925
B-#BearingAndShaft	5	0.999	0.000	0.000	0.000
B-#Centrifuge	3	0.999	0.400	1.000	0.571
B-#ColdStandBy	1.000	0.999	0.333	0.500	0.400
B-#Compressor	10	0.997	0.000	0.000	0.000
B-#ControlUnit	2	0.999	0.600	0.666	0.631
B-#Coupling	2	0.999	0.500	0.500	0.500
B-#DesalinizerUnit	7	1.000	0.000	0.000	0.000
B-#Document	70	0.997	1.000	1.000	1.000
B-#DownState	72	0.996	0.914	0.984	0.948
B-#ElectricGenerator	2	1.000	0.861	0.968	0.911

Continuação da Tabela A.16

Tag	Sup.	Accuracy, SD	Recall, SD	Precision, SD	F-score, SD
B-#ElectricalFailure	2	0.999	1.000	1.000	1.000
B-#ElectricalOutlet	1.000	0.999	0.000	0.000	0.000
B-#ElectrostaticSeparator	3	0.999	0.000	0.000	0.000
B-#ExternalInfluence	10	0.997	0.666	1.000	0.800
B-#Fan	2	1.000	0.300	0.750	0.428
B-#Flange	1.000	1.000	1.000	1.000	1.000
B-#FloterUnit	1.000	0.999	1.000	1.000	1.000
B-#Gas	1.000	1.000	0.000	0.000	0.000
B-#GasGenerator	1.000	0.999	1.000	1.000	1.000
B-#Gasket	3	0.999	0.000	0.000	0.000
B-#HeatExchanger	22	0.999	0.333	1.000	0.500
B-#Housing	3	0.999	0.954	0.913	0.933
B-#Hydrocyclone	3	0.999	0.666	1.000	0.800
B-#InstrumentFailure	7	0.998	0.333	0.500	0.400
B-#Inverter	1.000	1.000	0.571	0.666	0.615
B-#Lubricant	10	0.998	1.000	1.000	1.000
B-#Maintenance	14	0.997	0.700	0.875	0.777
B-#MaintenanceActivity	70	0.988	0.857	0.705	0.774
B-#MaterialFailure	23	0.997	0.571	0.833	0.677
B-#MechanicalFailure	55	0.990	0.739	0.944	0.829
B-#Membrane	1.000	1.000	0.727	0.727	0.727
B-#Motor	14	0.997	1.000	1.000	1.000
B-#NamedSystem	22	0.999	0.785	0.687	0.733
B-#Nozzle	1.000	1.000	0.909	0.952	0.930
B-#Oil	1.000	0.999	1.000	1.000	1.000
B-#OperatingBadly	22	0.996	0.000	0.000	0.000
B-#OperatingNormally	4	0.999	0.681	0.833	0.749
B-#Pipeline	7	0.998	0.750	1.000	0.857
B-#Piping	1.000	0.999	0.142	1.000	0.250
B-#Piston	1.000	0.999	0.000	0.000	0.000
B-#Plate	7	0.998	0.000	0.000	0.000
B-#ProductionPlant	1.000	0.999	0.714	0.625	0.666
B-#Pump	57	0.997	0.000	0.000	0.000
B-#Responsible	28	0.995	0.912	0.962	0.936
B-#RotorAndImpeler	1.000	0.999	0.642	0.750	0.692
B-#Seal	14	0.999	0.000	0.000	0.000
B-#Sensor	9	0.997	0.785	1.000	0.880
B-#Stage	3	0.999	0.555	0.625	0.588
B-#Stator	6	1.000	0.666	1.000	0.800
B-#StorageTank	5	0.998	1.000	1.000	1.000
B-#SubseaLine	2	0.999	0.200	1.000	0.333
B-#Transformer	1.000	1.000	0.500	1.000	0.666
B-#Tube	1.000	0.999	1.000	1.000	1.000
B-#Umbilical	1.000	1.000	0.000	0.000	0.000
B-#UninterruptiblePowerSupply	1.000	1.000	0.000	0.000	0.000
B-#UnknownFailure	3	0.999	1.000	1.000	1.000
B-#Valve	26	0.997	1.000	1.000	1.000
B-#Vessel	6	0.998	0.666	0.666	0.666
B-#Water	13	0.999	0.807	0.840	0.823
B-#Well	9	1.000	0.333	0.666	0.444
B-#XmasTree	2	0.999	0.846	1.000	0.916
B-xsd:dateTime	24	0.999	1.000	1.000	1.000
B-xsd:string	23	0.999	0.500	1.000	0.666
I-#BearingAndShaft	6	0.999	0.875	1.000	0.933
I-#Centrifuge	1.000	0.999	0.913	0.954	0.933
I-#ColdStandBy	1.000	0.999	0.000	0.000	0.000
I-#Compressor	9	0.998	0.833	1.000	0.909
I-#ControlUnit	4	1.000	1.000	0.500	0.666
I-#DesalinizerUnit	2	1.000	0.000	0.000	0.000
I-#Document	34	0.997	0.333	1.000	0.500
I-#DownState	15	0.997	1.000	1.000	1.000
I-#ElectricalFailure	1.000	0.999	1.000	1.000	1.000
I-#ElectrostaticSeparator	2	0.999	0.823	0.903	0.861
I-#ExternalInfluence	8	0.997	0.400	1.000	0.571
I-#Gasket	5	0.998	0.000	0.000	0.000
I-#InstrumentFailure	9	0.998	0.000	0.000	0.000
I-#Inverter	3	1.000	0.250	0.666	0.363
I-#Lubricant	7	0.999	0.000	0.000	0.000
I-#Maintenance	2	0.999	0.000	0.000	0.000
I-#MaintenanceActivity	13	0.996	0.555	0.714	0.625
I-#MechanicalFailure	25	0.994	1.000	1.000	1.000
I-#Motor	3	0.999	0.857	1.000	0.923
I-#OperatingBadly	40	0.993	0.000	0.000	0.000
I-#Pipeline	6	0.997	0.307	0.800	0.444
I-#Plate	2	0.998	0.000	0.000	0.000
I-#Pump	21	0.996	0.520	0.722	0.604
I-#Responsible	14	0.997	0.666	0.666	0.666
I-#Seal	7	0.998	0.675	0.794	0.729
I-#Sensor	11	0.995	0.166	0.333	0.222

Continuação da Tabela A.16

Tag	**Sup.**	**Accuracy, SD**	**Recall, SD**	**Precision, SD**	**F-score, SD**
I-#Stage	2	0.999	0.000	0.000	0.000
I-#Stator	6	1.000	0.619	0.764	0.684
I-#StorageTank	8	0.998	0.500	0.777	0.608
I-#SubseaLine	3	0.999	0.428	1.000	0.600
I-#Tube	1.000	0.999	0.727	0.421	0.533
I-#Umbilical	2	0.999	0.500	1.000	0.666
I-#Valve	19	0.996	1.000	1.000	1.000
I-#Vessel	5	0.998	0.375	1.000	0.545
I-#Water	5	0.999	0.000	0.000	0.000
I-xsd:string	49	0.997	0.000	0.000	0.000
Average, SD		**0.997, 0.010**	**0.527, 0.375**	**0.650, 0.404**	**0.563, 0.373**