Pontifícia Universidade Católica
do Rio de Janeiro

Luciana Rosa Redlich

Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da Web

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação em Informática da PUC-Rio como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Hélio Côrtes Vieira Lopes
Co-Orientador: : Prof. Marco Antonio Casanova

Rio de Janeiro
Setembro de 2013

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Luciana Rosa Redlich

Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da Web

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
Co-Orientador
Departamento de Informática - PUC-Rio

Prof. Ruy Luiz Milidiú
Departamento de Informática - PUC-Rio

Marcelo Tilio Monteiro de Carvalho
Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico - PUC-Rio

Rio de Janeiro, 4 de setembro de 2013

Luciana Rosa Redlich

Graduou-se em Engenharia da Computação na PUC-Rio(Pontifícia Universidade Católica do Rio de Janeiro) em 2010 onde trabalhou com pesquisas na área de Hipertexto e Multimídia e principalmente em TV digital no laboratório TeleMídia. Em 2011 ingressou no mestrado na mesma Universidade onde participou de projetos nas áreas de TV digital, Engenharia de Software, Banco de dados e Cloud Computing.

Ficha Catalográfica

Redlich, Luciana Rosa

Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da Web / Luciana Rosa Redlich ; orientadores: Hélio Côrtes Vieira Lopes, Marco Antonio Casanova. – 2013.

54 f. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2013.

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de máquina. 3. Ontologias. 4. Eventos. 5. Processamento de linguagem natural. I. Lopes, Hélio Côrtes Vieira. II. Casanova, Marco Antonio. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Agradecimentos

Aos meus pais, irmãs e avós que me apoiaram em todos momentos da minha vida e como não podia ser diferente não me deixaram sozinha neste momento tão importante. Mesmo que muitas vezes não estivessem presente fisicamente, estou certa que estavam olhando por mim.

Ao meu Orientador Hélio pela enorme paciência e incentivo na orientação deste projeto, mesmo em frente a tantos obstáculos.

Ao meu co-orientador Casanova por todo o auxilio dado durante o desenvolvimento deste projeto e durante todo o curso.

Aos amigos e colegas que tive o prazer de conhecer durante o curso e em especial ao Amigo Fábio Albuquerque pela grande ajuda durante todo o desenvolvimento da dissertação.

Aos professores do curso e a todos os professores que tive durante a vida, por me ensinarem a ser a pessoa que sou hoje.

E por fim, agradeço a PUC-Rio, a CAPES e a FAPERJ pelo auxilio dado a mim e sem o qual não seria possível a realização deste trabalho.

Resumo

Redlich, Luciana Rosa; Lopes, Hélio Côrtes Vieira (Orientador); Casanova, Marco Antonio (Co-Orientador). **Modelagem de eventos de trânsito com base em clipping de grandes massas de dados da Web.** Rio de Janeiro, 2013. 54p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho consiste no desenvolvimento de um modelo que auxilie na análise de eventos ocorridos no trânsito das grandes cidades. Utilizando uma grande massa de dados publicados na Internet, em especial no twitter, por usuários comuns, este trabalho fornece uma ontologia para eventos do trânsito publicados em notícias da internet e uma aplicação que use o modelo proposto para realizar consultas aos eventos modelados. Para isso, as notícias publicadas em linguagem natural são processadas, isto é, as entidades relevantes no texto são identificadas e depois estruturadas de tal forma que seja feita uma analise semântica da notícia publicada. As notícias publicadas são estruturadas no modelo proposto de eventos e com isso é possível que sejam feitas consultas sobre suas propriedades e relacionamentos, facilitando assim a análise do processo do trânsito e dos eventos ocorridos nele.

Palavras-chave

Aprendizado de máquina; ontologias; eventos; processamento de linguagem natural

Abstract

Redlich, Luciana Rosa; Lopes, Hélio Côrtes Vieira (Advisor); Casanova, Marco Antonio (Co-Advisor). **Traffic events modeling based on clipping of huge quantity of data from the Web.** Rio de Janeiro, 2013. 54p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This work proposes a traffic event model to assist the analysis of traffic events on big cities. This paper aims to provide not only an ontology for traffic events considering published news over the Internet, but also a prototype of a software architecture that uses the proposed model to perform queries on the events, using a huge quantity of published data on the Internet by regular users, especially on twitter. To do so, the news published in natural language is processed, and the relevant entities in the text are identified and structured in order to make a semantic analysis of them. The news reported is structured in the proposed model of events and thus the queries about their properties and relationships could be answered. As a consequence, the result of this work facilitates the analysis of the events occurred on the traffic process.

Keywords

Machine learning; ontologies; events; Natural language processing.

Sumário

Lista de figuras

Lista de tabelas

1
Introdução

1.1.
Motivação

Um dos maiores problemas enfrentados pelas grandes cidades hoje em dia é a questão do trânsito. O número excessivo de veículos nas ruas, juntamente com variados incidentes nas vias acarretam em atrasos e transtornos aos motoristas, causando não só problemas de bem estar, mas também problemas econômicos para a população. Solucionar o problema do trânsito não é uma tarefa simples e requer um planejamento urbano eficiente do governo das cidades. Entretanto com a ajuda de novas tecnologias, principalmente da Internet, a vida dos motoristas e das pessoas que precisam se deslocar pelas ruas congestionadas ganha um aliado que pode ajudar principalmente na escolha das melhores rotas de deslocamento. Cada dia mais pessoas compartilham dados do trânsito na Internet. Tanto Web sites de empresas especializadas neste tipo de notícias como usuários de redes sociais postam acontecimentos do trânsito. Entretanto não há ainda aplicações que consigam utilizar esta imensa quantidade de dados disponíveis de forma unificada e que gere dados padronizados que permitam consultas para análises sobre os eventos acontecidos no trânsito.

Assim, este trabalho fornece um modelo de dados sobre notícias de trânsito que padronize os dados publicados na Internet com atributos relevantes para sua análise. Permitindo então que notícias publicadas pelos mais diversos veículos e usuários diferentes, em linguagem natural, sejam processadas de tal maneira que possam ser pesquisados eventos acontecidos no trânsito e as consequências que estes acarretaram em cada localidade, sendo possível assim fazer uma análise sobre os incidentes que estão acontecendo e suas consequências, para que novas rotas para os motoristas possam ser escolhidas, assim como soluções para que alguns dos incidentes sejam evitados.

1.2. Contribuições

Este trabalho tem duas principais contribuições:

1. Uma modelo que represente os eventos ocorridos no trânsito e publicados no *twitter*.
2. Uma aplicação que, dado um *tweet* sobre acontecimentos no trânsito, é capaz de identificar tais acontecimentos transformando-os para o modelo proposto e processando-os de tal maneira que seja possível fazer consultas sobre esses eventos e seus relacionamentos. Ou seja, este protótipo mostra uma aplicação do modelo.

1.3. Organização do texto

Este texto esta estruturado em seis capítulos. O capítulo 2 apresenta os trabalhos relacionados a este projeto e utilizados como base para seu desenvolvimento. Primeiro, fala sobre ontologias e os trabalhos existentes na literatura que endereçam problemas ligados a ontologias de eventos e eventos no trânsito. Em seguida apresenta um resumo sobre o trabalho de estruturação de texto que serviu como base para o desenvolvimento deste projeto, apresentando seus conceitos importantes e que serão reutilizados aqui, assim como os problemas encontrados que serão também tratados nesta dissertação.

O capítulo 3 fala sobre a proposta de uma ontologia como modelo para os acontecimentos publicados no *twitter*, que é a contribuição principal desta dissertação.

O capítulo 4 apresenta uma arquitetura para uma aplicação de estruturação de notícias sobre o trânsito e análise dos eventos gerados.

O capítulo 5 fala sobre as tecnologias utilizadas para a implementação do protótipo da arquitetura e da ontologia proposta e também expõe os resultados obtidos nos testes feitos com a implementação.

O capítulo 6 apresenta a conclusão, salientando as contribuições feitas por esta dissertação, assim como os resultados obtidos nos testes. Também enumera os trabalhos futuros que podem ser feitos.

2
Trabalhos Relacionados

2.1.
Ontologias

Neste trabalho utilizaremos a classe de ontologias de domínio, que conceitualizam domínios particulares. Em especial, estamos interessados nos eventos ocorridos no trânsito das grandes cidades e publicados em notícias da Internet.

Para a modelagem destes eventos buscamos na literatura referências sobre ontologias de eventos e sobre ontologias de eventos no trânsito, mais especificamente. As referências encontradas serão apresentadas nas próximas duas subseções.

2.1.1.
Ontologia de Eventos

De forma geral, eventos são usados para descrever ocorrências de ações e mudanças no mundo real. Muitos projetistas de ontologias consideram objetos e propriedades como entidades estáticas e eventos como entidades dinâmicas do mundo real.

Worboys et al. [1] apresenta um modelo de objetos geoespaciais como objetos situados em um conjunto de propriedades puramente espaciais. A evolução de objetos no tempo pode ser vista através de imagens instantâneas feitas do objeto no tempo associadas a marcas de tempo. Seguindo esta mesma linha Guizzardi et al. [2] fala sobre objetos como entidades inteiramente presentes em qualquer instante do tempo que estiverem presentes, isto é, se em uma circunstância c_1, um objeto O possui a propriedade P_1 e em uma circunstância c_2 possui a propriedade P_2 (possivelmente incompatível com P_1), ele continua sendo o mesmo objeto O em ambas as situações.

O modelo de eventos é apresentado como uma evolução do modelo de objetos. Eventos são objetos com propriedades temporais. Segundo Guizzardi et

al. [2] eventos são possíveis transformações de uma situação para outra realidade, alterando o estado das coisas onde esses são modelados ontologicamente como entidades dependentes, já que para existirem dependem existencialmente de seus participantes, onde esses podem ser outros eventos.

Para Sowa [3], eventos são mudanças em etapas discretas que ocorrem em processos, onde um processo pode ser descrito pelos seus pontos de inicio e fim e pelas mudanças que ocorrem entre esses pontos. Adicionalmente, processos possuem períodos de inatividades chamados de estados.

Kaneiwa et al. [4] também falam sobre a diferenciação entre evento e objeto. Cada objeto possui um identificador, mas cada instância de evento possui um tempo e uma localização. Uma instância de um evento é a ocorrência de um evento que pertence a um tipo. Assim se um mesmo tipo de evento ocorre várias vezes podemos dizer que são instâncias de um mesmo evento. Toda instância também pode ter um ou mais componentes como ator e objeto.

Eventos possuem funções semânticas que formalmente indicam que cada evento implica em uma mudança funcional de um objeto no mundo real. Cada evento afeta um objeto ou ambiente e então muda suas propriedades ou estado no próximo tempo.

Eventos também podem se relacionar uns com os outros. Essas relações são importantes para descrever a sequência de vários eventos. Por exemplo, a sequências de causa e efeito.

Kaneiwa et al. [4] descrevem alguns tipos de relações que podem conectar dois eventos: causal, temporal ou espacial. Distingue relações entre instâncias de eventos e entre classes de eventos:

- Relação entre instâncias de eventos: considere e_1 e e_2 como sendo duas instâncias de eventos. Então a relação binária $r(e_1,e_2)$ entre e_1 e e_2 é chamada de relação entre instâncias de eventos e é definida por:
    1. Se a instância de evento e_1 causa a instância do evento e_2, então existe a relação causal $e_1 \rightarrow_{causa} e_2$.
    2. Se a instância do evento e_2 ocorre depois da instância e_1, então existe a relação $e_1 \rightarrow_{próximo} e_2$.
    3. Se um evento e_1 inclui temporalmente o evento e_2, e e_1 ocorre no espaço de e_2, então a relação “parte de” $e_1 <_{po} e_2$ existe.

- Relação entre classes de eventos: sejam E_1 e E_2 duas classes de eventos. Então a relação $R(E_1, E_2)$ é chamada de relação de classe de evento e é definida por:
    1. Se cada instância de evento de E_1 e E_2 não podem ocorrer simultaneamente, então existe uma relação disjunta. $E_1 \parallel E_2$.
    2. Se toda instância de E_1 pertence a E_2, então a relação de subclasse $E_1 \subseteq E_2$ ocorre.
    3. Se para toda instância e de E_1, existe uma instância e' de E_2 que e' $<_{po}$ e, então a "relação de classe de evento parte de" $E_1 <_{po} E_2$ existe.
    4. Se para toda instância e de E_1, existe uma instância e' de E_2 que $e \rightarrow_{causa} e'$ então a relação de classe de evento de causa $E_1 \rightarrow_{causa} E_2$ existe.

Worboys et al. [1] definem relações entre eventos e objetos. Eventos e objetos estão intimamente ligados. São citadas as seguintes relações:

- Criação: um evento que resulta na criação de um objeto. Como por exemplo, o evento de criação de uma ponte resulta em um objeto ponte.
- Manutenção: Um evento que resulta na continuação da existência de um objeto. Por exemplo, o evento de pintar a ponte resulta na continuação da existência da ponte.
- Reforço ou Degradação: Eventos que tem efeito positivo ou negativo na existência de um objeto.
- Destruição: Um evento que resulta na destruição de um evento.

Ou seja, essas relações nada mais são do que uma formalização de como os eventos agem sobre os objetos.

Worboys et al. [1] também definem relações entre eventos. Eventos podem possuir relações geoespaciais onde as propriedades espaciais são relacionadas. Entretanto eventos também podem possuir relações espaço-temporal, ou seja, relações sobre o próprio evento. Algumas dessas relações:

- Inicialização: A ocorrência do evento A inicializa o evento B.

- Perpetuação/ facilitação: A ocorrência do evento A gera uma papel positivo na continuação do evento B.
- Impedimento/bloqueio: A ocorrência do evento A tem um papel positivo no enfraquecimento ou término do evento B.
- Finalização: A ocorrência do evento A força ou permite que o evento B já iniciado, termine.

2.1.2. Ontologias de Eventos no trânsito

Existem na literatura alguns trabalhos que endereçam conceitos sobre ontologias de eventos no trânsito.

Dongli et al [5] definiram em seu trabalho uma ontologia para acidentes de trânsito. Esta ontologia tinha como objetivo oferecer interoperabilidade entre os diferentes sistemas de gestão de tráfego e fornecer a eles informação semântica suficiente para que computadores possam ler e automaticamente analisar e processar os dados sobre os acidentes de trânsito. Sua ontologia pode ser resumida pela figura 1.

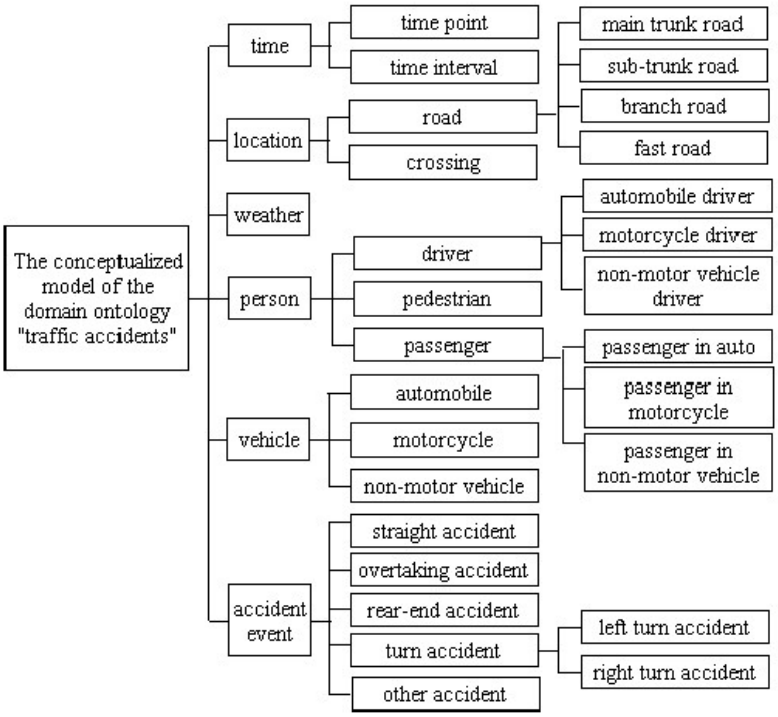

Figura 1 Modelo conceitualizado de ontologia de domínio "Traffic accidents" [5].

Essa ontologia possui uma semântica muito expressiva sobre o domínio dos acidentes de trânsito e por isso ela não pode ser reutilizada na ontologia definida nesta dissertação. Como trataremos de eventos publicados em notícias, esta semântica não é alcançada e tampouco necessária quando tratamos de eventos do tipo acidente, como veremos mais adiante.

Wang [6] também apresenta em seu trabalho uma ontologia chamada de TADO (traffic accident domain ontology), cujo domínio são os acidentes de trânsito e tem como objetivo permitir que usuários possam realizar buscas semânticas sobre os conceitos dos acidentes de trânsito.

Haggaf e Mahmoud [7] apresentam um sistema (OnTraJaCS) que gera para todos os veículos registrados uma recomendação de rota que minimize os congestionamentos das vias quando essas rotas forem executadas juntas. Ele utiliza ontologias para detectar e prever a existência de congestionamentos a partir de múltiplos dados coletados em placas instaladas em todos os veículos registrados. O modelo simplificado da ontologia de mapas pode ser visto na figura 2:

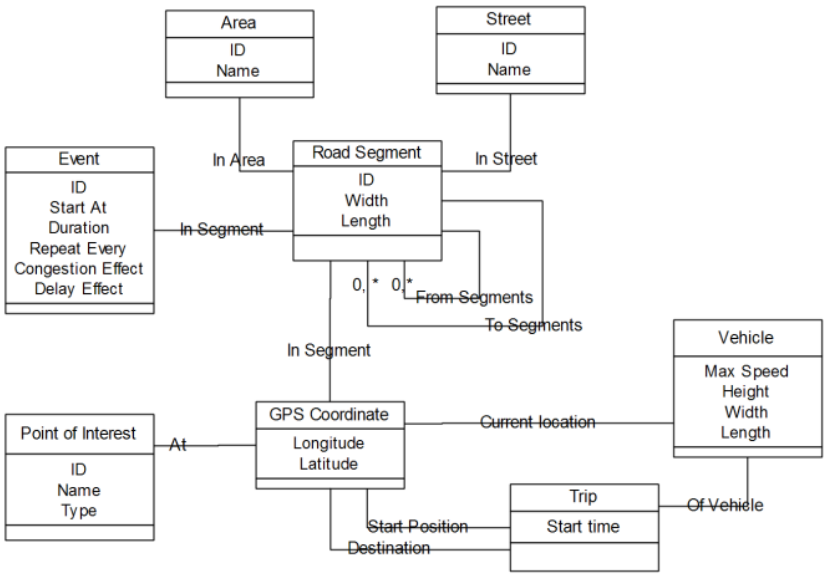

Figura 2 Modelo de representação de mapas do sistema OnTraJaCsis [7].

Um ponto importante neste trabalho são as predições e estimativas feitas pela ontologia. Inferências baseadas em regras lógicas são feitas sobre a ontologia

para que deem apoio a conclusões complexas do sistema sobre a ocorrência de congestionamentos.

2.2. Estruturação de texto

Albuquerque [8] apresenta uma aplicação que lê *tweets* de usuários especiais que postam notícias sobre o trânsito e faz uma estruturação desses *tweets*. Esse processo de estruturação tem como objetivo identificar fatos que tenham sido apresentados no texto e o local onde eles ocorreram.

A solução proposta por Albuquerque [8] para a aplicação descrita acima é baseada em Inteligência Artificial, usando processamento de linguagem natural e técnicas de aprendizado de máquina. Duas tarefas são propostas para a solução do problema: a primeira é a de identificação das entidades relevantes no texto, problema esse conhecido na literatura como Reconhecimento de Entidades (NER). A segunda é a tarefa de identificação e interpretação dos fatos relevantes, conhecido na literatura como problema de Extração de Informação (IE).

Para o Reconhecimento de Entidades são propostas as entidades descritas de forma resumida a seguir:

- **Location** (LOC): uma localização georreferenciável;
- **Point of reference** (REF_N): representa uma proximidade a uma localização mais específica;
- **Lane direction** (DIR_N): indica a direção do fluxo do trânsito;
- **Both lanes** (DIR_NS): indica se o evento ocorreu em ambas as direções;
- **Co-reference** (COREF_LOC): indica uma correferência a localização principal;
- **Restrictive location** (ABSL_N): indica uma localização que restringe geograficamente a área relacionada ao local principal;
- **Traffic intensity** (TR_INT): indica a intensidade do trafego e é subclassificada em:
    - **Good traffic** (TI_1): boas condições de tráfego;
    - **Heavy traffic** (TI_2): tráfego intenso;

- **Slow traffic** (TI_3): tráfego parado.

- **Fact** (FACT): indicam eventos ou fatos que causam algum impacto sobre o transito e são subclassificados em:
  - **Accident** (ACC): acidentes;
  - **Emergency** (EMER): emergências;
  - **Hindering fact** (HINDER): interdições.
- **End of a fact** (OPEN): Indica o final do fato, a solução do evento;
- **Other** (O): Entidades não relevantes para o problema.

Para a segunda tarefa, Extração de Informação, a solução do problema é modelada como uma árvore de dependências. Cada nó da árvore (Figura 3) é uma entidade (pertencente ao conjunto de entidades propostas na tarefa 1 da solução do problema, exceto a entidade "Other") e cada aresta representa uma relação entre as entidades.

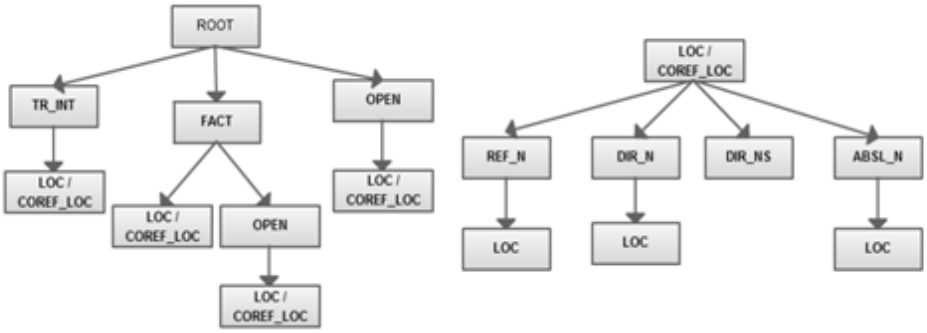

Figura 3 Relacionamentos entre entidades relevantes [8].

Foram feitos testes com um corpus de 475 *tweets*. O resultado obtido foi uma media de 90,41% da medida f-measure para o reconhecimento de entidades e acurácia de 73,37% para a tarefa de extração de informação. Os resultados sobre os reconhecimentos de fatos, ou seja, o reconhecimento das entidades FACT ficou com media de 60% e foi inferior aos 70% esperado. Esses resultados insatisfatórios da classificação dos fatos encontrados por Albuquerque [8] são explicados pelo autor pelo pequeno tamanho do corpus.

3
Proposta de Ontologia de Eventos no Trânsito

3.1.
Introdução

Nesta dissertação visamos a construção de um modelo de eventos de trânsito. Para tanto, uma ontologia cujo domínio são os eventos que ocorrem no trânsito e são publicados em notícias da Internet é apresentada. Esta ontologia será chamada de TEDO (Traffic Event Domain Ontology).

Para a modelagem destes eventos foram utilizados como base as referências encontradas na literatura sobre ontologias de eventos e sobre ontologias de eventos no trânsito que foram citadas no capítulo anterior.

A definição de uma ontologia neste trabalho é de suma importância, pois como estamos tratando de eventos publicados em notícias descritas em linguagem natural, as palavras podem ter semânticas totalmente diferentes conforme o seu contexto e uma ontologia se faz necessária para preencher o "gap" semântico existente nas notícias publicadas em linguagem natural e permitir que máquinas possam interpretar estas notícias tirando informações importantes sobre elas.

Assim sendo, o objetivo principal da definição desta ontologia é que ela gere uma padronização das notícias publicadas sobre o trânsito para que essas possam ser consultadas.

Como já mencionado a modelagem desta ontologia foi baseada nos modelos de ontologia de eventos anteriormente descritos, mas também foi levada em conta uma análise do domínio, ou seja, foi feita uma análise das características existentes em um conjunto de notícias sobre o *twitter*. Em torno de 700 *tweets* foram analisados.

É importante salientar que o domínio desta ontologia são eventos publicados em notícias da Internet por humanos e que possuem alguma relação com o trânsito. Este domínio traz algumas especificidades à ontologia que está sendo definida e que serão explicadas mais a diante.

3.2.
TEDO – Traffic Event Domain Ontology

3.2.1.
Classes

A TEDO tem como finalidade apresentar os acontecimentos do trânsito e seus relacionamentos com outros acontecimentos.

Trataremos o trânsito como um processo, onde ocorrem vários eventos discretos durante o tempo, e esses eventos geram mudanças nesse processo. Ou seja, o trânsito é um conjunto de eventos que ocorrem e que se relacionam entre si no tempo e no espaço.

Cada um desses *tweets* ou notícia, é modelado como um conjunto de eventos do trânsito. Esses eventos podem ser classificados de acordo com o seu tipo. Nesta ontologia cada tipo de evento foi modelado como uma classe diferente, já que cada uma dessas classes representam um grupo de eventos diferentes. Foram observados que alguns tipos de eventos são reportados nas notícias com maior frequência. Para cada um destes eventos foi criada uma classe diferente, detalhadas abaixo e que podem ser visualizadas na figura 4:

Interdição (Hinder): São eventos reportados em que a via está interditada para o tráfego. Em geral é representado pelas palavras “interdição” e “fechada” no texto da notícia;

Acidentes (Accident): Indica a ocorrência de um acidente. Pode ser uma colisão, engavetamento, choque entre carros, capotamento, queda de motociclista, atropelamento, entre outras coisas. Representam apenas acidentes de trânsito onde existem veículos envolvidos. No caso, por exemplo, do evento queda de árvore, este não é classificado como acidente;

Enguiço (Breakdown): Indica a existência de um veículo parado no trânsito por alguma pane (elétrica ou mecânica) e que esteja atrapalhando o fluxo do tráfego;

Engarrafamento (Traffic Jam): São fatos descritos nas notícias relacionados a existência ou não de engarrafamento. Por exemplo, retenção, congestionamento e lentidão, estão relacionados à condição do fluxo do tráfego em algum local. Esses eventos são representados nas notícias por substantivos

como por exemplo, as palavras: trânsito, tráfego, congestionamento, lentidão, retenção, entre outros;

Evento Climático (Weather Condition): São eventos relacionados ao clima como por exemplo, chuva, chuva forte, alagamento, bolsão d'água. Em geral, não foi muito observado nas notícias analisadas, entretanto como foi observado em ontologias já descritas sobre acidentes de trânsito [5] e [6] as condições climáticas possuem grande participação na ocorrência de acidentes de trânsito, portanto a classe de evento *Weather Condition* foi criada para eventos climáticos que possam interferir no tráfego;

Além desses cinco tipos de eventos descritos, muitos outros eventos também foram verificados nas notícias, mas eles não possuem uma frequência muito alta e por isso não foi criada uma classe para cada um desses tipos. Além disso, eventos de tipos ainda não cadastrados e que tenham alguma interferência no trânsito também podem ser noticiados e para tanto uma nova classe foi criada:

Outros (Other Events): Qualquer tipo de evento que afete o tráfego e que não se encaixe em nenhuma das classes citadas acima. Por exemplo: Bueiros sem tampas, sinal com problema, falta de energia elétrica e manifestações.

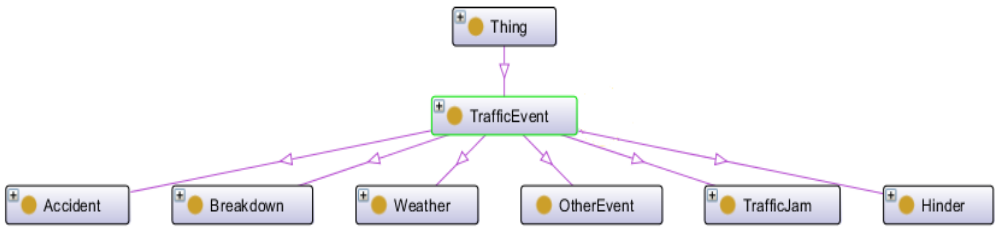

Figura 4 Classes de eventos

Nas publicações analisadas muitas vezes são postados quando um fato ocorreu e também quando um fato terminou. Por exemplo, um usuário posta a presença de um veículo com pane mecânica na via e também posta a retirada de um veículo em pane da via. Essas duas notícias publicadas representam dois eventos diferentes no trânsito e não um só evento com inicio e fim. A explicação para isso é que um evento é algo que possa trazer uma mudança no trânsito, possa dar inicio a um outro evento. Assim, esses dois acontecimentos podem ter consequências diferentes. E então, as classes de evento ***Accident***, ***Breakdown***, ***Wheather***, ***Hinther*** e ***OtherEvent*** foram subclassificadas em classes que

representam seu inicio e fim , esta subclassificação pode ser visualizada na figura 5.

Inicio de Interdição (*HinderStart*): Evento que mostra o acontecimento de uma via interditada;

Fim de Interdição (*HinderEnd*): Liberação da vida previamente interditada;

Inicio de Acidente (*AccidentStart*): Fatos que indicam a ocorrência de um acidente;

Fim de Acidentes (*AccidentEnd*): Engloba fatos que finalizam um acidente, como por exemplo, a retirada de veículos acidentados da pista;

Inicio de Enguiço (*BreakdownStart*): Fatos que indicam a existência de veículos enguiçados na via;

Fim de Enguiço (*BreakdownEnd*): São fatos que indicam a retiradas dos veículos enguiçados na pista;

Inicio de Evento Climático (*WeatherStart*): São fatos que indicam ocorrência de eventos climáticos que podem afetar o tráfego;

Fim de Evento Climático (*WeatherEnd*): Indicam a finalização de algum evento climático;

Inicio de Outros Eventos (*OtherEventStart*): São fatos que indicam ocorrência de eventos do tipo outros;

Fim Outros Eventos (*OtherEventEnd*): Indicam a finalização evento do tipo outros;

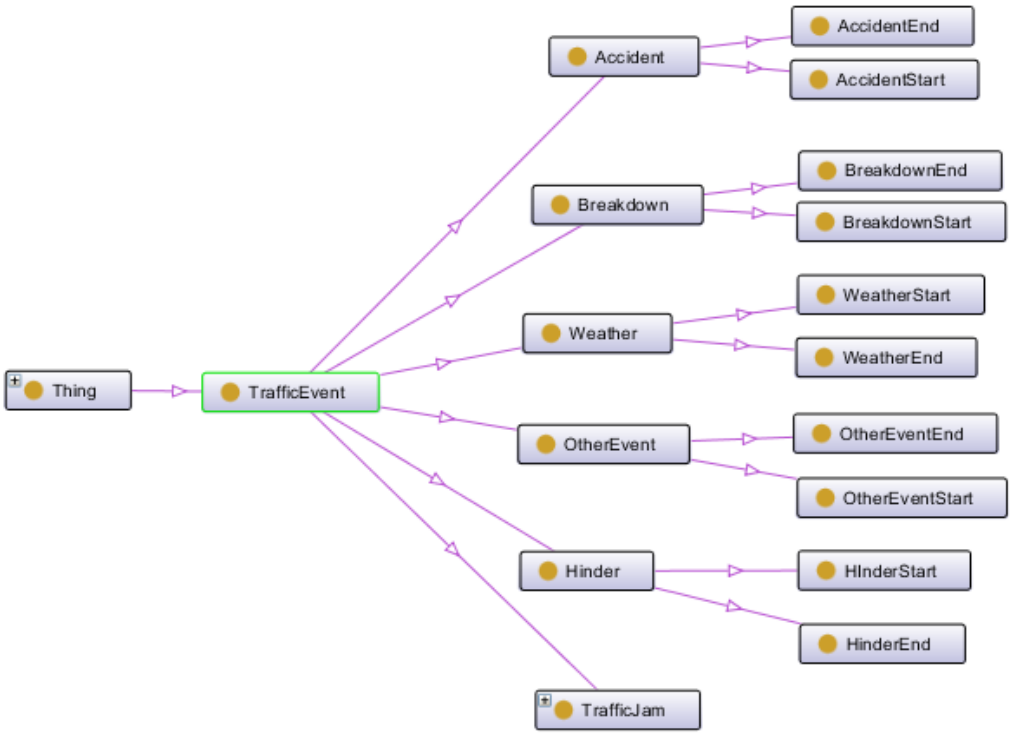

Figura 5 Subclassificação das classes de eventos do trânsito.

A Classe ***TrafficJam*** não foi subclassificada, pois ela representa o estado do tráfego e para isso, como será visto mais a frente, possui uma propriedade para esse estado.

Em uma ontologia as classes possuem propriedades. Essas propriedades representam relações entre dois objetos diferentes e servem para especificar objetos de uma determinada classe. Em owl (formato em que está sendo definido a TEDO) existem dois tipos principais de propriedades: propriedades de objetos, que fazem a ligação entre dois objetos diferentes e propriedades de dados, que fazem a ligação de um objeto a um valor primitivo qualquer (inteiros, floats, strings, boleanos e etc).

Para a classe ***TrafficEvent*** foram definidas três propriedades, apresentadas na figura 6, duas dessas são propriedades de objetos : ***hasTime*** e ***hasLocation***. A terceira propriedade é a propriedade ***Actor***.

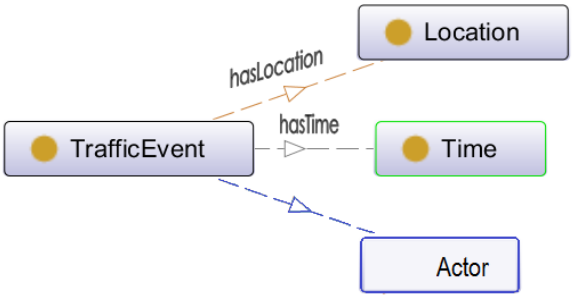

Figura 6 Propriedades de trafficEvent (Eventos de trânsito).

A propriedade ***hasLocation*** liga um objeto da classe ***TrafficEvent*** a um objeto da classe ***Location*** que representa o local em que o evento ocorreu. Ela possui uma cardinalidade 1*, isto é, um ou mais objetos ***Location*** podem ser ligados a um evento.

A propriedade ***hasTime*** liga um objeto da classe ***TrafficEvent*** a um objeto da classe ***Time*** que representa uma marcação de tempo sobre a ocorrência do evento. Ela possui cardinalidade 1, ou seja, todo evento está ligado a um e somente um objeto do tipo ***Time***.

A propriedade ***Actor*** liga um objeto da classe ***TrafficEvent*** a uma String que representa os atores deste evento. Essa propriedade possui cardinalidade 0*, ou seja, um evento pode possuir um os mais atores. Esses atores são os agentes envolvidos na ocorrência do fato. Por exemplo, na notícia, “Acidente entre caminhão e automóvel na Av. Brasil na altura do INTO.”, o evento é um acidente que possui dois atores: caminhão e automóvel.

Por essa ser uma ontologia definida para eventos publicados em notícias da Internet em linguagem natural, esses atores podem ser publicados de formas diferentes. Além disso, podem existir um conjunto enorme de atores e portanto não foi criada uma classe que representasse esses atores. Eles apenas são preenchidos pelo seu valor literal retirado da notícia. Alguns exemplos são: veículos, carro, moto, taxi, automóvel, motocicleta, pedestre.

As três propriedades escolhidas foram observadas na maioria das notícias analisadas. Além disso, de acordo com o que foi visto nas ontologias de eventos estudadas, essas são as três principais propriedades que definem um evento.

A classe ***TrafficJam***, possui além das classes herdadas de sua superclasse, a propriedade ***hasIntensity***, que liga um objeto ***TrafficJam*** a um objeto do tipo ***TrafficIntensity.***As propriedades dos eventos da Classe TrafficJam podem ser vistas na figura 7.

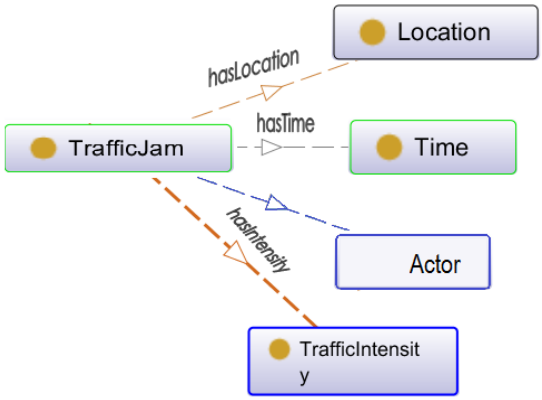

Figura 7 Propriedades de evento da classe traffiJam.

A classe ***TrafficIntensity*** representa a intensidade do congestionamento no momento do evento. Toda vez que um evento do tipo engarrafamento é publicado, deseja-se saber qual a intensidade desse engarrafamento, ou seja, as condições do trânsito no local do evento publicado. Por isso essa classe foi criada. Ela foi subclassificada em três outras classes (Figura 8), especificando assim os possíveis estados do tráfego:

GoodTraffic, representa boas condições de tráfego, e é normalmente publicado nas notícias da seguinte maneira: “boas condições”, “sem congestionamento”,”sem retenção”, “fluxo bom de veículos”, entre outras;

HeavyTraffic, representa um tráfego intenso, mas não parado da via. Normalmente representado nas notícias pelas seguintes expressões: “intenso”, “com retenção” e etc;

HeavyTraffic, representa o tráfego parado, lento e é publicado nas notícias com as seguintes expressões: “parado”, “lento”, “congestionado”.

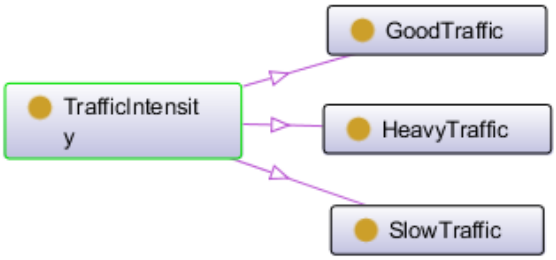

Figura 8 Subclassificação da classe TrafficIntensity.

Como já mencionado a classe ***TrafficEvent*** possui propriedades que a ligam a objetos das classes ***Time*** e ***Location***. Essas duas classes também foram especificadas nessa ontologia.

A classe ***Time*** é composta por três propriedades: ***timeStamp***, ***publicationTime*** e ***eventTime***. A propriedade ***timeSpamp*** é uma string que representa alguma marca de tempo publicada no texto da notícia. Como por exemplo, as palavras, "neste momento", "agora" e "próxima semana". Ou seja, qualquer palavra publicada na notícia que remeta ao tempo. Já as propriedades ***publicationTime*** e ***eventTime*** estão no formato W3C XML DateTime format (yyyy-mm-ddThh:mm:ssZ). A primeira representa o momento em que a notícia foi publicada e a segunda é uma compilação entre o momento em que a notícia foi publicada e a marca de tempo da notícia, ou seja, a data real da ocorrência do evento.

O local em que o evento ocorreu é publicado em linguagem natural como uma string, entretanto para que possam ser feitas consultas sobre as relações entre os locais onde os eventos ocorreram é necessários que eles sejam armazenados como coordenadas. Essas coordenadas podem representar somente um ponto ou uma área, neste caso estamos utilizamos coordenadas de dois pontos para representar essa área e damos o nome destes dois pontos de ***Envelope***.

Para que represente corretamente o local que o evento ocorreu, a classe ***Location*** é composta por quatro propriedades:

LocationValue, uma string com o conteúdo da descrição do local publicada na notícia;

hasCoordinate, propriedade que liga o objeto *Location* a um objeto da classe *Coodinate* que possui duas strings com o valor de longitude e latitude da coordenada;

hasEnvelope, uma propriedade que liga um objeto ***Location*** a um objeto da classe ***Envelope***, que por sua vez, possui dois objetos ***Coordinate***, uma para cada ponto do ***Envelope***;

hasDistance, que liga um objeto ***Location*** a um objeto da classe ***Distance***. A classe ***Distance*** possui duas propriedades, uma ***targetLocation*** (o local com o qual a distância esta sendo calculada) e ***distanceValue*** (o calor da distância entre o objeto ***Location*** e o objeto ***targetLocation***). Um objeto ***Location*** deverá estar ligado a varias distâncias, uma para cada um dos outros objetos ***Location*** instanciados.

3.3. Relações entre eventos

Algumas relações podem ser definidas entre eventos. Elas são importantes para descrever as sequências de acontecimentos dos eventos e assim permitir que o processo do trânsito seja descrito. Essas relações são relações binárias entre eventos:

Relação entre Eventos: <Evento, Evento>

As relações de interesse nesta ontologia são de quatro tipos:

Relações Entre Eventos do Trânsito:
Relações de Causalidade;
Relações Temporais;
Relações Espaciais;
Relação Espaço-temporal.

Em uma ontologia as relações entre objetos podem ser feitas por propriedades como já mencionado anteriormente. Para a definição destas relações foram definidas algumas propriedades entre objetos que serão apresentadas juntamente com uma definição de cada tipo de relação.

Relações de Causalidade:

Relação Causal: Sejam a e b instâncias de eventos, então a relação causal RC (a, b) existe se a causa b;

Relação Causado por: Sejam a e b instâncias de eventos, então a relação causal RC (a, b) existe se a é causado b;

Para as relações causais foram definidas duas propriedades que ligam as classes ***TrafficEvent***: ***causes*** e ***isCausedBy***. A primeira liga o evento causador ao evento causado e a segunda o inverso.

Relações Temporais:

Relação Temporal: Sejam *a* e *b* duas instâncias de eventos e cada uma dessas instâncias possuam marcas de tempos t_a e t_b, então a relação temporal $R_T(a, b)$ existe e pode ser subclassificada em:

1. **Relação Temporal disjunta, s**e $t_a \neq t_b$;
2. **Relação temporal simultânea**, se $t_a = t_b$;
3. **Relação temporal continua**, se $t_a = (t_b + \Delta)$ ou $t_b = (t_a + \Delta)$, onde Δ representa um período de tempo curto previamente definido.

Para as relações temporais foram definidas propriedades que ligam objetos da classe ***Time****:* ***sameAs***, ***isBefore***, ***isAfter***, ***isContinuous***, ***aroundTime***, que significam respectivamente, quando dois eventos possuem o mesmo *eventTime*, quando um evento possui um *eventTime* anterior ao segundo, quando um evento possui um *eventTime* posterior ao segundo, quando um evento possui um *eventTime* que é a sequência do segundo e quando um *eventTime* é próximo ao outro.

Relações Espaciais: Sejam *a* e *b* duas instâncias de eventos e cada uma dessas instâncias possuam marcas geoespaciais l_a e l_b. Esses espaços são os locais onde cada instância de evento ocorreu e devem ser definido em uma área e não somente um ponto no espaço. Então a relação espacial Re (a, b) existe e pode ser subclassificada em:

1. **Relação espacial disjunta: S**e $l_a \neq l_b$;
2. **Relação espacial sobreposta:** Se $l_a \subseteq l_b$ ou se $l_b \subseteq l_a$, ou seja, quando dois eventos ocorrem no mesmo local, ou quando um local esta contido no outro.

Para as relações temporais foram definidas propriedades que ligam objetos da classe ***Location***. Essas propriedades são: ***samePoint*** que indica se dois objetos possuem a mesma coordenada e ***isPartOf*** que indica se um objeto possui coordenadas que estão contidas na área delimitada pelo objeto ***Envelope***.

Relação Espaço-Temporal: Sejam a e b duas instâncias de eventos, então existe uma relação Espaço-Temporal R_{ET} (a, b) se existe uma relação R_T (a, e) e R_E (a, b).

4
Arquitetura

4.1.
Visão Geral

Este capítulo descreve uma arquitetura proposta para o processamento e análise de notícias sobre o trânsito de acordo com o modelo de eventos proposto no capítulo 3.

A arquitetura é dividida em módulos. A notícia retirada do *twitter* é passada para o módulo **Analisador de notícias** onde é processada e analisada. Fornecida para o analisador em linguagem natural, a notícia é processada de tal forma que entidades importantes do texto são identificadas e depois analisadas de maneira que as entidades sejam estruturadas semanticamente.

Ao sair desse módulo os dados não estão mais em linguagem natural. Já processados, eles são passados para o módulo **Gerenciador de Ontologia.** Neste módulo, os dados brutos são modelados de acordo com a ontologia de domínio definida (TEDO) e armazenado em um banco de dados de Triplas.

O **Gerenciar de análise de eventos do trânsito** é o responsável por fornecer as consultas sobre os eventos e executá-las no banco de dados. Essas consultas são feitas de acordo com o modelo de ontologia TEDO.

As próximas seções descrevem em detalhe os módulos mostrados na figura 9.

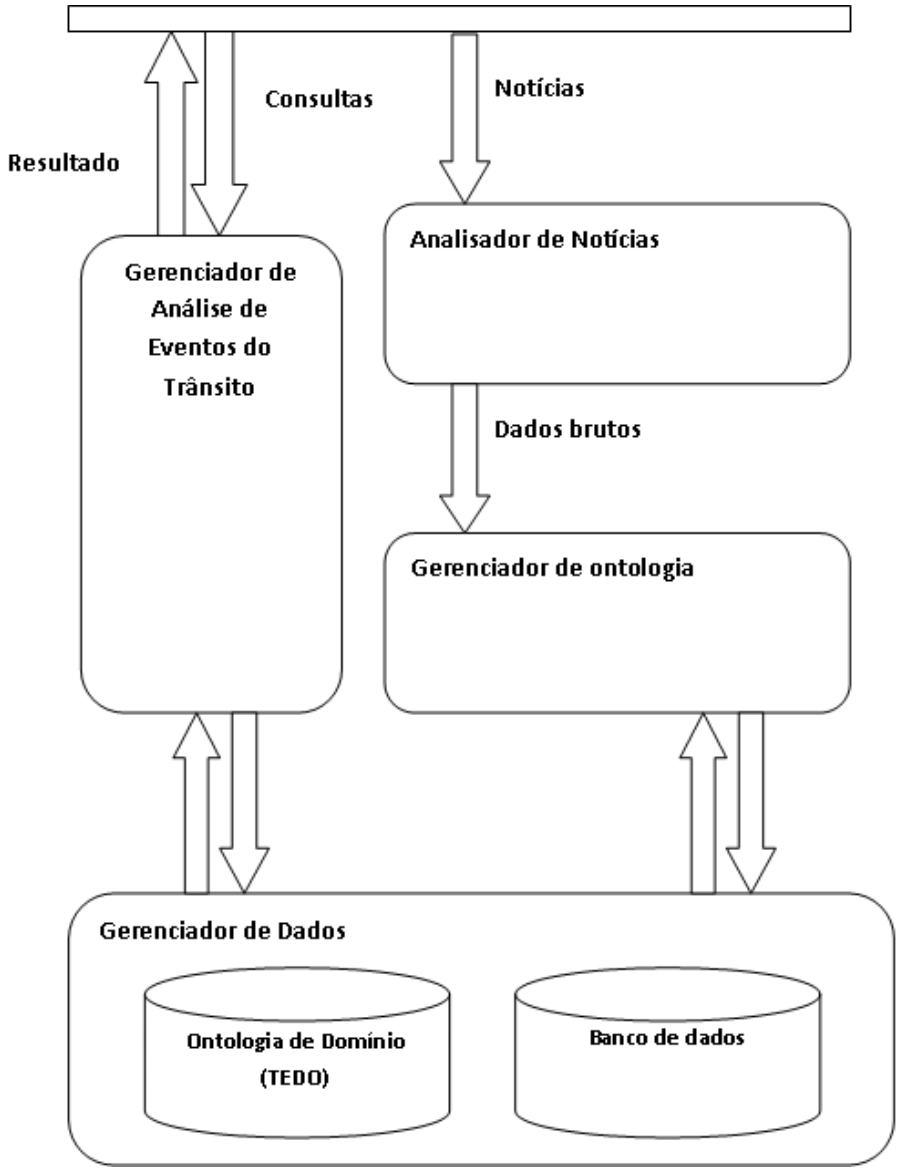

Figura 9 Arquitetura proposta

4.2. Analisador de notícias

O módulo **Analisador de Notícias** é responsável pelo processamento da notícia escrita em linguagem natural. A notícia é passada como entrada para o módulo, processada e estruturada de tal forma que, ao final, uma árvore de dependência com as entidades reconhecidas é passada para o módulo seguinte. Seu desenvolvimento foi baseado na solução proposta por Albuquerque [8] de estruturação de texto, descrita no capítulo 2.2. Albuquerque [8] utiliza dois passos

para a estruturação de texto, Reconhecimento de Entidades nomeadas e Extração de Informação. A forma como esses dois passos são feitos neste trabalho é descrita a seguir:

4.2.1. Reconhecimento de Entidades Nomeadas

A primeira etapa do processo é o reconhecimento das entidades nomeadas das notícias. Essa etapa tem como objetivo identificar no texto, escrito em linguagem natural, palavras que podem ser relevantes para que as notícias sejam classificadas de acordo com a ontologia definida. Para tanto, além das entidades mencionadas por Albuquerque [8], novas entidades foram acrescentadas ao modelo de reconhecimento proposto e algumas foram modificadas. São elas:

- **Hindering fact (HINDER):** O real fechamento e interdição da via, representado pelas palavras "interdição", "fechamento";
- **Breakdown (BREAK):** Ocorrência de enguiço de algum veículo;
- **Other (OTHER):** Qualquer evento que ocorre no trânsito e que não está classificado em nenhuma das demais classes. Inclui agora eventos que podem interditar a via, como manifestações;
- **TrafficJamm (TRAFFIC):** Ocorrência de eventos relacionados diretamente com o trânsito. Englobam congestionamentos, retenções, e trânsito livre.
- **Weather (WEATHER):** Ocorrência de eventos relacionados ao tempo. Englobam chuva, bolsão d'água, alagamento.
- **Actors (ACTOR):** Indica os personagens envolvidos na ocorrência de um evento, por exemplo, em um acidente os envolvidos no acidente (ex: carro e ônibus, motociclista, etc.) são os seus atores e em um enguiço o veiculo enguiçado.
- **Consequence (CONS):** São palavras que podem indicar relações de causa entre eventos. Por exemplo, causou, gerou, ocasionou.
- **TIME:** São palavras presentes na notícia que especificam o tempo em que o evento ocorreu. Palavras como: neste momento, agora, em 1 hora.

- **Traffic intensity (TR_INT):** Classifica as palavra, em geral adjetivos, relativos as intensidades:
  - **Good traffic (TI_1):** boas, livre;
  - **Heavy traffic (TI_2):** intenso, com retenção;
  - **Slow traffic (TI_3):** parado, lento.

Exemplos de notícias classificadas durante esta etapa do processo são:

1. Data: 24/03/2012 18:37:23; Fonte: odia24horas

Interdição	#AvBrasil,	em	Manguinhos	Complica	trânsito	na	#LinhaAmarela.
HINDER	**LOC**	**ABSL_N**	**LOC**	**CONS**	**TRAFFIC**		**LOC**

2. Data: 5/03/2012 07:07:01; Fonte: odia24horas

Acidente	entre	#2 carros,	Na	Av.das Américas	na	pista sentido
ACC		**ACTOR**		**LOC**		**DIR_N**

Grota Funda	próximo ao	número 19880.
LOC	**REF_N**	**LOC**

3. Data: 26/03/2012 07:19:27; Fonte:odia24horas

Queda	de	motociclista	na	pista central	da	#AvBrasil,	altura
ACC		**ACTOR**		**LOC**			**ABSL_N**

de	Bonsucesso	sentido	Centro	Trânsito	é lento
	LOC	**DIR_N**	**LOC**	**TRAFFIC**	**TI_3**

no local.

COREF _LOC

4. Data: 28/03/2012 06:39:40; Fonte:odia24horas

Caminhão	com	pane mecânica	Complica	o trânsito	na
ACTOR		**BREAK**	CONS	**TRAFFIC**	

Av Rodrigues Alves	altura	Rua Rivadária Correia,	No	sentido	Praça Mauá.
LOC	**REF_N**	**LOC**		**DIR_N**	**LOC**

5. Data: 17:28:50; Fonte:operacoesrio:

Trânsito	lento	na	Avenida Brasil,	altura	de
TRAFFIC	TI_1		LOC	**REF_N**	
Manguinhos,	sentido	Centro,	devido	a um	bolsão d´água
LOC	**DIR_N**	**LOC**	**CONS**		**EMER**

4.2.2. Extração de Informações

Depois das notícias terem as palavras classificadas de acordo com as entidades nomeadas, o próximo passo é a extração de informações dessas notícias.

As entidades são agrupadas de tal forma que geram uma árvore de dependência entre elas.

Uma nova modelagem da árvore de dependência foi feita para esta etapa do processo de tal forma que as informações geradas neste passo fiquem de acordo com as informações necessárias para o uso da ontologia definida.

Como definido durante a descrição da ontologia, um *tweet* é um conjunto de eventos. Assim sendo, cada um dos *tweets* pode gera uma árvore de dependência com 1 nó root. Esse nó root só pode ser ligado a nós do tipo ***FACT.***

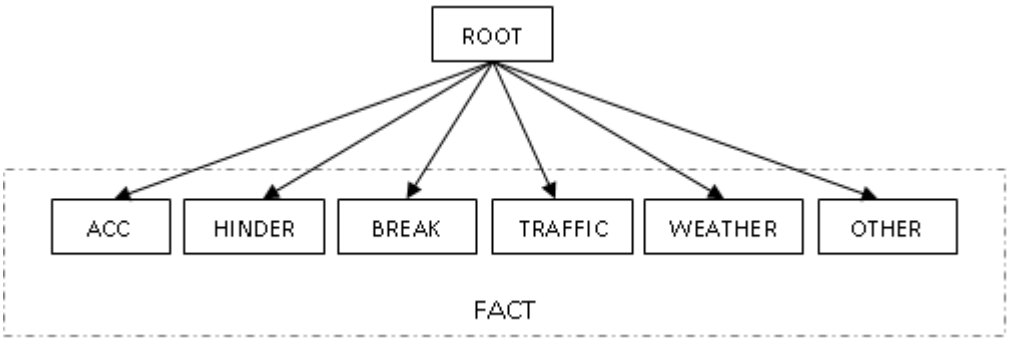

Figura 10 Modelagem do Fato.

Os fatos são subclassificados em acidente, interdição, enguiço, engarrafamento, clima, e outros, como apresentado na figura 10. Cada um dos nós do tipo FACT pode estar ligado a outros nós por arestas que representam uma

relação de propriedade entre esses nós. Neste documento esta relação é representada por uma seta (⟶).

As subárvores dos nós do tipo FACT estão representada abaixo na figura 11:

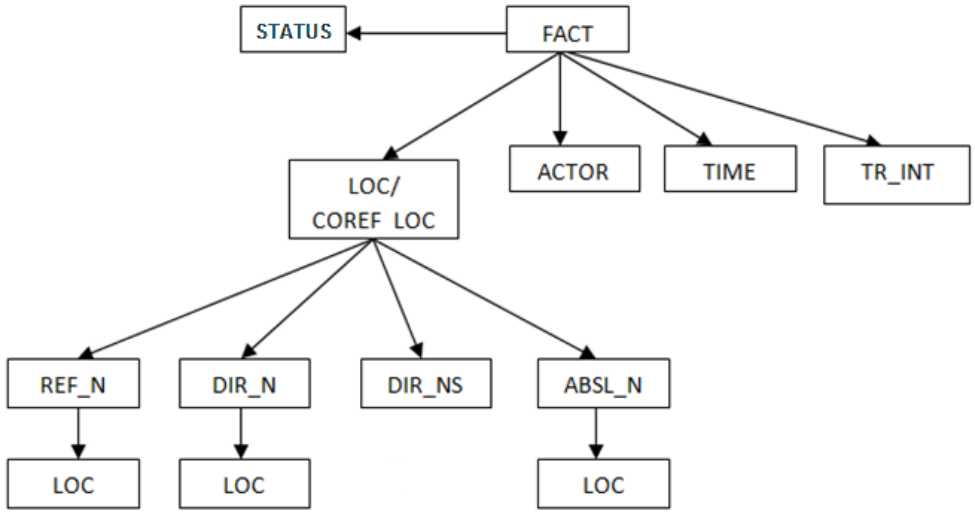

Figura 11 Estruturação das entidades nomeadas.

A árvore de dependências também deve mostrar o relacionamento causal entre os fatos. Um fato pode estar ligado a outro por uma aresta que representa esta relação, onde o fato B é consequência do fato A. Este relacionamento é uma tradução direta da entidade CONS e é representado por uma seta pontilhada.

Figura 12 Relacionamento entre Fatos.

Nas figuras 13, 14 e 15 são apresentados exemplos de árvores de dependência gerados durante esta etapa do processo :

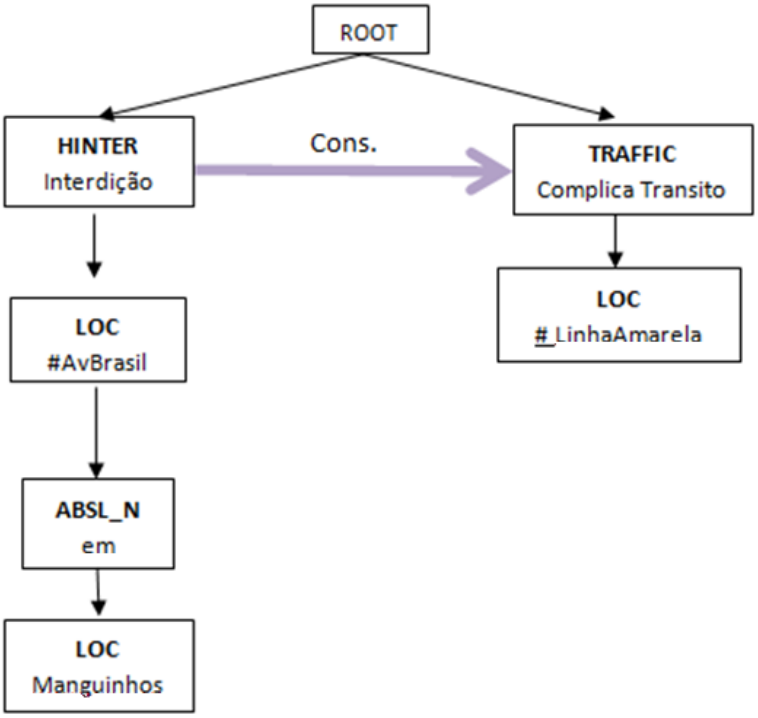

Figura 13 Exemplo 1 de árvore gerada.

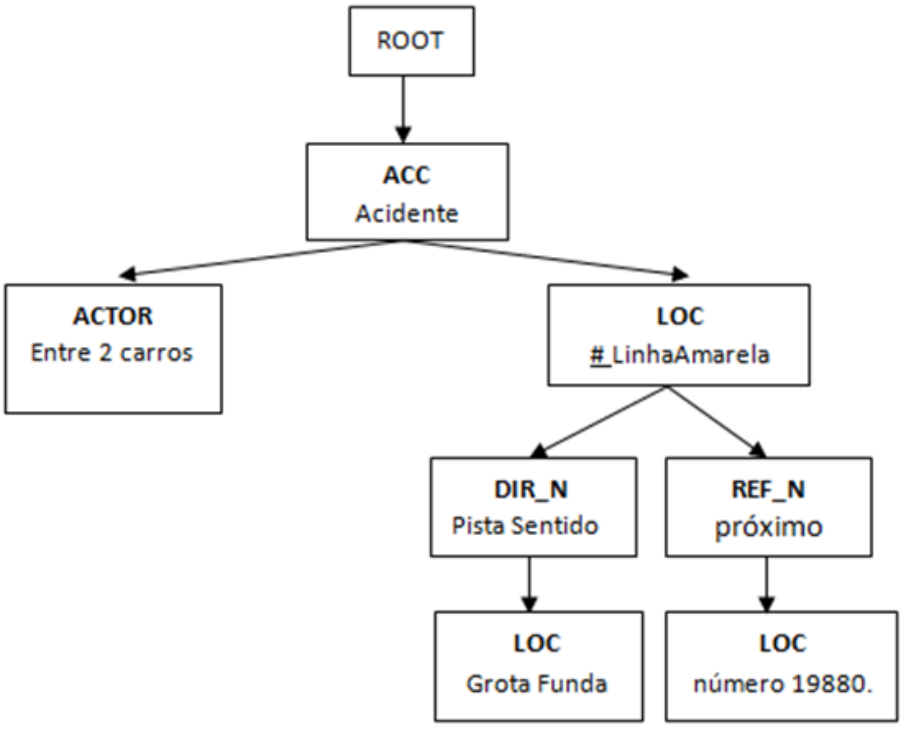

Figura 14 Exemplo 2 de árvore gerada.

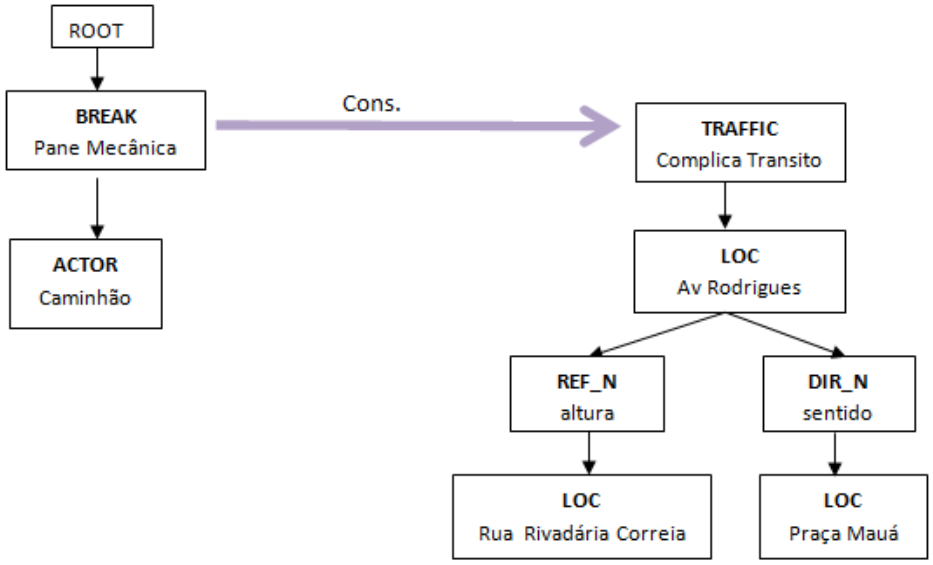

Figura 15 Exemplo 3 de árvore gerada.

4.3. Gerenciador de Ontologia

O Gerenciador de Ontologia é o módulo responsável por gerar os eventos de acordo com a ontologia TEDO e armazená-los no banco de dados. Os dados brutos resultantes do módulo analisador de notícias são passados para este módulo na forma de uma árvore de dependência. Além da árvore de dependência também são passadas as informações da notícia do *twitter: a* notícia original antes do processamento do texto, fonte e data em que a notícia foi postada.

A árvore de dependência representa as dependências das entidades encontradas em uma notícia inteira. Assim esta árvore pode gerar mais de um evento.

Como já mencionado anteriormente, a raiz de uma árvore está ligada a elementos do elementos do tipo fato. Cada um desses elementos FATO possui uma sub árvore. E cada uma dessas sub árvores representam diferentes eventos.

A essas sub árvores são aplicadas alguma regras para que os eventos possam ser gerados de acordo com a ontologia de domínio. As regras são descritas a seguir.

A propriedade classe do evento é preenchida pelos campos FACT e STATUS.

A propriedade **tempo** é composta por três atributos, *timestamp, publicationTime e eventTime* como já descrito. *TimeStamp* e *publicationTime* possuem um mapeamento direto. A primeira é a entidade TIME reconhecida no texto da notícia, e a segunda é a data de publicação da notícia no twitter. O atributo *publicationTime* é dado pela junção de *timeStamp* e *publicationTime*. O *timeStamp* é passado por um processador de marcas de tempo para que seu significado seja traduzido para medidas de tempo e adicionado a marca da publicação do evento e com isso seja descoberto o tempo do evento. Um evento que não possua uma entidade TIME em sua árvore de dependência terá os atributos *publicationTime* e o *eventTime* com o mesmo valor.

Caso a árvore de dependência de um fato possua um local para o mesmo o valor literal deste local é utilizado para preencher o campo *LocationValue* do objeto *location* relacionado ao evento em questão. Caso não exista um local, o *LocationValue* deve ser preenchido com o usuário fonte do *tweet*, por exemplo, o usuário linhaAmarelaRJ, quando nenhum local é apresentado na notícia.

Os objetos *Coordinate* e *Envelope* que se relacionam com o atributo *Location* do evento são preenchidos com o valor das coordenada do Local (*LocationValue*). A implementação feita hoje utiliza um serviço do CloudMap [11] que dado uma String que representa um local retorna uma coordenada composta pela latitude e longitude deste local. Outros serviços podem ser utilizados neste ponto no futuro. Estes serviços podem retornar além de uma coordenada, uma área e por isso o objeto Envelope também foi colocado na ontologia apesar de hoje não ser utilizado. O serviço utilizado hoje apenas reconhece as coordenadas dos locais principais.

Cada vez que um novo local é adicionado, sua distância para todos os demais locais armazenados é calculada e armazenada para que possam ser feitas as consultas sobre essas distâncias. O cálculo dessas distancias será mostrado mais a frente.

Caso uma árvore de dependência seja decomposta em mais de um evento, mas somente um local seja identificado na árvore, este deve preencher os locais de todos os eventos gerados.

Se for identificado um local do tipo COREF_LOC, este deve ser preenchido pelo local citado no evento imediatamente anterior.

Depois de identificados os eventos eles são armazenados no banco de dados em formato de triplas.

4.4. Gerenciador de Análises de Eventos

O Gerenciador de Análises de Eventos é proveê uma interface ao usuário com algumas funções de consultas ao banco de dados de eventos. Essas consultas são feitas baseadas nas propriedades da ontologia TEDO.

A implementação fornece ao usuário algumas consultas que, em uma primeira instância, podem ser úteis para a análise dos eventos. Entretanto, qualquer pessoa que conheça a ontologia poderá implementar outras consultas caso seja necessário.

As consultas implementadas são as seguintes:

1. Busca por eventos de uma classe passada como parâmetro;
2. Busca por eventos causados por outros eventos*;*
3. Dado um período de tempo, busca por um evento com publicação dentro deste período;
4. Dado um evento, buscar eventos que foram causados por ele;
5. Dado um evento, buscar eventos que o causaram;
6. Dado um evento, buscar eventos no mesmo horário;
7. Dado um horário, buscar um evento com aquele horário;
8. Dado um evento, buscar eventos em um local próximo, onde a distância é passada como parâmetro;
9. Dado um evento buscar seus atores;
10. Dado um evento buscar seus locais;
11. Dado um evento, buscar o tempo que ocorreu.

Este módulo também é capaz de gerar graficamente os eventos presentes no banco de dados, utilizando a biblioteca GraphViz [10].

4.5. Gerenciador de dados

Os dados são representados como triplas RDF e armazenados no OpenLink Virtuoso [14], seguindo a ontologia TEDo definida neste Módulo.

4.6. Implementação

Para o desenvolvimento da ontologia descrita no capitulo 3, foi utilizada a ferramenta Protégé 4.3 [12] [13], que é uma ferramenta livre, open-source para construir modelos de domínio e aplicações baseadas em conhecimentos com ontologias. O modelo da ontologia foi criado e exportado na linguagem OWL.

A implementação da arquitetura descrita no capítulo 4 foi feita utilizando Java como linguagem de programação.

No módulo News Analyser a arquitetura desenvolvida em [8] foi utilizada com algumas modificaçãos.

O Banco de dados utilizado para o armazenamento dos dados foi o OpenLink Virtuoso[14], um servidor hibrido que permite o armazenamento de dados em formato de triplas RDF. As consultas foram feitas em SPARQL [16] no grafo criados no Virtuoso.

A API do framework Virtuoso Jena Provider [15] foi utilizada a comunicação com o banco de dados.

Cloud Made service [9] foi utilizado para descoberta das coordenadas geográficas dos locais dos eventos.

5
Resultados

5.1.
Organização dos Experimentos

Para testar a implementação da arquitetura proposta, utilizamos como estudo de caso o *twitter* como fonte de notícias sobre o trânsito. Em específico, nos testes foram utilizados os usuários @odia24horas e @operacoesrio.

No módulo de Análise de Notícias, no processo de reconhecimento de entidades nomeadas foi usado apenas o algoritmo SMO [16], já que conhecidamente apresenta melhores resultados, comprovados nos experimentos feitos em [8].

Os mesmos testes feitos por Albuquerque em [8] foram feitos neste trabalho para medir os resultados nas etapas de Reconhecimento de Entidades Nomeadas e Extração de Informações. A técnica de ten-fold cross-validation foi utilizada. O corpus foi dividido em dez partes iguais e testadas dez vezes e em cada teste uma parte foi usada para o conjunto de teste e as outras nove para o conjunto de treinamento. O resultado apresentado então é uma media do resultado das dez execuções. As medidas de qualidade utilizadas foram acurácia (*accuracy*), abrangência (*recall*), precisão (*precision*), Medida f (*f-measure*), desvio padrão (*standard deviation of f-measure*), definidas como:

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_p + f_n}$$

$$Accuracy == \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

Onde,

		Anotação	
		Positivo	Negativo
Resultado do teste	Positivo	Verdadeiro Positivo (tp)	Falso Positivo (fp)
	Negativo	Falso Negativo (fn)	Verdadeiro Negativo (tn)

Tabela 3: Resultado de teste.

Para a etapa de Reconhecimento de Entidades Nomeadas foram utilizado um corpus com 690 tweets. Estes tweets foram anotados manualmente de acordo com a ontologia proposta para que o algoritmo fosse treinado.

Para a etapa de Extração de informação, 200 tweets foram anotados.

5.2. Resultados

A seguir serão apresentados os resultados encontrado neste trabalho e sua comparação com os resultados encontrados por Albuquerque [8]. De maneira geral os resultados neste trabalho foram melhores devido aos seguintes fatos:

O primeiro é o maior número de tweets utilizados no processo de treinamento do algoritmo, já que conhecidamente, quanto maior o conjunto de dados de treinamento, melhor o resultado do algoritmo.

O segundo é o uso de uma ontologia para a classificação dos dados. Uma ontologia bem definida para o modelo dos dados melhora o resultado porque elimina possíveis ambiguidades. Durante o processo de anotação manual dos dados, de posse de um modelo bem definido, a classificação dos dados é feita de acordo com regras claras, previamente definidas, fazendo com que não haja duvida durante a classificação. Assim sendo, com os dados anotados de forma não ambíguas, ou seja, sem que um conjunto de palavras completamente diferentes tenham a mesma classificação e nem que as mesmas palavras sejam classificadas de maneiras diferentes o algoritmo "aprende" melhor como fazer a classificação, deixando assim mais preciso o resultado.

A tabela a seguir , apresenta a média dos resultados da etapa de classificação de entidades nomeada. Em um primeiro momento foi medido o resultado da classificação de cada palavra individualmente, ou seja, os resultados mostrados a

seguir é da classificação palavra por palavra. A acurácia média de todas as entidades foi de 95,95 %.

Entity	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
ABSL_N	99,86	92,21	92,62	91,46	7,35
ACC	99,94	98,74	96,35	97,45	3,43
ACTOR	99,72	96,92	89,79	92,79	7,24
BREAK	99,97	99,07	98,61	98,77	2,35
CONS	99,83	88,24	89,71	88,29	11,08
COREF_LOC	99,95	97,24	97,28	97,06	3,54
DIR_N	99,67	95,78	93,83	94,65	5,21
DIR_NS	99,98	100	98,68	99,30	1,83
OTHER	99,61	90,58	78,29	81,11	10,66
HINDER	99,76	93,85	76,51	82,03	17,02
LOC	98,42	96,87	97,12	96,99	1,82
O	96,46	95,48	97,47	96,46	1,72
STATUS	99,92	97,91	90,65	93,44	8,54
REF_N	99,83	97,54	95,53	96,45	3,28
TIME	99,79	83,98	75,91	78,79	15,71
TI_1	99,86	95,77	88,04	91,25	6,60
TI_2	99,81	97,00	93,10	94,90	2,81
TI_3	99,89	97,32	91,55	93,54	9,33
TRAFFIC	99,62	95,77	95,31	95,49	2,81
WEATHER	99,90	92,49	92,84	91,11	11,27
Média	99,59	95,14	91,46	92,57	6,68

Tabela 4 Resultados da classificação das entidades palavra por palavra

A seguir é mostrado os resultados encontrados para as entidades acidente, fato impeditivo e outros, entidades que também existiam no trabalho de Albuquerque [8] e que tiveram uma melhora significativa no resultado. As primeiras linhas das tabelas que serão apresentadas a seguir mostram o resultado encontrado no presente trabalho.

ACC					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	99,94	98,74	96,35	97,45	3,43
Albuquerque	99,83	94,64	88,6	89,75	11,55
Aumento (%)	0,11	4,33	8,75	8,58	

Tabela 5 Comparação de resultados da medição palavra por palavra da entidade acidente

HINDER					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	47,09	62,61	51,5	54,55	28,93
Albuquerque	35,51	60	43,48	48,5	20.13
Aumento (%)	32,61	4,35	18,45	12,47	

Tabela 6 Comparação de resultados da medição palavra por palavra da entidade fato impeditivo

OTHER					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	49,6	75,99	54,27	60,44	28,01
Albuquerque	9,01	33,11	12,15	16,06	5,63
Aumento (%)	450,50	129,51	346,67	276,34	

Tabela 7 Comparação de resultados da medição palavra por palavra da entidade Outros

As entidades ator, enguiço, consequência, tempo, trafego e evento climático que não existiam no trabalho de Albuquerque[8] também tiveram resultados bastante satisfatórios na classificação de palavra por palavra:

Entidade	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
ACTOR	99,72	96,92	89,79	92,79	7,24
BREAK	99,97	99,07	98,61	98,77	2,35
CONS	99,83	88,24	89,71	88,29	11,08
TIME	99,79	83,98	75,91	78,79	15,71
TRAFFIC	99,62	95,77	95,31	95,49	2,81
WEATHER	99,9	92,49	92,84	91,11	11,27

Tabela 8 Resultados da medição palavra por palavra das novas entidades

A comparação das médias dos resultados da classificação individual das palavras nos dois trabalhos é apresentada a seguir. Um aumento de 12,75% para a acurácia foi observado.

Média da classificação de todas as entidades					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão

Redlich	99,59	95,14	91,46	92,57	6,68
Albuquerque	88,33	88,56	83,85	85,16	5,78
Aumento (%)	12,75	7,43	9,08	8,70	

Tabela 9 Comparação da média dos resultados da medição palavra por palavra

Também foram medidos os resultados por conjunto de palavras. Isto é, o resultado real do problema proposto já que a classificação de uma entidade em um tweet é feita em relação a um conjunto de palavras. Assim sendo, na tabela a seguir é apresentada a média dos resultados encontrados, utilizando a técnica Ten-fold.

Entity	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
ABSL_N	91,87	100	91,87	95,02	9,70
ACC	95,25	98,66	96,16	97,37	4,74
ACTOR	79,10	93,94	82,01	86,41	15,79
BREAK	95,23	97,77	96,29	96,96	8,5
CONS	82,93	94,41	86,34	89,75	10,40
COREF_LOC	97,28	100	97,28	98,53	3,03
DIR_N	93,68	99,76	93,82	96,53	4,74
DIR_NS	92,18	93,75	95	94,31	15,03
OTHER	49,60	75,99	54,27	60,44	28,01
HINDER	47,09	62,61	51,5	54,55	28,93
LOC	87,53	94,49	92,002	93,21	3,87
O	98,17	100	98,17	99,07	0,53
STATUS	84,16	91,48	85,35	87,94	23,39
REF_N	93,91	99,14	94,66	96,78	2,88
TIME	56,99	62,28	62,70	61,80	35,79
TI_1	75,46	89,35	80,11	84,01	15,89
TI_2	87,95	96,54	90,00	93,07	7,82
TI_3	88,94	95,09	90,53	92,59	14,78
TRAFFIC	90,84	96,98	93,22	95,00	4,62
WEATHER	75,76	84,72	81,19	82,22	23,38
Média	83,20	91,35	85,62	87,78	13,09

Tabela 10 Resultados da classificação das entidades na medição do conjunto de palavras.

As entidades acidentes também foram analisadas individualmente e comparadas com o trabalho de Albuquerque [8] na classificação do conjunto e

tiveram um resultado superior no presente trabalho, como é mostrando nas tabelas a seguir:

ACC					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	95,25	98,66	96,16	97,3716	4,74
Albuquerque	86,57	100	86,57	90,89	15,79
Aumento (%)	10,03	-1,34	11,08	7,13	

Tabela 11 Comparação dos resultados da medição do conjunto de palavras da entidade acidente

HINDER					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	47,09	62,61	51,5	54,55	28,93
Albuquerque	35,51	60	43,48	48,5	20,13
Aumento (%)	32,61	4,35	18,45	12,47	

Tabela 12 Comparação dos resultados da medição do conjunto de palavras da entidade fato impeditivo

OTHER					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	49,6	75,99	54,27	60,44	28,01
Albuquerque	9,01	33,11	12,15	16,06	5,63
Aumento (%)	450,50	129,51	346,67	276,34	

Tabela 13 Comparação dos resultados da medição do conjunto de palavras da entidade Outro

Assim como nos resultados obtidos por palavra, as entidades ator, enguiço, consequência, tempo, trafego e evento climático que não existiam no trabalho de Albuquerque [8] também tiveram resultados bons. Com acurácia acima de 70%em quase todos os casos menos na entidade tempo.

Entidade	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
ACTOR	79,108	93,9431	82,0146	86,4156	15,7954
BREAK	95,23	97,77	96,29	96,96	8,57
CONS	82,93	94,41	86,34	89,75	10,4
TIME	56,99	62,28	62,7	61,8	35,79
TRAFFIC	90,84	96,98	93,22	95	4,62

WEATHER	75,76	84,72	81,19	82,22	23,38

Tabela 14 Resultados da medição do conjunto de palavras da novas entidades

A média comparativa dos dois trabalhos na medição do conjunto de palavras é mostrada a seguir,

Média da classificação de todas as entidades					
	Accuracy	Precision	Recal	F-Measure	F-Measure Desvio Padrão
Redlich	83,2	91,35	85,62	87,78	13,09
Albuquerque	79,32	89,73	81,5	84,29	9,37
Aumento (%)	4,89	1,81	5,06	4,14	

Tabela 15 Comparação dos resultados da medição do conjunto de palavras

Para o processo de Extração de Informações foi utilizado um corpus com 200 *tweets* e os resultados encontrados foram os seguintes:

Acurácia média dos resultados utilizando Ten-fold: 93.51583774519278%.

Acurácia por exemplo utilizando Ten-fold: 73.0%

Desvio padrão por exemplo: 13.820274961085252%

Nos testes feitos com o módulo de análise eventos foram adicionados 520 eventos ao banco de dados. Alguns exemplos da forma gráfica de visualização dos eventos são apresentadas nas figuras 18, 19 e 20, a seguir:

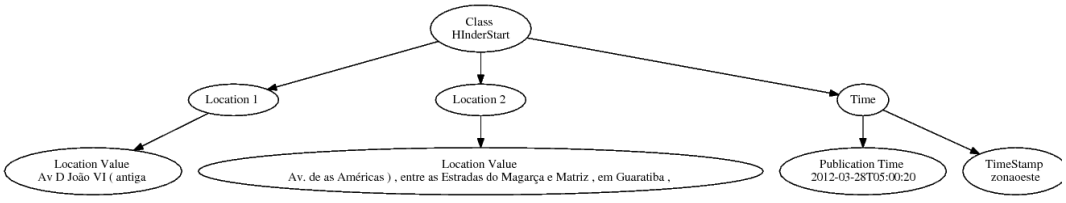

Figura 16 Exemplo 1 da visualização de um evento de gerado.

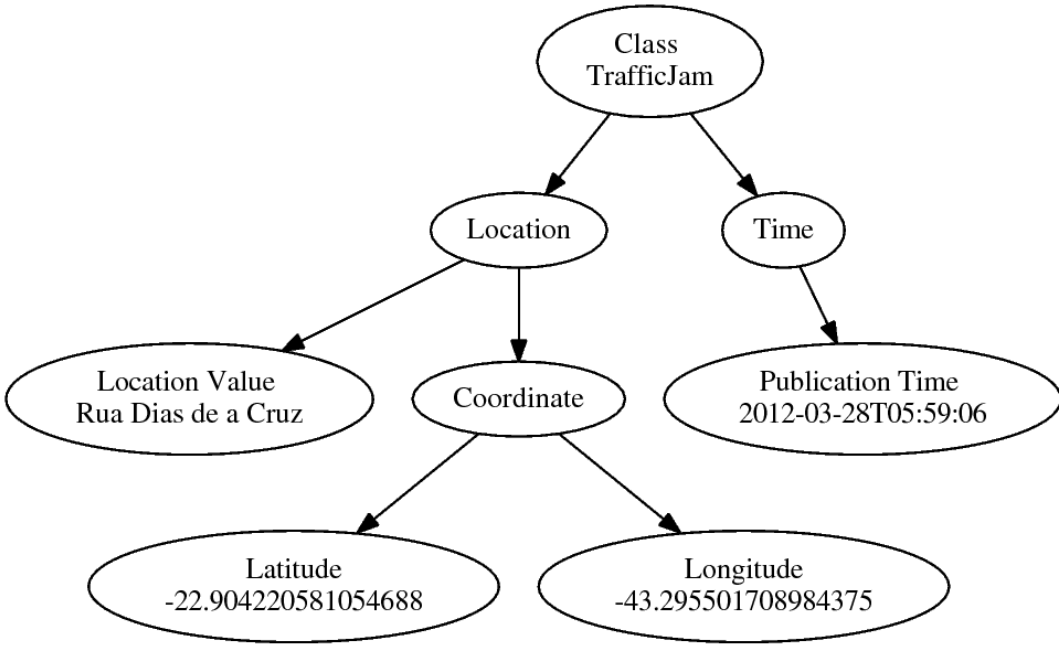

Figura 17 Exemplo 2 de visualização de um evento gerado

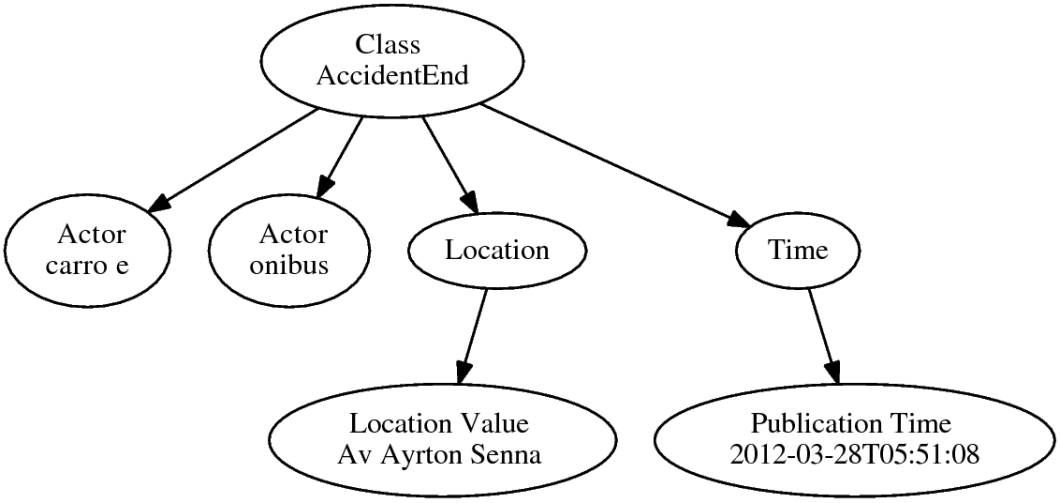

Figura 18 Exemplo 3 da visualização de um evento de gerado.

6
Conclusão e Trabalhos Futuros

6.1.
Conclusões

Esta dissertação assumiu como objetivo principal a criação de um modelo de eventos de trânsito publicados em notícias e presentes na Internet em grande escala.

Para este modelo foi desenvolvido TEDO - Traffic Event Domain Ontology, que fornece definições sobre o domínio de eventos no trânsito. Caso seja partilhada entre pessoas e aplicações e publicada na Web, máquinas poderão entender e interpretar os dados da Web de forma a raciocinar sobre eles provendo informações e serviços importantes ao usuário. Além disso, se diferentes agentes e aplicações tiverem conhecimento sobre a ontologia, a interligação entre eles pode ser facilitada. Ela também pode servir como uma base de dados e ser alvo de interrogações, a fim de procurar instâncias de eventos dentro delas e relacionamentos entre estas instâncias.

A criação de uma ontologia não é um processo estático, é um processo iterativo, ou seja, o modelo da ontologia não é imutável, ele pode e deve ser modificado ao longo do tempo de acordo com as necessidades verificadas devido ao uso do modelo. A Ontologia TEDO criada é, portanto, apenas uma primeira versão de um modelo. Com o seu uso pode ser verificada a necessidade de mudanças neste modelo.

Além da ontologia, foi implementado também uma aplicação para mostrar a usabilidade da TEDO. Esta aplicação utiliza a ontologia para a análise de dados publicados em linguagem natural no *twitter*.

Nos testes do módulo Análise de Notícias, os resultados encontrados no processo de estruturação do texto foram melhorados se comparados aos encontrados por Albuquerque [8], que utilizava dez classes na classificação. Neste trabalho mais classes foram utilizadas na classificação, aumentando o nível de detalhamento, o que permite que o modelo seja mais especifico. Foram medidos

os resultados para 20 classes, e a média encontrada foi de 87,7 % da f-measure. Embora matematicamente o resultado seja menor, considerando seu maior detalhamento, os resultados foram bastante satisfatórios e melhores do que os já encontrados anteriormente com a especialização dos Fatos e Intensidade do trânsito. Esta melhora se dá pelo fato de o conjunto de testes ter sido estendido e pela definição mais formal da classificação, decorrente da existência de uma ontologia definida, o que retirou algumas ambiguidades durante a classificação do conjunto de testes.

6.2. Trabalhos futuros

Ao final desta dissertação muitas ideias de trabalhos futuros para a melhoria deste trabalho surgiram. Alguns pontuais, outros amplos que podem originar trabalhos tão grandes quanto este.

Alguns destes trabalhos são explicitados nas próximas subseções.

6.2.1. Extensão da ontologia

A ontologia criada nesta dissertação representa um domínio bastante especifico que é o domínio dos eventos de trânsito publicados em notícias, mas também pode ser considerado muito amplo, pois a Internet possui muitas fontes de notícias feitas em diferentes formatos.

Assim sendo, esta ontologia representa as notícias que foram analisadas para o seu desenvolvimento, mas durante a sua utilização pode ser verificada que existem informações importantes que são publicadas nestas notícias que não estão representadas corretamente. Para tanto seria interessante que fosse criado um mecanismo de feedback desta ontologia onde novas informações pudessem ser acrescentadas automaticamente a essa ontologia à medida que seus objetos fossem sendo instanciados e que sua necessidade fosse averiguada.

Por exemplo, o caso das subclasses de *TrafficEvent*. As classes são definidas para cada tipo diferente de eventos e uma classe *Others* é utilizada para os eventos que não possuem classe especifica serem classificados. No caso de um tipo de evento ainda não classificado passe a ocorrer repetidas vezes ele passasse a ter

uma importância maior e seria interessante um mecanismo que gerasse na ontologia uma classe para este novo tipo de evento.

6.2.2. Reutilização de ontologias existentes e conhecidas

Apesar de não haver uma forma correta para a criação de uma ontologia, é comum entre as metodologias de criação um passo onde é feita a reutilização das ontologias já existentes. No caso da ontologia TEDO, ela poderia reutilizar ontologias do tipo superior ou geral (upper ontology ou foundation ontology) que são modelos de objetos comuns geralmente aplicáveis a uma grande variedade de ontologias de domínio. Essas ontologias podiam ser reutilizadas nas classes Time e Location a principio, mas antes deveria haver algum estudo para verificar esta possibilidade.

Existem algumas bibliotecas de ontologias nas Web como, por exemplo, Dublin Core, GFO, OpenCyc/ResearchCyc, SUMO e DOLCE.

Existem estudos a respeito de técnicas para mesclar ontologias, porém esta área de pesquisa ainda é muito teórica.

6.2.3. Criação de regras que permita a detecção de relações causais entre eventos não publicados na mesma notícia.

No protótipo apresentado nesta dissertação as relações de causalidade são detectadas apenas em eventos que são publicados em uma mesma notícia na Internet e que possuam uma palavra no texto que indique que houve uma relação de causa entre eles.

Seria interessante que fossem estudadas outras formas de detectar relações de causalidade. Podiam ser estudadas regras para serem aplicadas aos eventos de forma que essas relações pudessem ser deduzidas. Também poderia ser estudada a viabilidade de utilização na ontologia de lógica descritiva como foi feito em [7] onde a partir de inferências possa ser detectadas relações de causalidade.

7
Referências Bibliográficas

[1] Worboys, M. e Hornsby, K. **From objects to events: Gem, the geospatial event model.** In Proc. of GIScience 2004, páginas 327–344, LNCS 3234, 2004.

[2] Guizzardi,G., Falbo, R.A. e Guizzardi, R.S.S. **A importância de Ontologias de Fundamentação para a Engenharia de Ontologias de Domínio: o caso do domínio de Processos de Software.** Revista IEEE América Latina, v. 6, n.3, p. 244-251, 2008.

[3] Sowa. J.F. **Ontology**. Disponível em : http://www.jfsowa.com/ontology/ [Acessado em 23/04/20012]

[4] Kaneiwa, K., Iwazume M. , Fukuda, K. **An upper ontology for event classifications and relations.** In AI'07 Proceedings of the 20th Australian joint conference on Advances in artificial intelligence, páginas. 394-403 . 2007.

[5] Dongli, Y, Suihua, Z., Ailing, W.. **Traffic Accidents Knowledge Management based on Ontology.** In Sixth International Conference on Fuzzy Systems and Knowledge Discovery. Páginas: 447 – 449, 2009.

[6] Wang, J., Wang, X**. An ontology-based Traffic Accident Risk Mapping Framework.** In 12th International Symposium, SSTD 2011, Minneapolis, MN, USA, August 24-26, 2011, Proceedings.

[7] Haggag, M.H, Mahmoud D.R. **OnTraJaCS: Ontology based Traffic Jam Control System**. In International Journal of Computer Applications. Volume 60– No.2, December 2012.

[8] Albuquerque, F. **Environment changes detection: A proactive system to monitor moving objects.** Rio de Janeiro, 2012. 65p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

[9] **Cloud Made Map.** Disponível em: http://cloudmade.com/. [Acesso em 27/07/2013.]

[10] **GraphViz.** Disponível em: http://www.graphviz.org/ . [Acesso em 01/08/2013]

[11] **OpenLink Virtuoso.** Disponível em: http://virtuoso.openlinksw.com/ [Acesso em 27/07/2013.]

[12] Pfoser, D., Tryfona, N. **The Use of Ontologies in Location-based Services:The Space and Time Ontology in Protégé.** In Towards a General Theory of Action and Time. Allen, 1984

[13] **Protégé**. Disponível em: http://protege.stanford.edu/. [Acesso em 27/07/2013.]

[14] W3C. **SPARQL 1.1 Query Language**. Disponível em: http://www.w3.org/TR/rdf-sparql-query/ . [Acesso em 01/08/2013]

[15] **Virtuoso Jena Provider.** Disponível em : http://virtuoso.openlinksw.com. Acesso em 01/08/2013]

[16] Platt, John C. **Fast Training of Support Vector Machines Using Sequential Minimal Optimization.** In Advances in Kernel Methods - Support Vector Learning . 1998.